

# HAP Technology White Paper

# Table of contents:

- [Home](#)
- [Roadmap](#)
- [Technology White Paper](#)
- [Available Platforms](#)
- [Supported Components](#)
- [Recommended Servers](#)
- [Parameters for Environment Detection](#)
- [Security Protection](#)
- [Common Commands](#)
- [Environment Variable](#)
- [Offline Resources](#)
- [Cross-origin Resource Sharing](#)
- [Proxy Configuration](#)
- [HTTP](#)
- [HTTPS](#)
- [Multiple Access Address Configuration](#)
- [Configure Separate Access Address for Files](#)
- [Quick Installation](#)
- [Manual Installation and Deployment\(AMD version\)](#)
- [Manual Installation and Deployment\(ARM version\)](#)
- [Data Management](#)
- [Backup](#)
- [Restore](#)
- [Data Backup and Restore in MySQL](#)
- [Upgrade Microservices](#)
- [Upgrade Document Preview Service](#)
- [Upgrade Storage Component Service](#)
- [Upgrade MongoDB 3.4 to 4.4](#)
- [Same-server Deployment of Data Pipeline Service](#)
- [Standalone-server Deployment of Data Pipeline Service](#)
- [Same-server Deployment of Data Pipeline Service](#)
- [Standalone-server Deployment of Data Pipeline Service](#)
- [Upgrade Manager](#)
- [Strong Passwords for Database](#)
- [Change Default Password for MySQL](#)
- [Change Default Password for Redis](#)



- [MongoDB Add Authentication](#)
- [Ops Platform](#)
- [Kubernetes](#)
- [Microservices Upgrade](#)
- [Non-Cross Version](#)
- [Cross Version](#)
- [Upgrade MongoDB 3.4 to 4.4](#)
- [Mount Configuration File](#)
- [MongoDB Deployment \(Single Node\)](#)
- [Convert MongoDB Single Node to Replica Set](#)
- [Create MongoDB Database](#)
- [Common Commands in MongoDB](#)
- [Slow Query Optimization](#)
- [Application Data Storage Architecture](#)
- [Multi-cluster Storage for Application Data](#)
- [Multi-cluster Storage for Row Record](#)
- [Application Data Cleaning](#)
- [Clean Log Data](#)
- [Reclaim Disk Space](#)
- [Execution History of Workflow](#)
- [Logs of Row Record in Worksheet](#)
- [Application Behavior Logs](#)
- [Exclusive Computing Power](#)
- [Exclusive Database](#)
- [Open web](#)
- [Quick Start](#)
- [Catalog Description](#)
- [Replace Text](#)
- [New Page](#)
- [Reference Components](#)
- [API Calls](#)
- [Parameter Description](#)
- [MD\\_Structure](#)
- [StructureLog](#)
- [Calendar\\_Category](#)
- [Calendar\\_Member](#)
- [Calendar](#)
- [ChildCalendarRecurTime](#)

- ChildCalendarRecurTimeMember
- User\_CalCategory\_Order
- Account\_Device
- Application
- OAuth2\_Access\_Token
- Project\_Application
- User\_Application\_Order
- User\_Application
- Account\_ActionLog
- apk
- apksort
- appapply
- appauthorize
- appentity
- appitemrecoveryrecord
- applog
- appnew
- apprecoveryrecord
- appsettings
- appshare
- appsort
- groupsort
- homesetting
- markedapp
- markedgroup
- personalgroup
- projectgroup
- worksheetsetting
- wsaliasrelation
- activitylog
- field
- role
- view
- worksheet
- wsbtn
- wscollection
- wscontrols
- wsfilter

- [wslogs](#)
- [wsrowindexconfigs](#)
- [wsshare](#)
- [wsview](#)
- [ws\[wsid\]](#)
- [wslogs](#)
- [Deployment Issues](#)
- [Third-Party Applications](#)
- [Forgot Login Password?](#)
- [Dependent Cloud Services](#)
- [SSO](#)
- [OAuth 2.0](#)
- [CAS](#)
- [OIDC](#)
- [Ticket](#)
- [SAML2](#)
- [How to enable email service](#)
- [How to enable SMS service](#)
- [How to enable message integration](#)
- [How to enable custom file object storage](#)
- [How to push session status to external system](#)
- [How to extend the dependencies of code blocks](#)
- [How to extend the execution environment of code blocks](#)
- [Format Conversion \(WPS\)](#)
- [Format Conversion \(Self Integration\)](#)
- [LibreOffice](#)
- [WPS](#)
- [Based on OpenAI](#)
- [Autonomous integration](#)
- [Based on OpenAI](#)
- [Based on the Alibaba Cloud Bailian platform](#)
- [Based on Tencent Cloud](#)
- [Autonomous integration](#)
- [How to enable data pipeline](#)
- [How to independently deploy data pipeline service](#)
- [How to enable aggregate table](#)
- [Microsoft ADFS-SAML Instructions](#)
- [Google-SAML Integration Guide](#)

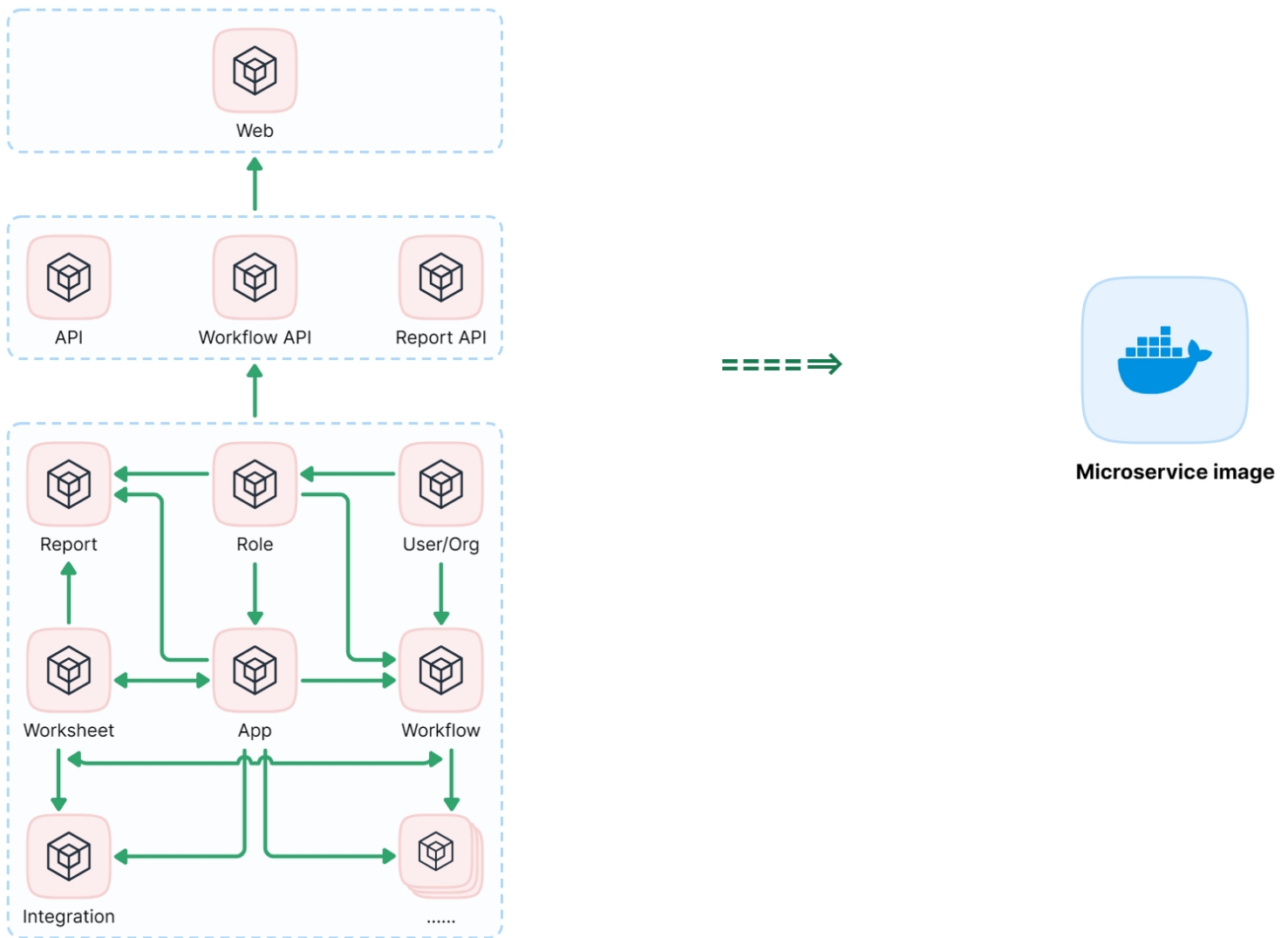
- [Google-OIDC Integration Guide](#)
- [Workflow](#)
- [File](#)
- [Continuous Queuing of Workflows](#)
- [Icon Not Showing](#)
- [Unable to Access Page](#)
- [Changelog](#)
- [SBOM](#)

# Home

**HAP** is a tool for quick design and development of enterprise software. It requires no code and can be mastered and used by ordinary business people. With flexible functional components, personalized systems such as CRM, ERP, OA, and project management can be built for managing corporate activities like production, sales, procurement, etc. HAP Server is the container deployment edition of HAP SaaS service, which empowers enterprise users to build no-code application platform on their own cloud computing facilities.

## Design

HAP is based on microservice architecture and uses Docker mirrors for deployment. In the private deployment mode, in order to ensure the stability of the system while facilitating deployment, HAP adopts a monolithic program model based on the microservice architecture, merging each independent microservice mirror into one big mirror (microservice application mirror). The configuration files are completely built-in and expandable. The mirror also contains relevant data storage components (available by default in standalone deployment mode), such as MySQL, MongoDB, Redis, Elasticsearch, Kafka, file object storage based on **MinIO** and online file preview module based on **OnlyOffice**. You can eventually start HAP service with a single command.



The monolithic program mode is inseparable from the single point of failure, but the monolithic mode of the HAP Server is more for simplifying deployment, and the internal is still a collection of microservices (**similar to the multi-container Pod in Kubernetes**), so in order to ensure the availability of each service process within the container, a health check thread is preset in the container, which allows automatic recovery when a service fails. [Standalone Deployment Mode](#)

Cluster Deployment Mode is a more granular clustering of microservices and dependent components, including clusters of storage components (like MySQL, MongoDB, Redis, Elasticsearch, Kafka, etc.) and clusters of microservice applications. Microservice application mirrors themselves are already adaptive to standalone and cluster deployment modes. [Cluster Deployment Mode](#)

## Architectural features

### Microservice Architecture and Containerized Deployment

HAP adopts a microservice architecture, which divides system functions into multiple independent and scalable service units, enabling each module to run autonomously. This design enhances the system's response speed, fault tolerance, and maintenance efficiency, and supports agile development for cross team collaboration. Each microservice can be managed and scheduled separately, achieving functional isolation and efficient invocation.

By combining Docker and Kubernetes containerization technologies, HAP's microservice components can run efficiently across environments, greatly simplifying system deployment and management.

Containerization abstracts the underlying hardware resources and, through Kubernetes' elastic scaling, automatic scaling, and load balancing capabilities, enables the system to remain stable in response to sudden business demands. In addition, HAP integrates middleware such as cache and message queue, further improving system performance and stability in high concurrency scenarios

## **Multi level deployment mode**

HAP Server provides flexible deployment environment support, allowing users to choose suitable cloud computing resources based on their own resources. Whether it is virtual machines, physical machines, or cloud hosts, HAP is compatible. Under such diverse deployment options, HAP can better adapt to the IT budgets and infrastructure conditions of different enterprises.

In order to adapt to the resource investment of enterprises of different scales, HAP supports both single machine and cluster deployment modes. Single machine deployment is suitable for small and medium-sized enterprises or testing environments, while cluster mode provides flexible resource selection for larger enterprises. The cluster mode is divided into three levels of configuration schemes: streamlined, standardized, and specialized. Enterprises can flexibly choose according to their current business needs and budget.

In addition, HAP's cluster architecture supports dynamic resource adjustment and mode switching. During business development or expansion, users can seamlessly upgrade from streamlined configuration to higher specifications, or choose to appropriately reduce configuration when resource utilization decreases, ensuring optimization of resource investment.

## **Service instance optimization and scenario adaptation**

HAP provides service instance management functionality based on actual business needs, allowing enterprises to adjust the number of instances as needed to better adapt to different scenarios. By pre configuring templates, enterprises can flexibly choose configuration strategies based on business load, such as configuring more replicas for high-frequency access services, reducing resource consumption of non core modules, and optimizing overall costs.

This design enhances the scalability and resource utilization of the system, ensuring that HAP can quickly respond to new functional requirements and adapt to diverse application scenarios. The independent scalability of service modules enables HAP to achieve rapid iteration in developing and integrating new features, thereby helping enterprises maintain technological advantages and cope with business fluctuations.

## **Visual and community-based installation experience**

In order to help users get started and experience the product faster, HAP Server provides a free community version, where users can complete product installation and configuration through simple visual guidance steps without the support of a professional IT team. Through a series of intuitive interfaces and detailed installation instructions, the Community Edition greatly reduces users' learning curve and provides a convenient initial experience.

At the same time, the design of the community version is not just a basic product, but also has support for future upgrades. Users can upgrade to the commercial version at any time based on business growth when trying out the community version, and achieve smooth upgrades through a visual management interface without the need for reinstallation or reconfiguration.

## **Comprehensive operation and maintenance management and system observability**

HAP is equipped with a powerful operations manager that provides support for system monitoring, alerting, backup, and recovery from multiple perspectives, ensuring the observability and robustness of the system. include:

**Monitoring:** Real time monitoring of the system's resource usage, including key indicators such as CPU, memory, and network traffic. Through graphical display, operations personnel can promptly identify and address performance bottlenecks.

**Alarm:** Supports multi-level alarm rule configuration, and notifies key events through email, SMS, or third-party tools for quick response by operation and maintenance personnel.

**Data management:** Provides data archiving and backup functions to ensure the persistence and security of critical data, while also supporting dynamic adjustments of multiple storage architectures to meet business expansion needs.

**Performance monitoring:** Conduct independent performance monitoring and optimization suggestions for the core functions of the system to ensure the smooth operation of the system's processes.



# Iteration

HAP Server and HAP SaaS Edition are basically the same in terms of functionality, while the SaaS Edition iterates more frequently, and it generally takes about 2 to 3 weeks to synchronize the features to the HAP Server, which is because it takes some time for edition merging, testing, bug fixing, etc. All of this is also to ensure the stability of the new edition of the Private Deployment.

# Roadmap

► More planned needs

## v6.1.0

### ! DEVELOPING

- Ops Platform v1.1.0 (alarm, database slow query, index suggestion)
- Support for configuring the validity period of login sessions (internal users/external portals/apps) in system configuration
- Support minimum installation mode (minimum resource requirements)
- Sync HAP SaaS Edition v6.1.0

## v6.0.0

### 💡 RELEASED(2025-01-13)

- Existing applications support migration to dedicated database
- Support for configuring password policy descriptions in multiple languages in system configuration
- Support for configuring AI service, which can be **based on OpenAI** or **autonomous integration** in system configuration
- Support for configuring OCR service, which can be **based on Tencent Cloud** or **autonomous integration** in system configuration
- Online document editing support **WPS cloud edition** and private deployment edition
- **Ops Platform** v1.0.0 (monitoring capability), simultaneously adapting to both standalone and cluster deployment architectures
- Sync HAP SaaS Edition **v6.0.0**

## v5.8.0



**RELEASED(2024-11-05)**

- Application logs support archiving
- Support configuring customized app push channels
- Support configuring map services (Gaode, Google)
- Sync HAP SaaS Edition [v5.8.0](#)

## v5.7.0



**RELEASED(2024-09-29)**

- Support viewing archived data (workflow execution history, worksheet record modification logs)
- Workflow AIGC node supports [configuring OpenAI models](#)
- Sync HAP SaaS Edition [v5.7.0](#)

## v5.6.0



**RELEASED(2024-07-24)**

- Support configuring hidden plugin modules
- Support setting to enter designated application after successful login
- Single sign on support for using Google accounts
- Support enabling new version release reminder
- Support [enabling aggregation table functionality](#)
- Support [AI mode for configuring workflow AIGC related nodes](#)
- Support [configuring Amap API Key](#)
- Sync with HAP SaaS [v5.6.0](#)

## v5.5.0



**RELEASED(2024-06-24)**

- Storage component: [MongoDB upgraded to v4.4](#), Kafka upgraded to v3.6
- Support enabling AI features (worksheet field suggestions, automatic generation of workflow code blocks, and application of multilingual intelligent translation)
- Single sign on support for configuring ADFS identity source

# Technology White Paper

## Abstract

HAP is designed to meet the needs of enterprises in digital construction. Focusing on the complexity of backend applications, it extracts the data structure of high granularity, user roles, permissions, workflow and other definition capabilities, providing both developers and users a flexible platform to implement various enterprise applications.

HAP is a modern enterprise software product based on container technology. The public cloud edition runs on nocoly.com, while the container edition can run on the customer's own computing facilities via Docker.

### Core Values:

**Build enterprise apps without coding, Release the user power by providing utter flexibility;**

**Get rid of DevOps process, Enable business engineering;**

**Speed up application implementation dramatically;**

**Make dataset unified by integrated application platform;**

**Provide common open platform for all applications created;**

**Intuitive and fluid interactive design;**

**Fully implemented with cloud-native technology stack.**

## 1. Customer Value

HAP product is aimed at addressing key pain points in the digital construction of enterprises. Each value corresponds to a problem that enterprise IT users have been trying to solve. The no code/low code application platform integrates mature technologies in the fields of cloud computing, big data and application development and deployment environments, and is the main solution for enterprises in their digital transformation efforts.

### (1) Ultimate flexibility and implementation

For information system construction around business management processes, such requirements are usually difficult to be met by off-the-shelf software products, but have to rely on tedious custom development. Custom development inevitably involves complex requirements communication, prototype validation and costly development, testing, deployment and improvement processes. APaaS, with paradigmization and high-granularity configurability, enables **the entire process to be achieved through a no-code or low-code approach**, replacing costly custom development and dramatically reducing the cost. In this way, not only is the first implementation fast and flexible, but APaaS still provides convenience when the system adjusts as the business changes.

## **(2) Data interconnectivity**

There are **data silos** of applications existing among medium and large-scale enterprises. Some are from custom systems developed with different technology stacks, and some are from application software products of different brands. In the process of digital management, there is a need to **connect these data silos** to enable cross-application access to data or to precipitate application data into a data middleware. Thereby, a more flexible digital collaboration can be achieved.

## **(3) Open design**

Enterprise applications built through HAP are automatically established with standard open interfaces (REST), and complete interfaces for all data objects to add, delete, check and modify. Webhook triggers and action nodes in the workflow can also be docked with other application APIs with no code. The front-end interface of HAP can be embedded in other application pages, and the pages of HAP can also be embedded in other applications.

## **(4) Easy-to-learn application building**

Most of the application scenarios can be built in a no-code way. Although the product capabilities are in-depth, the interaction interface is simple and intuition-driven. Non-developers who are familiar with business processes can also get started.

## **(5) Cloud-native technology stack and cross-cloud deployment capability**

HAP adopts a microservice architecture and makes extensive use of cutting-edge cloud-native technology stacks and mainstream open source frameworks, including Java, Node.js, MongoDB, MySQL, Redis, gRPC, Elasticsearch and Kafka. Through container technology and automatic operation and maintenance orchestration, it provides users with a highly resilient and reliable application operating environment. HAP can both run on public clouds and on any customer-owned computing facility. The functional features of both are the same.

## 2. Application Scope

With HAP products, users can build various enterprise middle and back end applications with no code, which, in general, includes the following application types:

### **(1) Core business management applications based on relational data structure**

For the business management systems built around the core business in different industries, such as the purchase-sales-inventory system in the distribution industry, production execution, material management and equipment management systems in the manufacturing industry, project management system in the modern service industry, etc., the alternative to these scenarios can be to purchase a specialized suite of software products or to resort to custom development. While an APaaS product like HAP can provide the both of personalization, high flexibility and high development efficiency.

### **(2) Applications that rely on mobile applications to collect data**

Application scenarios in manufacturing, engineering, retail, and other industries that require specific functional staff to collect data from the front line.

### **(3) Applications that write data and build management board with API interface**

Extract data from multiple heterogeneous systems, precipitate them into a unified data center, and build a management cockpit in combination with the custom dashboard of this product. This application scenario replaces the BI+ETL solution with another way.

### **(4) Department-level applications for specific business segments**

Because of the ease-to-use and code-free characteristics of no-code system, user companies can have non-developers from business departments directly build or lead some simple applications. With the unified application management capability, it can also prevent shadow IT problems.

### **(5) Applications built for process automation**

The automated workflow based on HAP can bridge the past interrupted processes that require manual coordination, such as the automatic connection of orders, deliveries and invoices; time-triggered checklist generation, equipment maintenance reminders, contract expiration reminders, etc.

### **(6) Applications built to enable data flow, filling and approval processes**

Build manually controlled workflow applications based on form data, approval and filling nodes for complex data collaboration.

Below is a list of over 30 common categories of enterprise applications and an assessment of the replacement capabilities of APaaS product.

Name	Full Name	Description	APaaS Replacement Capability
ABM	Account Based Marketing	ABM is one of the more segmented of modern marketing strategies, and the related software products are a branch of MAS. It changes the traditional way of creating content and marketing campaigns, but generates highly personalized content based on each customer. It is more about the quality of customer communication than the quantity, so it is often used in B2B, luxury goods, airlines, high-end hotels, insurance and other industries.	6
APS	Advanced Planning and Scheduling	APS refers to an application that helps to plan and schedule for manufacturing processes, focusing on optimal use of production materials and equipment resources, with the goal of maximizing capacity utilization.	6
ATS	Application Tracking System	It is an application that manages the entire recruitment process of a company. Its main value is to integrate resume submission from external job websites and other resume pools, and to manage the entire process from resume submission to hiring of candidates. Most ATS systems come with the ability to help companies build their own talent pools.	5
Accounting	Accounting Software	Financial software in a narrow sense refers to the recording of financial documents through double-entry bookkeeping, forming a financial general ledger and providing analysis of financial indicators. A complete financial suite also	8



Name	Full Name	Description	APaaS Replacement Capability
		includes modules for purchasing and inventory tracking, invoice processing, the receivable and payable management, etc. An ERP suite in the complete sense should include a financial management component.	
BI	Business Intelligence	BI is an important classification of enterprise software that is responsible for extracting data from various business data sources and building analytical databases, providing business analysis to management decision makers through visualization capabilities. Full-stack BI suites include data warehouses, ETL (data extraction, transformation, and loading) modules, OLAP query, and visual representation layers. BI is often integrated with other enterprise applications through databases and APIs.	5
BPM	Business Process Management	BPM is a collection of abstract tools for managing and driving various process activities in an enterprise, enabling the design, operation and analysis of various business processes across functions. In practice, BPM is a tool to meet the needs of approval control and business automation. Most BPM software is designed to follow the BPMN specification, allowing visual creation of workflows, monitoring of workflow running, and providing the user interaction interfaces necessary for workflow operation. Emerging low-code development platforms have features that are crossover with BPM.	9
CAD	Computer Aided Design	This is an old software category that has continued from the PC era to the present. In the	1

Name	Full Name	Description	APaaS Replacement Capability
		enterprise software category, it refers to design suites for the architectural and engineering, manufacturing, and entertainment media industries. Mature CAD suites provide not only design tools, but also components such as component libraries, material libraries, and design file management. The main users of CAD are engineers, interior designers, animators, architects and engineers.	
CAM	Computer Aided Manufacturing	It is a twin product of CAD. Usually input design files, optimized material and equipment parameters in CAD applications and give final execution instructions to digital machinery and equipment. It is typically used in machining industries that use CNC machine tools and equipment. Some CAD vendors also have joint CAM products, which are also known as CAE (Computer Aided Engineering).	1
CEM	Customer Experience Management	CEM is an emerging category of enterprise software. It includes customer service systems in the general sense, but with a greater emphasis on touchpoints around the full customer lifecycle and omnichannel. Its product capabilities therefore include personalized delivery of customer portals, convenient and consistent customer service support systems, and digital experiences, with the goal of measuring and improving customer satisfaction at different stages.	5
CMS	Content Management System	Content management systems mainly serve the management and deployment of enterprise website content. Conventional enterprise CMS	3

Name	Full Name	Description	APaaS Replacement Capability
		includes management of news, web pages, blogs and e-commerce platforms, where business users can directly create and edit online content and publish without code writing. The domestic web environment attaches more importance to the management of mini programs and official accounts in the WeChat ecology, and related products also belong to the category of CMS in a broad sense.	
CRM	Customer Relationship Management	It is a very important category in enterprise software that is responsible for managing the business aspects of customer capture and sales. CRM in a broad sense includes sales automation, marketing automation and customer service automation, while in a narrow sense it refers to the sales funnel from lead acquisition to customer conversion. CRM products are numerous and are important players in the enterprise software market, so many CRM products are more or less extended to some business aspects other than sales. CRM in the standard sense requires the management of leads, opportunities, customers, contacts, products, orders, price lists, contracts and other basic business objects.	8
CS	Customer Support	It is also known as a work order management system or HelpDesk Software, which serves the customer service departments, providing cross-channel customer service platform capabilities and managing the full lifecycle of customer service work orders. Because of the specialized nature of phone support, there is also platform software that specifically serves call centers, and	7

Name	Full Name	Description	APaaS Replacement Capability
		they also belong to this category. A complete CRM suite usually contains a customer service module. CS also crosses over with CEM (Customer Experience Management).	
DMP	Data Management Platform	Although it is a seemingly generic software category, it primarily refers to audience data management for online advertising. DMPs can collect and manage data about customers, audiences and campaigns from advertisers, advertising agencies and distribution media, so as to provide advertisers with detailed analysis and planning tools for online campaigns.	0
DMS	Document Management System	It is an application used to manage documents shared within an enterprise, but it goes beyond basic file storage to include a series of enhanced features such as image scanning of documents, text recognition, version control, meta data management, format conversion, collaborative editing, permission control, template management, search, and electronic signature. So it is suitable for customers in industries with special or intensive requirements for document archive management.	4
ERP	Enterprise Resource Planning	It is the core application for manufacturing and distribution enterprises, covering almost all business processes of such enterprises, managing key business aspects such as customers, sales, purchasing, warehousing and finance. Due to its comprehensiveness, there are also many enterprises that use ERP to refer to all integrated business systems, even non-	9

Name	Full Name	Description	APaaS Replacement Capability
		manufacturing and trading industries, so in market practice, the boundary between it and CRM products is often not that clear .	
HRM	Human Resource Management	It is a comprehensive suite for managing HR-related business, including personnel records, compensation, benefits, performance, evaluation, training, attendance and many other sub-sections. It is used to carry organization information and export master data of personnel and department structure for other applications.	7
LCAP	Low Code Application Platform	LCAP, also known as LCDP (Low Code Application Development Platform), is a model-driven application implementation mode. It defines application elements including data model, views, rules and workflows through a deep visual configuration interface. It is suitable for implementing middle and backend applications of relationship database type. There are also some low-code products that provide the ability to customize front-end pages, so they can also be used for some front-end type applications. LCAP and the APaaS that emerged in the cloud computing era are gradually converging to become a unified category.	10
LIMS	Laboratory Information Management System	It primarily serves laboratories in R&D, healthcare services and manufacturing organizations, providing them with the management of laboratory equipment, materials, samples, test results and workflows. Users rely on such software to provide compliant data recording and reporting, to analyze experimental results.	9

Name	Full Name	Description	APaaS Replacement Capability
LMS	Learning Management System	In an enterprise, the LMS is used for training and exams for employees, customers and partners. It should be able to create and manage lesson plans online, manage the learning process of learners, and organize learning communities in groups or classes. In the education industry, the LMS is the core business system that enables paperless teaching and supports the online learning.	5
MAS	Marketing Automation Software	It is an application for companies with repetitive and scaled marketing activities, with a focus on three business objects, customers, leads and marketing campaigns. The marketing automation suite is capable of managing marketing campaigns across various channels, such as email, SMS, social media, landing pages, etc. MAS can be understood as part of a broad CRM and integrated with parts of SFA to bring leads and opportunities to the sales funnel. MAS for Internet companies can also be highly integrated with user products to capture user behavior in real time for dynamic user segmentation. Tools and software such as Email, SMS marketing also fall into this category.	5
MES	Manufacturing Execution System	MES in the manufacturing industry is responsible for managing the process from the production of raw materials to the finished product manufacturing. Specifically, it creates and tracks manufacturing work orders, determines manufacturing procedures based on process routes, captures and records raw material usage and work hours during manufacturing, and	8

Name	Full Name	Description	APaaS Replacement Capability
		analyzes production efficiency. MES generally requires integration with ERP or PLM applications.	
OA	Office Automation	It is a mainstream domestic enterprise software category of great inclusiveness and comprehensiveness. It generally refers to the supporting systems for internal communication, documents, tasks, processes and so on. There are also OA products of complexity to provide a more complete portfolio of features, covering the management of internal personnel files, orders, contracts, suppliers and other business data. Because of its relatively high degree of integration, OA is crossed with specialized applications such as knowledge management, collaborative applications, process management, personnel management, etc.	9
PDM	Product Data Management	PDM, also known as PIM, integrates the management of technical and marketing information about products. Enterprises with a large number of products or product versions generally need to use PDM to organize product information, share it internally or among channel partners, to keep product information up-to-date and consistent.	9
PLM	Product Lifecycle Management	PLM helps R&D-oriented organizations manage the product development process, including innovative ideas, prototypes, iterations, and retirements. PLM can manage various design and process changes, features and requirements, drawing documentation, quality metrics and	9

Name	Full Name	Description	APaaS Replacement Capability
		testing, BOM (materials), compliance requirements, supply chain, etc.	
PMS	Property Management System	PMS is the core business software for industries such as rental properties and hotels. It manages property assets, tenants, leases and the business activities. The advanced PMS suite also has the ability to dynamically manage room prices to maximize revenue. In addition to the generic PMS, there are management applications around various specific commercial property forms.	6
POS	Point of Sales	POS is a combination of hardware and software system used at the retail end of the industry to collect payments and record sales flow. In the new retail context, POS has been gradually replaced by more integrated solutions. Cash registers have also been replaced by mobile payment terminals. In a related area, there are also various self-service sales hardware that assume the common functions of POS and sales clerks in the past.	1
PSA	Project Service Automation	PSA is also known as Professional Service Automation because most professional services are provided on a project basis. The PSA suite helps professional services teams stop using various tools to manage customers, time, costs, contracts, collections, materials, etc. in a fragmented manner, instead adopting a unified platform to create an automated interface between sales, service and financial processes.	8



Name	Full Name	Description	APaaS Replacement Capability
SCM	Supply Chain Management	<p>SCM manages the raw material planning, procurement, logistics, warehousing, and integrated production process of a manufacturing company and then goes on to manage the distribution of finished goods to customers.</p> <p>Compared to ERP suites, SCM manages the supplier sourcing process in more detail, usually consisting of modules for supply chain planning, procurement, order management, logistics and transportation, warehouse management, and demand management. Because of industry differences, the product market for SCM is fragmented and there is rarely a single solution that can fit all industries.</p>	7
SCRM	Social CRM	<p>SCRM is a typical category that has impacted the shape of enterprise software in the social media era. It can be understood as a business application that revolves around sales and marketing management, and integrates the ability to interact with customers through social media.</p> <p>In western countries, social media mainly refers to Facebook, Twitter, Instagram, etc. while in the Chinese market, it refers to WeChat. Therefore, sales and marketing applications built around WeChat and WeCom can be classified into SCRM.</p>	3
TMS	Transportation Management System	<p>TMS provides a more sophisticated management capability in the logistics aspect of supply chain management, which includes the management of freight rates, transportation contracts, transportation scheduling, cargo tracking, route planning and optimization, etc. TMS is generally used in the freight industry and large</p>	6

Name	Full Name	Description	APaaS Replacement Capability
		manufacturing industry with high logistics density.	
WMS	Warehouse Management System	WMS is an advanced application of general inventory management that serves large warehousing, retail and logistics industries to meet their in-depth needs in reducing personnel errors and paperless operations. A complete WMS system has many advanced features, including integration with IoT devices for identification, picking, labeling, automatic replenishment, automatic task assignment, counting systems, returns management, and the ability to integrate with e-commerce platforms, ERP software, and more.	6

Objectively speaking, there are also enterprise applications that do not work well with zero-code or low-code application platforms for implementation. They are:

- (1) Front-end type applications for consumers.
- (2) Professional tools such as CAD/CAM.
- (3) Some areas that are already covered by cost-effective standardized products, such as purchase-sales-inventory system for micro-enterprises.
- (4) Vertical applications for specialized scenarios with special views and algorithms, such as cashier management for the restaurant industry and dynamic room rate forecasting for the hotel industry.

### 3. Implementation Principle

The fundamentals for building enterprise middle and back end business applications with no and low code lies in the fact that most of these applications are built on the two core goals of data management (CRUD) and workflow. For a typical enterprise application building, there are similar parts in the design and

development process. Application Platform as a Service (APaaS) extracts these similarities and replaces the front-end and back-end design and code development processes of a project with an application-based approach.

In order to deliver a friendly and efficient APaaS product, different vendors may adopt slightly different implementation mechanisms. The solution chosen by HAP can be broadly described as: build a data model through form visualization, create data views for different roles on the model, and implement custom data actions and automated processes through workflows with trigger and action nodes, and built-in data visualization capabilities. It consists of the following as a whole:

(1) **Data structure.** Create a data model through form visualization, where the controls directly correspond to field types. And the necessary relationship data structure is implemented through associated worksheets, formulas, and fields in associated worksheets. Controls such as signature, positioning, member, department, and autonumber are designed to support specific business scenarios.

(2) **View.** Define the form of data presentation with different types of views, including grid, board, org, etc. Each view can be configured with flexible data filtering conditions, showing or hiding rules of fields, and sorting logic.

(3) **Statistics.** Create charts in worksheets, including bar charts, line charts, pie charts, etc. The created charts can be for personal use or published as shared charts in application.

(4) **Custom page.** In addition to worksheets and views, users can create a blank page where they can visually configure different types of components. It is quite common to implement a management dashboard by adding charts.

(5) **Users, roles and permissions.** Users can customize the roles in the application and configure permissions for each role to access, edit, delete and add data. The permission definition can be down to the field level, enabling flexible application distribution by adding users or departments to roles, and fully visualizing RBAC (Role-based access control).

(6) **Workflow.** Automated workflow consisting of triggers and action nodes can be understood as IFTTT (If this, then that) in enterprise applications. The workflow can be triggered by data changes, dynamic time and Webhook methods. Action nodes enable data query, locate, updating, adding, and deleting, as well as performing manual control processes such as approving and filling, sending personalized notifications, and pushing data to external APIs. Workflows can also be triggered by custom buttons attached to specific views.

With these six parts, users can build any enterprise application based on data management and workflow with no code.

## Differences between visual IDEs and rapid development tools (RAD)

The HAP product and many visual IDE products, as well as traditional rapid development tools, are often referred to together as low-code platforms. All products share a common goal in terms of achieving efficiency in applications. However, in terms of technical path selection for implementation, HAP is fundamentally different from the low-code tools of IDE model.

- (1) As an APaaS product, HAP fully adopts the cutting-edge technology stack in the field of cloud computing, no more source code generation for single application, and no more code compilation process. The application runs directly on the platform with a model approach. Development requirements such as version control and debugging are also done through the application configuration interface.
- (2) Most of Mindao's capabilities are no-code based, involving low-code in code block nodes of workflows and docking and development work only. Therefore, HAP is able to target business developers without relying on the user's software programming skills.
- (3) HAP features application exchange capability. Any application built by any tenant can export models. Models can be redistributed or imported to other tenants' environments.
- (4) Compared with development tools, HAP is easy to learn and easy to use. Business developers can master some simple single-session applications through quick learning, while professional developers and information architects can realize complex applications with the high-granularity definition capability of this product.

## 4. Complex Application

Before delving into HAP products, users may have misconceptions about the application implementation capabilities of zero-code products, thinking that no-code APaaS is only suitable for building some simple applications but cannot be for core business problems.

However, this is not the fact.

The complexity of enterprise software may be attributed to the following characteristics of enterprise applications:

- (1) A large number of data objects, complex data attributes, and multiple levels of data association.
- (2) The application involves many roles, and each role requires a differentiated permission configuration, which is related to several elements such as data attributes and organizational structure.

- (3) The data management itself cannot meet the needs of the process, involving data flow, approval, filling, etc.
- (4) There are many dimensions for data analysis, provided in large quantities by application role in reports.
- (5) Involves special and highly difficult algorithms.
- (6) Involves integration with external applications.

The so-called complex projects, whether it is a file management system for military enterprises or a production execution system for SMEs, are essentially composed of such complexities.

## **How APaaS to respond**

**To break it down into relatively simple segments. This mindset is a fundamental tool for solving complex problems.**

For the above complexity, find the corresponding capability modules in the APaaS product and then implement a high-granularity configuration based on the custom capabilities that these modules can provide.

### **Worksheet**

- (1) Data tables need to support various data types and be configured with various appropriate presentation controls.
- (2) Data objects can be added freely.
- (3) Data objects can be associated with each other in all modes (one-to-one; one-to-many; associate current worksheet).

### **View**

- (1) It can present different dimensions of the data table to different user roles.
- (2) It supports flexible row filtering, column filtering, sorting, and configuration of showing or hiding fields on the form.
- (3) It supports configuring different custom buttons for different views.

### **Roles and Permissions**

- (1) Assign custom roles to the application.
- (2) Configure permissions of high granularity for different roles, including permissions of adding, deleting, viewing and modifying view and worksheets, even down to the field level.
- (3) Combine with the user's organizational reporting relationship to define the scope of data permissions.
- (4) Distribute function pages and custom buttons for different roles.

## **Workflow**

- (1) Trigger a workflow with any data change, time condition or manually.
- (2) The workflow should support query, locate, add, modify and delete data.
- (3) The workflow should have the ability to operate with bulk data and arrays.
- (4) The workflow should be able to couple the approval and filling work with manual participation.
- (5) The workflow should support high-level scripting language to implement special algorithms.
- (6) The workflow should be able to operate on any data type (text, numeric, multiple choice, file, etc.).

## **Reports**

- (1) Report visualization based on arbitrary data objects and associated data objects.
- (2) Diverse report types.
- (3) Combine different reports and distribute them to different user roles.

## **Open Interface**

- (1) Provide interfaces for any data object to add, delete, view and modify.
- (2) When executing the workflow, to call other applications' interfaces.
- (3) Provide data interfaces to external applications and be able to start workflows through these interfaces, and external data can be passed to the workflow.

The above elements and capabilities ensure the flexibility to solve each granular problem, and with such flexibility, systems can be assembled no matter how complex they are. Coupled with a rational application

building process, it is possible to provide complex applications to customers while still maintaining order.

And, because of the ability to freely distribute roles, each individual is not plagued by complexity when complex applications are provided to specific roles. For example, we can distribute a single view to a worker role in a production execution system, perhaps with just a few buttons on their personal interface. This is something difficult to do with traditional enterprise software development.

## **Case of Complex System**

### **(1) Production Execution System**

Production execution system is an application system that can help manufacturing enterprises to manage production and manufacturing, improving management efficiency and accelerating digital transformation of manufacturing enterprises. The production management system has a high degree of flexibility in system configuration. Users can freely configure business data and processes based on the original framework of the system and according to their own business characteristics.

- Connect with ERP system to obtain information on goods purchase, arrival and inventory.

### **(2) ERP System**

ERP system integrates purchasing management, sales management, inventory management and billing management of the enterprise. It is a whole process management business within the enterprise, involving multiple departments such as sales, procurement, finance, warehouse, etc. For such a system, traditional custom development may take 2-3 months, while HAP can complete the building and delivery in about 2 weeks, meeting users' individual needs.

- Purchase order consolidation, which allows generating one purchase order from the same supplier through process configuration;
- Sales, purchases, or returns, synchronized to inventory;
- Different categories of materials configured with different inbound and outbound rules;
- Realization of inventory transfer, FIFO;
- Special processing of accounts, such as red flush, etc;
- Real-time generation of multi-dimensional statistical reports;

## **ERP System Function Diagram**



Dashboard	Procurement Management	Sales Management	Inventory Management	Billing Management
Data Overview	Supplier Database	Customer/Contacts	Warehouse	Sales Invoice
Sales Board	Supplier Quotation	Quotation	Product Inventory	Purchase Invoice
Purchase Board	Purchase Request	Sales Order	Purchase Inbound	Accounts Receivable
Inventory Board	Purchase Order	Sales Invoice	Sales Outbound	Accounts Payable
Finance Board	Purchase Return	Sales Return	Inventory Transfer	Payment Transactions
	Goods Receipt		Return In/Out	

### (3) Configuration of roles and permissions involving multiple departments and levels

For enterprises with cumbersome business processes, there are often many departments and levels involved, and the roles and permissions are relatively complex. For complex permission requirements, we usually rely on the RBAC model (Role-Based Access Control) for analysis and design. In HAP's application, you can configure the permissions of a role within minutes. Through view distribution, you can flexibly configure the worksheets, records, and even fields that are visible to each role.

- Different roles can view or work with different forms, records, and fields;
- Different roles have different operation permissions for the same form, record, or field;
- The supervisor can view or work with the data of his subordinates;



User

User

Role

Expand

+ Create Role

Search Roles

System

Administrator

Operator

Developer

Custom

Sales

Role Name

Sales

Hide the application

Description

Which App Items are distributed?

Distribute all application items (simple)

Distribute selected application item

Accessible views and data permissions

Items

View

Application Instructions

Lead Opportunity

Contract Payments

Sales Performance Ranking

Lead

All

Opportunity

All

Opportunity Details

All

Follow-up Log

All

Action Plan

All

Save

Cancel

Manage Users

Lead

Record Range

Action

Data Field

Name

New

View

Edit

Basic Information

Subject

Source channel

Lead number

Lead pool

Customer name

Company size

Industry

Estimated number of users

Contact name

Mobile

Email

Department

Position

Requirement Notes

Marketing Notes

#### (4) Multi-party integration of data

A single business system is difficult to meet the multi-faceted information needs of enterprises. There are often multiple sets of information technology software within the enterprise, so new business systems need to be integrated with the original system, or to establish an information center. The situations are as follows:

- Some features or requirements cannot be implemented on HAP (e.g. logistics status information, spatial modeling, etc.);
- The original system has a large amount of business data left behind, and data migration is not convenient;
- The data from the various systems used cannot be connected, and a data middleware is needed as a bridge;
- Various other reasons make it impossible for the company to give up using the original system.

Each application of HAP has standard API documents. Through the open API interface or Webhook users can interact with data from other business systems. Worksheets, records, and fields all have corresponding IDs and can be retrieved flexibly.

MEGA CRM API documentation

Overview

Request format

Authorization management

IP Whitelist

Obtain app info

Worksheet

- Create Worksheet
- Get worksheet structure
- Leads
  - Fields comparison table

Get list POST

New Record POST

Create records in batch POST

Obtain the Record GET

Get record details POST

Update the Record POST

Update record details in batch P...

Delete Record POST

Get associated record POST

Get the shared link for the record...

Get the total number of rows in th...

Get list POST

Request URL:https://api.mingdao.com/v2/open/worksheet/getFilterRows

Parameter	Mandatory	Type	Description
appKey	Is	string	AppKey
sign	Is	string	Signature
worksheetId	Is	string	Worksheet ID
viewId	Is	string	[["All": "66699b723f02770494548f"]]
listType	No	string	Return data type: 0 for specified view format (l imited data amount if not table view), 1 for tabl e view format. Default is 0 if unspecified, but 1 is recommended.
pageSize	Is	number	Row Number
pageIndex	Is	number	Page
sortId	No	string	Sort Field ID
isAsc	No	boolean	Ascending or not
filters	No	list	Filter combinations. Parameters refer to <a href="#">Appen dix</a>
notGetTotal	No	boolean	Whether to improve performance by not counti ng the total rows
useControllId	No	boolean	Whether to return only controllid, default false

Example of submission

```
{  "appKey": "c089ebed70122423",  "sign": "Y2RkM2I5ODU1ZTgyY2ZkNTk3YjMxIDhjNTkyMGFkOTMyZDUzZWUzY2RkYWIzNzQ0YjQyY2daITk1ZTRk",  "worksheetId": "xiananuo",  "viewId": "View ID, can be empty",  "pageSize": 50,  "pageIndex": 1,  "listType": 0,  "sortId": "Sort Field ID",  "isAsc": "Ascending or not",  "filters": [    {      "controlId": "control1",      "dataType": 6,      "spliceType": 1,      "filterType": 13,      "value": "2"    },    {      "controlId": "control2",      "dataType": 6,      "spliceType": 1,      "filterType": 13,      "value": "2"    },    {      "controlId": "control3",      "dataType": 6,      "spliceType": 1,      "filterType": 13,      "value": "2"    }  ]}
```

At present, there are typical docking including Amazon order docking, Kingdee K3 docking, Kingdee Starry Sky, IoT platform docking, visualization large screen docking, Fanruan T100 system docking, WeCom& DingTalk docking and so on. As long as there is a standardized open interface, all can be docked through HAP to facalitate data pipeline. At the same time, HAP can also be served as a data center to help enterprises unify data asset management, activate data value and release enterprise productivity.

## Data Pipeline

As APaaS products with no/zero code begin to be adopted by more and more enterprises, there is an increasing need to utilize the natural openness of APaaS products to integrate with other existing business systems. The following provides a few basic models for integrating APaaS products with other applications for users in need.

In the introduction, the different integration methods are illustrated with a specific example:

Scenario: An e-commerce company uses three systems, Purchase-Sales-Inventory, APaaS (assumed to be HAP) and Financial Software. The goods-return process of the company is as follows: customer applying for return => customer service personnel to check => product returned to the warehouse => financial pesonnel to refund money => completed. For each complete return process, data needs to flow among the three systems.

### 1 Message integration mode (recommended)

It enables the transfer and synchronization of data among applications by calling APIs, and may be divided into the following three ways (or phases).

**One-to-one synchronization (request/response): messages request another application by calling the API and waiting for the response**

When a customer applies for a return, the e-commerce platform calls the API of the APaaS platform to notify the customer service personnel to start the approval process, and then call the API of the e-commerce platform to change the status of the return application after the approval, and retries every time the request fails. When the volume of requests is large and the business logic is complex, there will be long waiting time or overload of computing resources on the other side, then you need to take the following another way.

**One-to-one asynchronous (queue/callback): API receives the request into the queue, after the execution call back to notify the results**

After receiving the returned goods, customer service personnel check and pass the return application, calling the API of the inventory application to notify the inventory-in. Since the logic of inventory-in is relatively complex (change the original out-inventory order and process the inventory), you can put the operation into the message queue to gradually consume it, and when the return logic is completed, call back the API of another application to notify the result. Many times if a data change, you need to notify multiple applications and the notification address often changes, it is necessary to frequently adjust the code logic, in which case you can take the publish-subscribe approach.

**One-to-many asynchronous (publish/subscribe): provide event subscription, send message notification to all subscribed addresses when data changes**

After receiving the returned goods, the APaaS platform needs to notify the finance system for refund, to notify the warehouse system for inventory-in, and to notify the e-commerce platform for changing the status of the returned goods, which may be different for different customers' needs. At this time, APaaS can provide event subscription, when the data changes, according to the subscription list to notify other applications, and provide webhook callback address, and then callback after the execution of other applications logic.

The message integration approach can best achieve **real-time data synchronization**, but it requires a complete API on the application side, and also needs to pay attention to the retry mechanism for push failures and the sequence coverage caused by subsequent retries.

HAP (APaaS) implements integration through workflow:

**Workflow Configuration Panel: Get associated records**

It is a way of obtaining the sole piece of data that meets the conditions with filters and collations, which can be used by other nodes in the flow.

**Choose the object to obtain**  
 The Node Object  
 Button trigger Worksheet "Lead"

**Select field of association type**  
 The System will provide the first record the Field linked with for the Nodes in Flow  
 Leads (Link to Worksheet "Leads")

**Filter**  
 Set the filter to find the data that meets the conditions. If no filter is set, obtain sole piece of data by the collation.  
 + Filter

**Collation**  
☐ Ignore collation and get one at random  
 When multiple data are found, the first data will be obtained according to the following sorting rules. If no rule is set, the first data is returned according to the sorting rule configured in the field  
 Select fields Set rules

**If cannot obtain any data**  
☒ Continue the process  
 Nodes below will skip execution if using this node object or related data  
☐ Abort the process, or continue to judge the branch based on the finding results

## 2 Application Integration

### Specialized development of integration tools:

Develop a separate application, obtain authorization from other applications, pull data at regular intervals (**full/incremental**) according to configuration, and push it to other applications based on conditions and mapping rules. Using the integration tools to synchronize data without calling applications, can **reduce coupling** and also allow flexible selection of technical solutions based on data volume and application form. However, since it is not actively pushed by the application, it is not that perfect in terms of real-time performance.

### Robotic Process Automation (RPA)

It is a business process automation technology based on software robotics and artificial intelligence. It provides an alternative way to automate the end-user's manual process by mimicking the way the end-user operates on the computer.

## 3 OLAP Data Source Connection

Configure the data source connection address to drill down to the required data according to open permissions and filtering conditions, usually as a way of BI report analysis. The data drilled down to can be stored separately again after processing or aggregation as needed.

#### 4 File Integration (Export & Import)

Manually export the data to be processed from the application system as needed, import it into another set of application system, and then import it back to this application system after processing. The HAP system provides a relatively sophisticated interaction for importing Excel files, enabling users to create field mappings when importing. It can also automatically de-duplicate and create data associations during import.

### 5. Function Structure

HAP provides three major functional modules: no-code application, collaboration suite and account management.

**(1) No-code Application Building.** It shows HAP capability. Users can build enterprise applications with the provided basic components. For an application, the following encapsulated capability components are required.

- Worksheet: Build the data structure of the application
- View: Present data based on user roles
- Statistics: Data-based analysis
- Custom Page: Add components such as charts and distribute them to specific roles
- User Roles and Permissions: Configuration details of roles and permissions
- Workflow: Business automation with triggers and action nodes

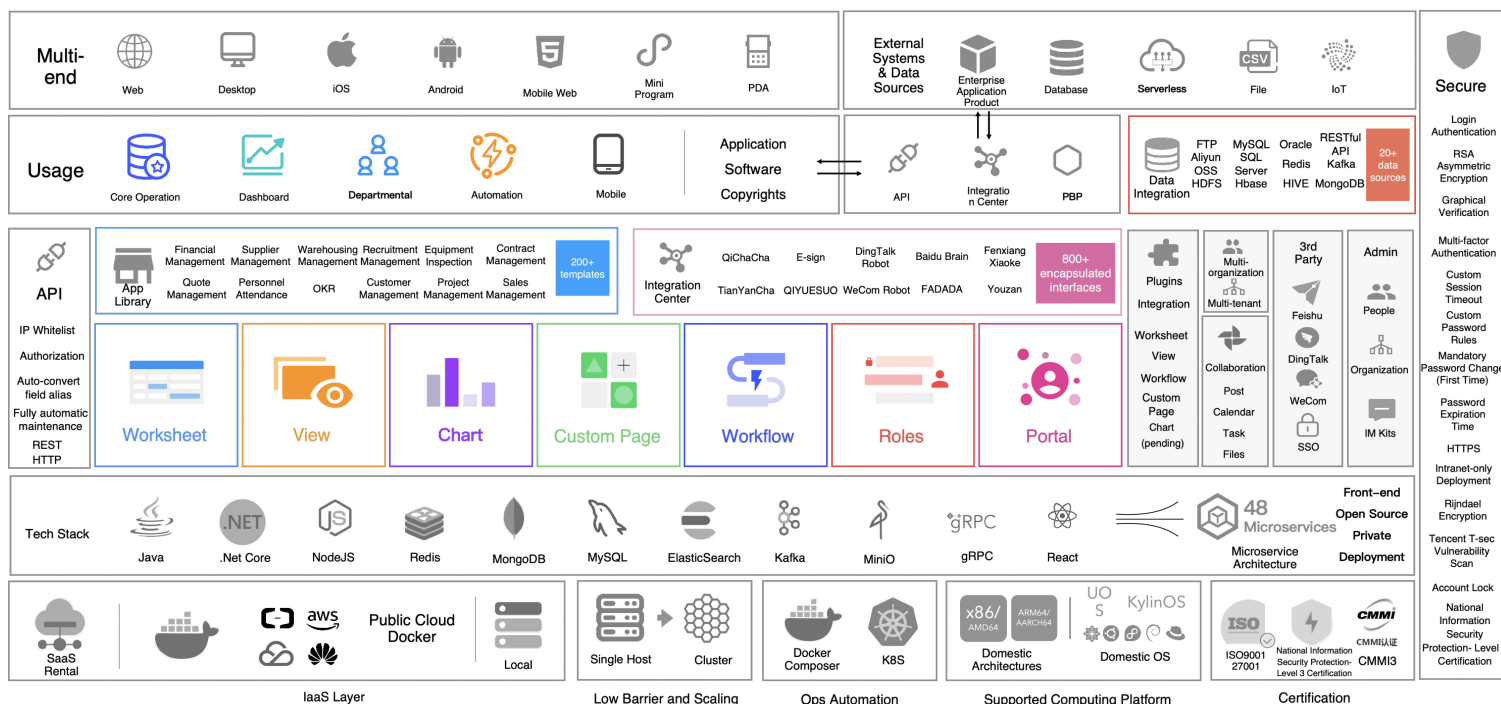
**(2) Collaboration Suite.** It meets the needs of sharing, communication, task collaboration, scheduling, and document management for enterprises, and can be connected with application platform capabilities.

- Post
- Task
- Calendar
- Files

**(3) Account Management.** It is divided into personal account management and enterprise account management.

- Personal Account
  - Modify personal information
  - Bind mobile phone and email
  - Modify password
  - Set language
  - Set information privacy
- Enterprise Account
  - Member/Department management
  - Reporting relationship setting
  - Handover
  - Group announcements
  - Data export
  - Organization information editing
  - Group management
  - Usage statistics
  - Roles and permissions management
  - Application management
  - Workflow management

## **6. Technology Architecture**



## Deployment Resources

- System deployment requirements are based on Linux operating systems, supporting virtual machines, physical machines, and cloud servers.
- Compatible with mainstream operating systems (such as Debian, Ubuntu, CentOS) and common CPU architectures (such as AMD64, ARM64).

## Deployment Methods

- Standalone mode (based on Docker Compose)
- Cluster mode (based on Kubernetes)

## Tech Stack

- Architecture: The platform is based on a microservices architecture, implemented using the gRPC framework, with over 80 microservices.
- Frontend technology: React
- Backend technology: Java, C#, Node.js
- Database: MySQL, MongoDB
- Middleware: Redis, Kafka, Elasticsearch, MinIO

## Database and Middleware Application Notes

- MySQL: Stores system basic data, such as user and organization information.



- MongoDB: Stores business data, such as application data, worksheet data, and workflow data.
- Redis: Provides caching support to enhance system performance. Multiple core features (such as user information, organization information, application data, worksheet data, workflow data, and application role permissions) utilize a caching mechanism, covering almost all parts of the system.
- Kafka: Key use cases include worksheet aggregation, worksheet data collection, and workflow events. Kafka's high throughput and multi-partition features ensure that events are executed in order and reduce latency when the consumer endpoints are appropriately configured. Kafka is also used in the super search module as a data transmission middleware.
- Elasticsearch: Primarily supports global search for cross-application worksheet data and collaboration suite data. It also provides periodic statistical dimension views for application usage statistics.
- MinIO: The default open-source file storage system is MinIO, which supports S3-compatible alternative solutions (such as Alibaba Cloud OSS, Tencent Cloud COS, Huawei Cloud OBS, Qiniu Cloud, etc.).
- Flink: Provides stream processing capabilities for data pipeline, supporting real-time synchronization from multiple data sources, and performing ETL data processing during synchronization.

**Core Function Modules:** Includes worksheets, views, statistics, custom pages, role permissions management, external portals, and more. These modules enable the platform to quickly build applications that meet business requirements. By offering APIs for integration and data pipeline functionality, the platform facilitates data exchange with external systems and addresses data silo issues.

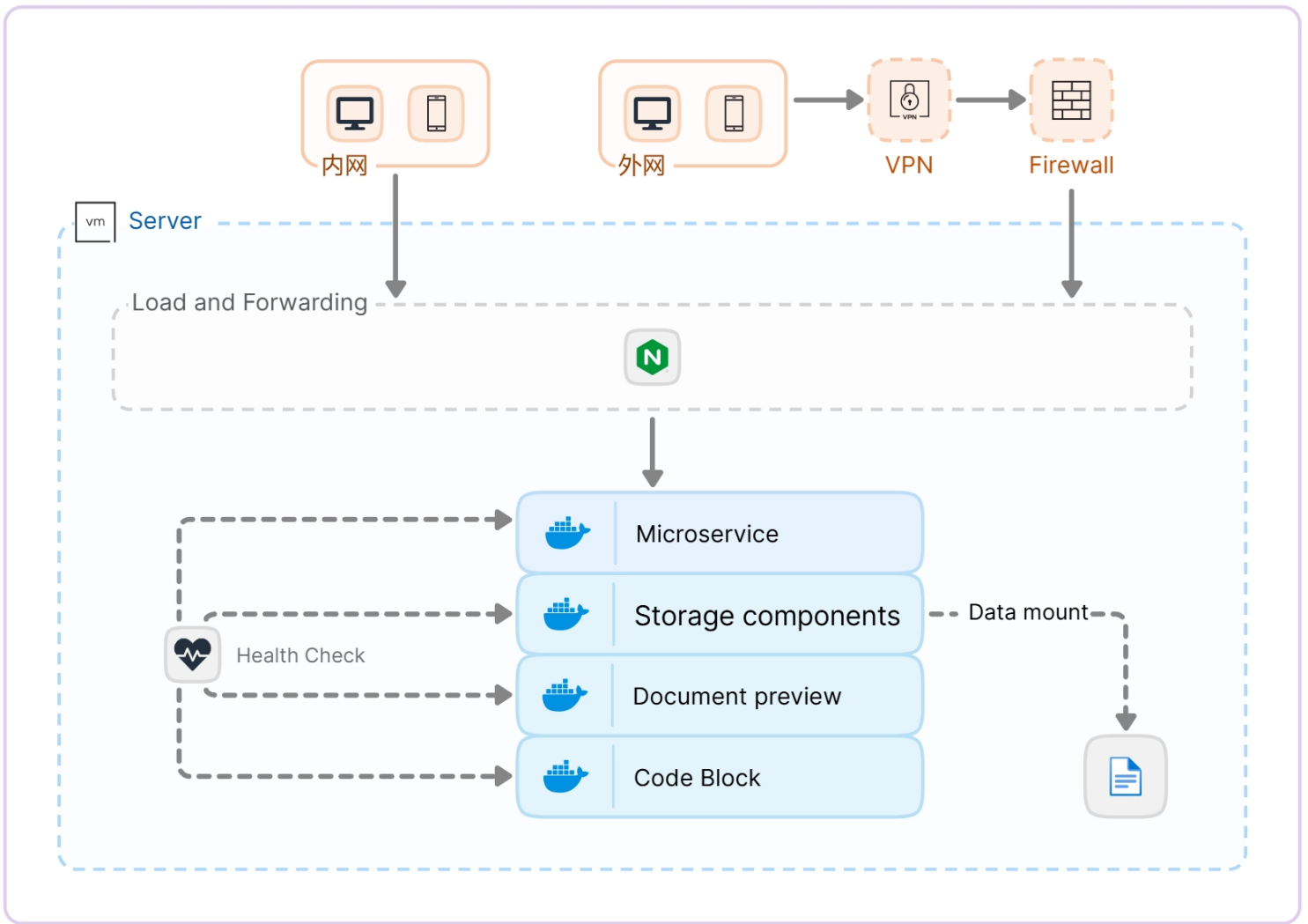
**Application Library and Plugin Center:** The platform comes with a rich set of application templates that cover various industries. Users can install and reconfigure templates from the application library, reducing the initial usage barrier. The Plugin Center also provides plugin design support for worksheet views and workflow nodes, further extending the platform's functionality.

**Basic Data Management and Integration Functions:** The platform offers basic data management features, including personnel and organization management, single sign-on (SSO), and supports integration with third-party systems like WeCom and DingTalk.

**Multi-Device Real-Time Visualization:** Applications built on the HAP platform support real-time synchronization across multiple devices, including Web PC, Web mobile (H5), mini-programs, and native apps (iOS, Android), ensuring a consistent experience for users across different devices.

## **Standalone Deployment Mode**





## Architecture Overview

In standalone deployment mode, only one server is required (server configuration depends on concurrency requirements). The system will start four containers, each serving different functions:

- **Microservice Container:** A collection of microservices handling the core business logic of the platform.
- **Code Block Container:** Functions for code blocks within workflows.
- **Storage Component Container:** Includes the database and middleware, with data persisted to a specified directory on the server. There is no need to manage the deployment and startup of individual components.
- **Document Preview Container:** Provides online preview capabilities for Office documents and PDFs.

Each service process within the containers includes a health check mechanism. If a service process crashes, the system will automatically restart it for recovery.

An optional proxy can be added at the service layer to route access to the microservice entry point.

Clients can decide whether to enable external network access based on requirements, supporting full access or secure entry via VPN.

**Strengths and Weaknesses of Deployment Architecture**

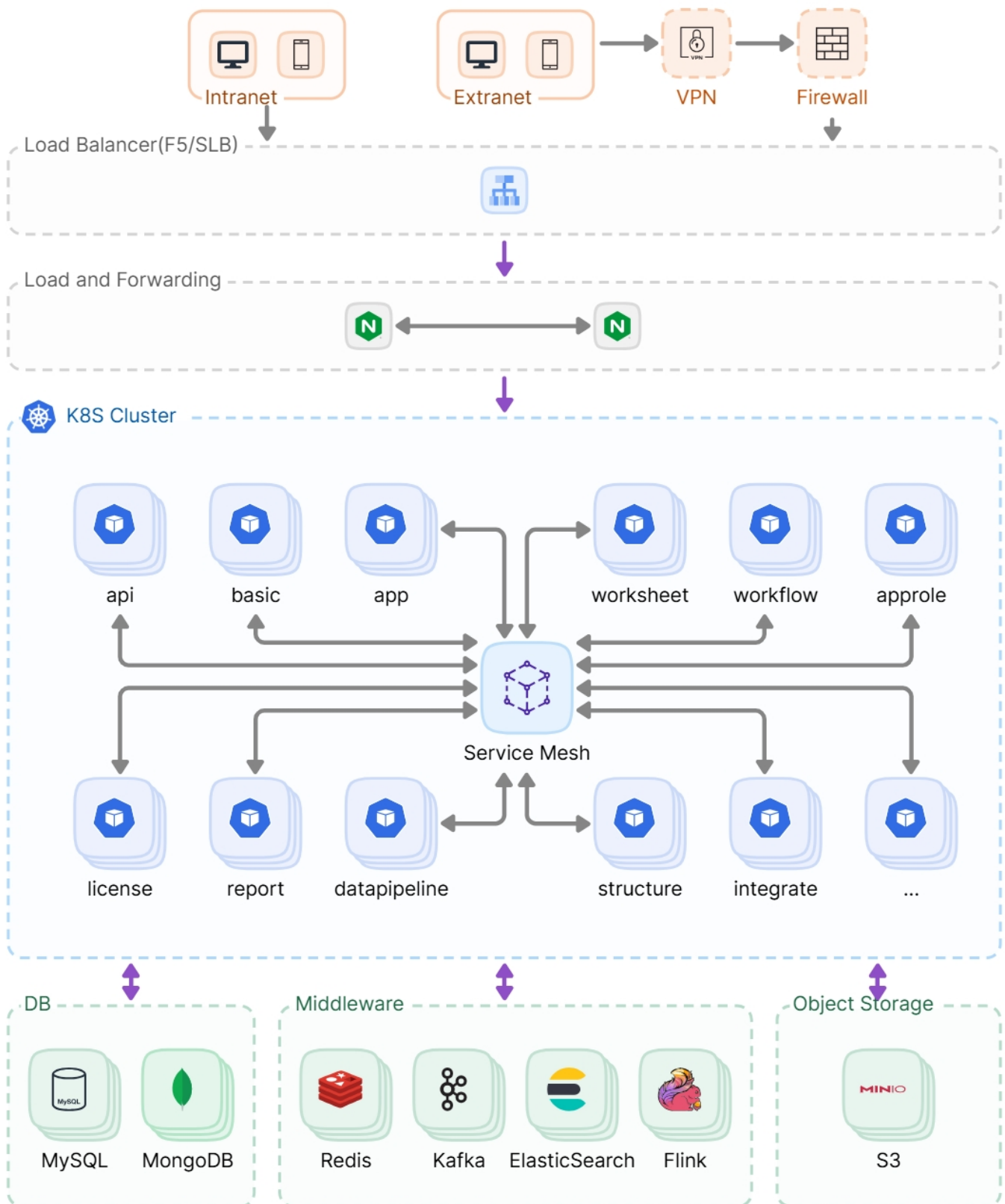
Strengths: simple and fast deployment, low server resources and O&M costs

Weaknesses: existing single point of failure, but there are self-contained internal health checks and self-recovery capabilities

**Recommended server configurations**

Concurrency	Configuration	OS
Within 100	8C/32G/40G system disk/100G data disk	Debian 10+
Within 200	16C/32G/40G system disk/100G data disk	Debian 10+
Within 300	24C/48G/40G system disk/100G data disk	Debian 10+
300+	Recommend cluster deployment mode	-

**Cluster Deployment Mode**



Architecture Overview

- The product's underlying architecture relies on multiple databases and middleware to ensure efficient operation. The appropriate deployment mode (single-node or multi-node) for each component is chosen based on concurrency requirements.
- The microservice layer is deployed using Kubernetes, supporting the adjustment of the number of instances for each microservice based on demand. Services with high traffic (e.g., worksheet service, workflow service) are configured with more replicas, while non-core modules optimize resource consumption by reducing the number of replicas.
- Implementing load and forwarding through multiple proxy services to ensure high availability of the system.
- Clients can decide whether to enable external network access based on requirements, supporting full access or secure entry via VPN.

#### Lite Edition (Concurrency 300+)

Strengths: save server resources and O&M costs, high availability of microservice clusters, and relatively independent deployment components, suitable for environments with limited resources or small-scale application scenarios

Weaknesses: some component coupling, possible single point of failure of storage-related services

Recommended server configurations are as follows:

Usage	Configuration	OS	Deployed Service	Number
Load and Forward Server	4C/8G/100G system disk/200G data disk	Debian 10+	Nginx	1
Microservice Application Server Cluster	8C/32G/100G system disk/200G data disk	Debian 10+	HAP Microservices	2
Document Preview Server	4C/16G/100G system disk/200G data disk	Debian 10+	Office & PDF File Online Preview	1
Middleware Server	4C/16G/100G system disk/500G data disk	Debian 10+	Kafka, Elasticsearch, File Object Storage	1
Data Storage Server	8C/32G/100G system disk/300G data disk	Debian 10+	MySQL, MongoDB, Redis	1

Usage	Configuration	OS	Deployed Service	Number
-------	---------------	----	------------------	--------

**Standard Edition (Concurrency 600+)**

Strengths: Having basic high availability capabilities, such as node redundancy and higher utilization of automatic fault tolerance, and being able to effectively utilize server resources. Suitable for most usage scenarios

Weaknesses: There is interference and resource competition between different storage components, and attention should be paid to resource allocation and management

Recommended server configurations are as follows:

Usage	Configuration	OS	Deployed Services	Number
Load and Forward Server	4C/8G/100G system disk/200G data disk	Debian 10+	Nginx	2
Microservice Application Server Cluster	8C/32G/100G system disk/200G data disk	Debian 10+	HAP Microservices	3
Document Preview Server	4C/16G/100G system disk/200G data disk	Debian 10+	Office & PDF File Online Preview	2
Cache Server	4C/16G/100G system disk/200G data disk	Debian 10+	Redis	2
Middleware Server	4C/16G/100G system disk/500G data disk	Debian 10+	Kafka, Elasticsearch, File Object Storage	4
Database Server	8C/32G/100G system disk/300G data disk (SSD)	Debian 10+	MySQL, MongoDB	3

**Professional Edition (Concurrency 1000+)**

Strengths: At the same time of high availability, a single application architecture is adopted (only one component instance runs on each server) to improve reliability and performance, avoid interference and resource competition between different components, and also facilitate independent expansion and management of each component

Weaknesses: high server resources and O&M costs

Recommended server configurations are as follows:

Usage	Configuration	OS	Deployed Service	Number
Load and Forward	4C/8G/100G system disk/200G data disk	Debian 10+	Nginx	2
Microservice Application	32C/64G/100G system disk/300G data disk	Debian 10+	HAP Microservices	3
Cache	16C/32G/100G system disk/200G data disk	Debian 10+	Redis	2
Message queue	16C/32G/100G system disk/500G data disk	Debian 10+	Kafka	3
Full-text search	16C/32G/100G system disk/500G data disk	Debian 10+	Elasticsearch	3
File Storage	16C/32G/100G system disk/500G data disk	Debian 10+	File Object Storage	4
RDB	16C/32G/100G system disk/500G data disk(SSD)	Debian 10+	MySQL	2
NoSQL	32C/64G/100G system disk/500G data disk(SSD)	Debian 10+	MongoDB	3

## 7. Open Source Module

HAP introduces many mainstream open source frameworks, including development languages and middleware.

### (1) .NET

.NET is an open source development platform, jointly maintained by Microsoft and the .NET community on GitHub, featuring **cross-platform, open source, and flexible deployment**. HAP's microservices are partially developed on .NET, currently based on the .NET 8 (LTS).

### (2) Java

Java is an object-oriented programming language of **simplicity, robustness, security, platform independence and portability, multi-threading, and dynamism**. Java, as a representative of static object-oriented programming languages, implements object-oriented theory well and allows programmers in an elegant mindset to perform complex programming. HAP primarily uses the mainstream Spring Boot framework, such as the workflow part.

### (3) Node.js

Node.js is a JavaScript runtime environment based on the Chrome V8 engine for easily building responsive and scalable web applications. Node.js is lightweight and efficient with an **event-driven and non-blocking I/O model**, making it ideal for running data-intensive and real-time applications on distributed devices. HAP's IM Web Socket communication service is built on the Node.js SocketIO framework.

### (4) MongoDB

MongoDB is a database based on distributed file storage, a product between relational and non-relational databases. It is the most feature-rich and relational-like among non-relational databases. It supports a very loose data structure like bson format in json, so it can store more complex data types. The highlight of Mongo is that it supports a very powerful query language, whose syntax is somewhat similar to an object-oriented query language, and it can achieve most of the functions similar to the single table query of a relational database. It also supports indexing of data. The data storage involved in the core module of HAP is based on MongoDB, and its flexible data structure greatly facilitates the fast iteration of the product.

### (5) MySQL

MySQL is a relational database management system that belongs to Oracle and is one of the most popular relational database management systems. MySQL software has a dual licensing policy for a community

version and a commercial version. Many enterprise applications choose MySQL as data storage for its small size, speed, low cost, and especially the open source feature. MySQL is used to store all the structured basic data of users and organizations involved in HAP.

## **(6) Redis**

Redis is an open source Key-Value database of high performance. The value types it supports for storage include string, list, set, zset, and hash. Redis operations are atomic, and data is cached in memory to ensure efficiency. Redis also periodically writes updated data to disk or writes modifications to an appended record file, and implements master-slave synchronization. HAP adopts Redis to provide data caching for many features, such as the basic user information and organization information, but also application information, worksheet information, row data in worksheet, workflow, roles and permissions in application, etc. For query performance, Redis is used basically for all.

## **(7) gRPC**

gRPC is an open source and general-purpose RPC framework of high performance designed for mobile and HTTP/2. It supports multiple development languages and is designed based on the HTTP/2 standard, enabling bi-directional streaming, flow control, header compression, multiplexing requests over a single TCP connection, and more. These features make it perform better on mobile devices, more power efficient and space saving. Most of the microservices of HAP are based on gRPC, including C#, Java, Node.js related services.

## **(8) Elasticsearch**

Elasticsearch is a Lucene-based search server. It provides a full-text search engine with distributed multi-user capabilities, which is based on RESTful interfaces, and distributed as open source under the Apache license terms. According to DB-Engines rankings, Elasticsearch is the most popular enterprise search engine. HAP's Super Search is based on Elasticsearch (including cross-application worksheet data and collaboration suite data) and is designed to give users faster response and more accurate matching. The usage statistics of applications are also based on the periodic statistical dimension of Elasticsearch to better track the usage status of applications.

## **(9) Kafka**

Kafka, an open source stream processing platform developed by the Apache Software Foundation, is a distributed messaging system that supports multiple partitions, multiple replicas, multiple subscribers, and zookeeper-based coordination. By introducing Kafka, it can induce decoupling between services and can withstand the access pressure during peak hours without crashing completely due to sudden overload of requests. For some functions of HAP APaaS, when data changes, it needs to trigger the



workflow. The implementation of this part relies on Kafka's message publishing and subscribing. Also based on Kafka's high throughput and multi-partitioning, events can be executed in an orderly manner and with low latency under the configuration of a reasonable number of consumer ends. In addition, HAP's super search module also uses Kafka as the data transfer middleware.

## **(10) MinIO**

MinIO is an object storage service based on the Apache License v2.0 open source protocol. It is compatible with the Amazon S3 cloud storage service interface and is ideal for storing large volume unstructured data such as images, videos, log files, backup data and container and VM mirrors, etc. An object file can be of any size, ranging from a few KB to a maximum of 5 TB. Files uploaded by users in HAP and some static resource files will be stored in MinIO.

## **(11) Flink**

Apache Flink is an open source processing framework for big data, designed for real-time stream processing and batch processing. It has high throughput, low latency, high scalability and powerful state management capabilities. Flink supports distributed computing and is capable of processing large amounts of data on large-scale clusters. It takes a data flow programming model, making it easy for users to build efficient data processing pipelines. The real-time synchronization of multiple data sources in HAP's data pipeline and the ETL data processing capability in the synchronization process are built on Flink.

# **8. Technical Points**

## **NOSQL Database Selection**

HAP adopts multiple NOSQL data technology stacks such as MongoDB, Redis and Elasticsearch. The main purpose of this is to fully utilize different modal database engines to bring more value to users in terms of feature extensibility, performance and user experience. Although HAP uses complex technology stacks, each technology stack is chosen to be mainstream, open source, runnable and compliant with industry standards. And users do not need to deal with the native technology stack, since the application is built and run in a no-code application environment.

## **Containerized Private Deployment**

HAP is the first publicly available and no code/low code platform that can be deployed via container technology in China. It supports standalone deployment and cluster deployment, and also supports

deployment under Kubernetes, reducing costs of O&M, installation and deployment. This capability enables customers to benefit from the scalability and high availability of cloud-native.

## **Front-end Open Source**

The front-end code of HAP Server supports complete open source (mingdao-openweb), allowing users to fork and secondary development.

## **Multi-platform Deployment**

It adapts to multiple architecture infrastructure, including Linux kernel servers with X86 CPU architecture and ARM64 CPU architecture, as well as domestic operating systems. It also has built-in quick operation commands, including stop, restart, upgrade, rollback, backup, etc.

## **Application Lifecycle Management**

Applications can be imported, exported, backed up, maintained, rolled back, updated and upgraded. The application builder can maintain it efficiently.

## **Open Design**

Enterprise applications built through HAP are automatically established with standard open interfaces (REST), providing complete interfaces for all data objects to add, delete, check and modify and automatically generate documentation. Webhook triggers, action nodes, and API integration in workflow can also be docked with other applications with no code.

## **Hyperautomation**

The innovative automated workflow, in addition to the traditional approval process, supports several triggering modes such as events from worksheet, timed trigger, custom buttons, interface trigger, etc. It also supports the data adding, deleting, checking and modifying in the workflow, as well as message notification.

## **Composability**

Based on the concept of PBC (Packaged Business Capabilities), HAP innovatively designed a packaged business process feature, which utilizes the automated data processing process orchestrated by workflow. It can reuse the data processing capability throughout the organization, accept the agreed parameters, and pass out the result parameters after processed by workflow. In addition to calling internal data, it can also interact with external data through API.

## Data Pipeline

HAP supports external data sources such as MySQL, SQL Server, Oracle, PostgreSQL, MongoDB, IBM DB2, Kafka, etc. as data sources or data destinations to establish synchronization pipelines for real-time data synchronization with second-level latency.

## 9. Intellectual Property

As the APaaS category is getting mature and the complexity and value of building applications on APaaS is also increasing. Many users and service providers are becoming concerned about the intellectual property issues associated with APaaS applications.

The relevant intellectual property laws are concentrated in the Copyright Law and the Contract Law. (In 2021, the Contract Law is repealed and the Civil Code comes into effect.)

This topic is also related to the Regulations on the Protection of Computer Software promulgated by the State Council, since the Copyright Law stipulates that the protection of computer software and information network transmission rights shall be separately regulated by the State Council.

## Formation and Attribution of Rights

### Formation of Rights

The "works" referred to in the Copyright Law include:

(VII) Graphic works and model works such as engineering design drawings, product design drawings, maps, schematics, etc.

(VIII) Computer software.

Therefore, the applications built on APaaS contain the intellectual work of the designers and builders, and there is no doubt that **the creators (natural and legal persons) are entitled to copyright. The copyright protection should refer to the application built by the user on the platform**, not the HAP platform itself. This is like the illustrator who creates with Photoshop, only enjoys the intellectual property of the illustration itself, but not the copyright of the Photoshop software, but the user who legally obtains the Photoshop software enjoys the right to use it according to the user agreement.

Specifically, the application copyright covers worksheet data structures, views, configuration of user roles and permissions, charts and workflow configurations, and aggregation of them.

## **Duty Work**

According to Article 16 of the Copyright Law, in an enterprise, applications built by employees with the APaaS platform purchased or rented by the enterprise should belong to duty works, and the creator enjoys the right of attribution as stipulated, and other copyright rights are entitled by legal persons and organizations. However, the Copyright Law allows the special case of contractual agreement, so the enterprise can establish other agreements with the creator in the labor contract or other contracts for the complete protection of relevant intellectual property rights.

## **Commissioned Creation (Outsourcing)**

Article 17 of the Copyright Law stipulates that the ownership of the copyright of a work commissioned to be created shall be agreed upon by the principal and the trustee by contract. If the contract is not explicitly agreed or no contract is concluded, the copyright belongs to the entrustee. About this point, the client side should pay special attention. If you wish to retain the copyright, you need to make additional agreement with the service provider in the contract of commissioned service.

# **10. Iteration**

HAP Server is a continuously iterative software product, with monthly updates and yearly upgrades. Monthly updates provide new features, feature improvements, and bug fixes, which are posted on the project page.

The code management of HAP Server shares the same cornerstone as HAP SaaS Edition, but with the necessary branching. Therefore, the Server Edition and SaaS Edition have almost the same features, but different means of implementation for supporting modules, such as object storage services. The update of HAP Server is slightly delayed by about two weeks from the SaaS Edition. During these two weeks, the obvious defects in the product will be fixed, so that a stable version more suitable for private deployment customers can be released.

Last update: 2024-11-12

# Available Platforms

HAP Server can be deployed on a wide range of release versions of Linux and mainstream CPU chip architectures:

Release Version	X86_64/AMD64	ARM64/AARCH64
Debian	✓ (8.2+)	✓ (10.2+)
CentOS	✓ (7.3+)	✓ (8.0+)
RedHat	✓ (7.3s+)	✓ (8.0+)
Ubuntu	✓ (14.04+)	✗
Fedora	✓ (30+)	✓ (29+)
UOS	✓ (20)	✓ (20)
Qilin	✓ (10)	✓ (10)

## Available Public Cloud Platforms

HAP Server can be deployed on all public cloud platforms.

- Microsoft AZure
- AWS
- Huawei Cloud (support Kunpeng series servers)
- Alibaba Cloud
- Tencent Cloud
- Others, such as Baidu Cloud, Jinshan Cloud, UCloud, Telecom Cloud, Mobile Cloud, Unicom Cloud, etc.

You can quickly install from the public cloud marketplace, and you need to choose the hardware configuration.

- [Alibaba Cloud Marketplace](#)

- Tencent Cloud Marketplace

# Supported Components

## Self-Built

Component	Standalone Default Version	Cluster Default Version	Supported Version	Supported Architecture	Notes	Alternative
MySQL	v5.7.37	v8.0.35	v5.7.x v8.x	Single node Master-slave	-	<a href="#">Dameng</a> <a href="#">KingbaseES</a> <a href="#">OceanBase</a> <a href="#">Xugu</a> <a href="#">HighGo</a> <a href="#">GBase</a>
MongoDB	v4.4.29	v4.4.29	v4.x v5.x	Single node Replica set	-	<a href="#">SequoiaDB</a>
Redis	v3.2.13	v6.2.16	v3.2.13+ v4.x v5.x	Single node Master-slave Sentinel	Sharding clustering is not supported.	<a href="#">TongRDS</a> <a href="#">Apusic AMI</a>
Kafka	v3.6.1	v3.6.2	v1.1.1+ v2.x v3.x	Single node Cluster	Only username and password methods are supported in authentication mode.	<a href="#">Apusic ADI</a>
Elasticsearch	v8.5.3	v8.5.3	v8.5.3+	Single node Cluster	1. The analysis-ik plugin needs to be installed; 2. SSL	-

Component	Standalone Default Version	Cluster Default Version	Supported Version	Supported Architecture	Notes	Alternativ
					certificate verification on the client and server is not supported.	
Flink	v1.17.1	v1.17.1	v1.17.1	Single node Cluster	-	-
Nginx	v1.26.1	v1.26.2	v1.16+	-	-	TongHttpS

## Cloud Product

### Microsoft AZure

Component	Version	Link	Notes
MySQL	v5.7, v8.0	<a href="https://azure.microsoft.com/en-us/products/mysql">https://azure.microsoft.com/en-us/products/mysql</a>	-
MongoDB	v4.2, v4.4, v5.0、 v6.0	<a href="https://azure.microsoft.com/en-us/solutions/mongodb">https://azure.microsoft.com/en-us/solutions/mongodb</a>	-
Redis	v4.x, v5.x, v6.x	<a href="https://azure.microsoft.com/en-us/products/cache">https://azure.microsoft.com/en-us/products/cache</a>	-
Kafka	v1.0+	<a href="https://azure.microsoft.com/en-us/products/event-hubs">https://azure.microsoft.com/en-us/products/event-hubs</a>	
Elasticsearch	-	-	-
Object Storage	-	<a href="https://azure.microsoft.com/en-us/products/storage/blobs">https://azure.microsoft.com/en-us/products/storage/blobs</a>	



Component	Version	Link	Notes
Flink	1.17.1	<a href="https://azure.microsoft.com/en-us/products/hdinsight-on-aks">https://azure.microsoft.com/en-us/products/hdinsight-on-aks</a>	

## AWS

Component	Version	Link	Notes
MySQL	v5.7, v8.0	<a href="https://aws.amazon.com/rds/mysql">https://aws.amazon.com/rds/mysql</a>	-
MongoDB	v4.0, v5.0	<a href="https://aws.amazon.com/documentdb">https://aws.amazon.com/documentdb</a>	-
Redis	v4.x, v5.x	<a href="https://aws.amazon.com/elasticache/redis">https://aws.amazon.com/elasticache/redis</a>	-
Kafka	v2.x, v3.x	<a href="https://aws.amazon.com/msk">https://aws.amazon.com/msk</a>	
Elasticsearch	-	-	-
Object Storage	-	<a href="https://aws.amazon.com/s3">https://aws.amazon.com/s3</a>	
Flink	1.17.1	<a href="https://aws.amazon.com/emr">https://aws.amazon.com/emr</a>	

## Alibaba Cloud

Component	Version	Link	Notes
MySQL	v5.7, v8.0	<a href="https://www.aliyun.com/product/rds/mysql">https://www.aliyun.com/product/rds/mysql</a>	-
MongoDB	v4.x, v5.x	<a href="https://www.aliyun.com/product/mongodb">https://www.aliyun.com/product/mongodb</a>	-
Redis	v4.x, v5.0	<a href="https://www.aliyun.com/product/kvstore">https://www.aliyun.com/product/kvstore</a>	Community Edition
Kafka	v2.2	<a href="https://www.aliyun.com/product/kafka">https://www.aliyun.com/product/kafka</a>	Professional Edition.

Component	Version	Link	Notes
			<p>Currently it does not support the automatic creation of Topics through Alibaba Cloud Kafka, you need to manually create all the Topics in advance, and select the local storage engine. (Note for later maintenance or upgrades, if new Topics are added, make sure to create them in advance.)</p> <p>The [Free Use of Group] in the Kafka Cloud Console needs to be enabled, otherwise there will be an error: Broker: Group</p>

Component	Version	Link	Notes
			authorization failed
Elasticsearch	v8.5	<a href="https://www.aliyun.com/product/bigdata/elasticsearch">https://www.aliyun.com/product/bigdata/elasticsearch</a>	General Enterprise Edition. [Automatic Index Creation] in the Cloud Console needs to be enabled.
Object Storage	-	<a href="https://www.aliyun.com/product/oss">https://www.aliyun.com/product/oss</a>	-
Flink	-	-	-

## Tencent Cloud

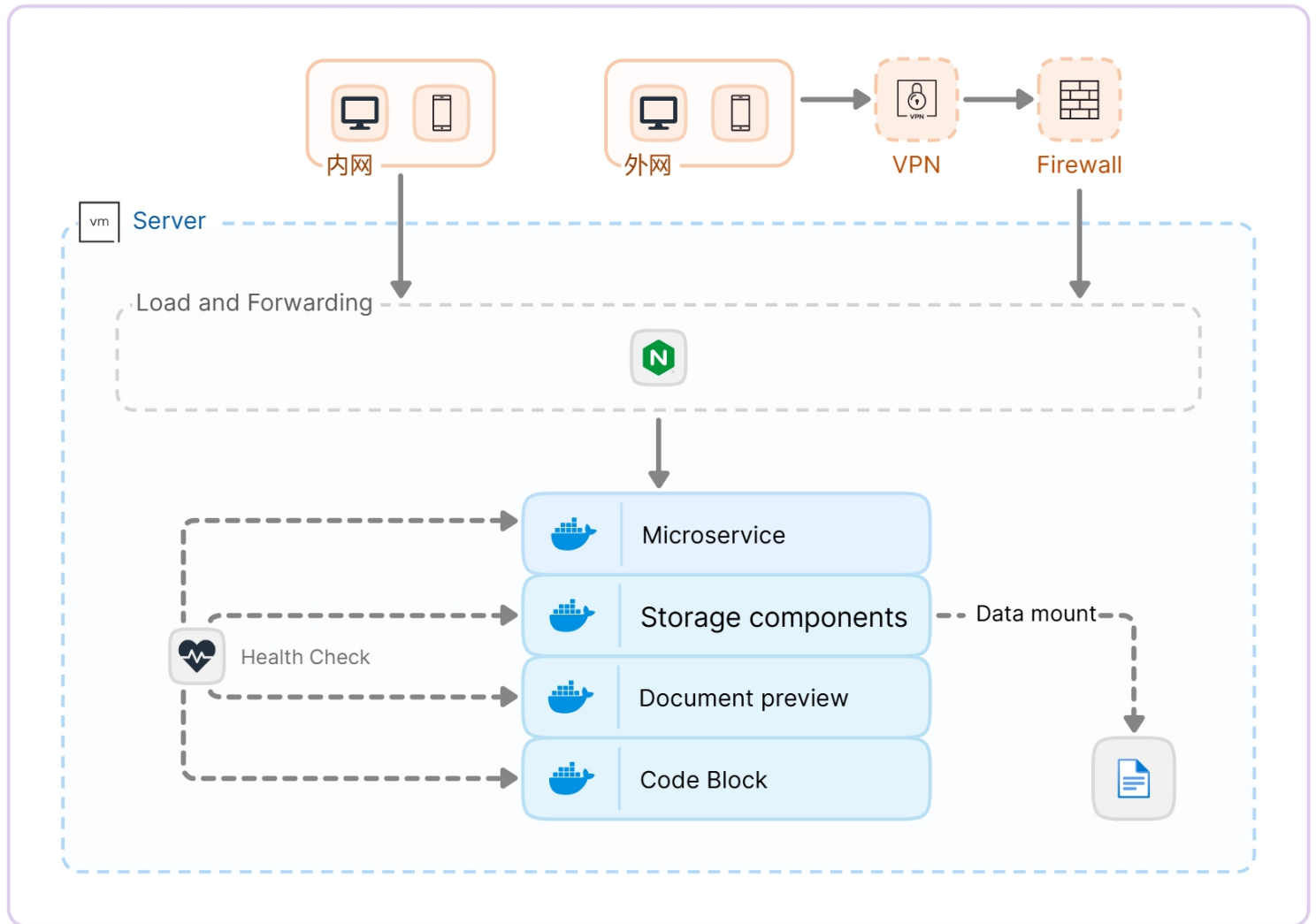
Component	Version	Link	Notes
MySQL	v5.7, v8.0	<a href="https://cloud.tencent.com/product/cdb">https://cloud.tencent.com/product/cdb</a>	-
MongoDB	v3.x, v4.x, v5.0	<a href="https://cloud.tencent.com/product/mongodb">https://cloud.tencent.com/product/mongodb</a>	-
Redis	v4.x, v5.0	<a href="https://cloud.tencent.com/product/crs">https://cloud.tencent.com/product/crs</a>	-
Kafka	v2.x, v3.x	<a href="https://cloud.tencent.com/product/ckafka">https://cloud.tencent.com/product/ckafka</a>	Professional Edition. Enable automatic creation of Topic.
Elasticsearch	v8.8.1	<a href="https://cloud.tencent.com/product/es">https://cloud.tencent.com/product/es</a>	-

Component	Version	Link	Notes
Object Storage	-	<a href="https://cloud.tencent.com/product/cos">https://cloud.tencent.com/product/cos</a>	
Flink	-	-	-

# Recommended Servers

## Standalone Deployment Mode

### Microservices



Strengths: Simple and fast deployment, low server resources and O&M costs

Weaknesses: single point of failure, but it supports health checks and usually has self-recovery capabilities

Recommended server configurations are as follows:

Concurrency	Configuration	OS
Within 100	8C/32G/40G system disk/100G data disk	Debian 10+
Within 200	16C/32G/40G system disk/100G data disk	Debian 10+
Within 300	24C/48G/40G system disk/100G data disk	Debian 10+
300 or more	Recommended cluster deployment mode	-

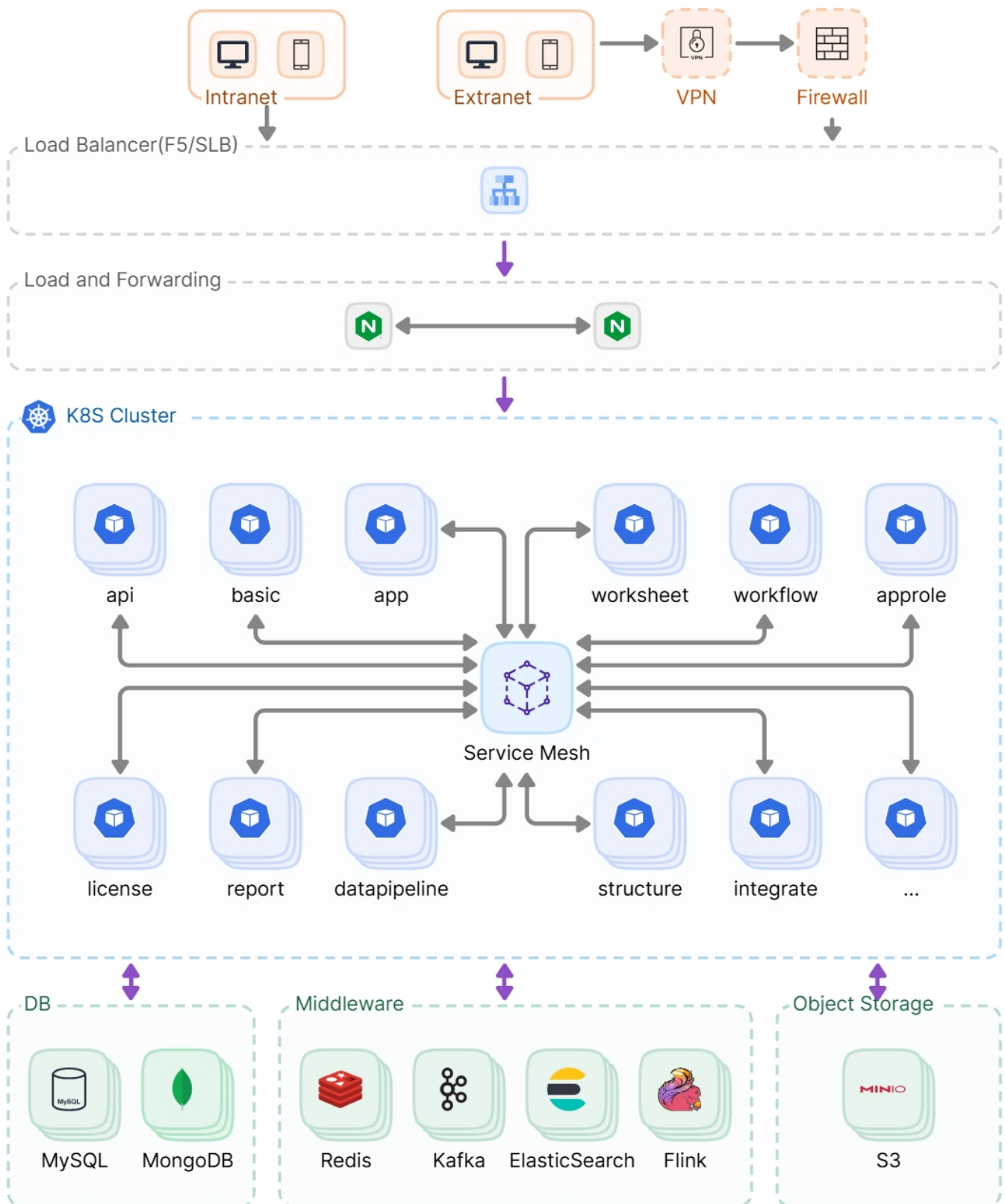
### Data Pipeline Service

Data Pipeline is an extension module in the HAP system, and users can choose whether to enable it or not. The recommended configuration for the data Pipeline server is as follows:

Number of synchronization tasks	Configuration	OS
1~20	8C/32G/40G system disk/100G data disk	Debian 10+
1~50	16C/64G/40G system disk/100G data disk	Debian 10+
50 or above	Recommend deploying in cluster mode for easy horizontal scaling	-

The larger the amount of data in the data source, the higher the required resource consumption may also need to be correspondingly increased

### Cluster Deployment Mode



**Lite Edition (Concurrency 300+)**

Strengths: save server resources and O&M costs, high availability of microservice clusters, and relatively independent deployment components, suitable for environments with limited resources or small-scale application scenarios

Weaknesses: some component coupling, possible single point of failure of storage-related services

Recommended server configurations are as follows:

Usage	Configuration	OS	Deployed Service	Number
Load and Forward	4C/8G/100G system disk/200G data disk	Debian 10+	Nginx	1
Microservice Application	16C/48G/100G system disk/200G data disk	Debian 10+	HAP Microservices	2
Middleware	8C/32G/100G system disk/500G data disk	Debian 10+	Kafka, Elasticsearch, File Object Storage	1
Data Storage	8C/32G/100G system disk/300G data disk (SSD)	Debian 10+	MySQL, MongoDB, Redis	1
Data Pipeline (optional)	8C/32G/100G system disk/200G data disk	Debian 10+	Data Pipeline Service	1

**Standard Edition (Concurrency 600+)**

Strengths: Having basic high availability capabilities, such as node redundancy and higher utilization of automatic fault tolerance, and being able to effectively utilize server resources. Suitable for most usage scenarios

Weaknesses: There is interference and resource competition between different storage components, and attention should be paid to resource allocation and management

Recommended server configurations are as follows:



Usage	Configuration	OS	Deployed Services	Number
Load and Forward	4C/8G/100G system disk/200G data disk	Debian 10+	Nginx	2
Microservice Application	16C/48G/100G system disk/200G data disk	Debian 10+	HAP Microservices	3
Cache	4C/16G/100G system disk/200G data disk	Debian 10+	Redis	2
Middleware	8C/32G/100G system disk/500G data disk	Debian 10+	Kafka, Elasticsearch, File Object Storage	4
Database	8C/32G/100G system disk/300G data disk (SSD)	Debian 10+	MySQL, MongoDB	3
Data Pipeline (optional)	8C/32G/100G system disk/200G data disk	Debian 10+	Data Pipeline Service	2

**Professional Edition (Concurrency 1000+)**

Strengths: At the same time of high availability, a single application architecture is adopted (only one component instance runs on each server) to improve reliability and performance, avoid interference and resource competition between different components, and also facilitate independent expansion and management of each component

Weaknesses: high server resources and O&M costs

Recommended server configurations are as follows:

Usage	Configuration	OS	Deployed Service	Number
Load and Forward	4C/8G/100G system disk/200G data disk	Debian 10+	Nginx	2

Usage	Configuration	OS	Deployed Service	Number
Microservice Application	32C/64G/100G system disk/300G data disk	Debian 10+	HAP Microservices	3
Cache	16C/32G/100G system disk/200G data disk	Debian 10+	Redis	2
Message queue	16C/32G/100G system disk/500G data disk	Debian 10+	Kafka	3
Full-text search	16C/32G/100G system disk/500G data disk	Debian 10+	Elasticsearch	3
File Storage	16C/32G/100G system disk/500G data disk	Debian 10+	File Object Storage	4
RDB	16C/32G/100G system disk/500G data disk(SSD)	Debian 10+	MySQL	2
NoSQL	32C/64G/100G system disk/500G data disk(SSD)	Debian 10+	MongoDB	3
Data Pipeline (optional)	16C/64G/100G system disk/200G data disk	Debian 10+	Data Pipeline Service	2

## Basic performance requirements for servers

### CPU

It is recommended to choose a CPU with a clock speed of 2.5 GHz or higher. High frequency can provide faster computation and response speed.

### Disk

For server disk performance, we have established the following minimum requirements. It is strongly recommended to configure disks that meet or exceed these standards; otherwise, business operations may be negatively affected.

Furthermore, mechanical hard drives are not recommended, as their IOPS typically reach only around 100, which falls far short of our minimum performance requirements and can significantly hinder business performance.

- Random write performance: IOPS  $\geq$  2000
- Random read performance: IOPS  $\geq$  2000
- Sequential write speed:  $\geq$  100 MiB/s
- Sequential reading speed:  $\geq$  100 MiB/s

## SSD hard drive

To evaluate the performance of SSD hard drives, we used common mid-range SSD hard drives provided by cloud server vendors as our testing benchmark.

The following are the benchmark indicators we have set, however, we strongly recommend that the actual hard drive configuration should not be lower than the following values. Of course, if we could surpass these values, it would be even more ideal.

- Random write performance: IOPS  $\geq$  15000
- Random read performance: IOPS  $\geq$  15000
- Sequential write speed:  $\geq$  200 MiB/s
- Sequential reading speed:  $\geq$  200 MiB/s

## Cluster Mode - Internal Network

In order to ensure the stability and efficient performance of the system under various workloads, we have set the following basic performance requirements for the internal network:

Internal packet sending and receiving rate (PPS): Each server should reach at least 200000.

Internal network bandwidth: The internal network bandwidth of each server should not be less than 2 Gbps.

These requirements help ensure that the system can maintain stable and efficient communication performance even under high load conditions.

# Bandwidth requirements

Bandwidth recommendation is **30M+**. If file storage related functions are frequently used, it may need to be expanded according to the actual situation

Accelerated access: The larger the bandwidth, the faster the data transmission speed, and the smoother the user access experience.

Supporting more users: High bandwidth can effectively avoid lagging or latency in high concurrency scenarios, supporting more concurrent access.

Coping with peak traffic: During peak traffic periods, sufficient bandwidth can ensure the normal operation of servers and will not be affected by traffic pressure.

Charged by data, higher cost-effectiveness: For cloud servers, it is recommended to purchase bandwidth. Charged by data is a flexible and cost-effective solution that can ensure speed and meet business needs of different scales.

## Example of bandwidth calculation

Assuming a bandwidth of 30Mbps is selected, if 30 people are downloading simultaneously, the bandwidth allocation is as follows:

Total bandwidth: 30Mbps=30000000 bits/second

Bandwidth allocation per user:  $30000000 \div 30 = 1000000$  bits/second

Download speed per user:  $1000000 \div 8 = 125000$  bytes per second  $\approx 125\text{KB/s}$

# Parameters for Environment Detection

*Take a CentOS 7.9 server as an example*

## User

Description: The user when running the script

Requirement: To root a user or a user with root permissions is required because docker is in use.

## Selinux

Description: The security system of Linux

Requirement: Selinux should be disabled because docker is in use.

Temporarily disabled: ``setenforce 0`

Permanently disabled: modify `SELINUX=enforcing` or `SELINUX=permissive` to `SELINUX=disabled` in `/etc/selinux/config`

## Firewalld

Note: It is a self-contained firewall tool. It is recommended to disable the firewall as it tends to conflict with iptables rules. Cloud servers can rely on security group rules for network control.

Temporarily disabled: `systemctl stop firewalld` (you need to restart the docker service after shutting down the firewall, otherwise the rules may be lost and lead to network anomalies)

Permanently disabled: `systemctl disable firewalld`

## CPU

Description: Processor

Requirements: Minimum 4 cores

# Memory

Description: Memory

Requirement: Minimum 32G memory capacity

## **docker**

Description: The docker command must be available in the environment variable and can be executed successfully.

Requirement: It will be added to the PATH environment variable automatically after installation via RPM. If it is a binary installation, you need to add it to the environment variable manually.

## **dockerd**

Note: The dockerd command must be available in the environment variable and can be executed successfully.

Requirement: It will be added to the PATH environment variable automatically after installation via RPM. If it is a binary installation, you need to add it to the environment variable manually.

## MaxMapCount

Description: It limits the number of VMAs (Virtual Memory Areas) that a process can have, and the built-in Elasticsearch needs to be adjusted to start successfully.

Temporary adjustment: `sysctl -w vm.max_map_count=262144`

Permanent adjustment: add the file `vm.max_map_count=262144` to `/etc/sysctl.conf`

## SysFileNr

Description: OS-level file descriptor limit

Temporary adjustment: `sysctl -w fs.file-max=2048000`

Permanent adjustment: add the file `fs.file-max=2048000` to `/etc/sysctl.conf`

## IPv4Forward

Note: It needs to be enabled for docker to provide external services.

Temporary adjustment: `sysctl -w net.ipv4.ip_forward=1`

Permanent adjustment: add the file `net.ipv4.ip_forward=1` to `/etc/sysctl.conf`

## DockerCgroupDrive

Description: The limit of Cgroup file descriptors in docker containers. If docker version is 20.10.16 or newer, it defaults to cgroupfs; if it is systemd, there may be insufficient file descriptors; if dockerd is not running, you need to start dockerd (systemctl start docker).

Permanent adjustment:

1. Upgrade Docker to version 20.10.16 or newer
2. If the check environment is still systemd after upgrade, you need to modify or add the configuration item "exec-opts": ["native.cgroupdriver=cgroupfs"] in `/etc/docker/daemon.json`

## DockerdFileNr

Description: The number of file descriptors in docker. If dockerd is not running, you need to start dockerd (systemctl start docker).

Permanent adjustment: For example, if you start the docker service via systemd, set LimitNOFILE=102400 in the file `docker.service`.

# Security Protection

## IMPORTANT REMINDER

To ensure system security, please pay close attention to and implement the following points. ⚠️⚠️⚠️

## Limit unnecessary access

Limit unnecessary access to ensure server security.

The HAP system deployment relies on a management tool that listens on port 38881 by default. This port is used for initial installation and provides functions like online upgrades and system restarts. Generally, except for system administrators, port 38881 should not be accessed by others. Therefore, after deployment, it is recommended to set an access policy for port 38881.

If external client software needs to connect to and access the storage components, ensure that exposed ports are controlled via a whitelist, especially in cases where cloud servers are deployed. Exposing commonly used ports (e.g., MySQL: 3306, MongoDB: 27017) to the internet, combined with weak passwords, can make the system highly insecure. In such cases, not only is data leakage likely, but the system is also vulnerable to hacker attacks. Hackers may delete database data (often leaving a README file asking for payment to recover the data).

## Use Strong Passwords

Regularly change complex passwords to prevent them from being cracked.

This applies to server authentication passwords and those used to connect to storage components, such as MySQL, MongoDB, Redis, etc. It is also important to strengthen protection for any externally exposed ports. In single-node deployment mode, refer to the [Database Strong Password Configuration](#) for guidance.

## Encrypt Data

Enable HTTPS and other encryption protocols to prevent data from being intercepted during transmission.

## Regular Backups



Automate the backup of system data to ensure it is recoverable and to prevent data loss.

In single-node deployment mode, refer to [HAP's backup method](#) for backup procedures. Additionally, it is recommended to take regular snapshots of the server.

In cluster deployment mode, regularly back up the data directories of both the **data storage servers** and **middleware servers**. It is also recommended to take regular snapshots of these servers. If the system is deployed by the HAP implementation team, the delivery documentation provided by HAP will specify the data directories.

## Vulnerability Patching

Update the system promptly and regularly apply patches to reduce security risks.

## Monitoring and Alerts

Configure anomaly detection and alerting systems to promptly detect potential threats and take necessary actions.

## DDoS Protection

If possible, enable DDoS protection to prevent malicious traffic attacks, ensuring better business continuity.

# Common Commands

## Start manager

```
bash ./service.sh start
```

## Start manager and HAP service

```
bash ./service.sh startall
```

## Stop manager and HAP service

```
bash ./service.sh stopall
```

## Restart manager and HAP service

```
bash ./service.sh restartall
```

## Enter the microservice application container

### Docker Compose

```
docker exec -it $(docker ps | grep community | awk '{print $1}') bash
```

### kubernetes

```
# Take the Default Namespace as an example, according to the actual pod name to  
modify hap-0.  
kubectl exec -it hap-0 bash
```

# Logs

## View health-checking logs of the microservice application

### Docker Compose

```
docker logs $(docker ps | grep community | awk '{print $1}')
```

### kubernetes

```
# Take the Default Namespace as an example, according to the actual pod name to  
modify hap-0.
```

## View running logs of the microservices application (abnormal or non-abnormal)

1. Enter the microservice application container
2. Execute the command to view the logs

```
source /entrypoint.sh && log log
```

# Environment Variable

The meanings of the environment variables used in the HAP Server container are as follows:

Environment Variable	Description
ENV_SERVERID	Instance number. Each instance cannot be the same (only)
ENV_MINGDAO_PROTO	Protocol. Support setting http or https
ENV_MINGDAO_HOST	Access address, such as hap.domain.com (if you configure a proxy first)

Environment Variable	Description
ENV_MINGDAO_PORT	Port, such as 80
ENV_MINGDAO_SUBPATH	Subpath, such as <code>https://www.domain.com/h</code>
ENV_MINGDAO_INTRANET_ENDPOINT	The corresponding intranet address of <code>ENV_MINGDAO_PORT</code> , such as <code>192.168.1.1:8880</code> , which needs to be configured
ENV_MINGDAO_WORKWXAPI	WeCom API interface host. It defaults to <code>https://</code> , using the private deployment version of WeCom, such as <code>https://work.weixin.qq.com/api</code>
ENV_MINGDAO_FEISHUAPI	FeiShu API interface host. It defaults to <code>https://</code> , using the private deployment version of FeiShu, such as <code>https://open.feishu.cn/open-apis</code>
ENV_DOCPREVIRE_ENDPOINTS	The address of the file preview service. It defaults to <code>https://</code> . If there are multiple instances in cluster mode, separate them with <code>,</code>
ENV_WEB_ENDPOINTS	Web front-end site address (required in secondary deployment). It defaults to <code>https://</code> . If there are multiple instances in cluster mode, separate them with <code>,</code> , such as <code>192.168.1.1:81,192.168.1.1:82</code>
ENV_MONGODB_URI	MongoDB connection address, such as <code>mongodb://192.168.1.1:27017,192.168.1.1:27018</code>
ENV_MONGODB_OPTIONS	MongoDB uri parameters, starting with <code>[?]</code>
ENV_MONGODB_CACHEGB	MongoDB's allowed maximum memory usage
ENV_MYSQL_HOST	MySQL address, such as <code>192.168.2.1</code> [VIP]
ENV_MYSQL_PORT	The MySQL port. It defaults to 3306
ENV_MYSQL_USERNAME	MySQL username. It defaults to root
ENV_MYSQL_PASSWORD	MySQL password. It can be empty and defaults to empty
ENV_REDIS_HOST	[Redis master-slave or standalone mode] Redis address

Environment Variable	Description
ENV_REDIS_PORT	[Redis master-slave or standalone mode] Redis port
ENV_REDIS_PASSWORD	[Redis master-slave or standalone mode] Redis password
ENV_REDIS_MAXMEMORY	[Redis Standalone Mode] Maximum allowed memory. Redis uses the LRU algorithm to clean it up. No limit by default.
ENV_REDIS_SENTINEL_ENDPOINTS	[Redis Sentinel mode] Sentinel addresses. If there are multiple, use English commas.
ENV_REDIS_SENTINEL_MASTER	[Redis Sentinel Mode] Master name
ENV_REDIS_SENTINEL_PASSWORD	[Redis Sentinel Mode] Connection password
ENV_KAFKA_ENDPOINTS	Kafka connection addresses, such as <code>192.168.1.4:9092,192.168.1.5:9092,192.168.1.6:9092</code>
ENV_KAFKA_SECURITY_PROTOCOL	Kafka security protocol, supports <code>Plaintext</code> (default) and <code>SSL</code>
ENV_KAFKA_SASL_MECHANISM	Only used when <code>ENV_KAFKA_SECURITY_PROTOCOL=SSL</code> . Value: <code>PLAIN</code>
ENV_KAFKA_SASL_USERNAME	Only used when <code>ENV_KAFKA_SECURITY_PROTOCOL=SSL</code> . Username
ENV_KAFKA_SASL_PASSWORD	Only used when <code>ENV_KAFKA_SECURITY_PROTOCOL=SSL</code> . Password
ENV_ELASTICSEARCH_ENDPOINTS	Elasticsearch connection addresses, such as <code>192.168.1.4:9200,192.168.1.5:9200,192.168.1.6:9200</code>
ENV_ELASTICSEARCH_PASSWORD	Elasticsearch connection authentication. It can be <code>username:password</code>
ENV_FILE_ENDPOINTS	File storage service addresses, such as



Environment Variable	Description
ENV_WORKFLOW_COMMAND_TIMEOUT	The timeout for the execution of the code block in seconds, default: 10
ENV_WORKFLOW_COMMAND_MAXMEMORY	The maximum memory for the code block execution, default: 1024MB
ENV_WORKFLOW_GRPC_TIMEOUT	The timeout duration for calling other service interfaces, default: 180, unit: seconds
ENV_WORKFLOW_PARALLELISM_THREADS	The size of the parallel consumption thread pool, default: 10
ENV_WORKFLOW_TRIGGER_DELAY_SECONDS	The delay of the workflow triggered by the worksheet, default: 0
ENV_WORKFLOW_IP_BLOCKLIST	Workflow sending custom requests, Data pipeline blacklisting, if more than one, separated by English commas
ENV_WORKSHEET_EXCEL_IMPORT_THREADS	Processing threads for importing Excel into worksheet, default: 2
ENV_WORKSHEET_REFRESH_ROWS_MINUTES	Calibration interval for data in worksheet, default: 10
ENV_WORKSHEET_CONSUMER_THREADS	Consumer threads of worksheet, default: 2
ENV_SESSION_TIMEOUT_MINUTES	Expiration time of session, in minutes, default: 10
ENV_SESSION_DISABLE_REFRESH	Whether to automatically refresh the session's expiration time, default: true <code>ENV_SESSION_TIMEOUT_MINUTES</code>
ENV_SESSION_PORTAL_TIMEOUT_MINUTES	Expiration time of sessions in external portal, in minutes, default: 10
ENV_WPS_CONVERT_APPID	WPS PDF Conversion Service appId
ENV_WPS_CONVERT_APPSECRET	WPS PDF Conversion Service appSecret
ENV_WPS_PREVIEW_APPID	WPS Document Editing Preview Service appId
ENV_WPS_PREVIEW_APPSECRET	WPS Document Editing Preview Service appSecret



Environment Variable	Description
ENV_OCR_SENDFILE	Whether to send files in base64 encoding. Default is false.
ENV_OCR_API_PROXY	Whether to enable forward proxy calls to the internal API.
ENV_SOCKET_POLLING	Whether to use polling mode for socket connections.

# Offline Resources



**TIP**

- After downloading the offline mirror image file and uploading it to the corresponding deployment server, you must import it to make it usable.
- For specific import steps, please refer to the examples provided at the bottom of this document.

**AMD64**

**ARM64**

Name	Download
hap-community Microservice	<a href="https://pdpublic.nocoly.com/offline/hap-community-linux-amd64-6.0.2.tar.gz">https://pdpublic.nocoly.com/offline/hap-community-linux-amd64-6.0.2.tar.gz</a>
hap-file File storage	<a href="https://pdpublic.nocoly.com/offline/hap-file-linux-amd64-1.6.0.tar.gz">https://pdpublic.nocoly.com/offline/hap-file-linux-amd64-1.6.0.tar.gz</a>
hap-flink Data pipeline	<a href="https://pdpublic.nocoly.com/offline/hap-flink-linux-amd64-1.17.1.530.tar.gz">https://pdpublic.nocoly.com/offline/hap-flink-linux-amd64-1.17.1.530.tar.gz</a>
hap-doc Document preview, based on OnlyOffice	<a href="https://pdpublic.nocoly.com/offline/hap-doc-linux-amd64-1.2.0.tar.gz">https://pdpublic.nocoly.com/offline/hap-doc-linux-amd64-1.2.0.tar.gz</a>
hap-ldoc Document preview, based on LibreOffice	<a href="https://pdpublic.nocoly.com/offline/hap-ldoc-linux-amd64-1.0.0.tar.gz">https://pdpublic.nocoly.com/offline/hap-ldoc-linux-amd64-1.0.0.tar.gz</a>
hap-sc Storage components	<a href="https://pdpublic.nocoly.com/offline/hap-sc-linux-amd64-3.0.0.tar.gz">https://pdpublic.nocoly.com/offline/hap-sc-linux-amd64-3.0.0.tar.gz</a>
hap-sc-upgrade Storage component upgrade secondary mirror	<a href="https://pdpublic.nocoly.com/offline/hap-sc-upgrade-linux-amd64-1.0.0.tar.gz">https://pdpublic.nocoly.com/offline/hap-sc-upgrade-linux-amd64-1.0.0.tar.gz</a>

Name	Download
hap-command Code Block	<a href="https://pdpublish.nocoly.com/offline/hap-command-linux-amd64-node2011-python312.tar.gz">https://pdpublish.nocoly.com/offline/hap-command-linux-amd64-node2011-python312.tar.gz</a>
hap-dataclean-app Application data clean	<a href="https://pdpublish.nocoly.com/offline/hap-dataclean-app-linux-amd64-1.0.0.tar.gz">https://pdpublish.nocoly.com/offline/hap-dataclean-app-linux-amd64-1.0.0.tar.gz</a>
hap-archivetools Log data archive	<a href="https://pdpublish.nocoly.com/offline/hap-archivetools-linux-amd64-1.0.4.tar.gz">https://pdpublish.nocoly.com/offline/hap-archivetools-linux-amd64-1.0.4.tar.gz</a>
ops-gateway Ops Platform - Gateway	<a href="https://pdpublish.nocoly.com/offline/ops-gateway-linux-amd64-1.0.0.tar.gz">https://pdpublish.nocoly.com/offline/ops-gateway-linux-amd64-1.0.0.tar.gz</a>
ops-prometheus Ops Platform - Prometheus Component	<a href="https://pdpublish.nocoly.com/offline/ops-prometheus-linux-amd64-1.0.0.tar.gz">https://pdpublish.nocoly.com/offline/ops-prometheus-linux-amd64-1.0.0.tar.gz</a>
ops-agent Ops Platform - Component monitoring data collection	<a href="https://pdpublish.nocoly.com/offline/ops-agent-linux-amd64-1.0.0.tar.gz">https://pdpublish.nocoly.com/offline/ops-agent-linux-amd64-1.0.0.tar.gz</a>
ops-nodeagent Ops Platform - Host monitoring data collection	<a href="https://pdpublish.nocoly.com/offline/ops-nodeagent-linux-amd64-1.0.0.tar.gz">https://pdpublish.nocoly.com/offline/ops-nodeagent-linux-amd64-1.0.0.tar.gz</a>

**After downloading and uploading the image package to the corresponding deployment server, import the image using the following command.**

**Docker**

Kubernetes

1. Load image

```
docker load -i xxx.tar.gz
```

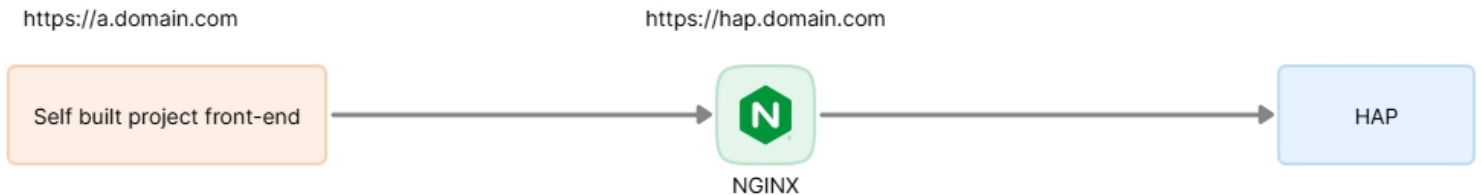
2. View image

docker images

# Cross-origin Resource Sharing

For some scenarios where self-built front-end projects are combined with HAP, since there are cross-domain issues, the front-end can't call the API of HAP system directly by default.

As shown below, the HTTP header parameters to allow cross-domain requests need to be set in the proxy tier configuration file (**additional deployment is required if there is no proxy tier. View more details in [Proxy Configuration](#)**).



An example configuration is shown below:

```
location / {  
  
    add_header Access-Control-Allow-Origin 'Addresses that allow cross-domain  
access, such as https://a.domain.com. Separate with English commas if multiple';  
    add_header Access-Control-Allow-Credentials 'true';  
    add_header Access-Control-Allow-Methods 'GET, POST, OPTIONS';  
    add_header Access-Control-Allow-Headers 'Adjust it according to the Header  
parameters passed by the front-end, e.g. X-Requested-With, Authorization';  
  
    if ($request_method = 'OPTIONS') {  
        return 204;  
    }  
    .....  
}
```

## ⚠ CAUTION

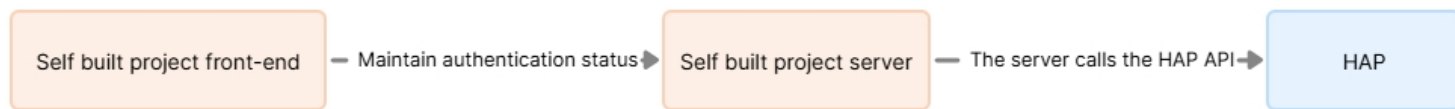
The above approach is not recommended if want to ensure the security of interface calls.

HAP has opened many standard APIs, such as [API Development Documentation] module for each application, and the interface calls are based on the authentication mode of `AppKey` and `Sign`.

If the front-end calls directly in the above way, it means that the `AppKey`, `Sign` and other information is made public. (Any user who can access the system can view the front-end code,) and with the values of these parameters, they can call the interface to get the data.

The recommended method is shown below:

`https://a.domain.com`



The front-end of the self-built project does not directly call the API of HAP, but through the API provided by the server side of the self-built project. The front-end of the self-built project maintains an authentication with the server side (e.g., it needs login authentication) and finally interacts with the API of HAP through the server side. The related `AppKey` and `Sign` will be configured in the server side of the self-built project, which is not visible to the front-end at all, so as to ensure the security of the interface calls.

# Proxy Configuration

To ensure that your HAP system is not directly exposing service ports to the internet, we strongly recommend further configuring Nginx proxy after deploying the system. This can greatly enhance the security of the system and meet the needs of users requiring certificates, who can refer to relevant documentation for configuration. Additionally, Nginx proxy can also provide load balancing and reverse proxy, improving the system's availability and stability.

1. Download the Nginx installation package.

Internet Access Available

Internet Access Unavailable

```
wget https://pdpublish.nocoly.com/offline/common/nginx-1.26.2-glibc2.17-  
amd64.tar.gz
```

2. Unpack the Nginx installation package to the installation directory.

```
tar -zxvf nginx-1.26.2-glibc2.17-amd64.tar.gz -C /usr/local/
```

3. Create storage directories for configuration and logs.

```
mkdir -p /usr/local/Nginx/conf/conf.d /data/logs/weblogs/
```

4. Write the main Nginx configuration file.

```
cat > /usr/local/Nginx/conf/Nginx.conf <<EOF  
user  nobody;  
worker_processes  auto;  
worker_cpu_affinity auto;  
worker_rlimit_nofile 204800;  
pid      Nginx.pid;  
events {  
    use epoll;  
    worker_connections 20480;  
}  
http {  
    include      mime.types;
```

```

default_type application/octet-stream;
server_tokens off;


log_format main "\$http_x_forwarded_for | \$time_local | \$request |
\$status | \$body_bytes_sent | "
                "\$request_body | \$content_length | \$http_referer |
\$http_user_agent | "
                "\$http_cookie | \$remote_addr | \$hostname |
\$upstream_addr | \$upstream_response_time | \$request_time";


server_names_hash_bucket_size 128;
client_header_buffer_size 8k;
client_max_body_size 10M;
large_client_header_buffers 4 32k;
sendfile on;
tcp_nopush on;
tcp_nodelay on;
proxy_buffer_size 64k;
proxy_buffers 4 128k;
keepalive_timeout 10;
open_file_cache max=102400 inactive=60s;
open_file_cache_valid 30s;
open_file_cache_min_uses 1;
resolver_timeout 10s;
underscores_in_headers on;


gzip on;
gzip_proxied any;
gzip_disable "msie6";
gzip_vary on;
gzip_min_length 1024;
gzip_comp_level 8;
gzip_buffers 16 8k;
gzip_types text/plain text/css application/json application/x-javascript
application/javascript text/xml application/xml application/xml+rss
text/javascript image/jpeg image/gif image/png;
proxy_http_version 1.1;
include conf.d/*.conf;
}
EOF

```

5. Configure the host proxy file.(Put the following configuration files in the directory /usr/local/Nginx/conf/conf.d/)

## HTTP Configuration File



## HTTPS Configuration File

### 6. Start Nginx.

Check the format of the Nginx configuration file.

```
/usr/local/Nginx/sbin/Nginx -t
```

Start Nginx.

```
/usr/local/Nginx/sbin/Nginx
```

### 7. Set up auto-start on boot.

```
echo "/usr/local/Nginx/sbin/Nginx" >> /etc/rc.local  
chmod +x /etc/rc.d/rc.local
```

## Scheduled rotation of Nginx logs

### 1. Create directories for configuration files and old logs.

```
mkdir -p /usr/local/logrotate-config  
mkdir -p /data/logs/weblogs/oldlogs
```

### 2. Create a configuration file.

```
cat > /usr/local/logrotate-config/Nginx <<EOF  
/data/logs/weblogs/*.log {  
  create 0664 nobody root  
  daily  
  dateext  
  dateformat -%Y-%m-%d  
  dateyesterday  
  rotate 180  
  missingok  
  ifempty  
  compress  
  delaycompress
```

```
olddir /data/logs/weblogs/oldlogs
sharedscripts
postrotate
    /bin/kill -USR1 \`cat /usr/local/Nginx/Nginx.pid 2>/dev/null\`
2>/dev/null || true
endscript
}
EOF
```

3. Check the configuration file.

```
logrotate -d -f /usr/local/logrotate-config/Nginx
```

- Pay attention to debug outputs; further action is needed if errors are encountered.

4. Execute `crontab -e` and add the following scheduled tasks to the configuration.

```
# Use Logrotate Cut Nginx Logs
0 0 * * * /usr/sbin/logrotate -f /usr/local/logrotate-config/Nginx
>/dev/null 2>&1
```

# HTTP

## NOTE

Due to the characteristics of HAP service, the configuration file `docker-compose.yml` needs to be modified additionally after the access address of the system is changed.

As shown below, change the environment variable value of `ENV_ADDRESS_MAIN` to the actual access address of HAP service, and restart HAP service to take effect.

```
app:
  image: registry.cn-hangzhou.aliyuncs.com/mdpublic/mingdaoyun-community
  environment:
    ENV_ADDRESS_MAIN: "http://mdy.domain.com:80"
```

If your environment was deployed earlier, there may not be the `ENV_ADDRESS_MAIN`. Similarly, changing the environment variable values of `ENV_MINGDAO_PROTO`, `ENV_MINGDAO_HOST`, and `ENV_MINGDAO_PORT` is the same.

## Configuration

If you need to proxy the intranet address of the back-end HAP service, you can refer to the configuration file for nginx http reverse proxy below.

```
upstream hap {
    server Server IP:8880; # Modify it to the intranet IP and port of your HAP
    system
}

server {
    listen 80;
    server_name hap.domain.com; # Modify it to your HAP system access address
    access_log /data/logs/weblogs/hap.domain.com.log main; # Customizable log
    path
    error_log /data/logs/weblogs/hap.domain.com.error.log; # Customizable log
    path

    underscores_in_headers on;

    # Size limit for uploading files
```

```

client_max_body_size    2048m;

# Enable browser compression to speed up requests
gzip on;
gzip_proxied any;
gzip_disable "msie6";
gzip_vary on;
gzip_min_length 512;
gzip_comp_level 6;
gzip_buffers 16 8k;
gzip_types text/plain text/css application/json application/x-javascript
application/javascript application/octet-stream text/xml application/xml
application/xml+rss text/javascript image/jpeg image/gif image/png;

location / {
    set $real_ip '';
    if ($http_x_real_ip) {
        set $real_ip $http_x_real_ip;
    }
    if ($http_x_real_ip = '') {
        set $real_ip $remote_addr;
    }
    proxy_set_header X-Real-IP $real_ip;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass http://hap;
}

# IM Requirements
location ~ /mds2 {
    proxy_set_header Host $http_host;
    proxy_hide_header X-Powered-By;
    proxy_set_header X-NginX-Proxy true;
    proxy_pass http://hap;
    proxy_redirect off;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection upgrade;
}
}

```

If you need to keep the original access to the system and expect it to be accessible via the new address as well, refer to [Multiple Access Address Configuration](#).

# HTTPS

## NOTE

Due to the characteristics of HAP service, the configuration file `docker-compose.yml` needs to be modified additionally after the access address of the system is changed.

As shown below, change the environment variable value of `ENV_ADDRESS_MAIN` to the actual access address of HAP service, and restart HAP service to take effect.

```
app:
  image: registry.cn-hangzhou.aliyuncs.com/mdpublic/mingdaoyun-community
  environment:
    ENV_ADDRESS_MAIN: "https://mdy.domain.com:443"
```

If your environment was deployed earlier, there may not be the `ENV_ADDRESS_MAIN`. Similarly, changing the environment variable values of `ENV_MINGDAO_PROTO`, `ENV_MINGDAO_HOST`, and `ENV_MINGDAO_PORT` is the same.

## Configuration

If you want to configure https domain access, you need to add another layer of proxy ( like nginx ) and configure the certificate, and then proxy to the intranet address of the back-end HAP service.

For reference, the following is an example of nginx, configuring an https reverse proxy.

```
upstream hap {
    server Server IP:8880; # Modify it to the intranet IP and port of your HAP
    system
}

# Force jump to https access
server {
    listen 80;
    server_name hap.domain.com; # Modify it to your HAP system access address
    rewrite ^(.*)$ https://$host$1 permanent;
}

server {
    listen 443 ssl;
```

```
server_name hap.domain.com; # Modify it to your HAP system access address
access_log /data/logs/weblogs/hap.domain.com.log main; # Customizable log
path
error_log /data/logs/weblogs/hap.domain.com.error.log; # Customizable log
path

ssl_certificate          /etc/cert/fullchain.pem; # Modify it to the file
path of SSL certificate
ssl_certificate_key       /etc/cert/privkey.pem; # Modify it to the file path
of the SSL certificate private key

underscores_in_headers on;

# Size limit for uploading files
client_max_body_size     2048m;

# Enable browser compression to speed up requests
gzip on;
gzip_proxied any;
gzip_disable "msie6";
gzip_vary on;
gzip_min_length 512;
gzip_comp_level 6;
gzip_buffers 16 8k;
gzip_types text/plain text/css application/json application/x-javascript
application/javascript application/octet-stream text/xml application/xml
application/xml+rss text/javascript image/jpeg image/gif image/png;

location / {
    set $real_ip '';
    if ($http_x_real_ip) {
        set $real_ip $http_x_real_ip;
    }
    if ($http_x_real_ip = '') {
        set $real_ip $remote_addr;
    }
    proxy_set_header X-Real-IP $real_ip;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass http://hap;
}

# IM Requirements
location ~ /mds2 {
    proxy_set_header Host $http_host;
```

```
    proxy_hide_header X-Powered-By;  
    proxy_set_header X-NginX-Proxy true;  
    proxy_pass http://hap;  
    proxy_redirect off;  
    proxy_http_version 1.1;  
    proxy_set_header Upgrade $http_upgrade;  
    proxy_set_header Connection upgrade;  
}  
}
```

If you need to keep the original access to the system and expect it to be accessible via the new address as well, refer to [Multiple Access Address Configuration](#).

# Multiple Access Address Configuration

## Dual Address

By default, an access exception occurs if the second access address of the system is directly proxy forwarded to the current original system address.

Due to the characteristics of HAP service, if you need to configure the second access address of the system, you need to forward the request to port 18880 in the container and specify the second access address in the configuration file before you can use the system through the second access address normally.

## Configuration

1. Modify the file `docker-compose.yml`.

Add a new variable under `environment` to specify the second access address.

```
ENV_EXT_MINGDAO_PROTO: "https"
ENV_EXT_MINGDAO_HOST: "hap2.domain.com"
ENV_EXT_MINGDAO_PORT: "443"
```

Add port mapping under `ports` to map port 18880 out of the container.

```
- 18880:18880
```

▶ [docker-compose.yml Example of Configuration File](#)

2. Restart the HAP service in the directory of Install Manager to take effect.

```
bash ./service.sh restartall
```

3. Forward the second access address request to port **18880** of the server.

## Multiple Address



If the system is to be configured with three, four, or more access addresses, the dual-address configuration method described above is not applicable, and you can refer to the following method of adding the `pdaddr` header to realize multi-address access to the system through the nginx proxy (Microservice Version: v3.5.0+).

Note: When configuring multiple addresses in this way, the configuration file `dockerc-compose.yml` cannot contain `ENV_EXT_MINGDAO_PROTO`, `ENV_EXT_MINGDAO_HOST` or `ENV_EXT_MINGDAO_PORT`.

## Configuration

1. Add port mapping under the app service `ports` in `docker-compose.yml` to map the **18880** port in the container.

```
- 18880:18880
```

2. Add a variable in the app service environment with the value of your extended access address. Separate multiple values with commas

```
ENV_ADDRESS_ALLOWLIST: "https://hap2.domain.com"
```

3. Restart the HAP service in the directory of Install Manager to take effect.

```
bash ./service.sh restartall
```

4. Configure the file for nginx. Reverse proxy the new access address to port **18880** of HAP microservice.

The reverse proxy configuration file for nginx is as follows:

- HTTP
- HTTPS

5. Add `proxy_set_header pdaddr` under `location` in the configuration file of nginx to specify the system access address.

Example:

```

location / {
    set $real_ip '';
    if ($http_x_real_ip) {
        set $real_ip $http_x_real_ip;
    }
    if ($http_x_real_ip = '') {
        set $real_ip $remote_addr;
    }
    proxy_set_header X-Real-IP $real_ip;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_pass http://hap;
    proxy_set_header pdaddr https://hap2.domain.com; # New, please note to
modify to your actual extended access address
}

# IM Requirements
location ~ /mds2 {
    proxy_set_header Host $http_host;
    proxy_hide_header X-Powered-By;
    proxy_set_header X-NginX-Proxy true;
    proxy_pass http://hap;
    proxy_redirect off;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection upgrade;
    proxy_set_header pdaddr https://hap2.domain.com; # New, please note to
modify to your actual extended access address
}

```

6. After reloading nginx, you can use HAP system with a new access address.

# Configure Separate Access Address for Files

## ! DESCRIPTION

In most browsers, the maximum number of concurrent requests to the same address is limited to 6. So if there are a large number of images in a worksheet view, a user accessing such a view with a click may result in a queue of browser requests. In this case, if the user switches to another page before the browser request has been fully loaded, he or she may experience a slowdown in browsing or even browser lag.

In response to the above, you first need to ensure that the file service instance resources and network bandwidth are sufficient. Second, you can follow this article to configure separate addresses for file requests to prevent file requests from blocking other interface requests.

## Stand-alone Environment

### ! TIP

If the separate address for file storage uses the default port 80/443 for http/https, you do not need to specify the default port 80/443 in its address when configuring the relevant environment variables.

## Single System Access Address

In this example, the system access address is `http://hap.domain.com` and the separate address for files is `http://file.domain.com`.

## Configuration Steps

1. Modify the configuration file `docker-compose.yaml` of the microservice.

```
version: '3'

services:
  app:
    image: nocoly/hap-community:6.0.2
    environment:
      ENV_ADDRESS_MAIN: "http://hap.domain.com"
```

```

ENV_APP_VERSION: "6.0.2"
ENV_API_TOKEN: "*****"
ENV_FILE_DOMAIN: "http://file.domain.com" # New variable, the
specified separate address for files
ENV_FILE_ADDRESS_MAIN: "http://file.domain.com" # New variable, the
specified separate address for files
ports:
  - 8880:8880
  - 9000:9000 # New port mapping, mapping port 9000 in the container to
the host
volumes:
  - ./volume/data:/data/
  - ../data:/data/hap/data
doc:
  image: nocoly/hap-doc:1.2.0
  environment:
    ENV_FILE_INNER_URI: "app:9000" # Modified it to the address of file
storage service

```

Configuration instructions for new variables:

- `ENV_FILE_DOMAIN`: Separate address for files specified by file storage service
- `ENV_FILE_ADDRESS_MAIN`: Separate address for files specified by microservice
- If the separate address for files uses the default port 80/443 of http/https, you do not need to specify its default port 80/443 when configuring the above two variables

Configuration instructions for new port mapping:

- `- 9000:9000` Map the port of file service in the container to the host to allow the nginx proxy to access the file service

Modification instructions for doc service variables:

- `ENV_FILE_INNER_URI`, Modify the default variable value to `app:9000`

2. Restart microservice

3. **Add nginx cross-domain proxy configuration** for the separate address for files, as shown below

```

upstream file {
    server 192.168.0.10:9000; # The specified IP of the back-end
microservice and the port of file storage service mapped to the microservice

```

```

host
}

server {
    listen 80;
    server_name file.domain.com; # The specified separate address for files
    access_log /data/logs/weblogs/file.domain.com.log main; # Log path,
    customizable
    error_log /data/logs/weblogs/file.domain.com.error.log; # Log path,
    customizable

    location / {
        if ($request_method = OPTIONS) {
            return 204 "";
        }
        proxy_set_header HOST $http_host;
        proxy_pass http://file;
        proxy_hide_header Access-Control-Allow-Origin;
        add_header Access-Control-Allow-Headers authorization,content-type;
        add_header Access-Control-Allow-Origin "http://hap.domain.com"; #
        The specified system access address of HAP
    }
}

```

4. After the proxy configuration is complete, clear the browser cache. You can then use the developer tools in the browser to check whether file-related requests are taking the separate address in Network.

## Multiple System Access Addresses

When configuring separate addresses for file services when there are multiple system access addresses, you need to configure a separate address for each system access address.

In this example, the primary system access address is `hap-external.domain.com` and the extended address is `hap-internal.domain.com`.

The separate addresses for files configured for the two access addresses are `file-external.domain.com` and `file-internal.domain.com`.

The correspondence between the system access address and the separate access address for files is as follows:

Address Type	Primary Address	Extended Address
System Access Address	hap-external.domain.com	hap-internal.domain.com
Seperate Address For Files	file-external.domain.com	file-internal.domain.com

## Configuration Steps

1. Modify the `docker-compose.yml` file of the microservice, add the relevant configuration and restart the microservice.

```
version: '3'

services:
  app:
    image: nocoly/hap-community:6.0.2
    environment:
      ENV_ADDRESS_MAIN: "http://hap-external.domain.com"
      ENV_APP_VERSION: "6.0.2"
      ENV_FILE_DOMAIN: "http://file-external.domain.com,http://file-internal.domain.com" # New variable, the specified multiple separate addresses for files
      ENV_FILE_ADDRESS_MAIN: "http://file-external.domain.com" # New variable, the specified primary separate address for files
    ports:
      - 8880:8880
      - 18880:18880
      - 9000:9000 # New port mapping, mapping port 9000 in the container to the host
    volumes:
      - ./volume/data:/data/
      - ../data:/data/hap/data

  doc:
    image: nocoly/hap-doc:1.2.0
    environment:
      ENV_FILE_INNER_URI: "app:9000" # Modify it to the address of file storage service
```

Configuration instructions for new variables:

- `ENV_FILE_DOMAIN` Specify multiple separate addresses for files for file storage service
- `ENV_FILE_ADDRESS_MAIN` Specify the **primary separate address for files** for microservice
- If the separate address for files uses the default port 80/443 of http/https, you do not need to specify its default port 80/443 when configuring the above two variables

Configuration instructions for new port mapping:

- `- 9000:9000` Map the port of file service in the container to the host to allow the nginx proxy to access the file service

Modification instructions for doc service variables:

- `ENV_FILE_INNER_URI`, Modify the default variable value to `app:9000`

## 2. Restart microservice

3. Add the nginx cross-domain proxy configuration for the **primary separate address for files**, and the proxy configuration file can follow the configuration below:

```
upstream file {
    server 192.168.0.10:9000; # The specified IP of the back-end
microservice and the port of file storage service mapped to the microservice
host
}

server {
    listen 80;
    server_name file-external.domain.com; # Specified main separate address
for files
    access_log /data/logs/weblogs/file-external.domain.com.log main; # Log
path, customizable
    error_log /data/logs/weblogs/file-external.domain.com.error.log; # Log
path, customizable

    location / {
        if ($request_method = OPTIONS) {
            return 204 "";
        }
        proxy_set_header HOST $http_host;
        proxy_pass http://file;
        proxy_hide_header Access-Control-Allow-Origin;
        add_header Access-Control-Allow-Headers authorization,content-type;
        add_header Access-Control-Allow-Origin "http://hap-
```

```
external.domain.com"; # Specified system main access address of HAP
    }
}
```

4. Add the nginx cross-domain proxy configuration for the **extended separate address for files**, the proxy configuration file can follow the configuration below:

```
upstream file {
    server 192.168.0.10:9000; # The specified IP of the back-end
microservice and the port of file storage service mapped to the microservice
host
}

server {
    listen 80;
    server_name file-internal.domain.com; # Specified extended separate
address for files
    access_log /data/logs/weblogs/file-internal.domain.com.log main; # Log
path, customizable
    error_log /data/logs/weblogs/file-internal.domain.com.error.log; # Log
path, customizable

    location / {
        if ($request_method = OPTIONS) {
            return 204 "";
        }
        proxy_set_header HOST $http_host;
        proxy_pass http://file;
        proxy_hide_header Access-Control-Allow-Origin;
        add_header Access-Control-Allow-Headers authorization,content-type;
        add_header Access-Control-Allow-Origin "http://hap-
internal.domain.com"; # System extended access address of HAP
    }
}
```

5. Modify the proxy for **system extended address** by adding the request header `pdfileaddr` field in `location /` and `location ~ /mds2` to specify the **extended separate address for files**, as follows:

```
.....
```

```
location / {
```



```

        set $real_ip '';
        if ($http_x_real_ip) {
            set $real_ip $http_x_real_ip;
        }
        if ($http_x_real_ip = '') {
            set $real_ip $remote_addr;
        }
        proxy_set_header X-Real-IP $real_ip;
        proxy_set_header Host $http_host;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header pdaddr http://hap-internal.domain.com;
        proxy_set_header pdfileaddr http://file-internal.domain.com; # New
request header pdfileaddr, the specified extended separate address for files
        proxy_pass http://hap-ext;
    }

    location ~ /mds2 {
        proxy_set_header Host $http_host;
        proxy_hide_header X-Powered-By;
        proxy_set_header X-NginX-Proxy true;
        proxy_pass http://hap-ext;
        proxy_redirect off;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection upgrade;
        proxy_set_header pdaddr http://hap-internal.domain.com;
        proxy_set_header pdfileaddr http://file-internal.domain.com; # New
request header pdfileaddr, the specified extended separate address for files
    }

    .....

```

6. After the proxy configuration is complete, clear the browser cache. You can then use developer tools in the browser to check if file-related requests in Network take the separate address.

## Cluster Environment

### TIP

If the separate address for file storage uses the default port 80/443 for http/https, you do not need to specify the default port 80/443 in its address when configuring the relevant environment variables.

## Single System Access Address

In this example, the system access address is `http://hap.domain.com` and the separate address for files is `http://file.domain.com`.

### Configuration Steps

1. Modify the configuration file `docker-compose.yaml` of the microservice.

Kubernetes

Docker Compose

- i. Modify the `config.yaml` of the microservice and add the environment variable `ENV_FILE_ADDRESS_MAIN` as the specified separate address for files.

```
ENV_FILE_ADDRESS_MAIN: "http://file.domain.com"
```

- ii. Modify the `service.yaml` of the microservice, and modify the `ENV_FILE_INNER_URI` of the doc service to the intranet address of file storage.

```
env:
- name: ENV_FILE_INNER_URI
  value: "192.168.0.10:9000" # Modify it to the intranet address of file storage
```

- iii. Restart microservice

2. Modify the configuration file `file.yaml` of the file storage service, add the relevant configuration, and restart the file storage service.

- i. Modify the configuration file `file.yaml` of the file storage service and add a new environment variable `ENV_FILE_DOMAIN` to specify a separate address for file storage service.

```
ENV_FILE_DOMAIN: "http://file.domain.com"
```

- ii. Restart the file storage service.

3. **Add nginx cross-domain proxy configuration** for the separate address for files, as shown below:

```

upstream file {
    server 192.168.0.10:9000; # Specified IP and port of file storage
service
}

server {
    listen 80;
    server_name file.domain.com; # The specified separate address for files
    access_log /data/logs/weblogs/file.domain.com.log main; # Log path,
customizable
    error_log /data/logs/weblogs/file.domain.com.error.log; # Log path,
customizable

    location / {
        if ($request_method = OPTIONS) {
            return 204 "";
        }
        proxy_set_header HOST $http_host;
        proxy_pass http://file;
        proxy_hide_header Access-Control-Allow-Origin;
        add_header Access-Control-Allow-Headers authorization,content-type;
        add_header Access-Control-Allow-Origin "http://hap.domain.com"; #
Specified system access address of HAP
    }
}

```

4. After the proxy configuration is complete, clear the browser cache. You can then use developer tools in the browser to check if file-related requests in Network take the separate address.

## Multiple System Access Addresses

When configuring separate addresses for file services when there are multiple system access addresses, you need to configure a separate address for each system access address.

In this example, the primary system access address is `hap-external.domain.com` and the extended address is `hap-internal.domain.com`.

The separate addresses for files configured for the two access addresses are `file-external.domain.com` and `file-internal.domain.com`.

The correspondence between the system access address and the separate access address for files is as follows:

Address Type	Primary Address	Extended Address
System Access Address	hap-external.domain.com	hap-internal.domain.com
Seperate Address For Files	file-external.domain.com	file-internal.domain.com

## Configuration Steps

1. Modify the yaml file of the microservice.

Kubernetes

Docker Compose

- i. Modify the configuration file `config.yaml` for microservice and add a new environment variable `ENV_FILE_ADDRESS_MAIN` to specify the **primary separate address for files** for microservice.

```
ENV_FILE_ADDRESS_MAIN: "http://file-external.domain.com"
```

- ii. Modify the configuration file `service.yaml` for the microservice and modify the `ENV_FILE_INNER_URI` variable for the doc service to the intranet address of file storage.

```
env:
- name: ENV_FILE_INNER_URI
  value: "192.168.0.10:9000" # Modify it to the intranet address for file storage
```

- iii. Restart microservice.

2. Modify the configuration file `file.yaml` of the file storage service, add the relevant configuration and restart the file storage service.

- i. Modify the configuration file `file.yaml` for the file storage service and add the environment variable `ENV_FILE_DOMAIN` to specify multiple separate addresses for the file storage service.

```
ENV_FILE_DOMAIN: "http://file-external.domain.com, http://file-internal.domain.com"
```

ii. Restart file storage service.

3. Add the nginx cross-domain proxy configuration for the **primary seperate address for files**, and the proxy configuration file can follow the configuration below:

```
upstream file {
    server 192.168.0.10:9000; # Specified IP and port of file storage
    service
}

server {
    listen 80;
    server_name file-external.domain.com; # Specified primary seperate
    address for files
    access_log /data/logs/weblogs/file-external.domain.com.log main; # Log
    path, customizable
    error_log /data/logs/weblogs/file-external.domain.com.error.log; # Log
    path, customizable

    location / {
        if ($request_method = OPTIONS) {
            return 204 "";
        }
        proxy_set_header HOST $http_host;
        proxy_pass http://file;
        proxy_hide_header Access-Control-Allow-Origin;
        add_header Access-Control-Allow-Headers authorization,content-type;
        add_header Access-Control-Allow-Origin "http://hap-
external.domain.com"; # Specified system main access address of HAP
    }
}
```

4. Add the nginx cross-domain proxy configuration for the **extended seperate address for files**, the proxy configuration file can follow the configuration below:

```
upstream file {
    server 192.168.0.10:9000; # Specified IP and port of file storage
    service
}

server {
    listen 80;
    server_name file-internal.domain.com; # Specified extended seperate
```

```

address for files
    access_log /data/logs/weblogs/file-internal.domain.com.log main; # Log
path, customizable
    error_log /data/logs/weblogs/file-internal.domain.com.error.log; # Log
path, customizable

    location / {
        if ($request_method = OPTIONS) {
            return 204 "";
        }
        proxy_set_header HOST $http_host;
        proxy_pass http://file;
        proxy_hide_header Access-Control-Allow-Origin;
        add_header Access-Control-Allow-Headers authorization,content-type;
        add_header Access-Control-Allow-Origin "http://hap-
internal.domain.com"; # System extended access address of HAP
    }
}

```

5. Modify the proxy for **system extended address** by adding the request header `pdfileaddr` field in `location /` and `location ~ /mds2` to specify the **extended separate address for files**, as follows:

```

.....

location / {
    set $real_ip '';
    if ($http_x_real_ip) {
        set $real_ip $http_x_real_ip;
    }
    if ($http_x_real_ip = '') {
        set $real_ip $remote_addr;
    }
    proxy_set_header X-Real-IP $real_ip;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header pdaddr http://hap-internal.domain.com;
    proxy_set_header pdfileaddr http://file-internal.domain.com; # New
request header pdfileaddr, the specified extended separate address for files
    proxy_pass http://hap-ext;
}

location ~ /mds2 {
    proxy_set_header Host $http_host;

```

```
proxy_hide_header X-Powered-By;
proxy_set_header X-NginX-Proxy true;
proxy_pass http://hap-ext;
proxy_redirect off;
proxy_http_version 1.1;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection upgrade;
proxy_set_header pdaddr http://hap-internal.domain.com;
proxy_set_header pdfileaddr http://file-internal.domain.com; # New
request header pdfileaddr, the specified extended separate address for files
}
```

.....

6. After the proxy configuration is complete, clear the browser cache. You can then use developer tools in the browser to check if file-related requests in Network take the separate address.

# Quick Installation

It is recommended to learn about [standalone deployment](#) first, and spend a few minutes watching the [installation instruction video](#). Some of the content in the video may differ slightly from the actual due to the earlier recording, while the overall process is basically the same.

## Cloud Marketplace Installation

It defaults to the base environment having been initialized. You can visit `http://{Server IP}:38881` directly in your browser to complete the subsequent steps (default path for the manager is `/usr/local/MDPrivateDeployment/`).

- [Alibaba Cloud Marketplace](#)
- [Tencent Cloud Marketplace](#)

## Manual Installation

### ⓘ NOTE

In case of lack of relevant experience, you may check [Manual Installation and Deployment](#) .

1. Prepare a [Linux server](#) with minimum configuration requirements: **8 cores, 32G RAM, 40G data disk** (excluding system disk). Make sure that ports `8880`, `38880`, `38881` are not occupied and that connectivity between the server and the client is normal ( [When used in a formal environment, it is recommended that access to port 38880 and 38881 be controlled through a security policy, and this address corresponds to some of the O&M functions](#));
2. Check the environment parameter. [Parameter Description](#)
3. Install docker. Official installation instructions for different Linux versions can be found [here](#);
4. Download the mirror for HAP Server ([Download Offline Package](#))

[Linux amd64](#)

[Linux arm64](#)

```
docker pull nocoly/hap-community:6.0.2
```



```
docker pull nocoly/hap-sc:3.0.0
```

```
docker pull nocoly/hap-command:node2011-python312
```

```
docker pull nocoly/hap-doc:1.2.0
```

#### 5. Download the manager

Linux amd64

Linux arm64

```
wget https://pdpublic.nocoly.com/6.0.2/hap_captain_linux_amd64.tar.gz
```

#### 6. Unzip the manager

Linux amd64

Linux arm64

```
tar -zxvf hap_captain_linux_amd64.tar.gz
```

7. Start the manager with root permissions `bash ./service.sh start` (**Make sure the manager remains in running state**);
8. After successful start, access `http://{serverIP}:38881` in the browser, and complete the setting of the HAP system access address and initialization (initialization takes about 3-5 minutes);
9. Once initialized, apply and bind a key on the current page, then create an administrator account to access the HAP workspace.

## SaaS Experience

The overall functionality of HAP Server is similar to HAP SaaS, you can [sign up for a HAP SaaS account](#) to experience it first.

# Manual Installation and Deployment(AMD version)

This document is deployed using the CentOS/RedHat system as an example, and can also be used as a reference for the Kirin V10 and Huawei Euler systems; The differences between the Debian series systems will be separately noted.

## Environment Preparation

1. Prepare a **Linux server** with minimum configuration requirements: **8 cores, 32G RAM, 40G data disk** (excluding system disk). **Recommended Servers**. Make sure ports `8880`, `38880`, `38881` are not occupied.
2. Log in to the server with the root user account
3. If the server has a data disk, please mount the data disk to the `/data` directory before deployment, or **modify the default storage path** of the data to the path where your data disk is mounted during installation
4. Shut down Firewalld

```
systemctl stop firewalld && systemctl disable firewalld
```

5. Shut down SELinux

```
setenforce 0
```

```
sed -i s/"^SELINUX=.*$"/"SELINUX=disabled"/g /etc/selinux/config
```

6. Adjust MaxMapCount

```
sysctl -w vm.max_map_count=262144
```

```
grep -q '^vm.max_map_count' /etc/sysctl.conf && sed -i  
's/^vm.max_map_count.*/vm.max_map_count=262144/' /etc/sysctl.conf || echo  
'vm.max_map_count=262144' >> /etc/sysctl.conf
```

## 7. Adjust SysFileNr

```
sysctl -w fs.file-max=2048000
```

```
grep -q '^fs.file-max' /etc/sysctl.conf && sed -i 's/^fs.file-max.*/fs.file-  
max=2048000/' /etc/sysctl.conf || echo 'fs.file-max=2048000' >>  
/etc/sysctl.conf
```

## 8. Adjust IPv4Forward

```
sysctl -w net.ipv4.ip_forward=1
```

```
grep -q '^net.ipv4.ip_forward' /etc/sysctl.conf && sed -i  
's/^net.ipv4.ip_forward.*/net.ipv4.ip_forward=1/' /etc/sysctl.conf || echo  
'net.ipv4.ip_forward=1' >> /etc/sysctl.conf
```

## 9. If the operating system is Debian 10 or above, please first check if there are any iptables using the `iptables --version` command;

If there is no iptables, it needs to be installed. The reference command is as follows (the server needs to be able to access the Internet):

```
apt-get update
```

```
apt install -y iptables
```

# Docker Installation

## 1. Download the installation package

Internet Access Available

Internet Access Unavailable

```
wget https://pdpublic.nocoly.com/offline/common/docker-27.3.1.tgz
```

2. Unzip the package and move the file to the binary file directory

```
tar -zxvf docker-27.3.1.tgz
```

```
mv -f docker/* /usr/local/bin/
```

3. Create docker configuration file

**The default docker data directory is /data/docker, if you need to change the default data directory, modify the data-root value in the configuration file.**

```
mkdir -p /etc/docker/
```

```
cat > /etc/docker/daemon.json <<EOF
{
  "registry-mirrors": ["https://uwlkeb6d.mirror.aliyuncs.com"],
  "data-root": "/data/docker",
  "max-concurrent-downloads": 10,
  "exec-opts": ["native.cgroupdriver=cgroupfs"],
  "storage-driver": "overlay2",
  "default-address-pools": [{"base": "172.80.0.0/16", "size": 24}]
}
EOF
```

4. Configure systemd to manage docker

```
cat > /etc/systemd/system/docker.service <<EOF
[Unit]
Description=Docker
After=network-online.target
Wants=network-online.target
[Service]
```

```
Type=notify
ExecStart=/usr/local/bin/dockerd
ExecReload=/bin/kill -s HUP \${MAINPID}
LimitNOFILE=102400
LimitNPROC=infinity
LimitCORE=0
TimeoutStartSec=0
Delegate=yes
KillMode=process
Restart=on-failure
StartLimitBurst=3
StartLimitInterval=60s
[Install]
WantedBy=multi-user.target
EOF
```

## 5. Start docker

```
systemctl daemon-reload && systemctl start docker && systemctl enable docker
```

# Standalone Installation

## 1. Download the mirror ([Offline package](#))

```
docker pull nocoly/hap-community:6.0.2
```

```
docker pull nocoly/hap-sc:3.0.0
```

```
docker pull nocoly/hap-command:node2011-python312
```

```
docker pull nocoly/hap-doc:1.2.0
```

## 2. Download manager

```
wget https://pdpublic.nocoly.com/6.0.2/hap_captain_linux_amd64.tar.gz
```

3. Create a directory and unzip the manager into the newly created directory

```
mkdir /usr/local/MDPrivateDeployment/
```

```
tar -zxvf hap_captain_linux_amd64.tar.gz -C /usr/local/MDPrivateDeployment/
```

4. Navigate to the directory where the manager is located, and start the manager.

```
cd /usr/local/MDPrivateDeployment/
```

```
bash ./service.sh start
```

5. After successful startup, access `http://{server IP}:38881` in your browser to complete the system access address setting and initialization (initialization takes about 5 minutes).

# Manual Installation and Deployment(ARM version)

This document is deployed using the CentOS/RedHat system as an example, and can also be used as a reference for the Kirin V10 and Huawei Euler systems; The differences between the Debian series systems will be separately noted.

## Environment Preparation

1. Prepare a **Linux server** with minimum configuration requirements: **8 cores, 32G RAM, 40G data disk** (excluding system disk). **Recommended Servers**. Make sure ports `8880`, `38880`, `38881` are not occupied.
2. Log in to the server with the root user account
3. If the server has a data disk, please mount the data disk to the `/data` directory before deployment, or **modify the default storage path** of the data to the path where your data disk is mounted during installation
4. Shut down Firewalld

```
systemctl stop firewalld && systemctl disable firewalld
```

5. Shut down SELinux

```
setenforce 0
```

```
sed -i s/"^SELINUX=.*$"/"SELINUX=disabled"/g /etc/selinux/config
```

6. Adjust MaxMapCount

```
sysctl -w vm.max_map_count=262144
```

```
grep -q '^vm.max_map_count' /etc/sysctl.conf && sed -i  
's/^vm.max_map_count.*/vm.max_map_count=262144/' /etc/sysctl.conf || echo  
'vm.max_map_count=262144' >> /etc/sysctl.conf
```

## 7. Adjust SysFileNr

```
sysctl -w fs.file-max=2048000
```

```
grep -q '^fs.file-max' /etc/sysctl.conf && sed -i 's/^fs.file-max.*/fs.file-  
max=2048000/' /etc/sysctl.conf || echo 'fs.file-max=2048000' >>  
/etc/sysctl.conf
```

## 8. Adjust IPv4Forward

```
sysctl -w net.ipv4.ip_forward=1
```

```
grep -q '^net.ipv4.ip_forward' /etc/sysctl.conf && sed -i  
's/^net.ipv4.ip_forward.*/net.ipv4.ip_forward=1/' /etc/sysctl.conf || echo  
'net.ipv4.ip_forward=1' >> /etc/sysctl.conf
```

## 9. If the operating system is Debian 10 or above, please first check if there are any iptables using the `iptables --version` command;

If there is no iptables, it needs to be installed. The reference command is as follows (the server needs to be able to access the Internet):

```
apt-get update
```

```
apt install -y iptables
```

# Docker Installation

## 1. Download the installation package



Internet Access Available

Internet Access Unavailable

```
wget https://pdpublic.nocoly.com/offline/common/arm64/docker-27.3.1.tgz
```

2. Unzip the package and move the file to the binary file directory

```
tar -zxvf docker-27.3.1.tgz
```

```
mv -f docker/* /usr/local/bin/
```

3. Create docker configuration file

**The default docker data directory is /data/docker, if you need to change the default data directory, modify the data-root value in the configuration file.**

```
mkdir -p /etc/docker/
```

```
cat > /etc/docker/daemon.json <<EOF
{
  "registry-mirrors": ["https://uwlkeb6d.mirror.aliyuncs.com"],
  "data-root": "/data/docker",
  "max-concurrent-downloads": 10,
  "exec-opts": ["native.cgroupdriver=cgroupfs"],
  "storage-driver": "overlay2",
  "default-address-pools": [{"base": "172.80.0.0/16", "size": 24}]
}
EOF
```

4. Configure systemd to manage docker

```
cat > /etc/systemd/system/docker.service <<EOF
[Unit]
Description=Docker
After=network-online.target
Wants=network-online.target
[Service]
```

```
Type=notify
ExecStart=/usr/local/bin/dockerd
ExecReload=/bin/kill -s HUP \${MAINPID}
LimitNOFILE=102400
LimitNPROC=infinity
LimitCORE=0
TimeoutStartSec=0
Delegate=yes
KillMode=process
Restart=on-failure
StartLimitBurst=3
StartLimitInterval=60s
[Install]
WantedBy=multi-user.target
EOF
```

## 5. Start docker

```
systemctl daemon-reload && systemctl start docker && systemctl enable docker
```

# Standalone Installation

## 1. Download the mirror ([Offline package](#))

```
docker pull nocoly/hap-community-arm64:-
```

```
docker pull nocoly/hap-sc-arm64:-
```

```
docker pull nocoly/hap-command-arm64:-
```

```
docker pull nocoly/hap-doc-arm64:-
```

## 2. Download manager

```
wget https://pdpublish.nocoly.com/-/hap_captain_linux_arm64.tar.gz
```

3. Create a directory and unzip the manager into the newly created directory

```
mkdir /usr/local/MDPrivateDeployment/
```

```
tar -zxvf hap_captain_linux_arm64.tar.gz -C /usr/local/MDPrivateDeployment/
```

4. Navigate to the directory where the manager is located, and start the manager.

```
cd /usr/local/MDPrivateDeployment/
```

```
bash ./service.sh start
```

5. After successful startup, access `http://{server IP}:38881` in your browser to complete the system access address setting and initialization (initialization takes about 5 minutes).

# Data Management

## Data Storage

In stand-alone deployment mode, the storage components are already built into the mirror, so users do not need to install them separately. The generated data is stored in a specified directory through mounting.

In HAP, the default data path is `/data/hap`, but if you have a [custom data path](#), use the actual path.

The data generated by storage components is by default mounted in the

`/data/hap/script/volume/data/` directory on the server. The directory structure is as follows:

```
/data/hap/script/volume/data/  
├─ elasticsearch  
├─ elasticsearch-8  
├─ kafka  
├─ logs  
├─ mongodb  
├─ mysql  
├─ redis  
├─ storage  
└─ zookeeper
```

Data stored in each directory:

- `elasticsearch`, for elasticsearch data prior to version 3.9.0
- `elasticsearch-8`, for elasticsearch data starting from version 3.9.0, mainly supporting super search and application logs
- `kafka`, for storing message queue data, expired data is automatically cleaned up by default
- `log`, for logs of component service
- `mongodb`, for storing business data (such as applications, worksheets, workflows, etc.)
- `mysql`, for storing system core data such as users, organizations, etc.
- `redis`, for caching data

- storage, for storing attachment data
- zookeeper, to provide services for kafka

## Data Backup

- [View more details about backup](#)
- If using a cloud server, you can utilize the disk snapshot feature provided by the cloud provider.

## Data Cleaning

In general, the data in logs, mongodb, and storage occupy relatively large disk space.

- For logs stored in the logs directory for component services, if you do not need this log data, you can clean it up.
- For business data stored in mongodb, currently only [log data archiving](#) and [cleaning deleted application data](#) are supported.
- For attachment data stored in storage, if the disk capacity cannot be supported, it is recommended to limit the file size on the upload page.
  - If you are using a cloud server, go to [enable object storage](#) to reduce disk capacity pressure.

If your environment has been upgraded, mirrors of history versions that have not been cleaned up will also take up some disk space.

► [Example of Cleaning up History Mirrors](#)

If the above methods fail to free up more disk space, consider expanding the disk capacity.

## Data Migration

- [More details about data backup](#)
- [More details about data restoration](#)

# Backup

- The following directory takes the time of 2022111184140 as an example, in the format of year, month, day, hour, minute, and second.
- The host uses the default data directory `/data/hap/` (which can be viewed by `cat /etc/pdcaptain.json` or `cat service.sh | grep installDir=` in the root of the manager).

## Dump (available in v3.7.1+)

### No need to stop HAP service

```
docker exec -it $(docker ps | grep community | awk '{print $1}') bash -c 'source /entrypoint.sh && backup mysql mongodb file'
```

The default backup file will be placed in the directory

`/data/hap/script/volume/data/backup/2022111184140/`, if you need to change the path, you can add a new mount in `docker-compose.yml` as follows:

```
volumes:  
  - /backup:/data/backup/
```

In this case, the backup file will be generated in the `/backup/` of the host.

```
backup/  
├── 20221201193829  
│   ├── backupFile.log  
│   ├── backupMongodb.log  
│   ├── backupMysql.log  
│   ├── file  
│   ├── ...  
│   ├── mongodb  
│   ├── mongodb | ...  
├── mysql  
│   └── ...
```

In the directory `backup`, execute the command `tar -zcvf 2022111184140.tar.gz ./20221201193829` to compress the dumped file.

## Copy File

In the directory `/data/hap/script/volume/data/`, the `mysql`, `mongodb`, `storage`, `kafka`, `zookeeper`, `redis`, `elasticsearch-8` folders all contain data that needs to be backed up.

**Stop HAP service** and execute the following command to pack the data:

```
mkdir -p /backup && cd /data/hap/script/volume/data/ && tar -zcvf  
/backup/20221111184140.tar.gz ./mysql ./mongodb ./storage ./kafka ./zookeeper  
./redis ./elasticsearch-8
```

# Restore

- The following directory takes the time 20221111184140 as an example for illustration, in the format of year, month, day, hour, minute, and second.
- The host uses the default data directory `/data/hap/` (which can be viewed via `cat /etc/pdcaptain.json` or `cat service.sh | grep installDir=` in the root of the manager).

## Preparation

1. Stop the HAP service. Execute the command `bash . /service.sh stopall` in the root of the manager;
2. Upload the backup file `20221111184140.tar.gz` to the data restore server, e.g. in the directory `/backup/`;
3. Remove the original data from the current environment (for security, use the mv command first, and if it is OK, delete it completely).

```
time=$(date +%Y%m%d%H%M%S) && mkdir -p /backup/$time/ && mv  
/data/hap/script/volume/data/* /backup/$time/
```

## Dump (available in v3.7.1+)

1. Decompress the zip package of the backup file in the directory `/backup/`.

```
tar -zxvf 20221111184140.tar.gz
```

2. Start the temporary container and mount the data directory.

```
docker run -it --rm --entrypoint bash -e ENV_MYSQL_HOST="127.0.0.1" -e  
ENV_MYSQL_PORT="3306" -e ENV_MYSQL_USERNAME="root" -e  
ENV_MYSQL_PASSWORD="123456" -e ENV_MONGODB_URI="mongodb://127.0.0.1:27017" -  
e ENV_MONGODB_OPTIONS="" -v /data/hap/script/volume/data:/data/ -v  
/backup:/data/backup/ nocoly/hap-sc:3.0.0
```

```
mkdir -p /data/{logs,mysql,mongodb,storage}  
mkdir -p /data/storage/data
```



3. Start the mysql server in a temporary container for restoring data.

```
source /entrypoint.sh && mysqlStartup &
```

4. Rebuild the mysql database. If there is important data, you need to backup it first before deleting.

```
source /entrypoint.sh && restore mysql /data/backup/20221111184140/mysql
```

5. Start the mongodb server in a temporary container for restoring data.

```
source /entrypoint.sh && mongodbStartup &
```

**Startup will automatically create the index, you need to execute the enter command manually after the creation.**

6. Rebuild the mongodb database. If there is important data, you need to backup it first before deleting.

```
source /entrypoint.sh && restore mongodb /data/backup/20221111184140/mongodb
```

7. Rebuild the file data.

```
source /entrypoint.sh && restore file /data/backup/20221111184140/file
```

8. Exit the temporary container.

9. Execute the command ``bash ./service.sh startall` in the root of the manager to restart the service.

## Copy File

1. Upload the zip package of the original backup file to the current server and unzip it to the data directory.

```
tar -zxvf /backup/20221111184140.tar.gz -C /data/hap/script/volume/data/
```

2. Execute the command `bash . /service.sh startall` in the root of the manager to restart the service.

# Data Backup and Restore in MySQL

## Data Backup

1. Execute the backup command.

```
docker exec -it $(docker ps | grep community | awk '{print $1}') bash -c  
'source /entrypoint.sh && backup mysql'
```

If successful, the following message will be output (the storage path in the container)

```
backup mysql saved to /data/backup/20240117130609/mysql  
backup log saved to /data/backup/20240117130609/backupMySQL.log
```

2. Copy the backup file to the directory in the host (/data/backup).

```
mkdir -p /data/backup  
docker cp $(docker ps | grep community | awk '{print  
$1}'): /data/backup/20240117130609 /data/backup/
```

## Data Restore

1. Stop the HAP service. Execute the following command in the root directory of the manager.

```
bash ./service.sh stopall
```

2. Remove the original data directory from MySQL (Default:

/data/hap/script/volume/data/mysql).

```
mv /data/hap/script/volume/data/mysql /data/backup/mysql_$(date  
+%Y%m%d%H%M%S)
```

3. Start the temporary container and mount the data directory (Replaced with the actual mirror address and version number).

Microservice Version $\geq$ 5.1.0

Microservice Version $<$ 5.1.0

```
docker run -it --rm --entrypoint bash -e ENV_MYSQL_HOST="127.0.0.1" -e  
ENV_MYSQL_PORT="3306" -e ENV_MYSQL_USERNAME="root" -e  
ENV_MYSQL_PASSWORD="123456" -v /data/hap/script/volume/data:/data/ -v  
/data/backup:/data/backup/ nocoly/hap-sc:3.0.0
```

4. Create directory.

```
mkdir -p /data/{logs,mysql}
```

5. Start MySQL in the temporary container.

```
source /entrypoint.sh && mysqlStartup &
```

6. Restore MySQL database (/data/backup/20240117130609/mysql, replaced with actual path).

```
source /entrypoint.sh && restore mysql /data/backup/20240117130609/mysql
```

7. Execution completed. Exit the temporary container.

```
exit
```

# Upgrade Microservices

## Upgrade Online

The super admin clicks the profile photo in the upper right corner, selecting [System Configuration] > [General] > [Management], and click [Update] on the settings page to upgrade.



## Upgrade on Terminal

1. Pull the microservice application mirror of the target version. [Offline Package](#)

Linux amd64

Linux arm64

```
docker pull nocoly/hap-community:version
```

2. Modify the version number of the mirror in the `/data/hap/script/docker-compose.yaml`;
3. Execute the command `bash . /service.sh restartall` in the root directory of the manager to restart the service and wait for it to finish;

# Upgrade Document Preview Service

1. Pull the mirror of document preview service of the target version ([Offline Package](#));

Linux amd64

Linux arm64

```
docker pull nocoly/hap-doc:version
```

2. Modify the version number of the corresponding mirror in the file `/data/hap/script/docker-compose.yaml`;
3. Execute the command `bash . /service.sh restartall` in the root directory of the manager to restart the service and wait for it to finish.

# Upgrade Storage Component Service

1. Pull the mirror of storage component service of the target version ([Offline Package](#))

Linux amd64

Linux arm64

```
docker pull nocoly/hap-sc:version number
```

2. Modify the version number of the corresponding mirror in the `/data/hap/script/docker-compose.yaml` file;
3. Execute `bash . /service.sh restartall` to restart the service in the root directory of the manager and wait for the command to finish execution;

# Upgrade MongoDB 3.4 to 4.4

## PROMPT

The upgrade duration is directly related to the number of collections in the MongoDB database

You can view the number of collections through `find`

```
/data/hap/script/volume/data/mongodb/ -name '*collection*' | wc -l
```

For example, it takes 1 minute for a collection of less than 10000, about 10 minutes for 100000, and about 30 minutes for 300000

## Stop service

Execute commands in the root directory of the manager:

```
bash ./service.sh stopall
```

## Backup data

**If the mongodb directory has been backed up, you can skip this step**

Before backup, it is necessary to confirm whether the remaining space on the server disk is sufficient

```
du -sh /data/hap/script/volume/data/mongodb
```

Create a backup, such as (adjusting the actual backup directory):

```
mkdir -p /backup && tar -zcvf /backup/mongodb3.4_$(date +%Y%m%d%H%M%S).tar.gz  
/data/hap/script/volume/data/mongodb
```

## Upgrade

MongoDB upgrade requires version by version upgrade. Upgrading from 3.4 to 4.4 requires sequentially upgrading to 3.6, 4.0, 4.2, and 4.4, with a total of 4 upgrade commands to be executed. If there is a version mismatch prompt during the upgrade process, the previous upgrade command



needs to be executed repeatedly(**For example, if the upgrade from 3.4 to 3.6 is successful and the upgrade from 3.6 to 4.0 prompts failure, the command from 3.4 to 3.6 needs to be executed again until the output of exit upgrade**). If the error persists after multiple attempts, the specific reason can be checked through `/data/hap/script/volume/data/mongodb/upgrade xxxx. log`.

Pull upgrade secondary mirror ([Offline Package] (.././../offline.md)):

```
docker pull nocoly/hap-sc-upgrade:1.0.0
```

Execute upgrade command:

3.4 to 3.6

```
docker run -i --rm -v /data/hap/script/volume/data/mongodb:/data/mongodb  
nocoly/hap-sc-upgrade:1.0.0 <<< 'upgradeMongodb.sh 3.4 3.6'
```

3.6 to 4.0

```
docker run -i --rm -v /data/hap/script/volume/data/mongodb:/data/mongodb  
nocoly/hap-sc-upgrade:1.0.0 <<< 'upgradeMongodb.sh 3.6 4.0'
```

4.0 to 4.2

```
docker run -i --rm -v /data/hap/script/volume/data/mongodb:/data/mongodb  
nocoly/hap-sc-upgrade:1.0.0 <<< 'upgradeMongodb.sh 4.0 4.2'
```

4.2 to 4.4

```
docker run -i --rm -v /data/hap/script/volume/data/mongodb:/data/mongodb  
nocoly/hap-sc-upgrade:1.0.0 <<< 'upgradeMongodb.sh 4.2 4.4'
```

The final output of **newRunVersion: 4.4** indicates that the upgrade is complete. 🙌 🙌

## Several common output explanations

## Successfully upgraded to the target version

```
oldRunVersion: 4.0
Upgrade 4.0 to 4.2...
{ "ok" : 1 }
newRunVersion: 4.2
```

## Successfully executed target version upgrade script

```
oldRunVersion: 4.2
exit upgrade 4.0 to 4.2
```

## Version mismatch, need to execute the previous successful upgrade command again

```
ERROR: child process failed, exited with error number 100
To see additional information in this output, start without the "--fork" option.
Failed
```

# Same-server Deployment of Data Pipeline Service

If your environment is deployed with data pipeline service along with microservices according to the document [How to Enable Data Pipeline](#), follow the steps here when upgrading data pipeline services.

1. Download the mirror of the new version

**Internet Access Available**

**Internet Access Unavailable**

```
docker pull nocoly/hap-flink:version number
```

2. Modify the version number of the mirror corresponding to the flink service in the `docker-compose.yaml` file.

```
flink:
  image: nocoly/hap-flink:version number
```

3. Execute `bash ./service.sh restartall` in the root directory of the manager to restart the service and wait for it to complete execution.

# Standalone-server Deployment of Data Pipeline Service

If your environment is deployed with data pipeline service on a standalone server according to the document [Standalone-server Deployment of Data Pipeline Service](#), follow the steps below when upgrading data pipeline service.

1. Download the mirror of the new version

Internet Access Available

Internet Access Unavailable

```
docker pull nocoly/hap-flink:version number
```

2. Modify the version number of the mirror corresponding to the flink service in the `flink.yaml` file for data pipeline service.

```
flink:
  image: nocoly/hap-flink:version number
```

3. Restart `Flink` service.

Stop first

```
docker stack rm flink
```

Start again when the flink container is not seen in `docker ps`

```
docker stack deploy -c flink.yaml flink
```

# Same-server Deployment of Data Pipeline Service

1. Download the mirror of the new version

Internet Access Available

Internet Access Unavailable

```
docker pull nocoly/hap-flink:version number
```

2. Enter the `hap-flink` container.

```
docker exec -it $(docker ps | grep flink | awk '{print $1}') bash
```

3. Backup and delete the `zookeeper` directory in the `Flink` data directory and empty the `Job` list in `Flink` in the old version.

```
mv /data/flink/zookeeper /data/flink/zookeeper.backup-$(date +%Y%m%d%H%M%S)
```

4. Modify the version number of the mirror corresponding to the `hap-flink` service in the `docker-compose.yaml` file.

```
flink:
  image: nocoly/hap-flink:version number
```

5. Execute `bash . /service.sh restartall` in the root directory of the manager to restart the service and wait for it to complete execution.

6. Go to [Integrate] > [Data Pipeline] > [Sync Task] to to re-open or publish the task.

# Standalone-server Deployment of Data Pipeline Service

1. Download the mirror of the new version

Internet Access Available

Internet Access Unavailable

```
docker pull nocoly/hap-flink:version number
```

2. Enter the `hap-flink` container.

```
docker exec -it $(docker ps | grep flink | awk '{print $1}') bash
```

3. Backup and delete the `zookeeper` directory in the `Flink` data directory and empty the `Job` list in `Flink` in the old version.

```
mv /data/flink/zookeeper /data/flink/zookeeper.backup-$(date +%Y%m%d%H%M%S)
```

4. Modify the version number of the mirror corresponding to the `hap-flink` service in the `flink.yaml` file.

```
flink:
  image: nocoly/hap-flink:version number
```

5. Restart `Flink` service.

Stop first

```
docker stack rm flink
```

Start again when the flink container is not seen in `dcoker ps`

```
docker stack deploy -c flink.yaml flink
```

---

6. Go to [Integrate] > [Data Pipeline] > [Sync Task] to to re-open or publish the task.

# Upgrade Manager

It is not necessary to upgrade the manager if not explicitly requested

1. Stop the service. Execute the command `bash . /service.sh stopall` in the root directory of manager (it will output stopped if successful).
2. Clean up the files associated with **HAP service** and **manager**:
  - Backup and remove the data and files related to the original microservice application, such as `mv /data/hap/ /backup/hapbak20200320/`. The target location can be customized. **This step is of high risk, so make sure it is done correctly.**
  - In the **root directory of manager**, execute `rm -rf . /*` (This operation is not allowed if the manager is not stored in a separate folder).
3. Reinstall. More details in [quick installation](#) (just complete 3~7 steps).
4. Stop the service. Execute the command `bash . /service.sh stopall` (it will output stopped if successful).
5. Execute the command `rm -rf /data/hap/script/volume/data/*` to clean up the data generated by the reinstallation, and restore the data previously backed up, such as `cp -r /backup/hapbak20200320/script/volume/data/* /data/hap/script/volume/data/`.
6. Compare `/data/hap/script/docker-compose.yaml` with `/backup/hapbak20200320/script/docker-compose.yaml` in the backup directory, and adjust it manually if there are additional custom configuration items.
7. Restart the service. Execute the command `bash . /service.sh startall` and wait for it to complete.



# Strong Passwords for Database

## CAUTION

💡 In standalone mode in a private deployment, by default, the database components are built into the mirror and start with the microservice, and the database in the container is not accessible externally.

💣 If there is a need to map the database port from the container to the host for access, it is highly recommended to do so when the following three conditions are met, otherwise it is extremely risky for the database port to be exposed to the Internet!!!

1. Back up data regularly (to prevent unrecoverable data loss due to database hacking).
2. Set up access whitelist for database ports ( to restrict source addresses from accessing database ports).
3. Change the password to a strong one (**Avoid special characters with regular meanings, such as "\$", "&", "@", etc., otherwise they cannot be passed into the container; you can choose - or \_**).

## Change Default Password for MySQL

- Using tC9S86SFWxga as an example for the root password, please remember to change it in your actual configura...

## Change Default Password for Redis

- Using f8K5ZT3aQXTb as an example for the new Redis password, please remember to change it before deploying.

## MongoDB Add Authentication

- When adding authentication, two users are created, the root user for the admin library and the hap user for all busi...

# Change Default Password for MySQL



TIP

- Using `tC9S86SFWxga` as an example for the root password, **please remember to change it in your actual configuration.**
  - To ensure compatibility and security, avoid special characters like "\$", "&", or "@" in your custom passwords. These characters can interfere with regular expression parsing and may not be correctly passed to the container. Use hyphens "-" or underscores "\_" instead.
- Advance data backup is recommended before operation.

1.

**microservice version**>=5.1.0

microservice version<5.1.0

Enter the hap-sc container and log in to MySQL

```
docker exec -it $(docker ps | grep hap-sc | awk '{print $1}') bash -c 'mysql -uroot -p123456 -h127.0.0.1'
```

2. Change MySQL password

```
GRANT ALL ON *.* to root@'%' IDENTIFIED BY 'tC9S86SFWxga';
```

3. Modify the file docker-compose.yaml to add environment variables and port mapping.

The default path for file docker-compose.yaml: `/data/hap/script/docker-compose.yaml`

**microservice version**>=5.1.0

microservice version<5.1.0

Add a new environment variable `ENV_MYSQL_PASSWORD` to specify the new password for MySQL under the app service.

```
ENV_MYSQL_PASSWORD: "tC9S86SFwxga"
```

Add port mapping under the sc service to map port 3306 out of the container (you can leave port mapping off if don't need external access).

```
- 3306:3306
```

► Configuration example of file docker-compose.yaml

4. Restart the microservice in the directory of Install Manager to take effect.

```
bash service.sh restartall
```

# Change Default Password for Redis



TIP

- Using `f8K5ZT3aQXTb` as an example for the new Redis password, **please remember to change it before deploying.**
  - To ensure compatibility and security, avoid special characters like "\$", "&", or "@" in your custom passwords. These characters can interfere with regular expression parsing and may not be correctly passed to the container. Use hyphens "-" or underscores "\_" instead.
- Advance [data backup](#) is recommended before operation.
- Microservices version: v3.7.0+.

1.

microservice version>=5.1.0

microservice version<5.1.0

Enter the hap-sc container and log in to Redis

```
docker exec -it $(docker ps | grep hap-sc | awk '{print $1}') bash -c 'redis-cli -a 123456'
```

2. Change Redis password

```
config set requirepass f8K5ZT3aQXTb
```

3. Modify the file docker-compose.yaml to add environment variables and port mapping.

The default path for file docker-compose.yaml: /data/hap/script/docker-compose.yaml

microservice version>=5.1.0

microservice version<5.1.0

Add a new environment variable `ENV_REDIS_PASSWORD` to specify the new password for redis under the app service.

```
ENV_REDIS_PASSWORD: "f8K5ZT3aQXTb"
```

Add port mapping under the sc service to map port 6379 out of the container (you can leave port mapping off if don't need external access).

```
- 6379:6379
```

► Configuration example of file docker-compose.yaml

4. Restart the microservice in the directory of Install Manager to take effect.

```
bash service.sh restartall
```

# MongoDB Add Authentication



## TIP

- When adding authentication, two users are created, the root user for the admin library and the hap user for all business libraries.
- The provided example uses `hTkfDMYJ7ZLs` for the root password and `tC9S86SFWxga` for the hap password. **Please replace these with your own, strong passwords in your actual setup.**
  - To ensure compatibility and security, avoid special characters like "\$", "&", or "@" in your custom passwords. These characters can interfere with regular expression parsing and may not be correctly passed to the container. Use hyphens "-" or underscores "\_" instead.
- Advance [data backup](#) is recommended before operation.
- Microservices version: v3.7.0+.
- If you've [enabled the aggregate table feature](#), please refer to the corresponding documents to complete the creation of the aggregate table database and the corresponding roles and users, as well as adjust the replica set related parameters

1.

microservice version  $\geq 5.1.0$

microservice version  $< 5.1.0$

First, use the `docker ps` command to find the hap-sc container

Then use the command `docker exec -it $(docker ps | grep hap-sc | awk '{print $1}')`  
`bash` to enter the hap-sc container

Execute the `mongo` command to log in to mongo shell in the hap-sc container

1. Create root users for the admin library and hap users for all business libraries in the mongo shell

```
use admin
db.createUser({user:"root",pwd:"hTkfDMYJ7ZLs",roles:
[{"role":"root",db:"admin"}]})
```

```
use MDLicense
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"MDLicense"}]})
use ClientLicense
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"ClientLicense"}]})
use commonbase
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"commonbase"}]})
use MDAalert
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"MDAalert"}]})
use mdactionlog
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdactionlog"}]})
use mdapproles
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdapproles"}]})
use mdapprove
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdapprove"}]})
use mdapps
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdapps"}]})
use mdattachment
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdattachment"}]})
use mdcalendar
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdcalendar"}]})
use mdcategory
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdcategory"}]})
use MDChatTop
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"MDChatTop"}]})
use mdcheck
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdcheck"}]})
use mddossier
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mddossier"}]})
use mdemail
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdemail"}]})
```



```
use mdform
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdform"}]})
use MDGroup
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"MDGroup"}]})
use mdgroups
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdgroups"}]})
use MDHistory
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"MDHistory"}]})
use mdIdentification
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdIdentification"}]})
use mdinbox
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdinbox"}]})
use mdkc
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdkc"}]})
use mdmap
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdmap"}]})
use mdmobileaddress
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdmobileaddress"}]})
use MDNotification
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"MDNotification"}]})
use mdpost
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdpost"}]})
use mdreportdata
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdreportdata"}]})
use mdroles
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdroles"}]})
use mdsearch
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdsearch"}]})
use mdservicedata
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdservicedata"}]})
```

```
use mdsms
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdsms"}]})
use MDSso
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"MDSso"}]})
use mdtag
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdtag"}]})
use mdtransfer
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdtransfer"}]})
use MDUser
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"MDUser"}]})
use mdworkflow
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdworkflow"}]})
use mdworksheet
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdworksheet"}]})
use mdworkweixin
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdworkweixin"}]})
use mdwsrows
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"mdwsrows"}]})
use pushlog
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"pushlog"}]})
use taskcenter
db.createUser({user:"hap",pwd:"tC9S86SFWxga",roles:
[{"role":"readWrite",db:"taskcenter"}]})
use mdintegration
db.createUser({user: "hap",pwd: "tC9S86SFWxga",roles: [{"role:
"readWrite",db: "mdintegration"}]})
use mdworksheetlog
db.createUser({user: "hap",pwd: "tC9S86SFWxga",roles: [{"role:
"readWrite",db: "mdworksheetlog"}]})
use mdworksheetsearch
db.createUser({user: "hap",pwd: "tC9S86SFWxga",roles: [{"role:
"readWrite",db: "mdworksheetsearch"}]})
use mddatapipeline
db.createUser({user: "hap",pwd: "tC9S86SFWxga",roles: [{"role:
"readWrite",db: "mddatapipeline"}]})
```

```
use mdwfpplugin
db.createUser({user: "hap",pwd: "tC9S86SFwxga",roles: [{role:
"readWrite",db: "mdwfpplugin"}]})
```

2. Modify the file docker-compose.yaml to add environment variables and port mapping.

Default path for file docker-compose.yaml: /data/hap/script/docker-compose.yaml

**microservice version>=5.1.0**

**microservice version<5.1.0**

Add environment variables `ENV_MONGODB_DAEMON_ARGS` and `ENV_MONGODB_URI` under the app service.

```
ENV_MONGODB_DAEMON_ARGS: "--auth"
ENV_MONGODB_URI: "mongodb://hap:tC9S86SFwxga@127.0.0.1:27017"
```

Add port mapping under the sc service to map port 27017 in the container (you can leave port mapping off if don't need external access).

```
- 27017:27017
```

► Configuration example of file docker-compose.yaml

3. Restart the microservice in the directory of Install Manager to take effect.

```
bash service.sh restartall
```

# Ops Platform

Starting from v6.0.0, HAP provides Ops Platform to assist in monitoring system operation status and operation and maintenance management.

## Basic requirements

1. Deploying servers requires reserving resources of at least 2C4G
2. When using built-in components, it is recommended to deploy the Ops Platform on the same host as the HAP service; When using external components, the connection method needs to be passed to the `agent` service
3. MySQL service address, users need to have the following permissions:
  - `SELECT` permission: allows reading tables and data from the database
  - `HOW DATABASES` permission: Allow viewing of all databases
  - `PROCESS` permission: allows viewing other users' threads to obtain performance data
  - `REPLICATION CLIENT` permission: Used to access replication status and information
4. MongoDB service address, users need to have the following permissions:
  - `root` role permission
5. Redis service address, users need to have the permission to `read` Redis data
6. Kafka service address, users need to have the permission to `read Kafka metadata`
7. Elasticsearch service address, users need to have `read and write index` permission
8. Flink service address, allowing access to the internal network environment of the Ops Platform

## Pull image (Offline Package)

```
docker pull nocoly/ops-gateway:1.0.0
docker pull nocoly/ops-prometheus:1.0.0
docker pull nocoly/ops-agent:1.0.0
docker pull nocoly/ops-nodeagent:1.0.0
```

## Create ops.yaml

```
cat > /data/hap/script/ops.yaml <<\EOF
version: '3'

services:
```

```
gateway:
  image: nocoly/ops-gateway:1.0.0
  ports:
    - "48881:48881"
  environment: &common_env
    ENV_OPS_TOKEN: "SS9PobGG7SDTpcyfSZ1VVmn3gCmy2P52tYk" #
Adjustment required (essential)
  ENV_PROMETHEUS_HOST: "hap_1/IP:59100" #
Need to adjust the internal IP address of the host computer, with port fixed at
59100
  ENV_PROMETHEUS_KAFKA: "kafka_1/agent:9308"
  ENV_PROMETHEUS_ELASTICSEARCH: "elasticsearch_1/agent:9114"
  ENV_PROMETHEUS_REDIS: "redis_1/agent:9121"
  ENV_PROMETHEUS_MONGODB: "mongodb_1/agent:9216"
  ENV_PROMETHEUS_MYSQL: "mysql_1/agent:9104"
  ENV_MYSQL_HOST: "sc" # If
using external components or modifying the default password, please adjust the
relevant parameter values
  ENV_MYSQL_PORT: "3306"
  ENV_MYSQL_USERNAME: "root"
  ENV_MYSQL_PASSWORD: "123456"
  ENV_MONGODB_URI: "mongodb://sc:27017"
  ENV_MONGODB_OPTIONS: ""
  ENV_REDIS_HOST: "sc"
  ENV_REDIS_PORT: "6379"
  ENV_REDIS_PASSWORD: "123456"
  ENV_KAFKA_ENDPOINTS: "sc:9092"
  ENV_ELASTICSEARCH_ENDPOINTS: "sc:9200"
  ENV_ELASTICSEARCH_PASSWORD: "md:ESPassWD1234"
  ENV_FLINK_URL: "http://flink:8081" #
Please adjust to an internal network accessible address, if Flink Web commenting
is not required
```

```
prometheus:
  image: nocoly/ops-prometheus:1.0.0
  environment: *common_env
  volumes:
    - ./volume/data/:/data/
```

```
agent:
  image: nocoly/ops-agent:1.0.0
  environment: *common_env
```

```
nodeagent:
  image: nocoly/ops-nodeagent:1.0.0
```

```
environment: *common_env
volumes:
  - /:/host:ro,rslave
network_mode: host
pid: host

networks:
  script_default:
    external: true
EOF
```

## Start Service

```
docker-compose -f /data/hap/script/ops.yaml up -d
```

To stop the service, the following command can be used:

```
docker-compose -f /data/hap/script/ops.yaml down
```

## Accessing the Ops Platform

Login Token as `ENV_OPS-TOKEN` in ops.yaml

```
http://IP:48881
```

# Kubernetes

In the latest cluster mode of HAP Server, microservices are structured in [Kubernetes + Istio](#), providing excellent scalability and load balancing.

If the microservices in your HAP Server cluster environment are structured in this architecture, you can view the documents in this directory for operation and maintenance.

# Microservices Upgrade

## Import Mirror

Importing mirrors is required for each microservice node.

Internet Access Available

Internet Access Unavailable

```
cricctl pull nocoly/hap-community:6.0.2
```

- When downloading, there is no output on the command line, just wait for the command to execute.

## Upgrade Microservices

### NOTES

- Microservices are running in [Kubernetes + Istio] mode, and can be restarted with a rolling update, during which page access is almost not affected.
- **Prerequisite for rolling updates: the memory of each microservice node server needs to be about 40% available.**
- If there is insufficient memory available, it is necessary to use a non-rolling update method to upgrade.
- Before upgrading microservices, go to [Changelog](#) to check whether there is a version marked with **additional operations** between the versions before and after the upgrade, and if there is, complete the related operations according to the guide documents.
- Before upgrading, please check whether the upgrade service in the license key is expired or not, when it is expired, please don't upgrade the version whose release date of the main version is later than the expiration date of the upgrade service. If you still upgrade after the expiration of the upgrade service, there will be a prompt that the upgrade is restricted.



## Rolling Update

## Non-Rolling Update

Execute the update command on the control node, in the directory where the configuration file is located.

It defaults to the `/data/hap/script/kubernetes` directory in the `microservices1` node.

```
bash update.sh update community 6.0.2
```

- 6.0.2 is the version you want to update. Import the mirror to each microservice node in advance.
- After execution, wait about 3 to 5 minutes for the restart to complete.

After restarting, you can check the status of each pod with the command `kubect1 get pod`, which is normally `2/2`.

# Non-Cross Version

## 1. Download the mirror of the new version

Operations are required on each node server in a kubernetes cluster.

Internet Access Available

Internet Access Unavailable

```
cricctl pull nocoly/hap-flink:version number
```

## 2. Modify the configuration file

Modify the version of the mirrors used by the `flink-jobmanager` and `flink-taskmanager` services in `flink. yaml`.

```
- name: jobmanager
  image: nocoly/hap-flink:version number
- name: taskmanager
  image: nocoly/hap-flink:version number
```

## 3. Restart service

```
kubectl apply -f flink.yaml
```

# Cross Version

1. Delete the `ConfigMap`, clean up the `Job` list and find the `ConfigMap` configuration, as shown in the following example:

```
kubernetes.cluster-id: md-flink
kubernetes.namespace: default
```

Execute the script to batch delete `ConfigMap`.

If something like `configmap "md-flink" deleted` is not output, it means that the namespace or the prefix of `ConfigMap` is not correct, check again.

```
# for i in $(kubectl -n [Replaced with value of kubernetes.namespace] get cm
| awk '$1~"[Replaced with value of kubernetes.cluster-id]"{print $1}');do
kubectl -n [Replaced with value of kubernetes.namespace] delete cm $i;done

for i in $(kubectl -n default get cm | awk '$1~"md-flink"{print $1}');do
kubectl -n default delete cm $i;done
```

2. Download the mirror of the new version

Operations are required on each node server in a kubernetes cluster.

**Internet Access Available**

**Internet Access Unavailable**

```
crictrl pull nocoly/hap-flink:version number
```

3. Modify the configuration file

Modify the version of the mirrors used by the `flink-jobmanager` and `flink-taskmanager` services in `flink. yaml`.

```
- name: jobmanager
  image: nocoly/hap-flink:version number
```

```
- name: taskmanager  
  image: nocoly/hap-flink:version number
```

#### 4. Restart service

```
kubectl apply -f flink.yaml
```

#### 5. Go to [Integrate] > [Data Pipeline] > [Sync Task] to to re-open or publish the task.

# Upgrade MongoDB 3.4 to 4.4

## NOTE

- The time required for the upgrade depends on the number of collections in the MongoDB database.

You can check the number of collections with the command `find /data/mongodb/ -name '*collection*' | wc -l`.

For example, upgrading with less than 10,000 collections may take 1 minute, 100,000 collections may take about 10 minutes, and 300,000 collections may take around 30 minutes; this can also be affected by server disk performance.

- In the example commands below, the path to the MongoDB program is `/usr/local/mongodb/`, and the data directory is `/data/mongodb/`. Modify the commands accordingly if different.

## Single Node

### Preparation

- Follow the details in [Docker Installation](#) to install Docker on the MongoDB server.
- Pull the upgrade auxiliary mirror ([Download Offline Package](#)).

```
docker pull nocoly/hap-sc-upgrade:1.0.0
```

- Download the installation package for MongoDB v4.4.29 and upload it to the server

**RedHat / CentOS 7.0 x64**

RedHat / CentOS 8.0 x64

Debian 10.0+ x64

Others

```
https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-4.4.29.tgz
```

- Unzip the installation package of MongoDB v4.4.29 and check if the `mongod` program in the package is missing dependencies. If dependencies are missing, install them accordingly.

RedHat / CentOS 7.0 x64

RedHat / CentOS 8.0 x64

Debian 10.0+ x64

Unzip the installation package of MongoDB v4.4.29.

```
tar xvf mongodb-linux-x86_64-rhel70-4.4.29.tgz
```

Check if the `mongod` program in the package is missing dependencies.

```
ldd mongodb-linux-x86_64-rhel70-4.4.29/bin/mongod
```

## Start Upgrading

Upgrading MongoDB must be done version by version. If you want to upgrade MongoDB from 3.4 to 4.4, you need to upgrade MongoDB to 3.6, 4.0, 4.2, and finally 4.4, executing the upgrade command 4 times.

During the upgrade, if it prompts that the versions do not match, you need to repeat the previous upgrade command (**for example, if upgrading from 3.4 to 3.6 is successful, but upgrading from 3.6 to 4.0 fails, then you need to execute the 3.4 to 3.6 command again until it outputs "exit upgrade"**). If the error persists after multiple attempts, you can check the reason in

`/data/mongodb/upgrade-xxxx.log`.

1. Stop the microservices
2. Back up the MongoDB data
3. Stop the existing MongoDB instance

```
systemctl stop mongod
```

4. Upgrade MongoDB version

3.4 to 3.6

```
docker run -i --rm -v /data/mongodb:/data/mongodb nocoly/hap-sc-upgrade:1.0.0 <<< 'upgradeMongodb.sh 3.4 3.6'
```

3.6 to 4.0

```
docker run -i --rm -v /data/mongodb:/data/mongodb nocoly/hap-sc-  
upgrade:1.0.0 <<< 'upgradeMongodb.sh 3.6 4.0'
```

4.0 to 4.2

```
docker run -i --rm -v /data/mongodb:/data/mongodb nocoly/hap-sc-  
upgrade:1.0.0 <<< 'upgradeMongodb.sh 4.0 4.2'
```

4.2 to 4.4

```
docker run -i --rm -v /data/mongodb:/data/mongodb nocoly/hap-sc-  
upgrade:1.0.0 <<< 'upgradeMongodb.sh 4.2 4.4'
```

5. Replace the existing package with the installation package of v4.4.29

```
mv /usr/local/mongodb /usr/local/mongodb.bak-v3.4  
cp -r mongodb-linux-x86_64-rhel70-4.4.29 /usr/local/mongodb  
chown -R mongodb:mongodb /usr/local/mongodb
```

- Copy the installation package for MongoDB v4.4.29, the actual name may vary

6. Modify the MongoDB startup command parameters, and add the `--bind_ip 0.0.0.0` parameter

```
vim /etc/systemd/system/mongodb.service  
  
# Append the `--bind_ip 0.0.0.0` parameter after the ExecStart command
```

7. Reload systemd and start MongoDB

```
systemctl daemon-reload  
systemctl start mongodb
```

8. Log in to MongoDB and query featureCompatibilityVersion

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase admin
> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
```

- If the output is `{ "featureCompatibilityVersion" : { "version" : "4.4" }, "ok" : 1 }`, the upgrade is successful.

## Replica Set

### Preparation

1. Download the necessary MongoDB installation packages for each version and upload them to the servers.

**RedHat / CentOS 7.0 x64**

RedHat / CentOS 8.0 x64

Debian 10.0+ x64

Others

```
https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-3.6.23.tgz
https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-4.0.28.tgz
https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-4.2.25.tgz
https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-4.4.29.tgz
```

2. Unzip the installation packages for each version and check if the `mongod` program is missing dependencies. If dependencies are missing, install them accordingly.

**RedHat / CentOS 7.0 x64**

RedHat / CentOS 8.0 x64

Debian 10.0+ x64

Unzip the installation packages for each version

```
tar xvf mongodb-linux-x86_64-3.6.23.tgz
tar xvf mongodb-linux-x86_64-4.0.28.tgz
tar xvf mongodb-linux-x86_64-rhel70-4.2.25.tgz
tar xvf mongodb-linux-x86_64-rhel70-4.4.29.tgz
```

Check if the `mongod` program in each package is missing dependencies



```
ldd mongodb-linux-x86_64-3.6.23/bin/mongod
ldd mongodb-linux-x86_64-4.0.28/bin/mongod
ldd mmongodb-linux-x86_64-rhel70-4.2.25/bin/mongod
ldd mongodb-linux-x86_64-rhel70-4.4.29/bin/mongod
```

## Start Upgrading

### NOTE

- Before upgrading, stop the microservices and backup the MongoDB data.
- Upgrade the replica set by upgrading each node, starting from the SECONDARY node and ending with the PRIMARY node.

### To upgrade from 3.4 to 3.6:

1. Upgrade the SECONDARY nodes of the replica set one at a time:

Stop the current MongoDB instance and replace the existing installation package with the package for version 3.6

```
systemctl stop mongodb
mv /usr/local/mongodb /usr/local/mongodb.bak-3.4
cp -r mongodb-linux-x86_64-3.6.23 /usr/local/mongodb
chown -R mongodb:mongodb /usr/local/mongodb
```

- Copy the installation package for MongoDB v3.6.23, the actual name may vary

2. Modify the MongoDB startup command parameters, and add the `--bind_ip 0.0.0.0` parameter

```
vim /etc/systemd/system/mongodb.service

# Add the `--bind_ip 0.0.0.0` parameter after the ExecStart command
```

3. Reload systemd and start MongoDB

```
systemctl daemon-reload
systemctl start mongod
```

- After starting, try logging in to the node. If you encounter login issues, check the program status or consider slow startup due to large data volume or low disk performance.

4. The upgrading for two SECONDARY nodes is the same

5. Degrade the PRIMARY node

Log in to the MongoDB primary node and use the `rs.stepDown()` command to downgrade the primary node and force the election of a new primary node.

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase
admin

> rs.stepDown()
```

6. Upgrade the old PRIMARY node following the same steps as upgrading the SECONDARY nodes:

- Stop the current MongoDB instance and replace the existing installation package with the package for MongoDB 3.6.
- Modify the MongoDB startup command parameters, and add the `--bind_ip 0.0.0.0` parameter
- Reload systemd and start MongoDB

7. Log in to the PRIMARY node and set the feature compatibility version to 3.6.

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase
admin

> db.adminCommand( { setFeatureCompatibilityVersion: "3.6" } )
```

- The normal output should be: `{ "ok" : 1 }`

Check the status of each node

```
> rs.status()
```

- It should ideally be one PRIMARY and two SECONDARY nodes.

Query featureCompatibilityVersion

```
> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
```

- If the output is `{ "featureCompatibilityVersion" : { "version" : "3.6" }, "ok" : 1 }`, the upgrade to version 3.6 is successful.

---

## To upgrade from 3.6 to 4.0:

1. Upgrade the SECONDARY nodes of the replica set one at a time:

Stop the current MongoDB instance and replace the existing installation package with the package for version 4.0

```
systemctl stop mongod
mv /usr/local/mongodb /usr/local/mongodb.bak-3.6
cp -r mongodb-linux-x86_64-4.0.28 /usr/local/mongodb
chown -R mongodb:mongodb /usr/local/mongodb
systemctl start mongod
```

- Copy the installation package for MongoDB v4.0.28, the actual name may vary.
- After starting, try logging in to the node. If you encounter login issues, check the program status or consider slow startup due to large data volume or low disk performance.

2. The upgrading for two SECONDARY nodes is the same

3. Degrade the PRIMARY node

Log in to the MongoDB primary node and use the `rs.stepDown()` command to downgrade the primary node and force the election of a new primary node.

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase
admin

> rs.stepDown()
```

4. Upgrade the old PRIMARY node following the same steps as upgrading the SECONDARY nodes
5. Log in to the PRIMARY node and set the feature compatibility version to 4.0.

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase admin  
  
> db.adminCommand( { setFeatureCompatibilityVersion: "4.0" } )
```

- The normal output should be: `{ "ok" : 1 }`

Check the status of each node

```
> rs.status()
```

- It should ideally be one PRIMARY and two SECONDARY nodes.

Query featureCompatibilityVersion

```
> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
```

- If the output is `{ "featureCompatibilityVersion" : { "version" : "4.0" }, "ok" : 1 }`, the upgrade to version 4.0 is successful.

---

## To upgrade from 4.0 to 4.2:

1. Upgrade the SECONDARY nodes of the replica set one at a time:

Stop the current MongoDB instance and replace the existing installation package with the package for version 4.3, and start MongoDB

```
systemctl stop mongod  
mv /usr/local/mongod /usr/local/mongod.bak-4.0  
cp -r mongod-linux-x86_64-rhel70-4.2.25 /usr/local/mongod  
chown -R mongod:mongod /usr/local/mongod  
systemctl start mongod
```

- Copy the installation package for MongoDB v4.2.25, the actual name may vary.

- After starting, try logging in to the node. If you encounter login issues, check the program status or consider slow startup due to large data volume or low disk performance.

2. The upgrading for two SECONDARY nodes is the same

3. Degrade the PRIMARY node

Log in to the MongoDB primary node and use the `rs.stepDown()` command to downgrade the primary node and force the election of a new primary node.

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase admin  
  
> rs.stepDown()
```

4. Upgrade the old PRIMARY node following the same steps as upgrading the SECONDARY nodes

5. Log in to the PRIMARY node and set the feature compatibility version to 4.2

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase admin  
  
> db.adminCommand( { setFeatureCompatibilityVersion: "4.2" } )
```

- The normal output should be: `{ "ok" : 1 }`

Check the status of each node

```
> rs.status()
```

- It should ideally be one PRIMARY and two SECONDARY nodes.

Query featureCompatibilityVersion

```
> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
```

- If the output is `{ "featureCompatibilityVersion" : { "version" : "4.2" }, "ok" : 1 }`, the upgrade to version 4.2 is successful.

## To upgrade from 4.2 to 4.4:

### 1. Upgrade the SECONDARY nodes of the replica set one at a time:

Stop the current MongoDB instance and replace the existing installation package with the package for version 4.4, and start MongoDB

```
systemctl stop mongod
mv /usr/local/mongodb /usr/local/mongodb.bak-4.2
cp -r mongodb-linux-x86_64-rhel70-4.4.29 /usr/local/mongodb
chown -R mongodb:mongodb /usr/local/mongodb
systemctl start mongod
```

- Copy the installation package for MongoDB v4.4.29, the actual name may vary
- After starting, try logging in to the node. If you encounter login issues, check the program status or consider slow startup due to large data volume or low disk performance.

### 2. The upgrading for two SECONDARY nodes is the same

### 3. Degrade the PRIMARY node

Log in to the MongoDB primary node and use the `rs.stepDown()` command to downgrade the primary node and force the election of a new primary node.

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase
admin

> rs.stepDown()
```

### 4. Upgrade the old PRIMARY node following the same steps as upgrading the SECONDARY nodes

### 5. Log in to the PRIMARY node and set the feature compatibility version to 4.4

```
/usr/local/mongodb/bin/mongo -u root -p password --authenticationDatabase
admin

> db.adminCommand( { setFeatureCompatibilityVersion: "4.4" } )
```

- The normal output should be: `{ "ok" : 1 }`

Check the status of each node

```
> rs.status()
```

- It should ideally be one PRIMARY and two SECONDARY nodes.

Query featureCompatibilityVersion

```
> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
```

- If the output is `{ "featureCompatibilityVersion" : { "version" : "4.4" }, "ok" : 1 }`, the upgrade to version 4.4 is successful.

# Mount Configuration File

In Kubernetes deployment mode, mounting a configuration file is done by default through variables, where the configuration file is base64-encoded into variable values, and mounted into the container.

If you want to mount the following to the

`/usr/local/MDPrivateDeployment/integrate/appextensions.json` in the container.

```
{
  "WebhookUrl": "https://api.domain.com/hooks/NjA0NzdjMDNjMGFjMTE3ZGUwMjRjN2Nl",
  "WebhookHeaders": {}
}
```

First, you need to encode the mounting path and the configuration file separately in Base64, either through the command (the steps in the example below) or an [online encoding tool](#).

```
[root@localhost ~]# echo -n
'/usr/local/MDPrivateDeployment/integrate/appextensions.json' | base64 -w0
L3Vzci9sb2NhbcC9NRFByaXZhdGVEZXBs3ltZW50L2ludGVncmF0ZS9hcHBleHRlbnNpb25zLmpzb24=
```

```
[root@localhost ~]# echo -n '{
  "WebhookUrl": "https://api.domain.com/hooks/NjA0NzdjMDNjMGFjMTE3ZGUwMjRjN2Nl",
  "WebhookHeaders": {}
}' | base64 -w0
ewogICJXZWJob29rVXJsIjogImh0dHBz0i8vYXBpLmRvbWVpbi5jb20vaG9va3MvTmpBME56ZGpNRE5qTU
```

After finishing the encoding mentioned above, splice the encoded content with a colon. **The left side of the colon is the encoded path and the right side is the encoded configuration file.**

```
L3Vzci9sb2NhbcC9NRFByaXZhdGVEZXBs3ltZW50L2ludGVncmF0ZS9hcHBleHRlbnNpb25zLmpzb24=:e
```

- What you get now is the variable value that needs to be mounted

Next, set custom variable names such as `ENV_SERVICE_CONFIG_INTEGRATE` ,

`ENV_SERVICE_CONFIG_SS0_JSON` are possible. But **the name of the environment variable must start with `ENV_SERVICE_CONFIG`.**



Configure the defined variable names and encoded variable values in the `config.yaml` file of the microservice (variable values must be enclosed in double quotes).

```
ENV_SERVICE_CONFIG_INTEGRATE:  
"L3Vzci9sb2NhbC9NRFByaXZhdGVEZXBs3ltZW50L2ludGVncmF0ZS9hcHBleHRlbnNpb25zLmpzb24=:"
```

After the configuration, restart the microservice. After this, you can enter any microservice container to check whether the mounting path and configuration file are correct.

---

The above is the way to mount a configuration file into a container in Kubernetes deployment mode, you can take this as a model to mount multiple configuration files as long as the custom variable names are not duplicated.

# MongoDB Deployment (Single Node)

## Start Deployment

1. Download the MongoDB installation package and unzip it to the installation directory

RedHat / CentOS 7.0 x64

RedHat / CentOS 8.0 x64

Debian 10.0+ x64

Other

```
wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-rhel70-4.4.29.tgz
tar -zxvf mongodb-linux-x86_64-rhel70-4.4.29.tgz
mv mongodb-linux-x86_64-rhel70-4.4.29 /usr/local/mongodb
```

2. Create MongoDB user

```
useradd -M -s /sbin/nologin mongodb
```

3. Create data and log directories and assign permissions

```
mkdir -p /data/mongodb/ /data/logs/mongodb
chown -R mongodb:mongodb /usr/local/mongodb/ /data/mongodb/ /data/logs
```

4. Configure the systemd management file

```
cat > /etc/systemd/system/mongodb.service <<EOF
[Unit]
Description=MongoDB
[Service]
User=mongodb
Group=mongodb
LimitNOFILE=1000000
LimitNPROC=1000000
ExecStart=/usr/local/mongodb/bin/mongod --logpath
/data/logs/mongodb/mongodb.log --dbpath /data/mongodb --auth --port 27017 --
bind_ip 0.0.0.0
ExecStop=/usr/bin/kill \${MAINPID}
Restart=on-failure
[Install]
```

```
WantedBy=multi-user.target
EOF
```

## 5. Auto-start on boot

```
# After installation, no user has been created yet, so don't use systemctl
start mongodb to start the service first
systemctl daemon-reload
systemctl enable mongodb
```

# Create Database User

## 1. Temporarily start a mongodb service without connection authentication enabled

```
su -c '/usr/local/mongodb/bin/mongod --fork --logpath
/usr/local/mongodb/mongodb.log --dbpath /data/mongodb --noauth --port
27017' -s /bin/bash mongodb
```

## 2. Create a user

```
/usr/local/mongodb/bin/mongo <<<'use admin
db.createUser({user:"root",pwd:"bxfC5J3HuYaY",roles:
[{'role':"root",db:"admin"}]})'
```

- The root user specified in the command is the administrator of MongoDB, with the password `bxfC5J3HuYaY`. Please replace it with the actual password during deployment.

## 3. Shut down the temporarily started MongoDB

```
kill $(pgrep -f 'mongod')
```

# Start MongoDB

```
systemctl start mongodb
```

# Convert MongoDB Single Node to Replica Set

Based on the HAP system architecture, a MongoDB single node involves either **built-in MongoDB** using a storage component mirror or **single-deployed MongoDB**. It also needs to consider whether authentication is enabled. In summary, there are the following 4 cases:

- Built-in MongoDB with authentication disabled
- Built-in MongoDB with authentication enabled
- External MongoDB with authentication disabled
- External MongoDB with authentication enabled

**Built-in MongoDB  
with authentication  
disabled**

**Built-in MongoDB  
with authentication  
enabled**

**External MongoDB  
with authentication  
disabled**

**External MongoDB  
with authentication  
enabled**

Built-in MongoDB refers to the MongoDB component included in the hap-sc:3.0.0 mirror.

1. Add `ENV_MONGODB_DAEMON_ARGS: "--replSet sc-mongodb"` in the docker-compose.yaml file as follows:

After adding `ENV_MONGODB_DAEMON_ARGS`, it will automatically convert to replica set mode. The built-in script is: `rs.initiate({_id: "sc-mongodb",members:[ {_id : 1, host : "sc:27017"} ]})` (If there are other clients connecting to this MongoDB, you need to add a host resolution record for the sc service name on the client.)

```
services:
  app:
    image: nocoly/hap-community:6.0.2
    environment: &app-environment
    ENV_MONGODB_DAEMON_ARGS: "--replSet sc-mongodb"
```

2. Restart the service and execute the following in the root directory of the manager:

```
bash ./service.sh restartall
```

# Create MongoDB Database

As iterates, it may adopt new databases for storage. If database access authentication is enabled, you will need to create the required database in advance.

The following are instructions on how to create a database in two different modes of user's role and permission settings:

- Separate user authentication for each database: it is necessary to create the same user as for the other databases using a user with the `admin` role, for example:

```
mongo -u username -p password --authenticationDatabase admin
```

```
use newdbname
```

```
db.createUser({ user: "Change to the same username as in the other  
databases", pwd: "Change to the same password as in the other databases",  
roles: [{ role: "readWrite", db: "newdbname" }] })
```

- The same user authentication for all databases: you need to modify the permissions of the user and authorize new databases.

# Common Commands in MongoDB

## Check Size of Tables in a Database

To check the size of tables in a database, use the following command to access the desired database:

```
use database_name
```

- Replace the database\_name with the actual name of the database you want to access, for example:

```
use mdworkflow.
```

```
function getReadableFileSizeString(fileSizeInBytes) {
  var i = -1;
  var byteUnits = [' kB', ' MB', ' GB', ' TB', ' PB', ' EB', ' ZB', ' YB'];
  do {
    fileSizeInBytes = fileSizeInBytes / 1024;
    i++;
  } while (fileSizeInBytes > 1024);

  return Math.max(fileSizeInBytes, 0.1).toFixed(1) + byteUnits[i];
};
var collectionNames = db.getCollectionNames(), stats = [];
collectionNames.forEach(function (n) {
  var stat = db[n].stats();
  var totalSize = stat.storageSize + stat.totalIndexSize;
  stats.push({
    ns: stat.ns,
    count: stat.count,
    totalSize: totalSize
  });
});
stats = stats.sort(function(a, b) { return b['totalSize'] - a['totalSize']; });
for (var c in stats) {
  print(stats[c]['ns'] + " , " + stats[c]['count'] + " , " +
  getReadableFileSizeString(stats[c]['totalSize']) + "");
}
```

- After running the above command, it will display the size of tables in the current database.

# Slow Query Optimization

## What is a slow query

A slow query refers to a query that takes a long time to execute.

Usually, database queries are completed quickly, but when the data volume in a table is very large (e.g. tens of thousands of rows or more), frequent querying from these tables can increase the burden on the database. This may lead to excessive use of hardware resources, such as high CPU usage, excessive memory usage, and high system load. Therefore, query operations become slow, taking several hundred milliseconds or even seconds to complete.

To optimize queries, we can take some measures, such as:

- Index optimization: Create appropriate indexes to speed up queries.
- Query optimization: Write efficient query methods to avoid unnecessary computations and data transfers.
- Hardware optimization: Increase the hardware resources of servers to handle larger data sets.
- Multi-cluster data storage: [Application data multi-cluster storage](#) and [Row records multi-cluster storage](#). Increase vertical scalability to alleviate the continuous pressure on a single database from growing business.

## Index Optimization

Visit the [HAP Help Center](#) to learn about basics of indexes and index creation and management.

## Query Optimization

When dealing with worksheets containing hundreds of thousands of rows or more, it is important to pay attention to the way queries are executed and to create indexes for the fields involved in the query conditions in advance.

Usually, the best practice is to use precise queries, such as using the "equal to" condition. Especially when the field values are non-repetitive and already indexed, this query method is extremely efficient.

Additionally, there are some query methods that do not involve indexes:

- Negative conditions such as "not equal to," "not contains," "not starting with" in the filter
- Conditions such as "contains," "is empty" in the filter
- Adding "or" condition in the filter
- The search on the top right of the worksheet is regular query and does not use the index
- Selection fields where a single option value has a high percentage, searches will not use the index
- When text fields are sorted by Pinyin, sorting indexes will not be effective

For worksheets with a large amount of data, it is advisable to minimize the use of the above-mentioned large batch query conditions to reduce database load.

## Hardware Optimization

- Increase CPU Cores:
  - MongoDB is a multi-threaded application, so increasing the CPU cores can improve concurrency.
  - Choose high-performance multi-core CPUs for better query and write handling.
- Increase Memory:
  - Memory is crucial for MongoDB performance. Increasing memory can improve cache efficiency and reduce disk I/O.
  - MongoDB uses memory as a working set cache, so having enough memory can reduce disk reads and improve query performance.
- Use High-Performance SSD Disk:
  - SSD disks are faster than traditional mechanical hard drives (HDD), with lower access latency and higher throughput.
  - MongoDB's default storage engine is WiredTiger, which has good support for SSD disks.

## Practical Tips for Index Optimization

### Searching Logs



If the MongoDB process is consuming high CPU, it is usually due to a large number of slow query statements. You can optimize further by checking the MongoDB logs to identify these slow queries.

**HAP Standalone Mode**

**HAP Cluster Mode**

The default path for MongoDB logs is  
`/data/hap/script/volume/data/logs/mongodb.log`.

If a custom data path is sett, please search for the `mongodb.log` file in the actual data path.

## Log Analysis Examples

Here are two examples for log analysis.

### Example 1

```
2023-03-08T15:50:43.075+0800 I COMMAND [conn38294] command
mdwsrows.ws63083dafdf551460042a73 command: find { find:
"ws63083dafdf551460042a73", filter: { 63e0bafdaf62a19751ee00: "4115fd005",
status: 1.0 }, sort: { utime: -1 }, projection: { _id: 0, status: 0, sharerange:
0, wsutime: 0, keywords: 0, discussunreads: 0, users: 0, owners: 0, unreads: 0
}, limit: 1 } planSummary: COLLSCAN keysExamined:0 docsExamined:381665
hasSortStage:1 cursorExhausted:1 numYields:2981 nreturned:1 reslen:1456 locks:{
Global: { acquireCount: { r: 5964 } }, Database: { acquireCount: { r: 2982 } },
Collection: { acquireCount: { r: 2982 } } } protocol:op_query 1023ms
```

Meaning of Fields:

- `mdwsrows`: database
- `ws63083dafdf551460042a73`: sheet name
- `command`: details of the operation command
- `find`: query criteria
- `63e0bafdaf62a19751ee00`: field (control ID)
- `COLLSCAN`: full sheet scan
- `docsExamined:381665`: total records in the sheet
- `op_query 1023ms`: duration

Log Interpretation: The log indicates a query on the ws63083dafdf551460042a73 worksheet for the field 63e0bafdaf62a19751ee00 with a value of 4115fd005, resulting in a total duration of 1023 milliseconds due to a full sheet scan.

Suggested Indexing: Consider adding an index to the field 63e0bafdaf62a19751ee00.

## Example 2

```
2023-04-25T08:12:10.736+0000 I  COMMAND [conn1696545] command
mdwsrows.ws629f242342bb5f060f3da4 command: aggregate { aggregate:
"ws629f242342bb5f060f3da4", pipeline: [ { $match: { 62c9c6076c186b941274189f:
"90123c1d-1498-43a4-b96b-001a69fd4bb9", 62ccea346123b943f747acf: { $in: [
"0cda5ce1-32139-4c52-b2f4-b324545ccfd1", "2e0e92ec-a935-43fa-a4a3-5fbdee85c0c0"
] } }, $and: [ { 62c9c6076c186b941274189f: "90123c1d-1498-43a4-b96b-001a69fd4bb9"
}, { 62ccea346123b943f747acf: { $in: [ "0cda5ce1-32139-4c52-b2f4-b324545ccfd1",
"2e0e92ec-a935-43fa-a4a3-5fbdee85c0c0" ] } } } ], status: 1.0 } }, { $group: {
_id: "null", count: { $sum: 1 } } } ], cursor: {}, $db: "mdwsrows", lsid: { id:
UUID("d09a5780-0115-4182-8eb2-9de7d76fd834") }, $clusterTime: { clusterTime:
Timestamp(1682410329, 2213), signature: { hash: BinData(0,
6B87048FC23C103EE23E4E026B9B246CD45F), keyId: 720303239516495874 } } }
planSummary: COLLSCAN keysExamined:0 docsExamined:468021 cursorExhausted:1
numYields:3656 nreturned:1 queryHash:97671D20 planCacheKey:97671D20 reslen:282
locks:{ ReplicationStateTransition: { acquireCount: { w: 3658 } }, Global: {
acquireCount: { r: 3658 } }, Database: { acquireCount: { r: 3658 } },
Collection: { acquireCount: { r: 3658 } }, Mutex: { acquireCount: { r: 2 } } }
storage:{ data: { bytesRead: 12110018, timeReadingMicros: 11575 } }
protocol:op_msg 1043ms
```

### Meaning of Fields:

- mdwsrows: database
- ws629f242342bb5f060f3da4: sheet name
- command: details of the operation command
- aggregate: aggregation operation
- \$match: filter
- 62c9c6076c186b941274189f: field (control ID)
- 62ccea346123b943f747acf: field (control ID)
- \$in: include
- COLLSCAN: full sheet scan
- docsExamined:468021: total records in the sheet

- op\_msg 1043ms: duration

Log Interpretation: In the ws629f242342bb5f060f3da4 sheet in the mdwsrows database, take aggregation operation to query data that meet the following conditions: the value of the 62c9c6076c186b941274189f field is "90123c1d-1498-43a4-b96b-001a69fd4bb9", and the value of the 62ccea346123b943f747acf field contains "0cda5ce1-32139-4c52-b2f4-b324545ccfd1" and "2e0e92ec-a935-43fa-a4a3-5fbdee85c0c0". Full sheet scan, and total time spent is 1043 milliseconds.

Suggested Indexing: Consider adding a composite index for the 62c9c6076c186b941274189f and 62ccea346123b943f747acf fields.

## Correspondence of Field IDs in Logs to Fields in Worksheet

After obtaining the worksheet ID, you can access the worksheet page by visiting

`systemURL/worksheet/worksheetID`.

For example, if the sheet name is known in the log as `ws643e4f73c70d5c2cc7285ce3`, remove the `ws` at the beginning and `643e4f73c70d5c2cc7285ce3` is the worksheet ID.

By visiting `systemURL/worksheet/643e4f73c70d5c2cc7285ce3`, you can access the corresponding worksheet.

Once in the worksheet, the App Admin can see the API documentation as shown below.

Mega CRM

DashboardSalesFinanceCustomerConfiguration

Receivable Plan

Receivable Write-Off

Payment Registration

Invoice Application

Commission Details

New

Edit Name and Icon

Navigation

Edit Description

Usage Analysis

Log

Copy

App Management

</>

API Documentation

Plan

Pending PaymentPartial PaymentFull Payment

select

Agreed Payment

Payment Status	Order	Receivable
Full payment	OR-202403170004	480,000.00
Full payment	OR-202403170011	72,000.00
Pending payment	OR-202403170014	800,000.00
Pending payment	OR-202403170010	62,900.00
Partial payment	OR-202403170013	1,836,000.00
Full payment	OR-202403280001	16,000.00

The API documentation includes a field table where you can see the correspondence between field IDs and field names.



## Overview

## Request format

## Authorization management

## IP Whitelist

## Obtain app info

## ▼ Worksheet

Create Worksheet

Get worksheet structure

## ▼ Leads

[Fields comparison table](#)

Get list POST

New Record POST

Create records in batch POST

Obtain the Record GET

Get record details POST

Update the Record POST

Update record details in batch P...

Delete Record POST

Get associated record POST

Get the shared link for the record...

Get the total number of rows in th...

## Leads

Worksheet ID.66699b723f027f70f4945483

Worksheet alias.xiansuo

[Setting](#)

## Fields comparison table

[Set field alias](#)

Field ID	Field Name	Type	Control Number	Description
65e5a0937f2f335ca4349609	Basic Info	Subsection	22	Read only, does not support writing and updating
65e5a023e94c805307fa9974	Topic	Text combination	32	Read only, does not support writing and updating
65eff5a0ef3a355dbfd6498a	Source channel	Table association	29	String, multiple records with rows separated by commas (,). Full coverage operation
65e816ca7059c43ba97f9263	Number	Auto number	33	Read only, does not support writing and updating
65f6af24e873a608f2d5fea4	Leads		29	String, multiple records with rows separated by commas (,). Full coverage operation
65e5a023e94c805307fa9975	Customer Name	Text box	2	String
65e5a023e94c805307fa9977	Company size	Single Choice	11	String
6613aed4790920ef1d2add06	Industry	Single Choice	11	String
65e6979faf2cbab9cc8ccaed	Expected number of users	Single Choice	11	String
65e5a023e94c805307fa9978	Contact Name	Text box	2	String
65e5a023e94c805307fa9979	Mobile phone	Mobile phone	3	String

## Add Indexes

## Way 1: Add in the worksheet settings page

After finding the field corresponding to the field ID in the API documentation, go to the worksheet settings page and add an index to the field.

←

Editing the form:Receivable Plan

Edit Fields

Form Setting

Extended functionality

Close

Basic

Submit

Set the text and actions for form submission

Naming

Set title field, record name and field alias

Function Switch

Set enabled system functions and usage scope

Advanced

Business Rules

When the condition is met, change the status of some field or prompt an error

Custom Action

Customize actions that can be performed on records

Printing Template

Customize your printing style

Search Acceleration

Customize worksheet indexes to speed up searches

Search Acceleration

Manually creating suitable indexes for worksheets with large data volumes can speed up worksheet retrieval, up to 5. [Help](#)

+ Create Index

No index

## Way 2: Add in the database

Add an index to the fields mentioned in the examples above, log in to the MongoDB database and execute the following command.

```
use mdwsrows
```

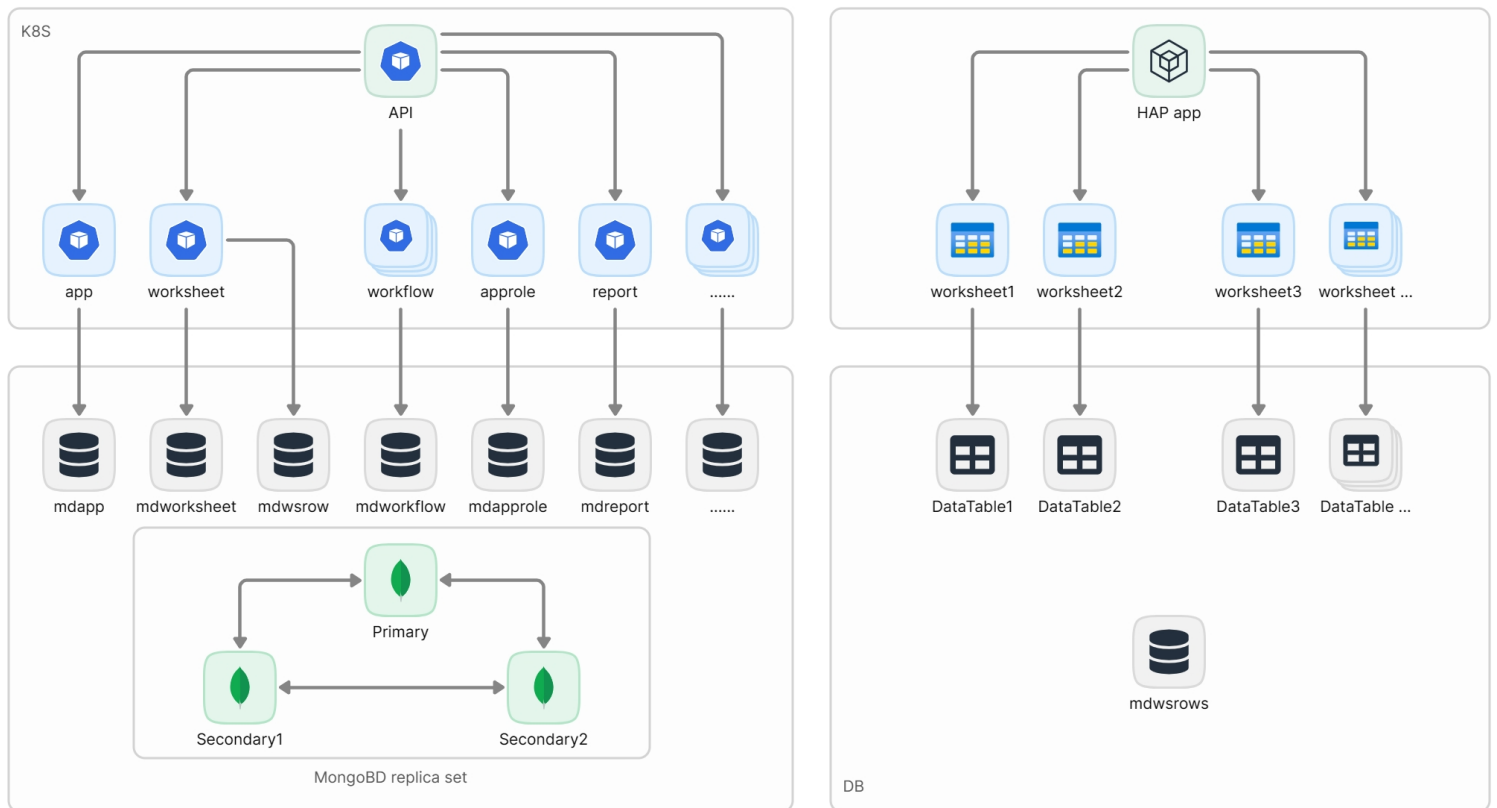
```
db.ws63083dafdf551460042a73.createIndex({"63e0bafdaf62a19751ee00":1},{background: true})

db.ws629f242342bb5f060f3da4.createIndex({"62c9c6076c186b941274189f":1,"62ccea3461274189f":1},{background: true})
```

# Application Data Storage Architecture

The application data in HAP includes modules such as worksheets, workflows, roles and permissions and so on. For application data storage, HAP chooses MongoDB, relying on its flexible storage structure (Bson), which greatly facilitates storage and retrieval performance.

The data of each module is stored in a separate database (You can also deploy an independent database cluster for each database. View more details in [Multi-cluster Storage for Application Data](#)). In addition to the separate database, the application structure data of the same module (non-worksheet) are logically isolated (based on the application id). There is a database table that corresponds to each worksheet in the application physically. Based on the visual field definition capability of the worksheet, drag and drop to dynamically adjust the database table structure. At the same time, the physical isolation of each table makes it more convenient to add necessary field indexes to improve the retrieval performance.

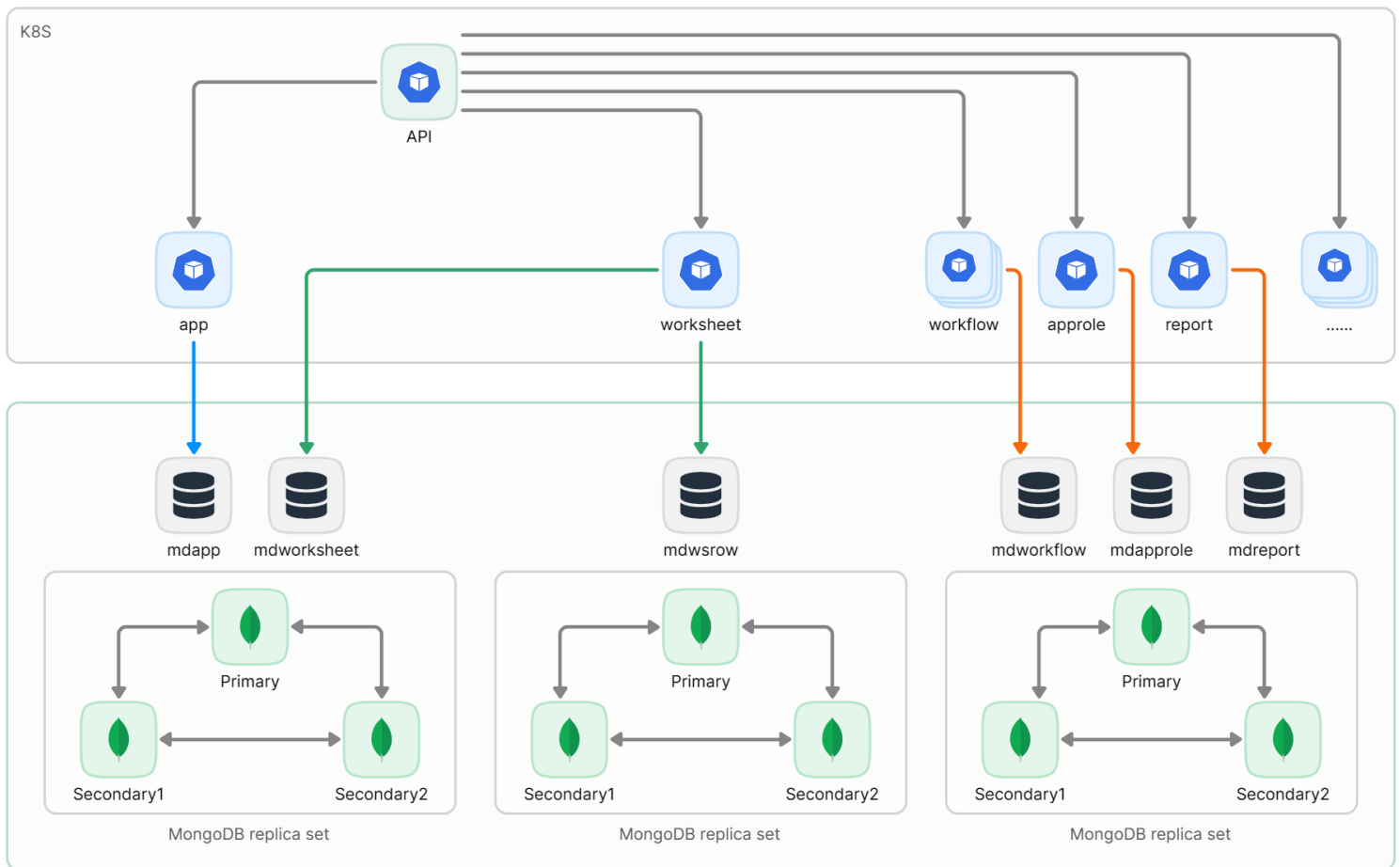


# Multi-cluster Storage for Application Data

In the default deployment mode, only one MongoDB cluster is required for HAP system. View more details in [Application Data Storage Architecture](#).

However, with the increasing amount of data, there is a gradual increase in database resource utilization, which can lead to bottlenecks in a single cluster. To better cope with this, HAP provides a multi-cluster mode, also known as vertical partitioning. In this mode, databases are arranged in different clusters, which can be one database corresponding to one cluster.

With the multi-cluster mode, users can better optimize the utilization of database resources and improve the performance and scalability of the system to better face the challenges of data growth.

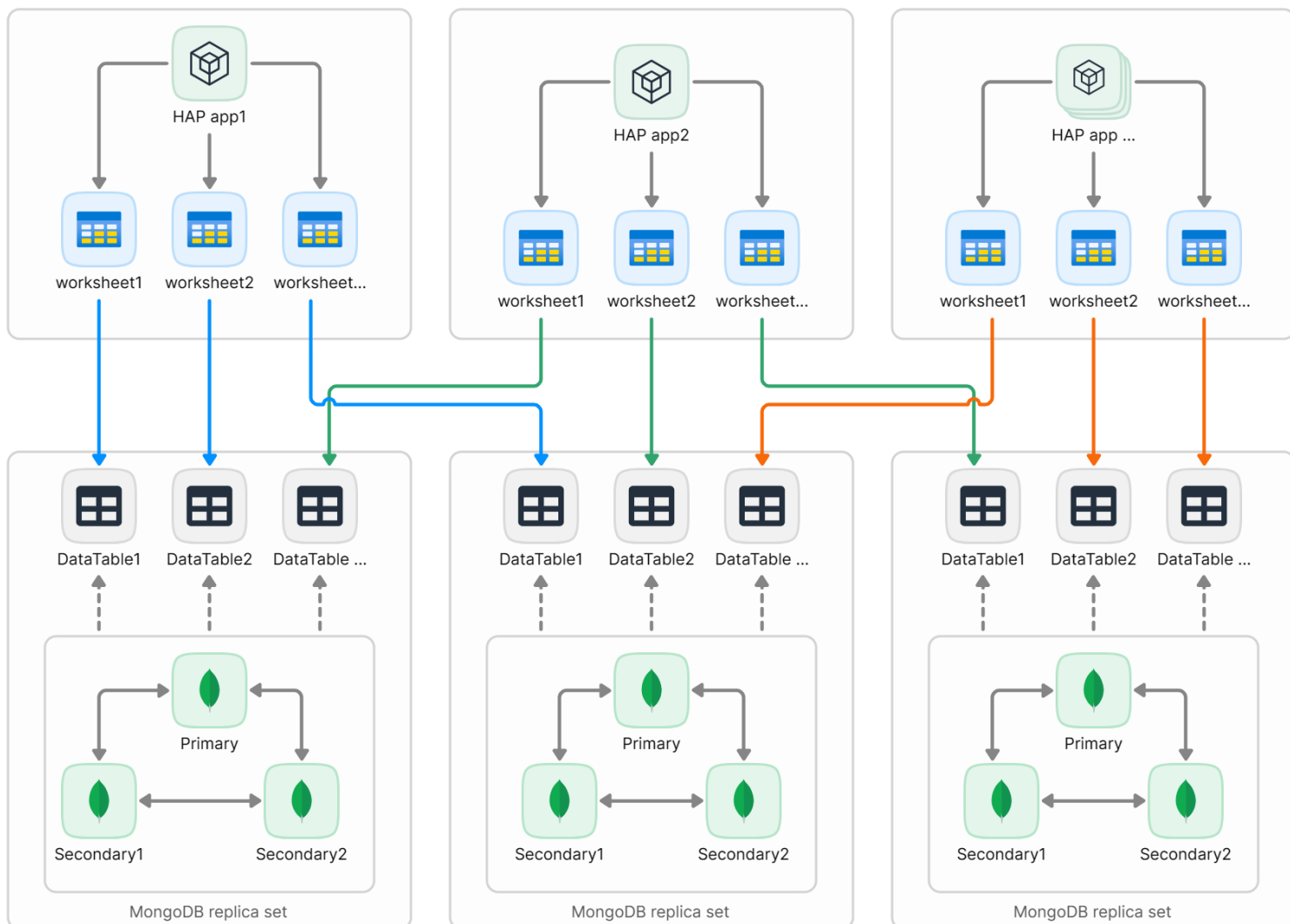




# Multi-cluster Storage for Row Record

Worksheet is the module with the largest volume of application core data in HAP, as described in [Application Data Store Architecture](#), you can see that each worksheet physically corresponds to a database table. In MongoDB, there is an upper limit to the number of tables (Collection) that can be created in each database instance. Though this upper limit can be modified, considering the resource consumption of the database instance itself, it is not recommended to set it too large. With the increase of tables, you can choose to separate the database in the current cluster or choose a new cluster, but separating databases in the same cluster still consumes resources of the current cluster, which is not a good choice in the long run, so HAP currently chooses a multi-cluster mode.

The implementation is based on a routing table (table id + cluster instance number of the row record database) that lists in which cluster instance the corresponding row record database table for each worksheet is stored.



# Application Data Cleaning

Application data includes application information, application grouping, application roles, worksheet information, worksheet data, and so on. In order to ensure data security, the deletion of important data in the system is logical deletion. However, with the increase of deleted data, in addition to occupying disk storage space, it may also have some impact on database performance. Therefore, HAP provides application data cleaning service to physically delete the deleted data regularly according to the set rules.

## 1. Download mirror

Internet Access Available

Internet Access Unavailable

```
docker pull nocoly/hap-dataclean-app:1.0.0
```

## 2. Docker Swarm initialization (skip this step if already initialized)

```
docker swarm init
```

## 3. Create the `dataclean.yaml` file in any directory with the following contents:

```
version: '3'

services:
  app:
    image: nocoly/hap-dataclean-app:1.0.0
    volumes:
      - ./appsettings.json:/app/Config/appsettings.json
```

## 4. Create the `appsettings.json` configuration file in the directory where the `dataclean.yaml` file is located, with the following contents:

```
{
  "WorksheetConnStr": "mongodb://127.0.0.1:27017/mdworksheet",
  "MD_WSService1": "mongodb://127.0.0.1:27017/mdwsrows",
  "MD_WSService2": "mongodb://127.0.0.1:27017/mdwsrows",
  "MD_WSService3": "mongodb://127.0.0.1:27017/mdwsrows",
  "MD_WSService4": "mongodb://127.0.0.1:27017/mdwsrows",
```

```

"MD_WSService5": "mongodb://127.0.0.1:27017/mdwsrows",
"MD_WSService6": "mongodb://127.0.0.1:27017/mdwsrows",
"MD_WSService7": "mongodb://127.0.0.1:27017/mdwsrows",
"MD_WSService8": "mongodb://127.0.0.1:27017/mdwsrows",
"MD_WSService9": "mongodb://127.0.0.1:27017/mdwsrows",
"MD_WSService10": "mongodb://127.0.0.1:27017/mdwsrows",
"MD_WSLogService1": "mongodb://127.0.0.1:27017/mdworksheetlog",
"ApkConnStr": "mongodb://127.0.0.1:27017/mdapps",
"AppRoleConnStr": "mongodb://127.0.0.1:27017/mdapproles",
"MD.Worksheet.DbName": "mdworksheet",
"MD.WorksheetRow.DbName": "mdwsrows",
"DeleteDay": 60,
"DeleteRowByNormal": 0,
"DeleteRowDay": 60,
"DeleteRowTotal": 10000,
"Corn": "0 0 2 * * ? "
}

```

### Configuration Notes:

- `mongodb://127.0.0.1:27017` Replaced with the actual MongoDB connection address
- `DeleteDay` Delete data from X day(s) ago
- `DeleteRowByNormal` Whether to clean up deleted row records when the worksheet is in normal status (0: No cleaning, 1: Cleaning)
- `DeleteRowDay` When `DeleteRowByNormal=1`, delete row records from X days ago; when `DeleteRowByNormal=0`, the configuration is invalid.
- `DeleteRowTotal` When `DeleteRowByNormal=1`, delete data only when the total number of worksheet row records is greater than X; when `DeleteRowByNormal=0`, the configuration is invalid.
- `Corn` Timed execution of the corn expression. See <https://51tools.info/cron/>, `0 0 2 * * ?` means execute at 2:00 every day.

5. Start data cleaning service and execute it in the directory where `dataclean.yaml` is located

```
docker stack deploy -c dataclean.yaml dataclean
```

The execution logs of timed tasks can be viewed with `docker logs $(docker ps | grep dataclean | awk '{print $1}')`.

If you need to remove the service, execute `docker stack rm dataclean`.

# Clean Log Data

In HAP Server, some log data is stored in MongoDB for a long time, which may lead to a large amount of this type of data occupying a significant portion of the database storage space.

You can use the `show dbs` command in MongoDB to check the size of each database and then use the [command to calculate table size](#) to identify tables that occupy a large amount of storage space.

We provide a solution for cleaning log data. Based on the set rules, data in relevant tables can be physically deleted. **After the deletion is completed, log data for the corresponding time on the page will no longer be displayed. For example, execution records of workflows, records of approval processes (approval processed are also within the scope of workflows), logs of row records in worksheets, and request logs in the integration center.**

Whitelisted Tables for Cleaning:

Database	Table Name	Direct Drop	Table Purpose
mdworkflow	code_catch	Yes	Cached data generated during code block execution
mdworkflow	hooks_catch	Yes	Temporary cache data for triggers
mdworkflow	webhooks_catch	Yes	Cached data generated during Webhook execution
mdworkflow	wf_instance	No	Associated data for main workflow execution history
mdworkflow	wf_subInstanceActivity	No	Associated data for sub-workflow execution history
mdworkflow	wf_subInstanceCallback	No	Associated data for sub-workflow execution history
mdworkflow	app_multiple_catch	No	Data stored when "Direct access" is checked in the "Get Multiple Data"

Database	Table Name	Direct Drop	Table Purpose
			node
mdworkflow	custom_apipackageapi_catch	No	Data returned from calling API integration interface
mdworksheetlog	wslog*	Yes	Log of row records in the corresponding month The format of the worksheet name is wslog+date (e.g., wslog202409)
mdintegration	wf_instance	No	Integration center - request logs
mdintegration	wf_instance_relation	No	Integration center - associated data for request logs
mdintegration	webhooks_catch	No	Integration center - data corresponding to "View details" in request logs
mdintegration	code_catch	No	Integration center - data corresponding to "View details" in request logs
mdintegration	json_catch	No	Integration center - data corresponding to "View details" in request logs
mdintegration	custom_parameter_catch	No	Integration center - data corresponding to "View details" in request logs
mdservicedata	al_actionlog*	Yes	Store the application behavior logs for the corresponding month The format of the worksheet name is

Database	Table Name	Direct Drop	Table Purpose
			al_actionlog+date (e.g., al_actionlog202409)
mdservicedata	al_uselog	No	Log for storage usage analysis

- Tables that can directly `drop` are recommended to be deleted using the `db.collection.drop()` command because dropping them will release the storage space occupied by the corresponding table directly.

For example, the following operation will delete the `code_catch` table under the `mdworkflow` database:

```
use mdworkflow
db.code_catch.drop()
```

- For tables that cannot directly `drop`, refer to the following steps to configure a cleanup task.

## Configure Data Cleanup Task

1. Download the mirror ([offline package download](#))

```
docker pull nocoly/hap-archivetools:1.0.4
```

2. Create a `config.json` configuration file with the following example content:

```
[
  {
    "id": "1",
    "text": "Description",
    "start": "2023-05-31 16:00:00",
    "end": "2023-06-30 16:00:00",
    "src": "mongodb://root:password@192.168.1.20:27017/mdworkflow?
authSource=admin",
    "archive": "",
    "table": "wf_instance",
```

```

        "delete": true,
        "batchSize": 500,
        "retentionDays": 0
    },
    {
        "id": "2",
        "text": "Description",
        "start": "2023-05-31 16:00:00",
        "end": "2023-06-30 16:00:00",
        "src": "mongodb://root:password@192.168.1.30:27017/mdworkflow?
authSource=admin",
        "archive": "",
        "table": "wf_subInstanceActivity",
        "delete": true,
        "batchSize": 500,
        "retentionDays": 0
    }
]

```

- According to the above configuration file format, adjust or add configuration content to clean the data tables as needed.
- Note: The time specified in the configuration file is in Coordinated Universal Time (UTC).
  - UTC: 2023-05-31 16:00:00
    - Converted to UTC+8 (East 8th Zone) time: 2023-06-01 00:00:00 (2023-05-31 16:00:00 + 8 hours)
  - UTC: 2023-06-30 16:00:00
    - Converted to UTC+8 (East 8th Zone) time: 2023-07-01 00:00:00 (2023-06-30 16:00:00 + 8 hours)

### Parameter Description:

```

"id": "Task Identifier ID",
"text": "Description",
"start": "Specify the start time of the archived data, in UTC time zone (if
the value of retentionDays is greater than 0, this configuration will
automatically become invalid), delete data greater than or equal to this
time.",
"end": "Specify the end time of the archived data, in UTC time zone (if the
value of retentionDays is greater than 0, this configuration will
automatically become invalid), delete data before this time.",
"src": "Connection address of the source database",
"archive": "Connection address of the target database (if empty, no

```

```
archiving will be done, only deletion according to the set rules)",  
"table": "Data table",  
"delete": "It is fixed to true; after the archiving task is completed, and  
the number of records verified is correct, clean up the archived data in the  
source database",  
"batchSize": "Number of entries and deletions in a single batch",  
"retentionDays": "It defaults to 0. If greater than 0, it means delete data  
X days ago and enable scheduled deletion mode, the dates specified in start  
and end will automatically become invalid, scheduled to run every 24 hours  
by default"
```

3. Start the archiving service by executing the following in the directory where the `config.json` file is located

```
docker run -d -it -v  
$(pwd)/config.json:/usr/local/MDArchiveTools/config.json -v  
/usr/share/zoneinfo/Etc/GMT-8:/etc/localtime nocoly/hap-archivertools:1.0.4
```

Other:

- Resource Usage: During program operation, there will be a certain amount of resource pressure on the source database, target database, and the device where the program is located. It is recommended to execute in the idle period of the business.
- Viewing Logs:
  - Running in the background (default): Use `docker ps -a` to find the container ID, then execute `docker logs container ID` to view the logs.
  - Running in the foreground: Remove the `-d` parameter, and the logs will be output in real-time to the terminal for easy progress tracking.
- Scheduled Tasks:
  - Set execution interval: You can modify the execution interval in milliseconds by customizing the `ENV_ARCHIVE_INTERVAL` variable, with a default value of 86400000.
- **Reclaim Disk Space**: When data is deleted using the cleanup tool, the disk space occupied by the deleted data is not immediately released, but it is usually reused by the same table.



# Reclaim Disk Space

After deleting data from a MongoDB instance, the storage space used by these deleted data will be marked as free, and subsequently, new data written by the same collection is usually stored directly in this free storage space. However, this free storage space cannot be reused by other collections. These unused free storage spaces are referred to as disk fragments. The more disk fragments there are, the lower the disk utilization.

There are two methods for reclaiming disk space: using the `compact` command and rebuilding data files.

- `compact` command: This is a collection-level operation and requires collection-by-collection compression.
- Rebuild data files: This is a database instance-level operation, performed on the entire database, and is generally more comprehensive.

## compact

### Precautions

- **Please make sure to have a complete backup of the database first.**
- For versions prior to MongoDB 4.4, executing the `compact` command may cause the database associated with the collection to be locked, and read and write operations on that database will be blocked. It is recommended to perform this operation during off-peak business hours or after upgrading the version. For more details on the blocking issue, refer to the [MongoDB official documentation](#).
  - The time required to reclaim disk fragments using the `compact` command is related to the data volume of the collection, system load, disk performance, etc. During execution, there will also be a certain increase in CPU and memory usage.
  - For versions below MongoDB 4.4.9, nodes currently executing the `compact` command will be forced into **RECOVERING** state. If this state persists for an extended period, the node may no longer be able to synchronize with the `PRIMARY` node's data.
  - For versions between MongoDB 4.4.9 and 4.4.17, nodes executing the `compact` command will remain in `SECONDARY` state but will still be unable to synchronize with the `PRIMARY` node's data.

- For versions above MongoDB 4.4.17, when executing the `compact` command, `SECONDARY` nodes will continue to replicate data from the `PRIMARY` node. **(It is recommended to execute the `compact` command on versions above MongoDB 4.4.17)**
- The following conditions may cause the `compact` command to be ineffective, for more details please refer to the [open-source code](#).
  - The size of the physical collection is less than 1 MB.
  - In the first 80% of the storage space in a file, the amount of free storage space is less than 20%; in the first 90% of the storage space in a file, the amount of free storage space is less than 10%.
- When executing the `compact` command, it is possible that the released storage space is less than the free storage space. If this occurs, you can try to repeat the `compact` command to release disk fragments, but it is not recommended to execute the `compact` command frequently.

## Estimated Reclaimed Disk Fragment Space

1. Switch the database to the database where the collection is located.

```
use database_name
```

- `database_name` is the name of the database where the collection is located.

2. View the disk fragment space to be reclaimed for the collection.

```
db.collection_name.stats().wiredTiger["block-manager"]["file bytes available for reuse"]
```

- `collection_name` is the name of the collection.

The returned result is as follows:

```
1485426688
```

This result indicates that the estimated disk fragment space to be reclaimed is 1485426688 bytes.

## Reclaim Disk Fragments for Single Node or Replica Set Instances

## Single Node

A single node instance has only one node, so you only need to execute the `compact` command for this instance.

## Replica Set

Replica set instances have multiple nodes, follow the following steps:

1. Execute the `compact` command on one of the `SECONDARY` nodes. After the `compact` command is completed, repeat this operation on each remaining `SECONDARY` node in sequence.
2. Reassign the primary node. Use the `rs.stepDown()` method on the `PRIMARY` node to trigger the re-election of a new `PRIMARY` node. Once the `PRIMARY` node changes to `SECONDARY` status and a new `PRIMARY` node is successfully elected, then execute the `compact` command.
  - If you need to force the execution of the `compact` command on the `PRIMARY` node, you will need to add the `force` parameter, for example:

```
db.runCommand({compact:"collection_name",force:true})
```

## compact Operation

1. Connect to the database node using the Mongo Shell.
2. Switch the database to the database where the collection is located.

```
use database_name
```

- `database_name` is the name of the database where the collection is located.
3. Specify the collection to execute the `compact` command and reclaim disk fragments.

```
db.runCommand({compact:"collection_name"})
```

- `collection_name` is the name of the collection.

If successful, the return result is as follows:

```
{ "ok" : 1 }
```

```
{ "ok" : 1 }
```

## Rebuild Data Files

### Precautions

- **Before proceeding, please make sure to have a complete backup of the database.**
- The time required to rebuild data files depends on the data volume of the collection, system load, disk performance, etc.

### Single Node

1. Stop the application service.
2. Stop the MongoDB database.
3. Use the `--repair` parameter of `mongod` to rebuild data files and reclaim disk space.

Example:

```
mongod --repair --dbpath /data/mongodb/
```

- `/data/mongodb/` is the MongoDB data storage directory.
- **Do not interrupt the operation during execution, as it may affect data integrity and prevent the database from starting.**

4. Start the MongoDB database.

### Replica Set

Reclaim disk space by deleting data on `SECONDARY` nodes and leveraging MongoDB replica set's internal resynchronization mechanism to rebuild data files.

1. Execute the following command on any `SECONDARY` node to delete the data on the current node (excluding the `keyfile`):

```
find /data/mongodb/ -mindepth 1 ! -name 'keyfile' -exec rm -rf {} +
```

- This command excludes the `keyfile` file in the `/data/mongodb/` directory and deletes all other files and subdirectories.

2. Restart the current MongoDB node

3. Use the `rs.status()` command to check the node status. During the synchronization process, the node status will display as `STARTUP2`, and once synchronization is complete, it will change to `SECONDARY`.

4. After the previous node has completed synchronization and the node status changes to `SECONDARY`, repeat the same operation on the remaining `SECONDARY` nodes in sequence.

5. Finally, on the `PRIMARY` node, use the `rs.stepDown()` method to trigger the re-election. When the `PRIMARY` node changes to `SECONDARY` status and successfully elects a new `PRIMARY` node, you can perform the same operation on that node.

# Execution History of Workflow

To alleviate the continuous storage growth pressure in the `mdworkflow` database of MongoDB, users can choose to archive the execution history of workflows and store them in a new MongoDB instance. The archived execution records can also be viewed on the page.

## Configuration Steps for Archiving

1. Deploy a MongoDB instance in advance for storing archived data
  - We provide a [MongoDB deployment document \(single node\)](#) for reference.
2. Download the mirror ([offline package download](#))

```
docker pull nocoly/hap-archivetools:1.0.4
```

3. Create a `config.json` configuration file with the following example content:

```
[
  {
    "id": "1",
    "text": "Description",
    "start": "2022-12-31 16:00:00",
    "end": "2023-12-31 16:00:00",
    "src": "mongodb://root:password@192.168.1.20:27017/mdworkflow?
authSource=admin",
    "archive":
    "mongodb://root:password@192.168.1.30:27017/mdworkflow_archive_2023?
authSource=admin",
    "table": "wf_instance",
    "delete": true,
    "batchSize": 500,
    "retentionDays": 0
  },
  {
    "id": "2",
    "text": "Description",
    "start": "2022-12-31 16:00:00",
    "end": "2023-12-31 16:00:00",
    "src": "mongodb://root:password@192.168.1.20:27017/mdworkflow?
authSource=admin",
```

```

        "archive":
"mongodb://root:password@192.168.1.30:27017/mdworkflow_archive_2023?
authSource=admin",
        "table": "wf_subInstanceActivity",
        "delete": true,
        "batchSize": 500,
        "retentionDays": 0
    },
    {
        "id": "3",
        "text": "Description",
        "start": "2022-12-31 16:00:00",
        "end": "2023-12-31 16:00:00",
        "src": "mongodb://root:password@192.168.1.20:27017/mdworkflow?
authSource=admin",
        "archive":
"mongodb://root:password@192.168.1.30:27017/mdworkflow_archive_2023?
authSource=admin",
        "table": "wf_subInstanceCallback",
        "delete": true,
        "batchSize": 500,
        "retentionDays": 0
    }
]

```

- According to the above configuration file format, adjust or add configuration content to clean the data tables as needed.
- Note: The time specified in the configuration file is in Coordinated Universal Time (UTC).
  - UTC: 2023-05-31 16:00:00
    - Converted to UTC+8 (East 8th Zone) time: 2023-06-01 00:00:00 (2023-05-31 16:00:00 + 8 hours)
  - UTC: 2023-06-30 16:00:00
    - Converted to UTC+8 (East 8th Zone) time: 2023-07-01 00:00:00 (2023-06-30 16:00:00 + 8 hours)

### Parameter Description:

```

"id": "Task Identifier ID",
"text": "Description",
"start": "Specify the start time of the archived data, in UTC time zone (if
the value of retentionDays is greater than 0, this configuration will
automatically become invalid), archive data greater than or equal to this

```

```
time.",
"end": "Specify the end time of the archived data, in UTC time zone (if the
value of retentionDays is greater than 0, this configuration will
automatically become invalid), archive data less than this time.",
"src": "Connection address of the source database",
"archive": "Connection address of the target database (if empty, no
archiving will be done, only deletion according to the set rules)",
"table": "Data table",
"delete": "It is fixed to true; after the archiving task is completed, and
the number of records verified is correct, clean up the archived data in the
source database",
"batchSize": "Number of entries and deletions in a single batch",
"retentionDays": "It defaults to 0. If greater than 0, it means delete data
X days ago and enable scheduled deletion mode, the dates specified in start
and end will automatically become invalid, scheduled to run every 24 hours
by default"
```

4. Start the archiving service by executing the following in the directory where the `config.json` file is located

```
docker run -d -it -v
$(pwd)/config.json:/usr/local/MDArchiveTools/config.json -v
/usr/share/zoneinfo/Etc/GMT-8:/etc/localtime nocoly/hap-archivetools:1.0.4
```

Other:

- Resource Usage: During program operation, there will be a certain amount of resource pressure on the source database, target database, and the device where the program is located. It is recommended to execute in the idle period of the business.
- Viewing Logs:
  - Running in the background (default): Use `docker ps -a` to find the container ID, then execute `docker logs container ID` to view the logs.
  - Running in the foreground: Remove the `-d` parameter, and the logs will be output in real-time to the terminal for easy progress tracking.
- In the example configuration file `config.json`, name the new database in the format of `source database name_archive_date`. Each time you execute, modify the target database name in



archive.

- Since the program will first count the amount of data in the target table after the archive is completed, if they are not equal, deletion will not occur; if the archive target database name is not modified in the second run, it may result in more data in the target table than in the current archive, preventing the deletion of the source data.
- **Reclaim Disk Space:** After archiving is completed, the corresponding data in the source database will be deleted. The disk space occupied by the deleted data will not be immediately released, but it is typically reused by the same table.

## Configure Visualization of Archived Data

1. Create the `application-www-ext.properties` configuration file

For example: `/data/hap/script/volume/workflow/application-www-ext.properties`

```
spring.data.mongodb.archive.group[0].id=0
spring.data.mongodb.archive.group[0].text=\u5f52\u6863-2023
spring.data.mongodb.archive.group[0].uri=mongodb://root:password@192.168.1.30:27020
authSource=admin
spring.data.mongodb.archive.group[0].start=2023-01-01
spring.data.mongodb.archive.group[0].end=2023-12-31
```

Parameter Description:

- `group[0]`: It defaults to 0 and is incremented for multiple archived data
- `id`: It defaults to 0 and is incremented for multiple archived data
- `text`: Name displayed on the page, Unicode encoding is required if in Chinese
- `uri`: Connection address of the archived database
- `start`: Start date of the archived data
- `end`: End date of the archived data

2. Mount the configuration file

**Standalone Mode**

**Cluster Mode**

Add the following to the `docker-compose.yaml` `volumes` corresponding to the microservice application:

```
- ./volume/workflow/application-www-  
  ext.properties:/usr/local/MDPrivateDeployment/workflow/application-www-  
  ext.properties
```

3. Restart the microservice

# Logs of Row Record in Worksheet

To alleviate the continuous storage growth pressure in the `mdworksheetlog` database of MongoDB, users can choose to archive the logs of row records in a worksheet and store them in a new MongoDB instance. The historical logs that have been archived can also be viewed on the page.

## Configuration Steps for Archiving

1. Deploy a MongoDB instance in advance for storing archived data
  - We provide a [MongoDB deployment document \(single node\)](#) for reference.
2. Download the mirror ([offline package download](#))

```
docker pull nocoly/hap-archivetools:1.0.4
```

3. Create a `config.json` configuration file with the following example content:

```
[
  {
    "id": "1",
    "text": "Description",
    "start": "2022-12-31 16:00:00",
    "end": "2023-12-31 16:00:00",
    "src": "mongodb://root:password@192.168.1.20:27017/mdworksheetlog?authSource=admin",
    "archive": "mongodb://root:password@192.168.1.30:27017/mdworksheetlog_archive_2023?authSource=admin",
    "table": "wslog*",
    "delete": true,
    "batchSize": 500,
    "retentionDays": 0
  }
]
```

- According to the above configuration file format, adjust or add configuration content to clean the data tables as needed.
- Note: The time specified in the configuration file is in Coordinated Universal Time (UTC).
  - UTC: 2023-05-31 16:00:00

- Converted to UTC+8 (East 8th Zone) time: 2023-06-01 00:00:00 (2023-05-31 16:00:00 + 8 hours)
- UTC: 2023-06-30 16:00:00
  - Converted to UTC+8 (East 8th Zone) time: 2023-07-01 00:00:00 (2023-06-30 16:00:00 + 8 hours)

### Parameter Description:

```
"id": "Task Identifier ID",
"text": "Description",
"start": "Specify the start time of the archived data, in UTC time zone (if the value of retentionDays is greater than 0, this configuration will automatically become invalid), archive data greater than or equal to this time.",
"end": "Specify the end time of the archived data, in UTC time zone (if the value of retentionDays is greater than 0, this configuration will automatically become invalid), archive data less than this time.",
"src": "Connection address of the source database",
"archive": "Connection address of the target database (if empty, no archiving will be done, only deletion according to the set rules)",
"table": "Data table",
"delete": "It is fixed to true; after the archiving task is completed, and the number of records verified is correct, clean up the archived data in the source database",
"batchSize": "Number of entries and deletions in a single batch",
"retentionDays": "It defaults to 0. If greater than 0, it means delete data X days ago and enable scheduled deletion mode, the dates specified in start and end will automatically become invalid, scheduled to run every 24 hours by default"
```

4. Start the archiving service by executing the following in the directory where the `config.json` file is located

```
docker run -d -it -v
$(pwd)/config.json:/usr/local/MDArchiveTools/config.json -v
/usr/share/zoneinfo/Etc/GMT-8:/etc/localtime nocoly/hap-archivetools:1.0.4
```

### Other:

- Resource Usage: During program operation, there will be a certain amount of resource pressure on the source database, target database, and the device where the program is located. It is

recommended to execute in the idle period of the business.

- Viewing Logs:
  - Running in the background (default): Use `docker ps -a` to find the container ID, then execute `docker logs container ID` to view the logs.
  - Running in the foreground: Remove the `-d` parameter, and the logs will be output in real-time to the terminal for easy progress tracking.
- In the example configuration file `config.json`, name the new database in the format of `source database name_archive_date`. Each time you execute, modify the target database name in archive.
  - Since the program will first count the amount of data in the target table after the archive is completed, if they are not equal, deletion will not occur; if the archive target database name is not modified in the second run, it may result in more data in the target table than in the current archive, preventing the deletion of the source data.
- **Reclaim Disk Space:** After archiving is completed, the corresponding data in the source database will be deleted. The disk space occupied by the deleted data will not be immediately released, but it is typically reused by the same table.

## Configure Visualization of Archived Data

1. Create the `appextensions.json` configuration file

For example: `/data/hap/script/volume/worksheet/appextensions.json`

```
{
  "Log.Archive": [
    {
      "id": "1",
      "text": "Archive-2023",
      "uri":
"mongodb://root:password@192.168.1.30:27017/mdworksheetlog_archive_2023?
authSource=admin;socketTimeout=30s",
      "start": "2023-01-01",
      "end": "2023-12-31"
    }
  ]
}
```

```
]
}
```

#### Parameter Description:

- id: It defaults to 1 and is incremented for multiple archived data
- text: Name displayed on the page
- uri: Connection address of the archived database
- start: Start date of the archived data
- end: End date of the archived data

#### 2. Mount the configuration file

**Standalone Mode**

**Cluster Mode**

Add the following to the `docker-compose.yaml` `volumes` corresponding to the microservice application:

```
-
  ./volume/worksheet/appextensions.json:/usr/local/MDPrivateDeployment/worksheet/
```

#### 3. Restart the microservice

# Application Behavior Logs

To alleviate the continuous storage growth pressure in the `mdservicedata` database of MongoDB, you can choose to archive the application behavior logs and store them in a new MongoDB instance. Archived historical logs can also be selected and viewed on the page.

## Configuration Steps for Archiving

1. Deploy a MongoDB instance in advance for storing archived data
  - We provide a [MongoDB deployment document \(single node\)](#) for reference.
2. Download the mirror ([offline package download](#))

```
docker pull nocoly/hap-archivetools:1.0.4
```

3. Create a `config.json` configuration file with the following example content:

```
[
  {
    "id": "1",
    "text": "Description",
    "start": "2022-12-31 16:00:00",
    "end": "2023-12-31 16:00:00",
    "src": "mongodb://root:password@192.168.1.20:27017/mdservicedata?authSource=admin",
    "archive": "mongodb://root:password@192.168.1.30:27017/mdservicedata_archive_2023?authSource=admin",
    "table": "al_actionlog*",
    "delete": true,
    "batchSize": 500,
    "retentionDays": 0
  }
]
```

- According to the above configuration file format, adjust or add configuration content to clean the data tables as needed.
- Note: The time specified in the configuration file is in Coordinated Universal Time (UTC).
  - UTC: 2023-05-31 16:00:00

- Converted to UTC+8 (East 8th Zone) time: 2023-06-01 00:00:00 (2023-05-31 16:00:00 + 8 hours)
- UTC: 2023-06-30 16:00:00
  - Converted to UTC+8 (East 8th Zone) time: 2023-07-01 00:00:00 (2023-06-30 16:00:00 + 8 hours)

### Parameter Description:

```
"id": "Task Identifier ID",
"text": "Description",
"start": "Specify the start time of the archived data, in UTC time zone (if the value of retentionDays is greater than 0, this configuration will automatically become invalid), archive data greater than or equal to this time.",
"end": "Specify the end time of the archived data, in UTC time zone (if the value of retentionDays is greater than 0, this configuration will automatically become invalid), archive data less than this time.",
"src": "Connection address of the source database",
"archive": "Connection address of the target database (if empty, no archiving will be done, only deletion according to the set rules)",
"table": "Data table",
"delete": "It is fixed to true; after the archiving task is completed, and the number of records verified is correct, clean up the archived data in the source database",
"batchSize": "Number of entries and deletions in a single batch",
"retentionDays": "It defaults to 0. If greater than 0, it means delete data X days ago and enable scheduled deletion mode, the dates specified in start and end will automatically become invalid, scheduled to run every 24 hours by default"
```

4. Start the archiving service by executing the following in the directory where the `config.json` file is located

```
docker run -d -it -v
$(pwd)/config.json:/usr/local/MDArchiveTools/config.json -v
/usr/share/zoneinfo/Etc/GMT-8:/etc/localtime nocoly/hap-archivetools:1.0.4
```

Other:



- Resource Usage: During program operation, there will be a certain amount of resource pressure on the source database, target database, and the device where the program is located. It is recommended to execute in the idle period of the business.
- Viewing Logs:
  - Running in the background (default): Use `docker ps -a` to find the container ID, then execute `docker logs container ID` to view the logs.
  - Running in the foreground: Remove the `-d` parameter, and the logs will be output in real-time to the terminal for easy progress tracking.
- In the example configuration file `config.json`, name the new database in the format of `source database name_archive_date`. Each time you execute, modify the target database name in archive.
  - Since the program will first count the amount of data in the target table after the archive is completed, if they are not equal, deletion will not occur; if the archive target database name is not modified in the second run, it may result in more data in the target table than in the current archive, preventing the deletion of the source data.
- **Reclaim Disk Space:** After archiving is completed, the corresponding data in the source database will be deleted. The disk space occupied by the deleted data will not be immediately released, but it is typically reused by the same table.

## Clean Up Historical Data in Elasticsearch

**Standalone Mode**

**Cluster Mode**

1. Enter the microservice container

```
docker exec -it $(docker ps | grep community | awk '{print $1}') bash
```

2. Clean up historical data in Elasticsearch

```
source /entrypoint.sh && deleteActionlog "2022-12-31 16:00:00" "2023-12-31 16:00:00"
```

- Please ensure the time range in the above command matches the time range configured in the corresponding archival task.

## Configure Visualization of Archived Data

1. Create the `application-www-ext.properties` configuration file

For example: `/data/hap/script/volume/actionlog/application-www-ext.properties`

```
spring.data.mongodb.archive.group[0].id=0
spring.data.mongodb.archive.group[0].text=\u5e94\u7528\u884c\u4e3a\u65e5\u5fd7-
spring.data.mongodb.archive.group[0].uri=mongodb://root:password@192.168.1.30:2
authSource=admin
spring.data.mongodb.archive.group[0].start=2023-01-01
spring.data.mongodb.archive.group[0].end=2023-12-31
```

Parameter Description:

- group[0]: It defaults to 0 and is incremented for multiple archived data
- id: It defaults to 0 and is incremented for multiple archived data
- text: Name displayed on the page, Unicode encoding is required if in Chinese
- uri: Connection address of the archived database
- start: Start date of the archived data
- end: End date of the archived data

2. Mount the configuration file

**Standalone Mode**

**Cluster Mode**

Add the following to the `docker-compose.yaml` `volumes` corresponding to the microservice application:

```
- ./volume/actionlog/application-www-
  ext.properties:/usr/local/MDPrivateDeployment/actionlog/application-www-
  ext.properties
```

3. Restart the microservice

# Exclusive Computing Power

## What is Exclusive Computing Power

Exclusive computing power is an independent resource service provided by the platform to ensure the stable and efficient operation of critical workflows. By adding important workflows to exclusive computing power, it avoids the impact of other process blockages on the platform.

Exclusive computing power does not increase the execution speed of the workflows themselves, but ensures that workflows in exclusive computing power can be executed stably without being affected by other workflows.

The core advantages of exclusive computing power are:

- **Independent Resources:** Allocate independent computing resources to each exclusive computing power to ensure that workflows have exclusive resources and are not affected by other workflows.
- **Stable Execution:** Workflows in exclusive computing power have independent Kafka Topic queues, avoiding messages competition in public queues, thus ensuring that workflows can be executed stably and efficiently.
- **Isolation & Protection:** By isolating critical workflows from regular workflows, it effectively reduces the impact of congestion caused by regular workflows.

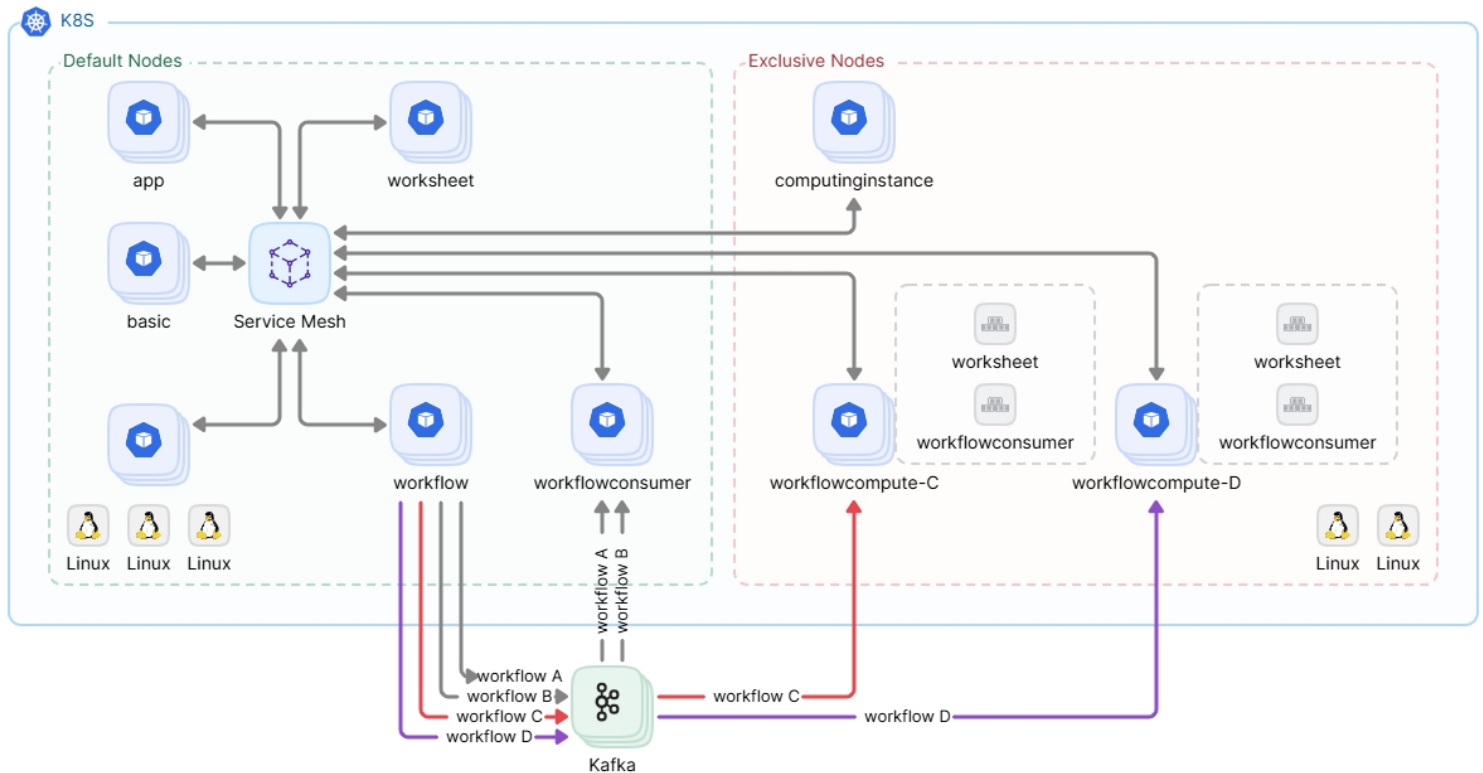
Operation principles:

- **Trigger & Execution:** When a workflow in exclusive computing power is triggered, it enters its independent Kafka Topic queue.
- **Independent Consumption:** The exclusive consumption service will consume messages in the Topic in real-time and execute the workflow according to the preset logic.
- **Resource Guarantee:** The computing resources and storage space used by exclusive computing power are all exclusive resources and will not be occupied by other workflows.

Applicable scenarios:

- **High-priority workflows:** Workflows that require quick response and timely processing.
- **Complex workflows:** Complex, computationally intensive, and bulk-triggered workflows. (Place them in independent high-specification exclusive resources to avoid these complex workflows queuing in public queues for ordinary workflows)

# Deployment Architecture



As shown in the above figure, two new nodes have been added to the existing Kubernetes cluster as dedicated resources for dedicated computing power services. Here, two dedicated computing powers, **workflowcompute-C** and **workflowcompute-D**, have been created. After configuring **workflow C** to use dedicated computing power **workflowcompute-C** for consumption, the execution of **workflow C** will use the resources allocated by **workflowcompute-C**. **workflow D** is the same.

## Use of Exclusive Computing Power

The exclusive computing power service requires an independent key and authorization, and can only be deployed and used in cluster mode.

Once the deployment is completed and the authorization is added, you can choose the instance specifications on the page below for creation.

Org Admin

HAP << < Create Dedicated Computing Power

Home

User

Organization

App Management

App

Workflow

Aggregate Table **beta**

Global Variable

**Proprietary Resources**

Usage Analysis

General Settings

Security

Log

Integration

**Select specifications**

<b>10 concurrency</b> 2 cores (vCPU)   4GiB	<b>20 concurrency</b> 4 cores (vCPU)   8GiB	<b>50 concurrency</b> 8 cores (vCPU)   16GiB	<b>100 concurrency</b> 16 cores (vCPU)   32GiB
--	--	---	---

Concurrency refers to the number of instances that can run simultaneously at a single time.

Active time

Expiration date of the organization

Number of creations - 1 + Number of instances

Confirm

Concurrency in exclusive computing power instance:

- **20 Concurrency:** Represents 20 partitions in an independent Kafka Topic, 20 threads in independent workflow consumption services, and a maximum of 4 cores and 8GB of resources that can be used by independent workflows and worksheets services.
- **100 Concurrency:** Represents 100 partitions in an independent Kafka Topic, 100 threads in independent workflow consumption services, and a maximum of 16 cores and 32GB of resources that can be used by independent workflows and worksheets services.
- In theory, the larger the Kafka Topic partition, the larger the thread size of workflow consumption services, and the more workflows can be processed in parallel.

## How to choose the specifications for deployment servers?

The exclusive computing power service utilizes the features of Kubernetes to run exclusively in servers isolated from microservices, avoiding the impact of resource occupation by other services. Therefore, deploying the exclusive computing power service requires adding independent servers.

First, plan the required number and specifications of exclusive computing power instances, and then plan the configuration and quantity of deployment servers.

For example: If you need to create 2 exclusive computing power instances of 20 concurrency and 2 instances of 50 concurrency, then a minimum of 2 servers with 16 cores and 32GB each is needed, with each server having a fixed disk of 200GB.

# Exclusive Database

## What is an exclusive database

An exclusive database is a database instance that isolates all worksheet data from a specific application from the worksheet data of other applications, protecting it from the influence of the default system database.

Exclusive databases can effectively reduce competition for database resources, allowing applications to have exclusive access to database resources, thereby improving the response time of query and update operations.

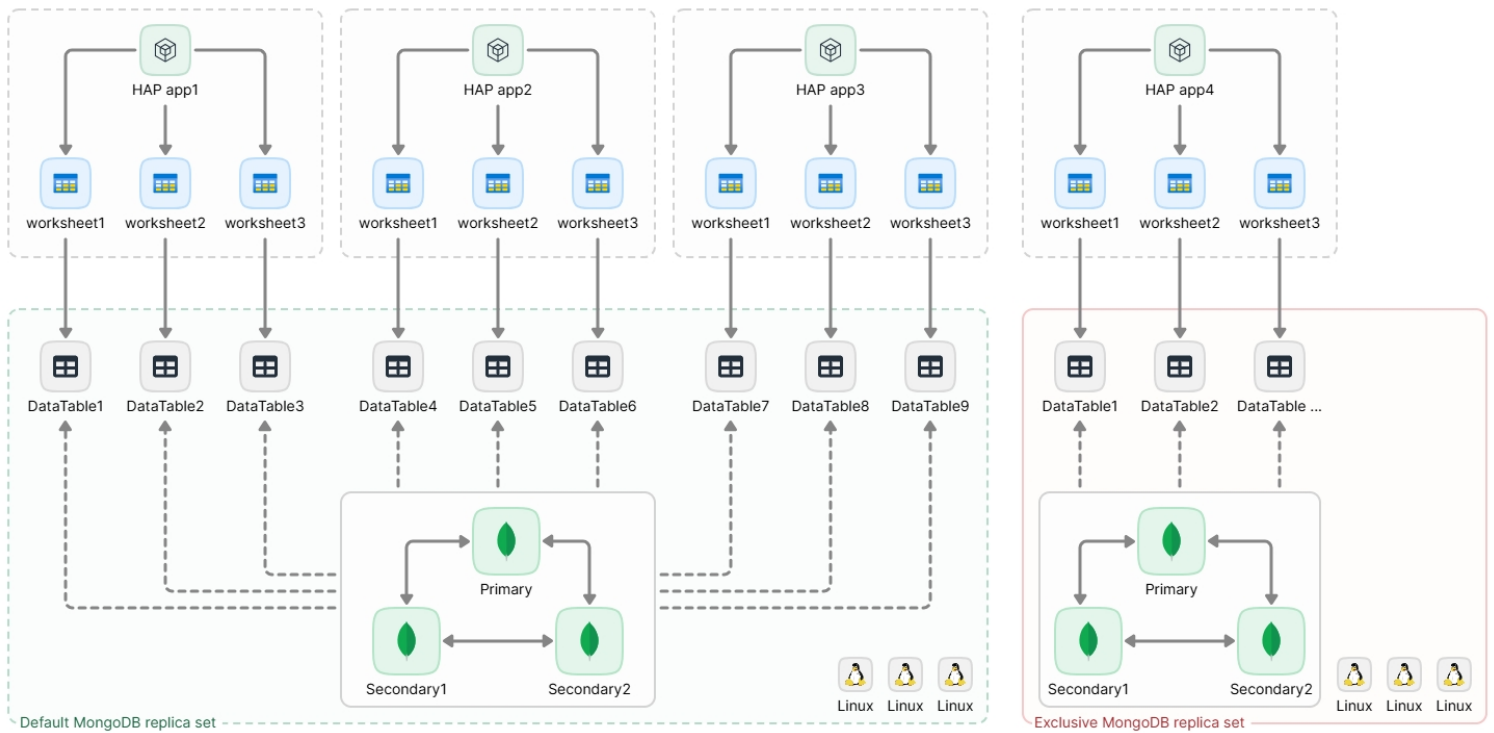
## Core Advantages

- **Data Isolation:** Exclusive databases provide independent physical data storage space for applications.
- **Resource Exclusivity:** Applications exclusively share the computing resources of exclusive databases, reducing competition for resources in the default system database, ensuring overall platform stability and smooth operation.
- **Performance Enhancement:** Due to resource exclusivity, exclusive databases can accelerate the response time of data query and update operations, especially when dealing with large data volume or frequent operations.

## Applicable Scenarios

- **Data Isolation:** When certain applications have high requirements for data isolation, exclusive databases can provide a completely independent storage environment to prevent data overlap.
- **High-load Applications:** Suitable for applications that frequently perform data query and update operations. Exclusive databases can ensure overall stability of the platform under high concurrency.
- **Performance-Sensitive Scenarios:** For applications with high requirements for query and update operation speed, using an exclusive database can significantly optimize performance.

## Deployment Architecture



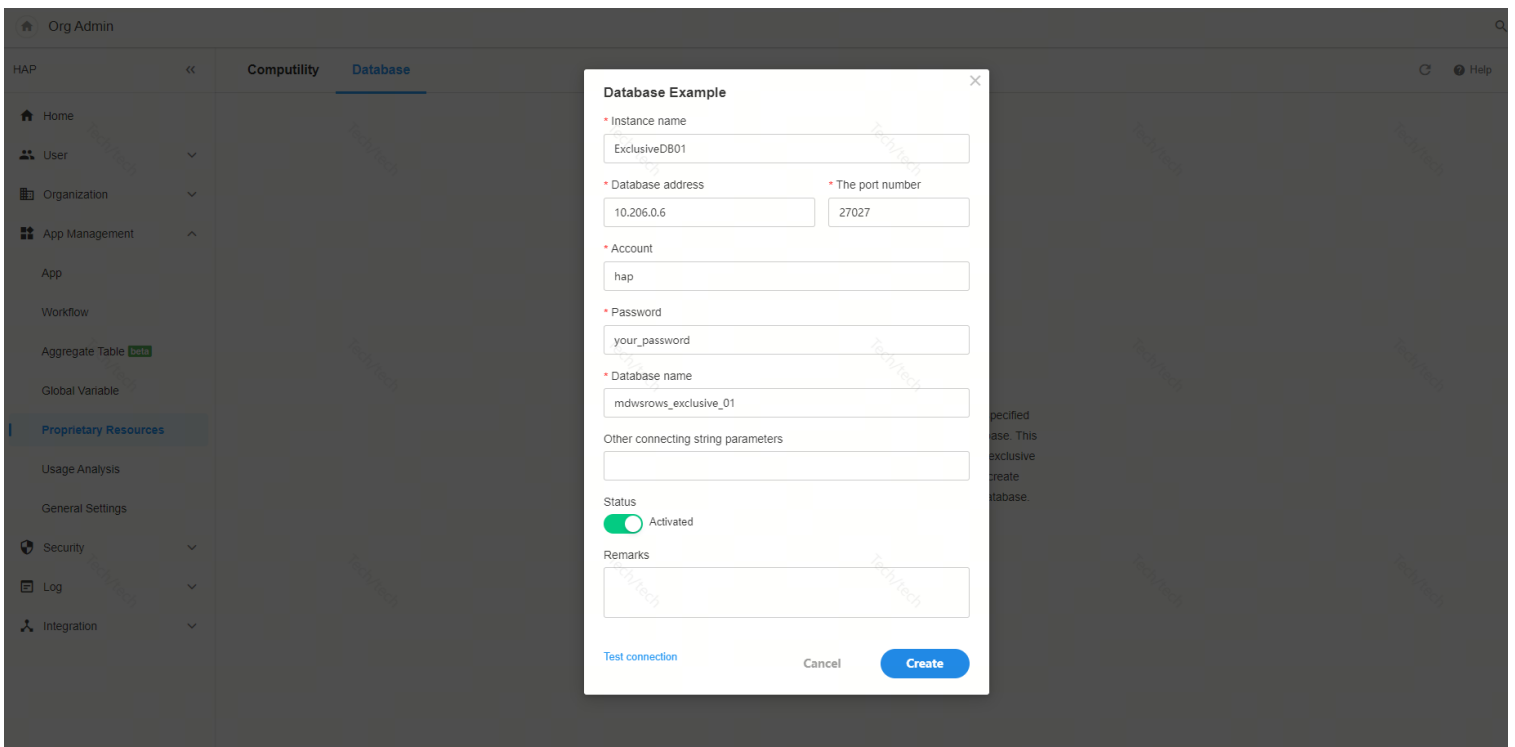
As shown in the above figure, the worksheets in app1, app2, and app3 are stored in a shared database, while the worksheets in app4 are stored in a exclusive database, and the exclusive database cluster is deployed using independent server resources.

## Use of Exclusive Database

### Create Exclusive Database Instances

Creating exclusive database instances requires an independent key and authorization. Prior to creation, it is necessary to deploy a new MongoDB instance, whether it is a single node or a replica set.

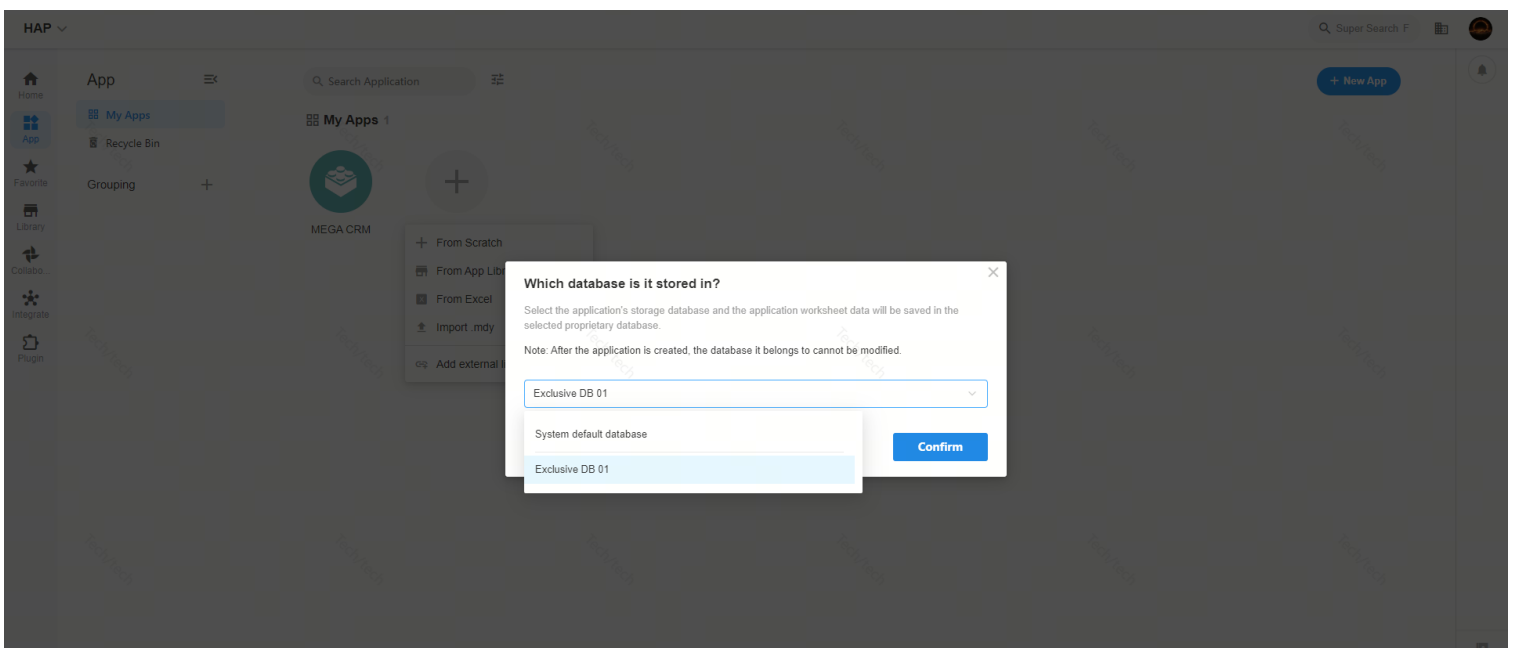
On the page below, create exclusive database instances within the organization. Fill in the configuration information of the database instance as required, perform connection testing to ensure successful connection, and then click "Create" to save.



## Store Application Data in Exclusive Databases

When creating an application from scratch, from Excel or by importing .mdy file, you can store all worksheet data of the application in an exclusive database instance.

By default, the system default database is used when creating an application. If an exclusive database is required, the corresponding database instance needs to be manually selected.





# Open web



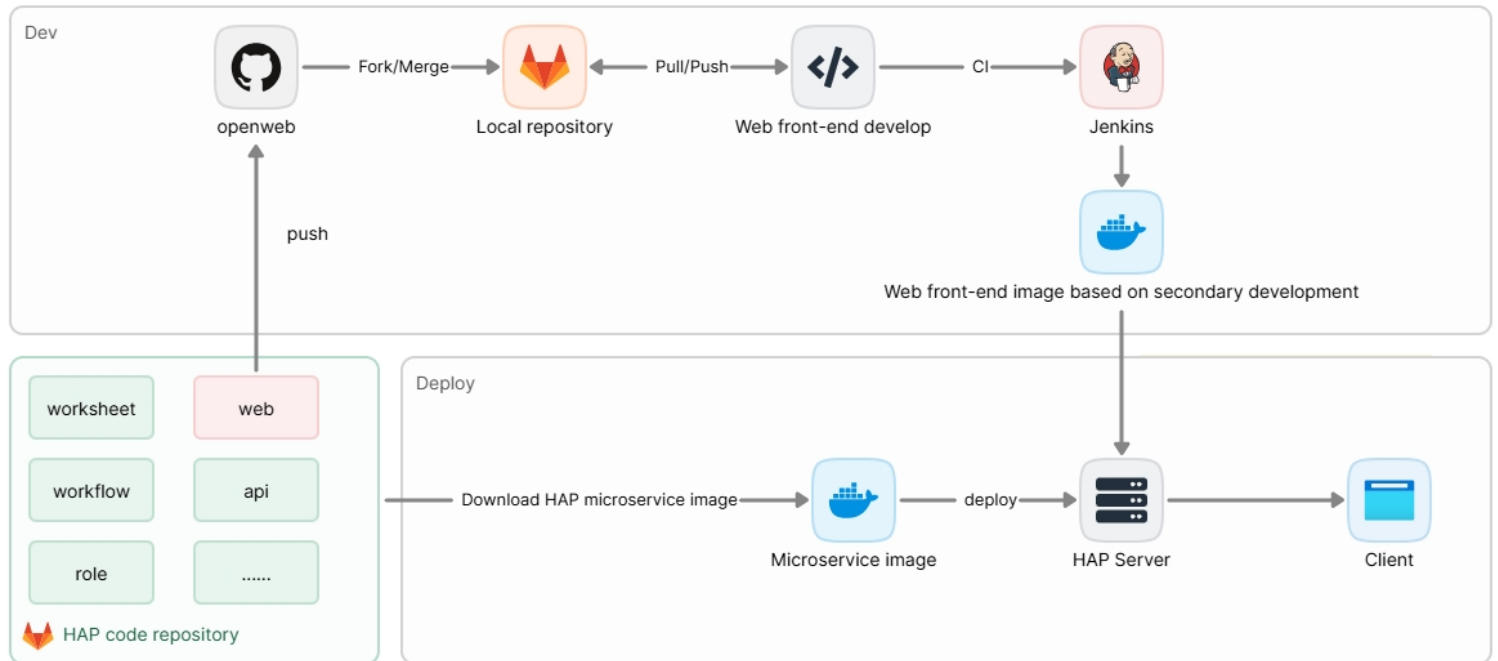
# open web

Open Source Repository for Web of HAP Server

- Github: <https://github.com/mingdaocom/pd-openweb>
- Gitee: <https://gitee.com/mingdaocloud/pd-openweb>

If need secondary development on the web side of HAP, you are welcome to fork and use. View more details in [Quick Start](#).

# Quick Start



## Install HAP Server

*Ignore this step if already installed*

The web side is one of the services in the HAP Server microservices collection, and cannot be used independently. Before secondary development, you need to deploy HAP Server (**v2.8.0+**) first. More details in [quick installation](#).

## Development

### Environment Preparation

It is recommended to install Node.js 12.18.3+ on a development device with more than 4G RAM.

### Development Tools

Recommended: [Visual Studio Code](#)

### Clone Project

```
git clone git@github.com:mingdaocom/pd-openweb.git
```

## Install Dependency Packages

Execute `yarn` (recommended) or `npm install`.

## Modify API address

Modify the value of the `API_SERVER` of the `start` command in `package.json`. The value of `API_SERVER` is the access address of the HAP system deployed in the first step, and the build tool will proxy the API requests to the configured `API_SERVER` address during development.

Example:

```
"start": "cross-env API_SERVER=http://172.17.30.60:8880 node --max-old-space-size=8192 ./node_modules/gulp/bin/gulp.js dev:main"
```

## Start Project

```
npm start
```

## Release

`package.json` is preconfigured with two commands `release` and `publish`. `API_SERVER` and `WEBPACK_PUBLIC_PATH` can be modified according to the actual situation.

```
"release": "cross-env NODE_ENV=production node --max-old-space-size=8192 ./node_modules/gulp/bin/gulp.js release",  
"publish": "cross-env NODE_ENV=production API_SERVER=/wwwapi/ WEBPACK_PUBLIC_PATH=/dist/pack/ node --max-old-space-size=8192 ./node_modules/gulp/bin/gulp.js publish"
```

`release`: compile the front-end code

`publish`: handle the templates and files needed for publishing

- `API_SERVER`: API address
- `WEBPACK_PUBLIC_PATH`: the prefix of the page script reference path. It defaults to `/dist/pack/`. You need to modify this parameter when you need to use CDN to accelerate access, such as `${CDN_HOST}/dist/pack/`

Please execute `npm run release` and `npm run publish` in order, after the release is done, all the built files will be exported to the `build` folder in the root directory.

## Deployment

Before building the mirror, you need to modify the file `docker/nginx.conf`. All records starting with `rewrite` need to have the system access address added to the target address (the default port 80 or 443 for http/https does not need to have the system access address added, if there is a subpath you need to add), such as `rewrite (?i)^/app/my http://172.17.30.60:8880/ mobile/appHome redirect;`

If you need Docker for front-end project deployment, you can use `Dockerfile` under the docker folder to build the mirror directly. The following is a demo of the mirror building (based on Linux Jenkins).

```
# Requires Nodejs environment dependency 12.18.3+
# PATH=/usr/local/node-12.18.3/bin:$PATH

# Submit logs
git log -n 1

# Clean up
git clean -fdx -e node_modules -e packages
rm -f yarn.lock

# Install dependency packages
npm install
# Build
npm run release
# Release
npm run publish

# Customize mirror addresses based on the actual mirror repository used
REGISTRY_PATH=hub.doamon.com/hap/web

# TAG
BUILD_DATE=$(date +%Y%m%d_%H%M)

IMAGE_NAME=$REGISTRY_PATH:$BUILD_DATE

# Build the mirror
docker build --no-cache -t $IMAGE_NAME -f ./docker/Dockerfile .
```

```
# Push to the mirror repository
docker push $IMAGE_NAME
```

After the successful push of the mirror, you can pull the mirror on the deployment server and launch it, such as `docker run -d --rm -p 80:80 hub.doamon.com/hap/web:20210801_1111`.

## Integration

After the front-end project is deployed, you need to configure the service address of the front-end site into the HAP microservice application container as follows, modify the corresponding `docker-compose.yml` of the microservice application (*multiple microservice instances in cluster deployment mode need to be modified*), add the environment variable `ENV_WEB_ENDPOINTS` (*if there are more than one, use English comma to separate them*), and restart the service to take effect after configuration.

```
services:
  app:
    environment:
      ENV_WEB_ENDPOINTS: "172.17.30.60:80,172.17.30.60:81"
```

# Catalog Description

CI # ci continuous integration and building configuration

docker # docker configuration files

locale # Multilingual configuration files

localeTool # Multilingual building configuration

src # Development source code

└─ api # api interface

└─ common # Global base styles, global base methods, preprocessing

└─ components # Global base components

└─ html-templates # Independent entry pages

└─ library # Collection of third-party plugins

└─ ming-ui # ui component library of HAP

└─ pages # Source code for major modules

| └─ Admin # Organization management backend

| └─ AppHomepage # Home page of the workspace

| └─ calendar # Calendar

| └─ chat # Chat

| └─ customPage # Custom page

| └─ feed # Post

| └─ FormSet # Form settings

| └─ hr # Personnel

| └─ kc # Files

| └─ Mobile # Code for mobile terminal

| └─ NewRecord # Add a new record

| └─ PageHeader # Summary of header for each module

| └─ Personal # Personal account

| └─ Print # Personnel printing

| └─ PublicWorksheet # Make forms visible to the public

| └─ PublicWorksheetConfig # Configuration for public queries

| └─ PublicWorksheetPreview # Preview of public forms

| └─ Roles # User roles

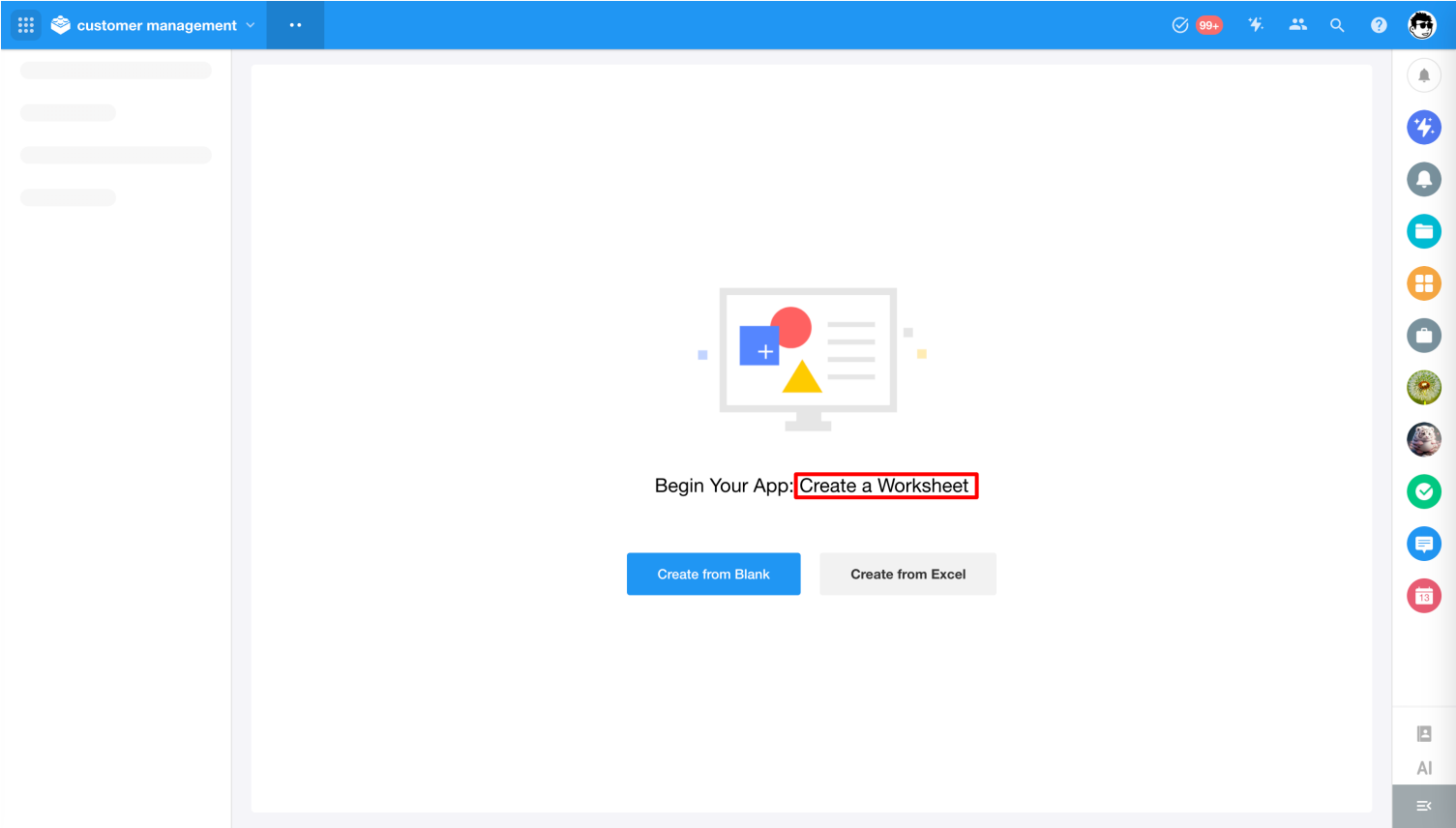
| └─ SmartSearch # Intelligent search

| └─ task # Task

- | | — UploadTemplateSheet # Make templates
- | | — UserProfile # Personal profile
- | | — widgetConfig # Edit fields
- | | — workflow # Workflow
- | | — worksheet # Worksheet
- | | — worksheetApi # API development documentation
- | — redux # store creation
- | — router # routing configuration file
- | — socket # socket configuration
- | — util # global tools staticfiles # Static pages, static resources
- .babelrc # babel configuration file
- .dockerignore # docker configuration for ignoring commit files
- .editorconfig # configuration for code editing styles and specifications
- .eslintignore # git configuration for ignoring commit files .eslintrc.json # configuration file for eslint code rule checking
- .gitattributes # git file attribute settings
- .gitignore # git configuration for ignoring commit files
- .prettierrc # configuration file for prettier code rule checking
- gulpfile.js # gulp building configuration file
- jsconfig.json # configuration for the root directory of the js project
- package.json # configuration for npm package management
- postcss.config.js # file for mobile adaptation

# Replace Text

Find the text to be replaced.



Find the corresponding file in the editor.

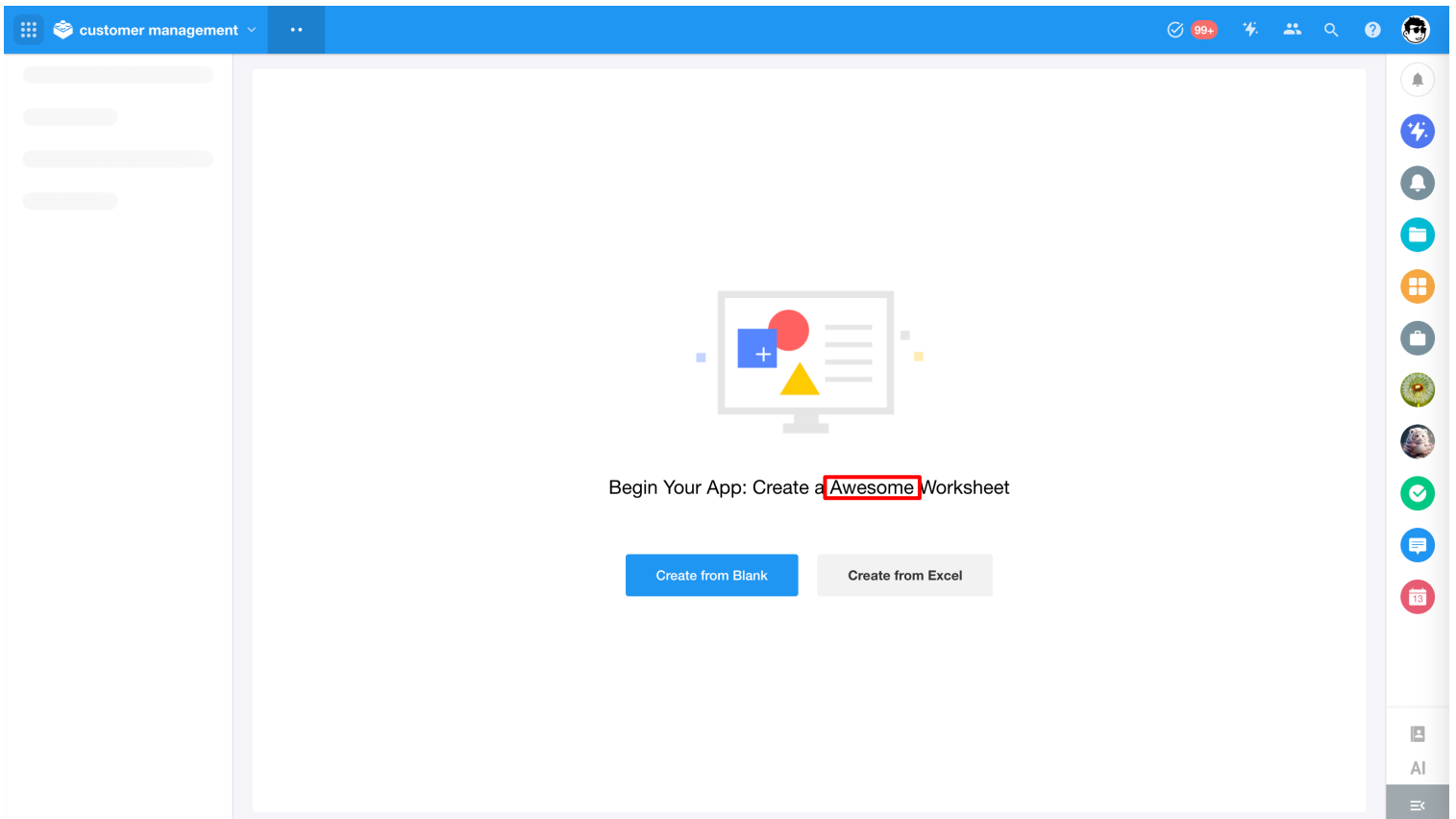


```
45 const { sheetList, appId, groupId, isCharge } = this.props;
46 const { createType, visible, dialogImportExcel } = this.state;
47 const { appGroups = [], projectId } = store.getState().appPkg;
48 const sheetCount = sheetList.length;
49 const isAdd = !(appGroups.length > 1 && !sheetCount);
50
51 if (!isCharge) {
52   return (
53     <div className="contentBox">
54       <i className="iconBox" />
55       <span className="Block TxtCenter Font20 Black">{_l('Empty')}</span>
56     </div>
57   );
58 }
59
60 return (
61   <div className="contentBox">
62     <i className={cx('iconBox', { add: isAdd })} />
63     <span className="Block TxtCenter Font20 Black">
64       {isAdd ? _l('Begin Your App: Create a Worksheet') : _l('Empty')}
65     </span>
66     {isAdd ? (
67       <div className="flexRow createOperate">
68         <Button
69           type="primary"
70           className="mRight20 fw500"
71           onClick={() => {
72             this.setState({ createType: 'worksheet' });
73           }}
74         >
75           {_l('Create From Empty')}
76         </Button>
77         <Button
78           className="excelCreateBtn bold"
79           onClick={() => {
80             this.setState({ dialogImportExcel: true });
81           }}
82         >
83           {_l('Create From Excel')}
```

Replace the text with what you want and save it.

```
MDPublic-bug
SEARCH
Begin Your App: Create a Worksheet
Replace
files to include
files to exclude
2 results in 2 files - Open in editor
> JS mdTranslation.js locale/en
> mdTranslation.po locale/en
src > pages > worksheet > common > WorksheetEmpty > WorksheetEmpty.jsx > renderCreate
45 const { sheetList, appId, groupId, isCharge } = this.props;
46 const { createType, visible, dialogImportExcel } = this.state;
47 const { appGroups = [], projectId } = store.getState().appPkg;
48 const sheetCount = sheetList.length;
49 const isAdd = !(appGroups.length > 1 && !sheetCount);
50
51 if (!isCharge) {
52   return (
53     <div className="contentBox">
54       <i className="iconBox" />
55       <span className="Block TxtCenter Font20 Black">{_(('Empty'))}</span>
56     </div>
57   );
58 }
59
60 return (
61   <div className="contentBox">
62     <i className={cx('iconBox', { add: isAdd })} />
63     <span className="Block TxtCenter Font20 Black">
64       {isAdd ? _l('Begin Your App: Create a Awesome Worksheet') : _l('Empty')}
65     </span>
66     {isAdd ? (
67       <div className="flexRow createOperate">
68         <Button
69           type="primary"
70           className="mRight20 fw500"
71           onClick={() => {
72             this.setState({ createType: 'worksheet' });
73           }}
74         >
75           {_(('Create From Empty'))}
76         </Button>
77         <Button
78           className="excelCreateBtn bold"
79           onClick={() => {
80             this.setState({ dialogImportExcel: true });
81           }}
82         >
83           {_(('Create From Excel'))}
```

Refresh the page when finished.



# New Page

Pages are divided into **internal pages** and **external pages**.

Internal pages are used within the site and require login. Page switching is based on front-end routing rules, and all pages point to index.html.

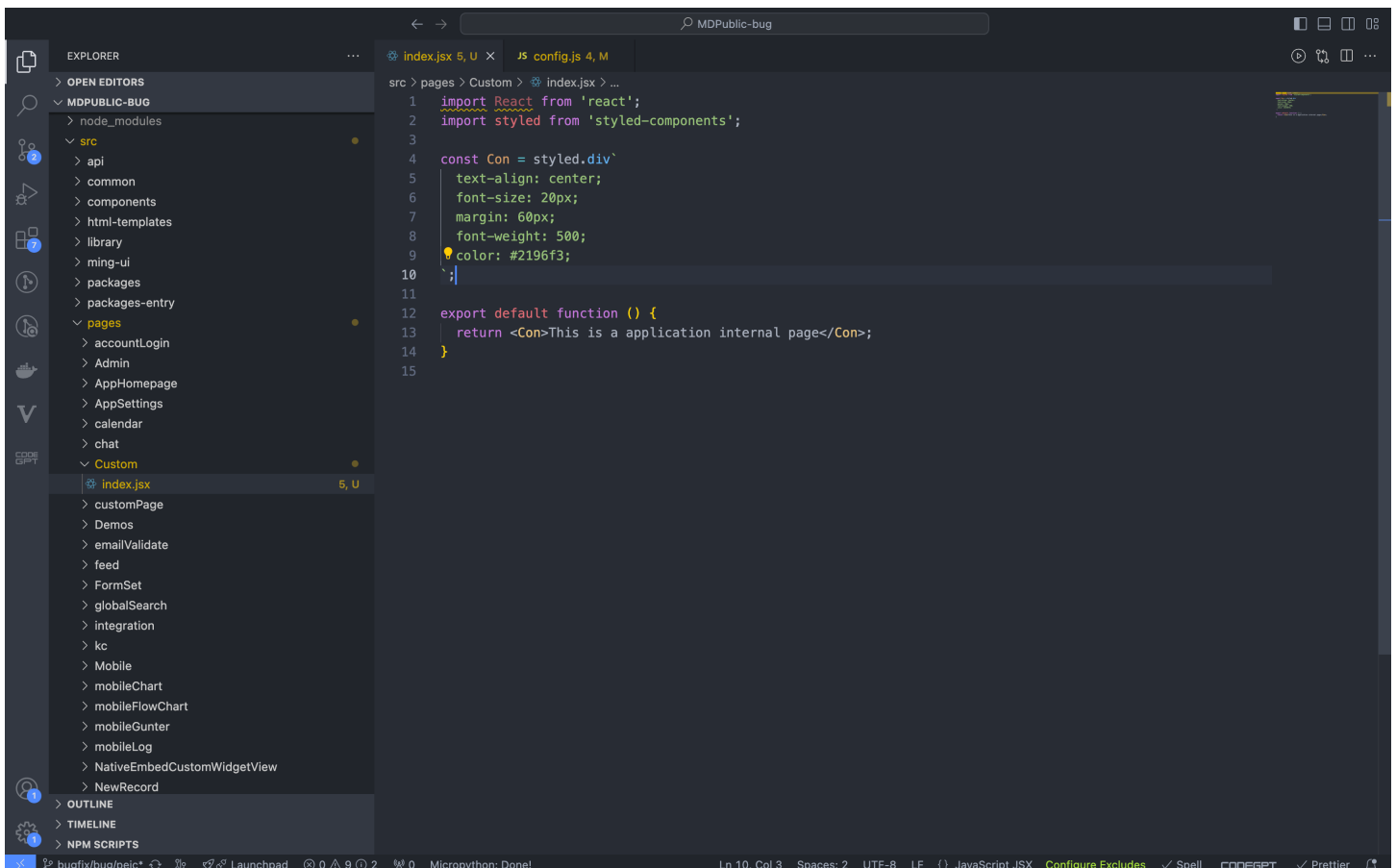
External pages are separate html entry files. It is often used for external pages for public forms, record sharing, etc. These pages generally do not require login.

## Add Internal Page

Add the components of the new page under `src/pages`. (Only the components need to be exposed here. Rendering of the components to the dom is processed externally and uniformly.)

### In-Application Page

1. Add the page



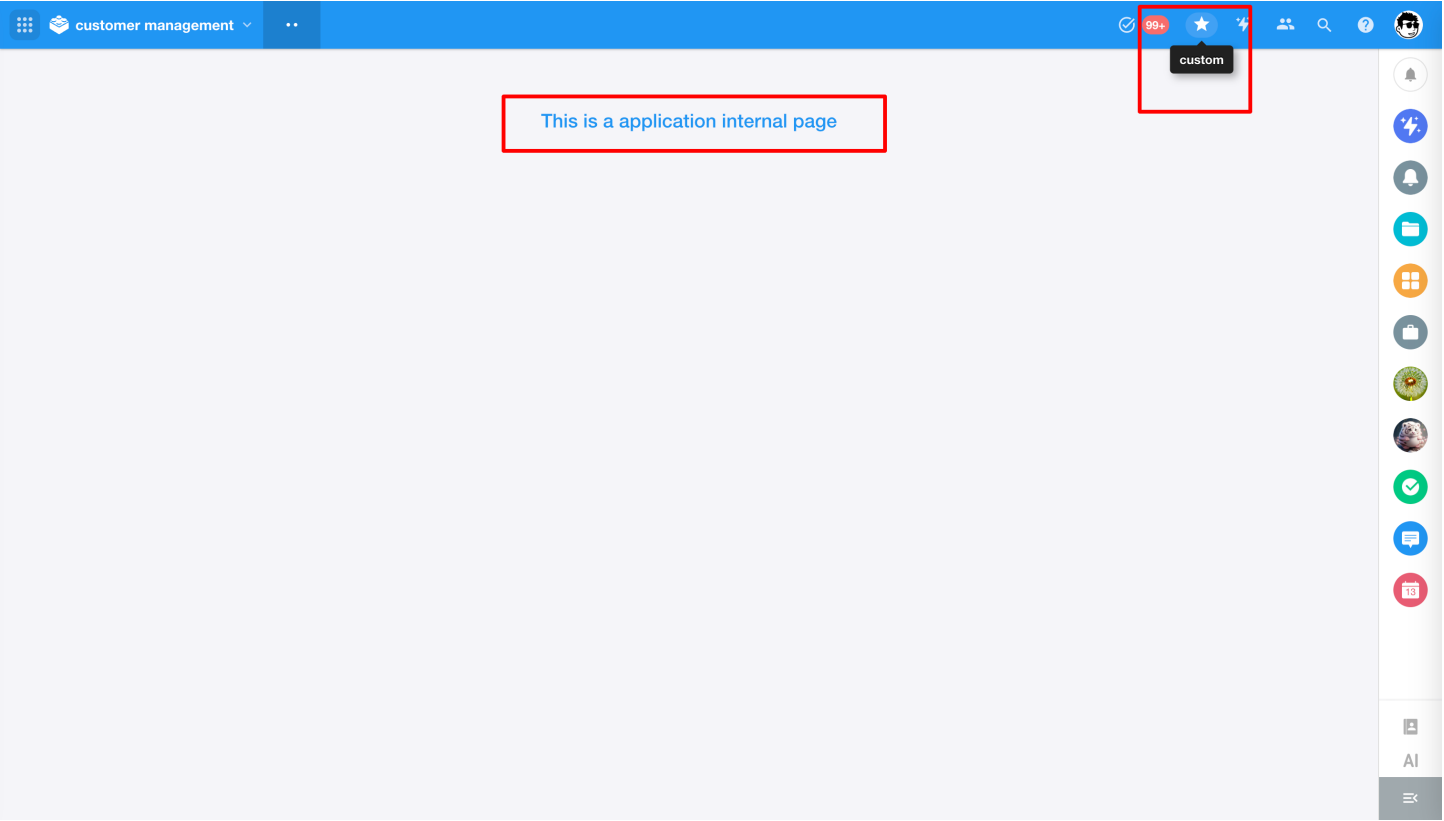
2. Add the route

```
src > router > Application > JS config.js > ROUTE_CONFIG > role
...
1 import { addSubPathOfRoutes } from 'src/util';
2
3 export const ROUTE_CONFIG = addSubPathOfRoutes(
4   // This is a application internal page
5   {
6     custom: {
7       path: '/app/:appId/custom',
8       component: () => import('pages/Custom'),
9       sensitive: true,
10    },
11
12    workflow: {
13      path: '/app/:appId/workflow',
14      component: () => import('pages/workflow/WorkflowList/AppWorkflowList'),
15      sensitive: true,
16    },
17
18    role: {
19      path: '/app/:appId/role/:editType?/:listType?',
20      component: () => import('pages/Role'),
21      sensitive: true,
22    },
23
24    portaluser: {
25      path: '/app/:appId/portaluser/:id?',
26      component: () => import('pages/Role/PortalCon/portalUser'),
27      sensitive: true,
28    },
29
30    appLogs: {
31      path: '/app/:appId/logs/:projectId',
32      component: () => import('src/pages/Admin/logs/AppLog'),
33      sensitive: true,
34    },
35
36    analytics: {
37      path: '/app/:appId/analytics/:projectId',
38      component: () => import('src/pages/Admin/app/useAnalytics/components/AppAnalytics'),
39      sensitive: true,
40    },
41  },
42);
```

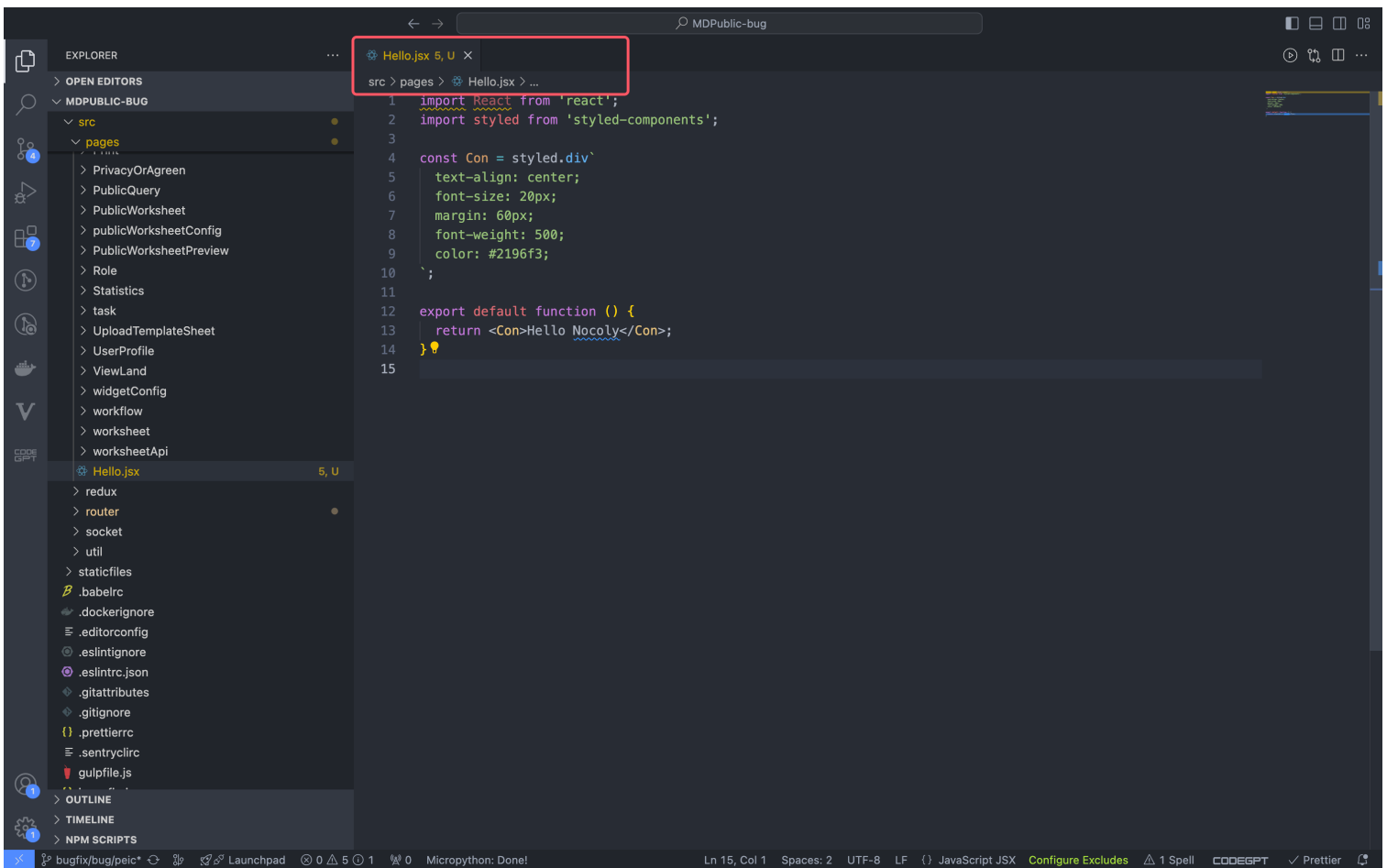
### 3. Add the button

```
src > pages > PageHeader > AppPkgHeader > AppExtension > index.jsx ...
131 </Tooltip>
132 </div>
133 </RoleDebugSelectWrap>
134 };
135
136 if (loading) return null;
137
138 return (
139   <div className="appExtensionWrap">
140     <MdLink to={`/app/${appId}/custom`}>
141       <div className="flexRow appExtensionItem" data-tip={_l('custom')}>
142         <Icon className="Font20" icon="star" />
143       </div>
144     </MdLink>
145
146     {otherAllShow && !window.isPublicApp && canEditApp(permissionType, islock) && (
147       <MdLink to={`/app/${appId}/workflow`}>
148         <div className="flexRow appExtensionItem" data-tip={_l('Workflow')}>
149           <Icon className="Font20" icon="workflow" />
150         </div>
151       </MdLink>
152     )}
153     {(roleEntryVisible || canDebug) && renderRoleDebugSelect()}
154   </div>
155 );
156
157
```

4. Finish



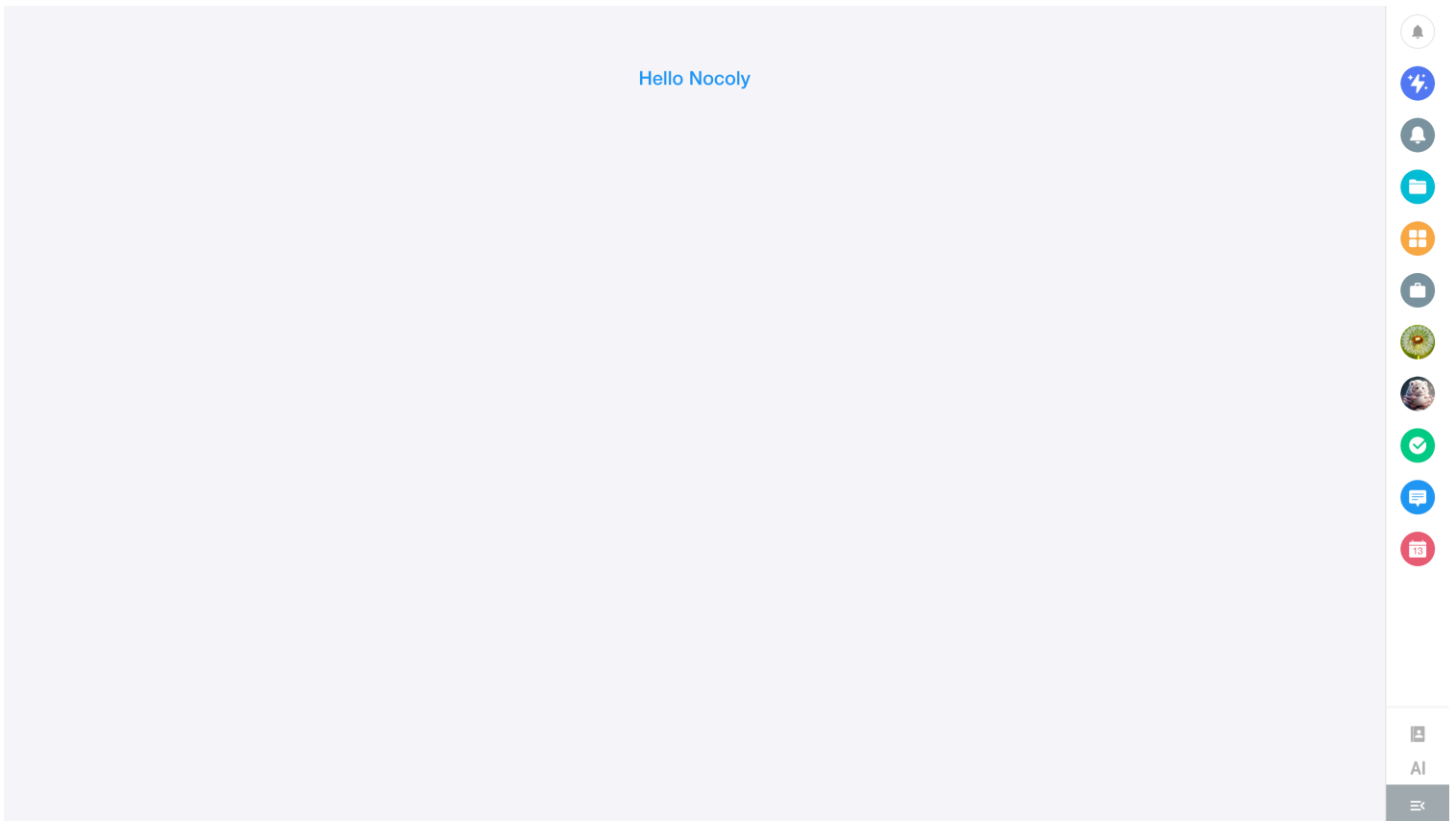
Non-In-Application Page



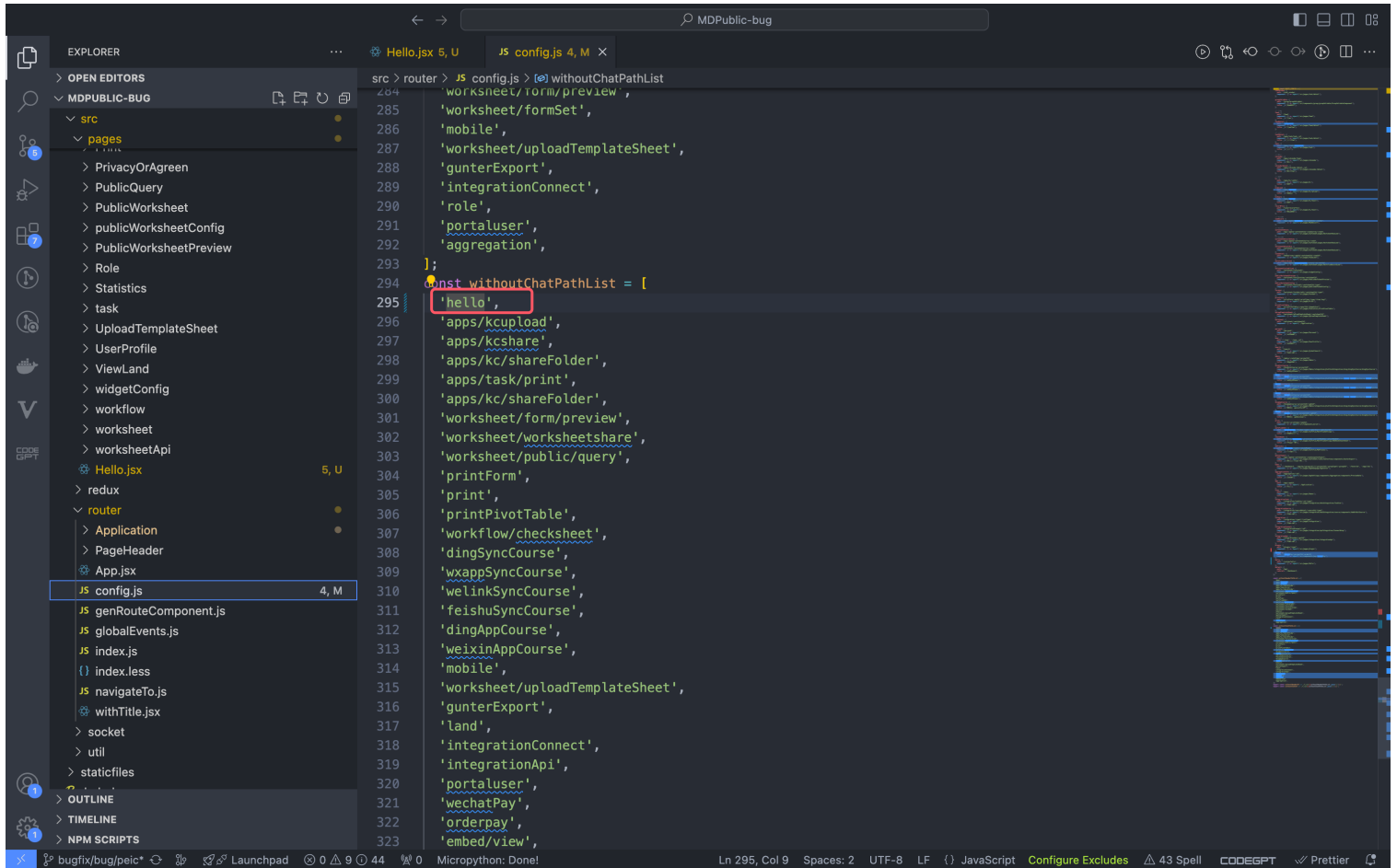
In the route configure file `src/router/config.js` to add a new page `custom/hello`. The new path should start with `custom/`. If you need to add other beginnings, add the corresponding rewrite rules in the `docker/rewrite.setting`.

```
249 },
250 plugin: {
251   path: '/plugin/:type?',
252   component: () => import('src/pages/plugin'),
253 },
254 wechatPay: {
255   path: '/wechatPay/:projectId:orderId',
256   component: () => import('src/components/pay/wechatPay'),
257 },
258 hello: {
259   path: '/custom/hello',
260   component: () => import('src/pages/Hello'),
261 },
262 default: {
263   path: '/app',
264   redirect: '/dashboard',
265 },
266 };
267
268 const withoutHeaderPathList = [
269   'demo',
270   'apps/kcupload',
271   'apps/kcshare',
272   'apps/kc/shareFolder',
273   'apps/task/print',
274   'apps/kc/shareFolder',
275   'worksheet/worksheetshare',
276   'worksheet/public/query',
277   'printForm',
278   'print',
279   'workflow',
280   'workflowEdit',
281   'workflow/checksheet',
282   'worksheet/field/edit',
283   'worksheet/form/edit',
284   'worksheet/form/preview',
285   'worksheet/formSet',
286   'mobile',
287   'worksheet/uploadTemplateSheet',
```

Visit <http://localhost:30001/custom/hello>, and you can see the new page is added successfully.



By default, there is a list of messages on the right side of the internal page. If you don't need it, you can add rules to `withoutChatPathList` in `src/router/config.js` to disable the message list.



Refresh the page and you can see the message list is no longer displayed on the right side.



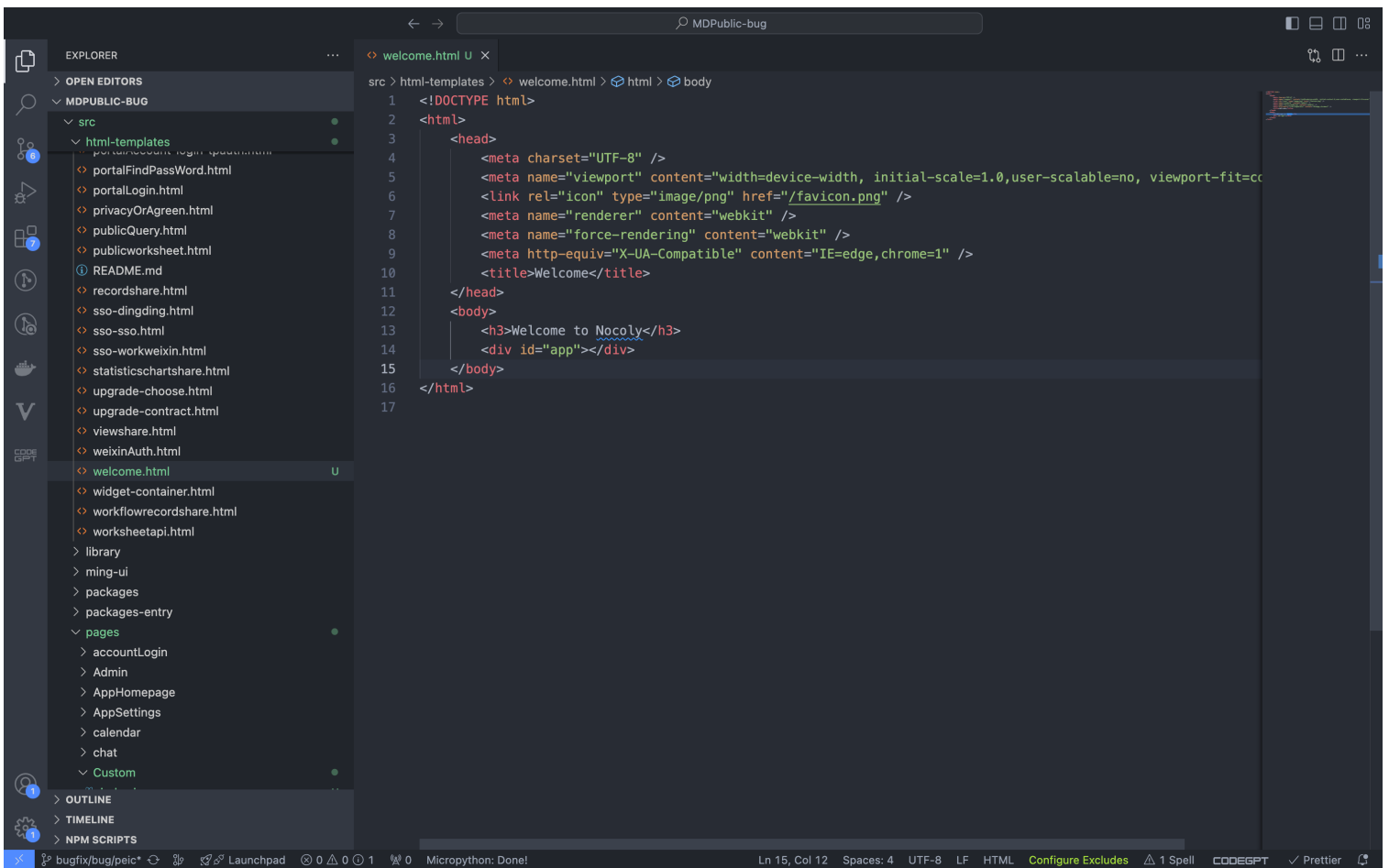
Hello Nocoly

## Add Standalone Page

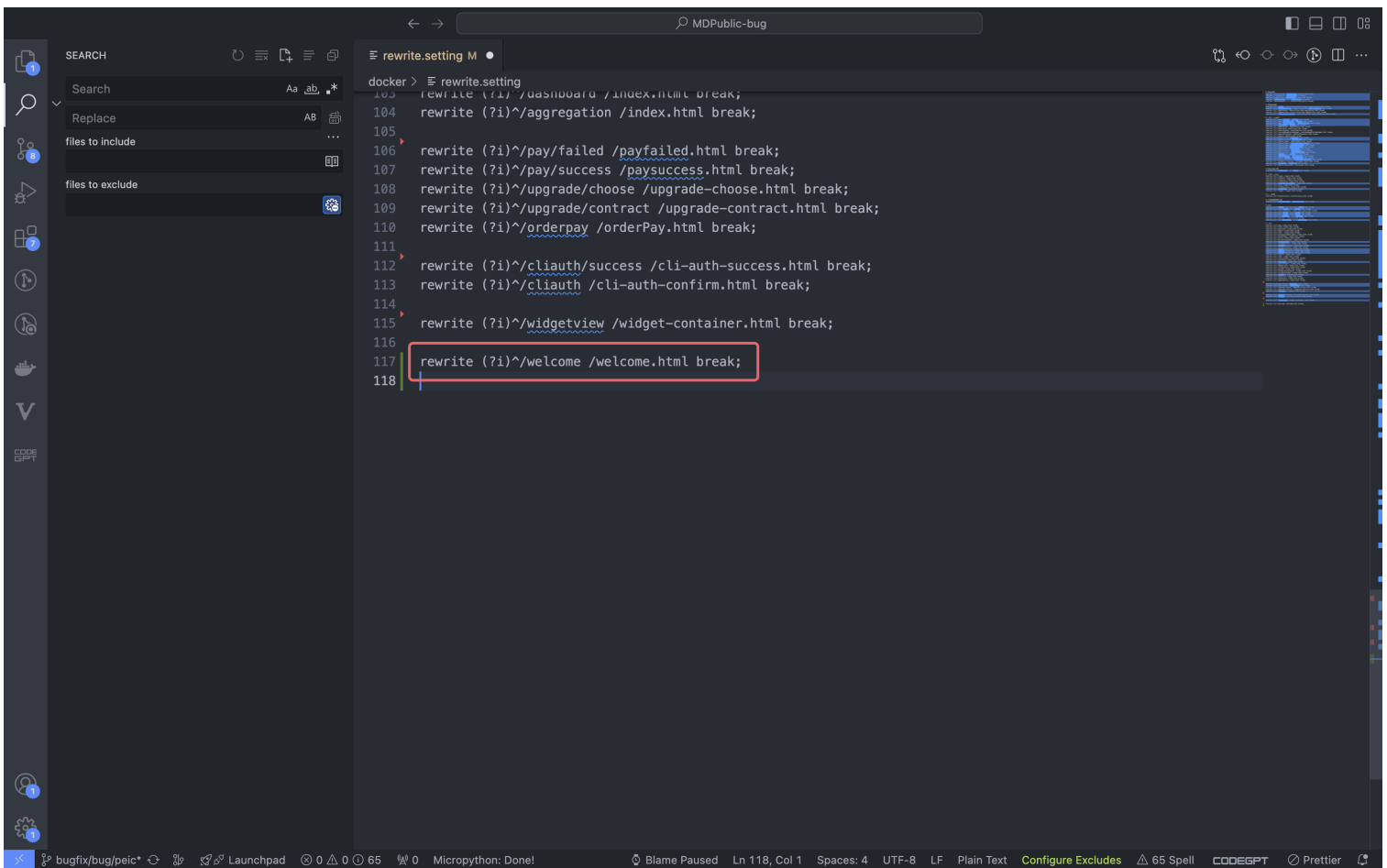
Add the html file welcome.html to the `src/html-templates` path. The html construction will call ejs compilation, so you can use ejs syntax in html. The ejs compilation will pass in two variables, `apiServer` and `publicPath`. `apiServer` is used to specify the address of the backend service. It is used when you need to access the backend interface. `publicPath` is used to specify the static file hosting address. This variable is required when there are scripts loaded asynchronously within the referenced module. So in most cases you need to keep these two lines of code in the html file and make sure that the variables

`__api_server__` and `webpack_public_path` are declared before all scripts are loaded.

```
<script>
  var __api_server__ = <%= apiServer %>;
  var __webpack_public_path__ = '<%= publicPath %>';
</script>
```



Resource mapping within the project relies on proxy rules, so you also need to configure the corresponding rewrite rules in `docker/rewrite.setting`.

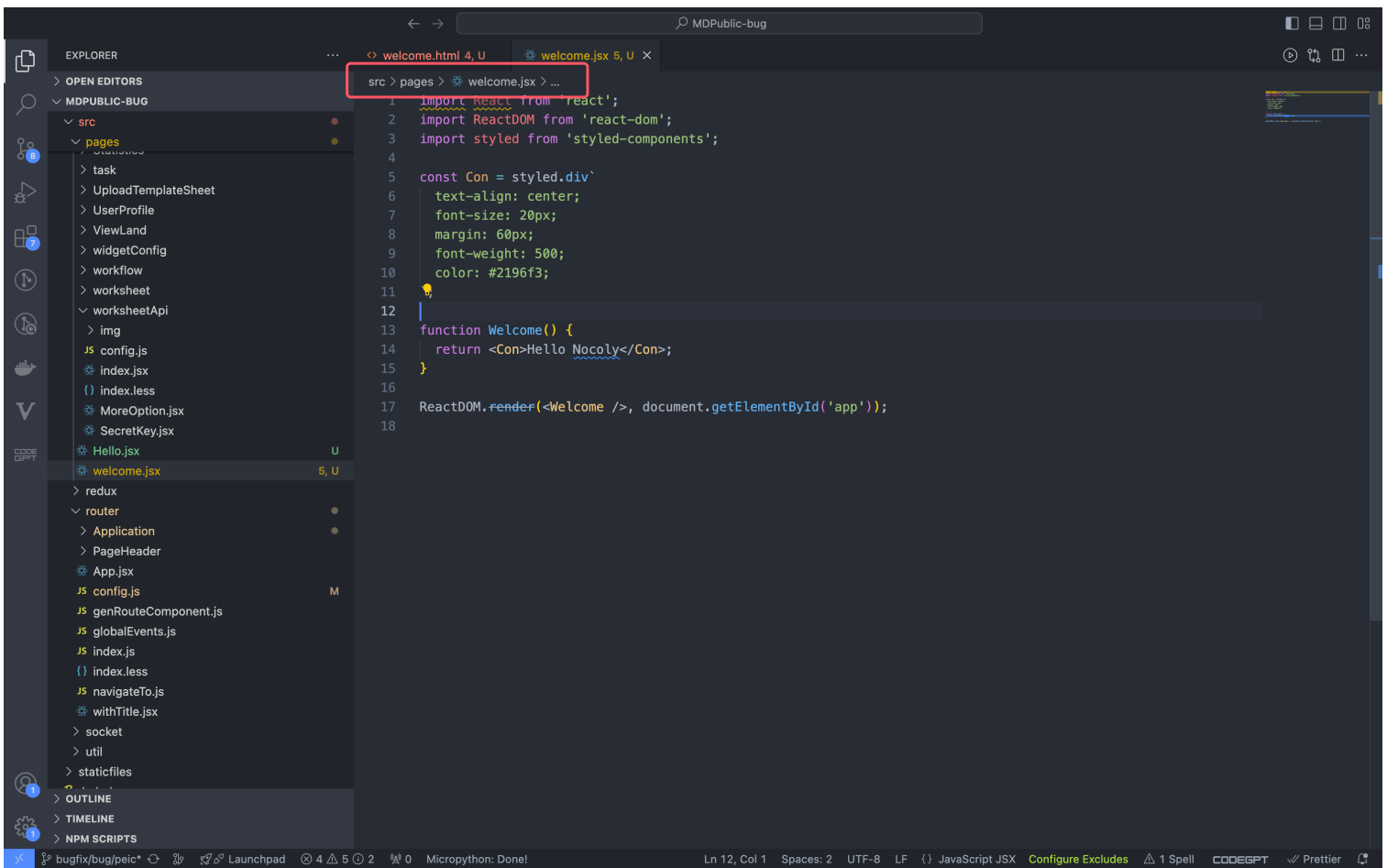


Restart the build tool or execute `gulp generate-mainweb` in the console and visit <http://localhost:30001/custom/hello>.

Welcome to Nocoly

At this point the page is still static, and the following is to render the React component into the page.

Add the corresponding React component. Unlike the internal pages, you need to render the components into the corresponding dom yourself.



Go back to the welcome.html and add `<script src="webpack[single]?src/pages/welcome.jsx"></script>` to the end of the body. The build tool will automatically extract the entry and replace the src with the published resource address when it is published. After restarting the build tool, visit <http://localhost:30001/custom/hello>, and you can see the React component has been successfully rendered into the page.

Welcome to Nocoly

Hello Nocoly

# Reference Components

## Component - Chart

```
import WidgetDisplay from
'src/pages/customPage/components/WidgetContent/WidgetDisplay'.

<WidgetDisplay
  ids={
    {
      worksheetId: '', // worksheet id
    }
  }
  value="" // chart id
  type={1} // fixed value: 1
/>
```

## Component - Member selection

```
import 'dialogSelectUser'.

$(this).dialogSelectUser({
  title: '', // title on the popup
  showMoreInvite: false, // show more invitations or not
  fromType: 4, // fixed 4
  SelectUserSettings: {
    filterAll: true, // filter all or not
    filterFriend: true, // filter friends or not
    filterOthers: true, // filter other collaborations or not
    filterOtherProject: true, // filter other networks or not
    filterAccountIds: [], // which users to be filtered
    projectId: '', // organization number
    unique: true, // single choice or not
    callback: (users) => {}, // return the selected users
  }.
}).
```

## Component - Department selection

---

```
import DialogSelectGroups from 'src/components/dialogSelectDept'.

new DialogSelectGroups({
  projectId: '', // organization number
  isIncludeRoot: false, // include all or not
  unique: true, // single selection or not
  showCreateBtn: false, // show the Create Department button or not
  selectFn: (departments) => {}, // return the selected departments
}).
```

## Component - Position selection

```
import DialogSelectJob from 'src/components/DialogSelectJob'.

new DialogSelectJob({
  projectId: '', // organization number
  onSave: jobs => {}, // return the selected positions
}).
```

## Create schedule

```
import 'createCalendar'.

$.CreateCalendar({
  Start: null, // end time String
  End: null, // start time String
  AllDay: false, // it is all day or not
  MemberArray: [{
    accountId: '', // user id
    avatar: '', // user avatar
    fullname: '', // user name
  }], // members
}).
```



# API Calls

## API files

### src/api

- └─ account.js # User information interface
- └─ appManagement.js # Application information interface
- └─ global.js # Global information interface
- └─ homeApp.js # Interface related to applications on the home page
- └─ worksheet.js # Worksheet-related interfaces

### src/pages/workflow/api

- └─ instance.js # Interface related to approvals in workflow
- └─ instanceVersion.js # Interface related to lists (pending, to be filled, etc.) in workflow

Take the file `worksheet.js` in the folder `src/api` as an example

Here's a demonstration of requesting and presenting data from a worksheet.

Go back to [add front-end page](#) and open the added page component.

Introduce the api file of the worksheet in the header.

```
import api from 'src/api/worksheet';
```

Add the code to request data and present data inside the component.

```
const Con = styled.div`
  width: 100%;
  height: 100%;
  padding: 20px;
`;
const Title = styled.div`
  font-size: 30px;
  color: #666;
  margin: 10px;
`;
const ViewTabs = styled.div`
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
```

```

`
;
const ViewTab = styled.div`
  text-align: center;
  background: #fff;
  border-radius: 6px;
  margin: 10px;
  line-height: 100px;
  color: #888;
  box-shadow: 3px 3px 6px rgb(0, 0, 0, 0.1);
`;

export default function Hello(props) {
  const [info, setInfo] = useState({views: []});
  useEffect(() => {
    const worksheetId = location.pathname.match(/^\/custom\/hello\/(\w+)/)[1];
    api.getWorksheetInfo({worksheetId, getViews: true}).then(setInfo);
  }, []);
  return (
    <Con>
      <Title>{info.name}</Title>
      <ViewTabs>
        {info.views.map((v) => (
          <ViewTab>{v.name}</ViewTab>
        ))}
      </ViewTabs>
    </Con>
  );
}

```

Accessed at <http://localhost:30001/custom/hello/{worksheet id}>

# Worksheet Demo

- All
- Grid
- Board
- Calendar
- Details

# Parameter Description

## Application

### Worksheet

`http://localhost:30001/app/{application id}/{group id}/{worksheet id}/{view id}`

### Custom Page

`http://localhost:30001/app/{application id}/{group id}/{custom page id}`

# MD\_Structure

## Reporting Relationship

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int		false		
StructureID	char(36)	PK	false		Unique ID
UserID	char(36)		true		User ID (deprecated)
ProjectID	char(36)		true		Network ID
IsTop	bit		true		Top executives or not
Manager_UserID	char(36)		true		Superior Account ID
CreateUserID	char(36)		true		Creator Account ID
CreateTime	datetime		true		Date created
UpdateUserID	char(36)		true		Account ID updated by
UpdateTime	datetime		true		Date updated
AccountID	varchar(64)		true		Account ID

# StructureLog

## Operation Log of Reporting Relationship

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int	PK	false		
AccountId	varchar(64)		true		Account ID
ProjectId	varchar(64)		true		Network ID
ParentId	varchar(64)		true		Parent ID
OldParentId	varchar(64)		true		Original parent ID
OperateId	varchar(64)		true		Operator Account ID
CreateTime	datetime		true	now()	Date created

# Calendar\_Category

## Calendar Category

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int		false		
CatID	varchar(64)	PK	false		Category ID
UserID	varchar(64)		true		User ID (deprecated)
Color	int		true		Color number
CatName	varchar(400)		true		Category name
ProjectID	varchar(64)		true		Network ID (deprecated)
CreateTime	datetime		true		Date created
AccountID	varchar(64)		true		Account ID

# Calendar\_Member

## Members of Calendar

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int	PK	false		
CalendarID	varchar(64)		false		Calendar number
UserID	varchar(64)		true		Member number
Email	varchar(128)		true		External member's email
MobilePhone	varchar(32)		true	NULL	Mobile phone. Save the phone number of the external member, and the default value is null
CatID	varchar(128)		true		Category ID
GroupID	varchar(64)		true		Participating group number
Types	tinyint		true		Type. LOV: 0 for user; 1 for external user-email; 2 for group; 3 for external user-mobile
Status	tinyint		true		Status. LOV: 0 for not confirmed; 1 for participation confirmed; 2 for participation denied;
ReMark	varchar(1024)		true		Notes. Description of the refusal to participate
CreateTime	datetime		true		Date created



Field	Type	Primary Key	Nullable	Default	Description
RemindTime	int		true		Reminder time, minute
RemindType	tinyint		true		Reminder type. 0: None , 1: By minute 2: By hour 3: By day
UpdateTime	datetime		true		Date updated
AccountID	varchar(64)		true		Account ID
ProjectID	varchar(64)		true		Network ID (deprecated)
VoiceRemind	bit		true	0	Voice reminder or not

# Calendar

## Calendar

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int		false		
CalendarID	varchar(64)	PK	false		Calendar number
CalendarName	varchar(512)		true		Calendar name
StartTime	datetime		true		Start time, to the minute
EndTime	datetime		true		End time, to the minute
IsAllDay	bit		true		All day
Address	varchar(512)		true		Address
Description	varchar(1024)		true		Description
Types	tinyint		true		Type. LOV: 0 for private; 1 for public ( you can invite other members to join); 2 for invisible calendar (push reminder) ( deprecated)
GroupID	varchar(512)		true		Private group, separated by an English comma (deprecated)
ProjectID	varchar(64)		true		Network ID (deprecated)
Sequence	int		true	0	Sequence. Increment by 1 when editing

Field	Type	Primary Key	Nullable	Default	Description
					start,end,recurRule, or member status. Default:1
IsRecur	bit		true	0	Repeated calendar or not. 1:Yes, 0:No. Default:0
Frequency	tinyint		true	0	Frequency. LOV: 0: No repeat, 1: Daily, 2: Weekly. 3: Monthly, 4: Yearly
Interval	int		true	1	Repeat interval. Default: 1
RecurCount	int		true	0	Repeat times. Default: 0
UntilDate	datetime		true		End date. If RecurCount is 0 and UntilDate is null, it is a permanent repeat
WeekDay	int		true		Repeat on which day. LOV: 0: non-weekly repeat, 1:Monday, 2:Tuesday, 4:Wednesday, 8:Thursday, 16:Friday, 32 Saturday, 64:Sunday
EventStart	datetime		true		Repeat start date
EventEnd	datetime		true		Repeat end date
CreateUser	varchar(64)		true		Creator (deprecated)
AppID	varchar(64)		true		Source of creating calendar
IsDelete	bit		true	0	Deleted or not
CreateTime	datetime		true		Date created

Field	Type	Primary Key	Nullable	Default	Description
UpdateTime	datetime		true		Date updated
Locked	bit		true		Locked or not (deprecated)
CreateAccountID	varchar(64)		true		Creator Account ID
IsPrivate	bit		false	0	Private calendar or not

# ChildCalendarRecurTime

## Sub-calendar

Field	Type	Primary Key	Nullable	Default	Description
CalendarID	varchar(64)	PK	false		Calendar ID
RecurTime	datetime		false		Repeat time
StartTime	datetime		false		Start time
Status	tinyint		true		Status. 1:Normal 0:Delete
CalendarName	varchar(512)		true		Calendar name
EndTime	datetime		true		End time, to the minute
IsAllDay	bit		true		All day
Address	varchar(512)		true		Address
Description	varchar(1024)		true		Description
Types	tinyint		true		Type. LOV: 0 for private; 1 for public (other members can be invited to join) (deprecated)
GroupID	varchar(512)		true		Private group, if there are multiple, separate them with English commas (deprecated)
Sequence	int		true		Sequence. Increment by 1 when editing

Field	Type	Primary Key	Nullable	Default	Description
					start,end,recurRule, or member status. Default:1
CreateTime	datetime		true		Date created
UpdateTime	datetime		true		Date updated
CreateUser	varchar(64)		true		Creator (deprecated)
Locked	bit		true		Locked or not
ProjectID	varchar(64)		true		Network ID
CreateAccountID	varchar(64)		true		Creator Account ID
AutoID	int		false		
IsPrivate	bit		false	0	Private calendar or not

# ChildCalendarRecurTimeMember

## Members of Sub-calendar

Field	Type	Primary Key	Nullable	Default	Description
CalendarID	varchar(64)		false		Calendar number
UserID	varchar(64)		true		Member number (deprecated)
RecurTime	datetime		false		Repeat time
Email	varchar(128)		true		External member's email
MobilePhone	varchar(32)		true		Mobile phone. Save the phone number of the external member, and the default value is null
GroupID	varchar(64)		true		Participating group number (deprecated)
CatID	varchar(64)		true		Category number
Types	tinyint		true		Type. LOV: 0 for user; 1 for external user-email; 2 for group; 3 for external user-mobile
Status	tinyint		true		Status. LOV: 0 for not confirmed; 1 for confirmed participation; 2 for declined;
ReMark	varchar(1024)		true		Notes. Notes for declining to participate
CreateTime	datetime		true		Date created

Field	Type	Primary Key	Nullable	Default	Description
RemindTime	int		true		Reminder time, minute
RemindType	tinyint		true		Reminder type. 0:None , 1:By minute, 2:By hour, 3:By day
UpdateTime	datetime		true		Date updated
AutoID	int	PK	false		
AccountID	varchar(64)		true		
ProjectID	varchar(64)		true		(deprecated)
VoiceRemind	bit		true	0	Voice message



# User\_CalCategory\_Order

## User Calendar Classifying and Sorting

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int	PK	false		
UserID	varchar(64)		true		User ID (deprecated)
CatID	varchar(64)		true		Category ID
DisplayOrder	int		true		Custom serial number, starting from 0
ProjectID	varchar(64)		true		Network ID (deprecated)
CreateTime	datetime		true		Date created
AccountID	varchar(64)		true		Account ID

# Account\_Device

## Device Information

Field	Type	Primary Key	Nullable	Default	Description
Autold	int	PK	false		
AccountID	varchar(64)		true		Account ID
AppID	varchar(64)		true		App ID
Device	varchar(256)		true		Device number
DeviceType	int		true	0	Device type
RegID	varchar(256)		true		Registration ID
NotifyType	int		true		Notification type (0: notification message, 1: custom message)
OSType	varchar(128)		true		Model, e.g. Xiaomi, Huawei
UpdateTime	datetime		true		Date updated
Language	varchar(50)		true		Language
AppVersion	varchar(30)		true		Current app version
Source	int		true		Attribution
Channel	int		true		Push channel, 0: default, 1: carrier pigeon

# Application

## Third Party Application

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int		false		
AppID	varchar(64)	PK	false		App ID
AppName	varchar(256)		true		App name
Avatar	varchar(128)		true		Avatar
Summary	varchar(256)		true		Profile
About	varchar(1024)		true		Description
IS_Personal	bit		true		Personal application or not
IS_Free	bit		true		Free or not
IS_Private	bit		true	0	Private or not
ProjectIDRange	varchar(1024)		true		ID of the installed network
ShowModel	tinyint		false	0	Display mode. (0: standalone page, 1:I-form, 2:T-form)
Pricing_Fee	decimal		true	0	Billing
Pricing_Unit	tinyint		true	0	Billing unit
Pricing_Time	tinyint		true	0	Time unit of billing
Pricing_Type	tinyint		true	3	Billing type. LOV: 1 for prepayment, 2 for postpayment

Field	Type	Primary Key	Nullable	Default	Description
					a combination of these two
Pricing_Mark	varchar(1024)		true		Billing description
NoticeUrl	varchar(256)		true		Notification of successful payment
AppUrl	varchar(128)		true		External access address of the application
HomeUrl	varchar(128)		true		Internal home page of the application
SettingUrl	varchar(128)		true		Application's setting address
Consumer_Key	varchar(128)		true		Application token
Consumer_Secret	varchar(128)		true		Application token key
Authorize_sign	tinyint		true	0	Authorized login method. (0: basic authorization)
Authorize_lic	tinyint		true	0	Authorization type. (0: basic authorization 1: advanced authorization)
Calback_Url	varchar(128)		true		Callback address
Rating	decimal		true		Application rating
Num_User	int		true		Number of users who installed the app
Num_Post	int		true		Number of comments posted
ProjectID	varchar(64)		true		Network ID

Field	Type	Primary Key	Nullable	Default	Description
CreateUser	varchar(64)		true		Creator Account ID
LastModifyUser	varchar(64)		true		Modifier Account ID
Status	smallint		true		Status. (-2:All, -1:Deleted, 1: published, 2:Published, 4:Reviewing, 5:Reviewing failed, 6:Debugging, 7: Withdrawn and reviewing, 8: Withdrawing and reviewing failed, 9:Present on line, 10: Withdraw)
Type	tinyint		true		Application type. (-1: All, 0: Common, 1: Basic, 2: Desktop, 3: Mobile, 4: Web, 5: Mobile and Web)
ReMark	varchar(512)		true		Notes
CategoryID	varchar(64)		true		Category ID
GeoID	int		false	1	Region ID
DeviceField	varchar(512)		true		For IOS: ios_appid; for ANDROID: android_downurl android_package(package name of Android application) android_activity (startup activity of Android application)
CreateTime	datetime		true		Date created
UpdateTime	datetime		true		Date updated
PublicTime	datetime		true		Date published

Field	Type	Primary Key	Nullable	Default	Description
ApplyStatus	smallint		true		Status. (-2:All, -1:Deleted, 1: published, 2:Published, 4:Reviewing, 5:Reviewing failed, 6:Debugging, 7: Withdrawn after reviewing, 8: Withdrawing after reviewing failed, 9:Present on the market, 10: Withdraw)
Is_Wap	bit		true	1	Wap display is supported or not
SubscribeUrl	varchar(256)		true		Notification address of the app subscription
AppCode	varchar(1024)		true		QR code
AppmdConnect	varchar(1024)		true		Integration items for HAP
AppCompanyName	varchar(1024)		true		Company name
AppContact	varchar(1024)		true		Customer service method
URLScheme	varchar(1024)		true		URL Scheme
AndroidSSO	bit		true		Breakthrough to HAP account on Android
IosSSO	bit		true		Breakthrough to HAP account on iOS
AppEmail	varchar(1024)		true		Email for feedback
H5Url	varchar(1024)		true		URL for mobile adaptation
AccountID	varchar(64)		true		Account ID

Field	Type	Primary Key	Nullable	Default	Description
WebHookUrl	varchar(500)		true		Webhook address
Modules	varchar(100)		true		Data access module. 1: Cont 2: Messages, 3: Post, 4: Task Calendar, 6: Knowledge, 7: Approval, 8: Attendance
WebHookCode	varchar(20)		true		
WebHookStatus	int		true	1	Webhook status. (1: support suspended)

# OAuth2\_Access\_Token

## Access Token

Field	Type	Primary Key	Nullable	Default	Description
Autold	int		false		
AccessToken	varchar(128)	PK	false		Access token
AccountID	varchar(128)		false		Account ID
AppID	varchar(128)		false		App ID
ProjectID	varchar(128)		true		Network ID
AccessTokenExpires	datetime		true		Expiration time of access token
RefreshToken	varchar(128)		true		Refresh token
RefreshTokenExpires	datetime		true		Expiration time of refresh token
CreateTime	datetime		true		Date created



# Project\_Application

## Installed Web Application

Field	Type	Primary	Nullable	Default	Description
AutoID	int	PK	false		
ProjectID	varchar(64)		true		Network ID
AppID	varchar(64)		true		App ID
AppCustomName	varchar(128)		true		Custom name
GroupID	varchar(1024)		true		Visible group ID
Rating	tinyint		true		Rating
Status	tinyint		true		Status (1: Installed)
CreateTime	datetime		true	now()	Date created
UpdateTime	datetime		true	now()	Date updated

# User\_Application\_Order

## Application Sorting

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int	PK	false		
UserID	varchar(64)		false		User ID (deprecated)
AppID	varchar(64)		false		App ID
DisplayOrder	int		false		Sorting
CreateTime	datetime		false		Date created
IsTop	bit		true		Sticky or not
TopOrder	int		true		Order of the sticky
AccountID	varchar(64)		true		Account ID
ProjectID	varchar(64)		true		Network ID

# User\_Application

## Installed Personal Application

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int	PK	false		
UserID	varchar(64)		true		User ID (deprecated)
AppID	varchar(64)		true		App ID
DeviceMark	varchar(128)		true		Device number (deprecated)
Rating	tinyint		true		Rating
Status	tinyint		true		Status (1: Installed)
CreateTime	datetime		true		Date created
UpdateTime	datetime		true		Date updated
DeviceType	int		false	0	Device type (deprecated)
RegID	varchar(128)		true		Registration ID (deprecated)
ProjectID	varchar(64)		true		Network ID
AccountID	varchar(64)		true		Account ID

# Account\_ActionLog

Account Actions Log

Field	Type	Primary Key	Nullable	Default	Description
AutoID	int	PK	false		
AccountID	char(36)		true		Account ID
Title	varchar(256)		true		Brief description
Description	varchar(512)		true		Detailed description
Type	tinyint		true		Log type. 1: Log in, 2: Log out, 3: Send SMS, 4: Send email, 5: Email prompt for login, 6: Client, 7: Search logs by keywords, 8: Log off user, 9: Log off network, 10: Remove user, 11: Block user, 12: Restore user, 13: Purchase extension package with balance
IP	varchar(32)		true		IP
ProjectID	char(36)		true		Network ID
CreateTime	datetime		true	now()	Date created

# apk

## Application Basic Information

```
{
  "_id":"ObjectID",
  "apkid":"string",           // App id
  "pid":"string",            // Organization id
  "apkName":"string",        // App package name
  "apkNPY":"string",         // Chinese phonetic alphabet of the package
name (sort and search)
  "avatar":"string",         // Icon
  "color":"string",          // Color
  "shortDescription": "string", // Abbreviation (obsolete)
  "description": "string",    // Short description
  "status":"int",            // Status. 0=disabled, 1=enabled, 2=deleted
  "isTemplate":"bool",       // Set as a template or not (obsolete)
  "templateType":int,        // App package type. 1=normal package,
101=bibble package ( obsolete, keep compatible )
  "apps":[],                 //Application id that belongs to this app
package
  "sourceType":1,            // 1=regular, 10=map distribution, 11=install
from application library, 12=local copy, 13=import
  "sourceId":"string",       // Source id
  "sourcePid":"string",      // Source organization number of the
application
  "isLock":"bool" ,          // Whether the application is locked
  "fixed":"bool",            // Maintaining
  "fixRemark":"string",      // Maintenance information
  "fixCaid":"string",        // Maintainer account ID
  "extLicence":[] ,          // Authorization information for the
application of distribution type
  "goodsId":"string",        // Map product package id
  "libraryId":"string",      // Map application library id
  "isImageApk":"bool",       // Whether it is the application package of
the mirror version when map creating application
  "distributeId":"string",    // Map distribution id
  "appNaviStyle":"int",      // Mobile navigation type. 0=default, 1=nine-
grid, 2=navigation bar
  "pcDisplay":"string",      // Display on PC
  "webMobileDisplay":"string", // Display on mobile web
  "appDisplay":"string",     // Display on app
  "openApiWhiteList":[],     // Whitelist of open api for the application
  "caid":"string",           // Creator account id
}
```

```
"uaid":"string",           // Account id updated by
"ctime":"datetime",        // Date created
"utime":"datetime",        // Date updated
}
```

# apksort

## Application Sorting

```
{
  "_id": "ObjectID",
  "type" : "int",          // Sorting type. (1=organization-wide starred
                           // application, 2=organization, 3=person, 4=external collaboration, 5=expired
                           // organization, 6=group in home organization, 7=starred application in current
                           // organization)
  "pid" : "string",        // Organization number
  "sortindex" : [          // Collection of sorting information
  {
    "apkid" : "string",    // Application ID
    "displayno" : "int"    // Sequence number
  }
  "caid": "string",        // Creator account ID
  "uaid": "string",        // Account ID updated by
  "ctime": "datetime",     // Date created
  "utime": "datetime",     // Date updated
}
```

# appapply

## Application Information

```
{
  "_id": "ObjectID".
  "appId" : "string", // Application ID (original application package ID)
  "pid" : "string", // Organization number
  "status" : "int", // Status (1=applying, 2=passed, 3=rejected)
  "applicantId": "string", // Applicant account ID
  "roleId": "string", // Role ID
  "remark": "string", // Notes
  "caid": "string", // Creator
  "ctime": "datetime", // Date created
  "utime": "datetime", // Date updated
  "isDel": "bool" // Deleted or not
}
```



# appauthorize

## Authorization Information

```
{
  "_id" : "ObjectId",
  "projectId" : "string",      // Organization ID
  "appId" : "string",         // App ID
  "appKey" : "string",
  "secretKey" : "string",
  "sign" : {                  // Signature
    "key" : "string",         // Signature key information
    "time" : "datetime",      // Signature generation time
    "type" : 1,               // Type (reserved information)
    "day" : 0,                // Days (reserved field)
    "ctime" : "datetime"      // Date created
  },
  "type" : 1,                 // Application authorization type. 1=all, 2=query
  "status" : 1,               // Status. 1=enabled, 2=disabled, 3=deleted
  "caid" : "string",          // Creator account ID
  "uaid" : "string",           // Modifier account ID
  "ctime" : "datetime",        // Date created
  "utime" : "datetime",
  "remark" : "string",         // Notes
  "default" : true             // Default or not (reserved field)
}
```

# appentity

## Grouping in Application

```
{
  "_id": "ObjectID",           // Application ID
  "apName": "string",          // Application name
  "shortDescription": "string", // Short description (obsolete, deprecated)
  "description": "string",      // Description
  "icon": "string",             // Icon
  "iconc": "string",            // Icon color
  "enabled": "bool",            // Enabled or not
  "isDel": "bool",              // Deleted or not
  "atype": "int",               // Application type. 3: worksheet
  "cfgs": [{                    // Collection of application entity
    configuration items
    "appcfgid": "string",        // Guid
    "dsName": "string",          // Show name
    "isNavigation": "bool",      // Whether it is a navigation or not
    "dsNo": "int",               // Show sorting
    "icon": "string",            // Icon
    "icc": "string",             // Icon color
    "wsType": "int",             // Type. (0=worksheet, 1=custom page)
    "wsid": "string",            // Worksheet ID
    "vtype": "int"               // View type in worksheet (reserved field)
  ]
  "caid": "string",             // Creator account ID
  "uaid": "string",             // Account ID updated by
  "ctime": "datetime",          // Date created
  "utime": "datetime",          // Date updated
}
```

# appitemrecoveryrecord

## Record Entity in Application Item Recycle Bin

```
{
  "_id": "ObjectID",      // Document ID
  "apkid": "string",      // Application ID
  "ctime": "datetime",    // Date created
  "icon": "string",       // AIcon of application item
  "iconcolor": "string",  // Color of application icon
  "isdelete": "bool",     // Deleted or not
  "item": "string",       // Application item ID
  "itemname": "string",   // Application item name
  "operator": "string",   // Account ID deleted by
  "projectid": "string",  // Organization number
  "secid": "string",      // Group ID
  "type": "int",          // Application item type. 0-worksheet, 1-custom page
  "uaid": "string",       // Modifier account ID
  "utime": "datetime"     // Date modified
}
```

# applog

## Application Log

```
{
  "_id" : "ObjectId",
  "projectId" : "string",    // Organization number
  "appIds" : [              // Application IDs
    "string"
  ],
  "entityid" : "string",    // Source ID (reserved parameter)
  "logType" : 1,            // Log type. 1=normal, 2=permissions
  "handleType" : 6,        // Actions type. 1=add, 2=enable, 3=disable,
4=delete, 5=export, 6=import, 7=delete completely, 8=restore
  "actionType" : 2,        // Log template
  "caid" : "string",       // Creator account ID
  "ctime" : "datetime",    // Date created
  "msg" : [                // log content, filler content for placeholder in
log template
    ""
  ]
}
```

Log Template	Description
0	[cid] created an application
1	[cid] exported 0 application
2	[cid] imported 0 application
3	[cid] enabled an application
5	[cid] disabled an application
7	[cid] deleted an application
8	[cid] completely deleted an application

Log Template	Description
9	[cid] restored an application
10	[cid] completely deleted
11	[cid] deleted
12	[cid] restored

# appnew

## New Corner Marker

```
{
  "_id" : "ObjectId",
  "caid" : "string",    // Account ID
  "appId" : "string"   // Application ID
}
```

# apprecoveryrecord

## Record Entity in Application Recycle Bin

```
{
  "_id": "ObjectID",          // ID
  "apkid": "string",          // Application ID
  "apkname": "string",        // Application name
  "appowner": "string",        // Creator account ID of application
  "ctime": "datetime",        // Date created
  "icon": "string",           // Application icon
  "iconcolor": "string",      // Application icon color
  "isdelete": "bool",         // Deleted or not
  "operator": "string",        // Operator account ID
  "projectid": "string",       // Organization number
  "uaid": "string",           // Modifier account ID
  "utime": "datetime",        // Date modified
  "wscount": "int"            // Number of worksheets
}
```

# appsettings

## Application Settings

```
{
  "_id": "ObjectID",
  "apkId": "string",           // Application ID
  "pid": "string",            // Organization number
  "memberStatus": "int",      // Status. (1=normal, 2=closed)
  "caid": "string",           // Creator account ID
  "ctime": "datetime",        // Date created
  "utime": "datetime",        // Date updated
}
```



# appshare

## Application Sharing

```
{
  "_id" : "ObjectId",
  "shareId" : "string",    // Sharing ID
  "sourceId" : "string",  // Sharing source ID
  "sourceType" : 22,      // Source type. 11=application library, 21=custom
                             page, 22=worksheet, 31=chart
  "appId" : "string",     // Application ID
  "isDel" : boolean,      // Deleted or not
  "caid" : "string",      // Creator account ID
  "uaid" : "string",      // Modifier account ID
  "ctime" : "datetime",   // Date created
  "utime" : "datetime"    // Date modified
}
```

# appsort

## Groupinng and Sorting in Application

```
{
  "_id": "ObjectID",
  "pid" : "string",      // Organization number
  "apkid" : "string",    // Application ID
  "appsortindex" : [     // Grouping and soritng in an application
    {
      "appid": "string", // Group ID in an application
      "displayno": "int" // Sequence number
    }
  ]
  "caid": "string",      // Creator account ID
  "uaid": "string",      // Account ID updated by
  "ctime": "datetime",   // Date created
  "utime": "datetime",   // Date updated
}
```

# groupsort

## Grouping and Sorting in Organization

```
{
  "_id" : "ObjectId",
  "projectId" : "string",    // Organization number
  "type" : 2,                // Sorting type. 1=starred group, 2=network
  "sortindex" : [           // Sorting entity
    {
      "_id" : "string",      // Group ID
      "no" : 0               // Sequence number
    }
  ],
  "caid" : "string",         // Creator account ID
  "ctime" : "datetime",     // Date created
  "utime" : "datetime",     // Date updated
}
```

# homessetting

## Display Settings of Personal Home Page

```
{
  "_id" : "ObjectId",
  "projectId" : "string",    // Organization number
  "displayType" : 0,
  "maDisplay" : 0,
  "exDisplay" : false,
  "caid" : "string",        // Creator account ID
  "ctime" : "datetime",    // Date created
  "utime" : "datetime",    // Date modified
  "isDel" : false          // Deleted or not
}
```

# markedapp

## Starred Application

```
{
  "_id":"ObjectID",
  "apkid" : "string",    // Application package ID
  "pid" : "string",      // Organization number
  "caid":"string",       // Account ID starred by (creator)
  "ctime":"datetime",    // Date created
  "utime":"datetime",    // Date updated
}
```

# markedgroup

## Starred Group

```
{
  "_id" : "ObjectId",
  "projectId" : "string",    // Organization number
  "groupId" : "string",      // Group ID
  "groupType" : 1,           // Grouping type. 0=personal, 1=organizational
  "displayNo" : 0,           // Sequence number
  "caid":"string",           // Account ID starred by (creator)
  "ctime":"datetime",        // Date created
  "utime":"datetime",        // Date modified
}
```

# personalgroup

## Personal Group

```
{
  "_id" : "ObjectId",           // Group ID
  "projectId" : "string",       // Organization number
  "name" : "string",            // Name
  "enName" : "string",          // English name
  "color" : "string",           // Color
  "icon" : "string",            // Icon
  "apps" : [],                  // Application in the group
  "displayNo" : 0,              // Sequence number
  "displayType" : 0,            // Display mode, 0=tile, 1=tab
  "caid":"string",             // Creator account ID
  "uaid":"string",             // Modifier account ID
  "ctime":"datetime",          // Date created
  "utime":"datetime",          // Date updated
  "isDel" : false               // Deleted or not
}
```

# projectgroup

## Organizational Group

```
{
  "_id" : "ObjectId",           // Group ID
  "projectId" : "string",       // Organization number
  "name" : "string",            // Name
  "enName" : "string",          // English name
  "color" : "string",           // Color
  "icon" : "string",            // Icon
  "apps" : [],                  // Application in the group
  "displayType" : 0,            // Display mode, 0=tile, 1=tab
  "caid":"string",             // Creator account ID
  "uaid":"string",             // Modifier account ID
  "ctime":"datetime",          // Date created
  "utime":"datetime",          // Date updated
  "isDel" : false               // Deleted or not
}
```



# worksheetsetting

## Application Item Hidden in Navigation Bar (Worksheet or Custom Page)

```
{
  "_id" : "ObjectId",
  "appId" : "string",           // Application ID
  "worksheetIds" : [           // IDs of the hidden application items
    "string"
  ],
  "caid":"string",             // Creator account ID
  "uaid":"string",             // Modifier account ID
  "ctime":"datetime",          // Date created
  "utime":"datetime",          // Date updated
  "isDel" : false              // Deleted or not
}
```

# wsaliasrelation

## Alias

```
{
  "_id": "ObjectId",
  "alias": "string",           // Alias
  "worksheetId": "string",    // Worksheet ID
  "appId": "string",          // Application ID
  "projectId": "string"       // Organization number
},
```

# activitylog

## Operation Log of Roles in Application

```
{
  "_id": "ObjectID",           //(Primary key)
  "actionId": "int",           // Action ID
  "roleId": "string",          // Role ID
  "message": "string",         // Log content
  "operatorId": "string",       // Operator account ID
  "createdTime": "DateTime",    // Date created (when the operation occurred)
}
```

# field

## Permissions for Field in Worksheet

```
{
  "_id": "ObjectID",           // (Primary key)
  "appId": "string",           // Application ID
  "sheetId": "string",          // Worksheet ID
  "fieldId": "string",          // Field ID
  "notRead": "bool",            // Whether the field is not visible
  "notEdit": "bool",            // Whether the field is not editable (when modifying)
  "notAdd": "bool",             // Whether the field is not editable (when creating)
}
```

# role

## Roles in Application

```
{
  "_id":"ObjectID",                // (Primary key)
  "projectId": "string",           // Organization number
  "appId":"string",                // Application ID
  "isDefault":"string",            // Whether it is a default role (an
exclusive feature for external roles)
  "name":"string",                 // Role name
  "description":"string",          // Role description
  "roleId":"int",                  // Role type. 0: custom role, 10: read
only, 25: visitor (hidden in business), 50: member, 100: Admin
  "roleCategoryId":"int",          // Category. 0: internal user role, 10:
external user role
  "permissionWayId":"int",         // Permission authorization mode (simple
mode and custom mode). 0: custom permissions (assign different permissions to
fields in the view), 20: view-only permissions for all records, 30: manage only
your own records (can only view and manage your own records), 40: manage only
your own records and those of your subordinates, 50: view all & manage your own
records, 60: view all & manage your own records and those of your subordinates,
80: manage all records
  "sortIndex":"int",              // Displaying order of roles (the smaller
they are, the higher they are)
  "createdAccountId":"string",     // Creator account ID
  "createdTime":"DateTime",        // Date created
  "updatedAccountId":"string",     // Modifier account ID
  "updatedTime":"DateTime",        // Date modified
  "deleted":"bool",               // Tombstone status
  "users":[{
    "accountId":"string",          // Authorized account ID
    "addedTime":"DateTime"         // Time of adding authorization
  }],
  "authorizedOrgs":[{
    "projectId":"string",          // Authorized organization number
    "addedTime":"DateTime",        // Time of adding authorization
  }],
  "departments":[{
    "projectId":"string",          // Authorized department ID
    "addedTime":"DateTime",        // Time of adding authorization
  }],
  "jobs":[{
    "jobId":"string",              // Authorized position ID
```

```

    "addedTime":"DateTime",          // Time of adding authorization
  }],
  "sheets":[{
    "sheetId":"string",              // Worksheet ID
    "canAdd":"bool",                 // Whether it can be added
    "readLevelId":"int",             // Level of viewng permission. 0: not
authorized, 20: only my own, 30: my and subordinates', 100: all
    "editLevelId":"int",             // Level of editing permission. 0: not
authorized, 20: only my own, 30: my and subordinates', 100: all
    "removeLevelId":"int",           // Level of deleting permission. 0: not
authorized, 20: only my own, 30: my and subordinates', 100: all
    "navigateHide":"bool",           // Whether it is hidden from navigation
  }],
  "pageIds":["string"],              // Set of custom page ID
  "navigateHidePageIds":["string"],  // Set of ID hidden in the navigation bar
on custom page
}

```

Set of ID hidden in the custom page navigation bar

# view

## Permissions for View

```
{
  "_id": "ObjectID",          // (Primary key)
  "appId": "string",          // Application ID
  "roleId": "string",          // Role ID
  "sheetId": "string",         // Worksheet ID
  "viewId": "string",          // View ID
  "notRead": "bool",           // Whether the view is not visible
  "notEdit": "bool",           // Whether the view is not editable
  "canRemove": "bool",         // Whether the view can be deleted
}
```

# worksheet

## Worksheet

```
{
  "_id": "ObjectID",           // Worksheet ID
  "templateid": "string",      // Template ID
  "serverurl": "string",       // Service library address of row data
  "name": "string",            // Worksheet name
  "maxautoid": "int",           // Maximum number
  "pyname": "string",           // Chinese phonetic alphabet of worksheet name
  "desc": "string",            // Worksheet description
  "status": "int",              // Status. 0: not enabled, 1: normal, 9: deleted, 999:
completely deleted
  "type": "int",                // Type. 1: normal worksheet, 2: blank sub-form, 3:
worksheet of external portal
  "entityname": "string",       // Record entity name
  "btnname": "string",          // Name of Add Record button
  "sourcetype": "int",          // 1: Manual, 2: Excel, 3: Project, 4: Template
  "projectid": "string",        // Organization number
  "ctime": "datetime",          // Date created
  "caid": "string",             // Creator account ID
  "utime": "datetime",          // Date updated
  "sharerange": "int",          // Sharing scope. 1: disable sharing, 2: visible to
public, 3: visible to members
  "sourceid": "string",         // Source worksheet ID
  "handlerelation": "bool",     // Enable new association storage method
  "closeautoid": "bool",        // Disable autoid logic
  "handleusers": "bool",        // Processed redundant member permissions
  "version": "int",             // Version number of control template
  "rownum": "long"              // Number of records
  "alias": "string",            // Worksheet alias
  "ogid": "string",             // Original worksheet ID, used for worksheet upgrade
  "adset", [{"key": "string", "value": "string"}], // Advanced settings
}
```

## AdvancedSetting(adset) Description

key	Description (string)
XXX	Desc



# wsbtn

## Custom Action

```
{
  "_id": "ObjectId",
  "wsid": "string",           // Worksheet ID
  "name": "string",          // Button name
  "views": [],               // Show view
  "showtype": "int",         // 1: Always, 2: Meet filter conditions, 9: Custom
page
  "filters": [],             // Filter items. Refer to filter parameters in
worksheet
  "clicktype": "int",        // 1: execute immediately, 2: double confirmation,
3: fill in
  "confirmmsg": "string",    // Confirmation
  "surename": "string",      // The OK button
  "cancelname": "string",    // The Cancel button
  "writeobj": "int",         // Object. 1: current record, 2: associated record
  "writetype": "int",        // Type. 1: fill in the field, 2: new associated
record
  "relationcontrol": "string", // Associated record ID
  "addrcontrol": "string",    // New associated record ID
  "controls": [{
    "controlid": "string",    // Control ID
    "type": "int",            // 1: Read only, 2: Fill, 3: Required
    "defsource": "[{"rcid": "Control ID in associated worksheet", "cid": "Control
ID", "staticValue": "string"}]"
  }],                        // The filling in control
  "workflowtype": "int",     // 1: Execute, 2: Not execute
  "workflowid": "string",    // Workflow ID
  "status": "int",           // Status. 1: normal, 9: deleted
  "color": "string",         // Color
  "icon": "string",          // Icon
  "ctime": "datetime",       // Date created
  "caid": "string"           // Creator account ID
}
```

# wscollection

## Option Set

```
{
  "_id": "ObjectId",
  "appid": "string",           // Application ID
  "cid": "string",            // Option set ID
  "name": "string",           // Name
  "ops": [{
    "k": "string",             // key
    "v": "string",             // Name
    "idx": "string",           // Sorting
    "color": "string",         // Color
    "score": "double",         // Score
    "isdel": "bool",           // Deleted or not
  }],                          // Option
  "color": "bool",             // Enable colorful
  "enscore": "bool",           // Enable score
  "wsids": ["string"],         // Worksheet IDs
  "status": "int",             // Status. 1: normal, 9: delete
  "caid": "string",            // Creator account ID
  "ctime": "date",             // Date created
}
```

# wscontrols

## Configuration of Controls in Worksheet

```
{
  "cid":"string",           // Control ID
  "wsid":"string",         // Worksheet ID
  "name":"string",         // Name
  "type":"int",            // Control type. Refer to enumeration
  "attribute":"int",       // Attribute. 1: Title
  "enumdef":"int",         // Subtype. Refer to the enumeration
  "enumdef2":"int",        // Subtype. Refer to the enumeration
  "dtsrc":"string",        // Source data. Refer to the description
  "sourcecid":"string",    // Source control ID
  "unit":"string",         // Unit
  "opts":[{
    "Key":"string",        // Unique key value
    "Value":"string",      // Text value
    "IsDeleted":"bool",    // Deleted or not
    "Index":"int",         // Sorting
    "Color":"string",      // Color value
  }],                      // Option (Positioning information)
  "noticeitem":"int",      // Notification item (control of members). 0: no
notification, 1: add notification
  "userpermission":"int",  // Permissions (control of members). 0: entry
only, 1: member, 2: owner
  "showcids":["string"],   // ID list of fields displayed in the associated
worksheet
  "unique":"bool",         // Unique value verification
  "covercid":"string",     // ID of cover control (single row association)
  "defmen":["string"],     // Default account ids, all in array, creator:
user-self
  "desc":"string",         // Field description
  "strdefault":"string",   // General string field. Refer to description
  "fieldpermission":"string", // Null or "111", the 1st digit represents whether
can view; the 2nd digit represents whether can edit (read-only); the 3rd digit
represents whether can add; 1: can, 0: cannot,
  "adset":[{
    "key":"string",        // Configuration key. Refer to the following
description
    "value":"string"       // Value. Refer to the following description
  }],                      // Advanced settings (dictionary<string,string>
for configuring transfers)
```

```
"alias": "string", // Alias (for API)
}
```

## Contol Type (type)

Control	Enum value	Description
Text	1	Text (Deprecated)
TextArea	2	Text
MobilePhone	3	Telephone
Landline	4	Landline
Email	5	Email
Number	6	Numeric
Certificates	7	ID Number
Amount	8	Amount
Radio	9	Choice
CheckBox	10	Choices
Select	11	Dropdown
Attachment	14	File
Date	15	Date
DateTime	16	Time
District	19/23/24	Region/City
Relation	21	Free Link

Control	Enum value	Description
SplitLine	22	Line
BigAmount	25	Uppercase
UserSelect	26	Members
DepartmentSelect	27	Department
Score	28	Level
RelationWorksheet	29	Relationship
RelationWSColumn	30	Foreign
WSFormula	31	Formula
FormulaString	32	Concat
Increase	33	Autonumber
DetailTable	34	Subform
Cascade	35	Cascading
CheckRadio	36	Check items
Summary	37	Rollup
FormulaDate	38	Formula (date)
ScanCode	39	Scan Code
Location	40	Positioning
RichText	41	Rich Text

Control	Enum value	Description
Signature	42	Signature
OCR	43	OCR
Role	44	Application Role
Embed	45	Embedded
Time	46	Time
Code	47	Barcode
OrgRole	48	O-roles
Btn	49	Query Button
Search	50	API Query

**DataSource Description(dtsrc)**

Control	Value Description
BigAmount	Referenced Field ID
RelationWorksheet	Associated Worksheet ID
RelationWSColumn	Associated Control ID
FormulaDate	Formula ID or End Date Field ID
Select/CheckBox/Radio	Option Set ID
Embed	URL or ID of Embedded Object
Code	Referenced Field ID

EnumDefault Description(enumdef)

Control/Value	0	1	2	3	4	5
TextArea	Default single line is multi-line	Multi-line	Single line			
MobilePhone	International numbers	Non-International				
UserSelect	Single	Multiple				
DepartmentSelect	Single	Multiple				
Score		1-5 star	1-10 level			
RelationWorksheet		Relationship	Subform			
Increase	Original number	Number of digits in format				
Summary	Refer to enumeration					
FormulaDate		Duration	Add/subtract a date	Time between a date and today		
Attachment		New ones in front	Old ones in front			
Location	Not show map	Show map				

Control/Value	0	1	2	3	4	5
Number	Show thousandths	Not show thousandths				
Select/CheckBox/Radio		Enable score				
OCR		General Recognition	ID Card	Invoice Recognition		
Embed		iframe	Chart			
Code		Barcode	QR code			

## EnumDefault2 Description (enumdef2)

Control	Value Description
TextArea	0: No format validation, 1: Format validation
Increase	Display digits
RelationWorksheet	Single digit for creation permission, ten digits for association permission. 1 for disable. 0: All allowed, 1: Disable addition, 10: Disable association, 11: Disable addition and association.
Summary	Enumeration value for the summary result. Refer to the control type
Radio/Select	0 or null: Not enabled, 1:Enable Colorful
FormulaDate	1: Ignore year of end date
Attachment	0: No restriction, 1: Take photo, 2: Take video, 3: Take photo or video
Location	Positioning range. 0: No restriction, 1: Current location
Select/CheckBox/Radio	1: Enable Colorful



Control	Value Description
UserSelect	0: No restriction on the selection range, 1: Restriction on the selection range
Code	QR code data source. 1: Internal link, 2: External link, 3: Field value

**StrDefault Description (strdefault)**

Control	Value Description
TextArea	The 1st digit indicates whether to disable album; the 2nd digit indicates whether to enable code scanning. "11": cannot select album, and enable code scanning
FormulaDate	Null or 0: start at 0:00, end at 0:00; 1: start at 0:00, end at 24:00
RelationWorksheet	The 1st digit indicates whether to verify view and permission; the 2nd digit indicates whether to disable album; the 3rd digit indicates whether to allow only scan code to associate. "111": Verify permissions, disable albums, and scan code only
Attachment	The 1st digit indicates whether to disable album; the 2nd digit indicates whether to allow only mobile input. "11": cannot select album, and can only input on mobile
ScanCode	The 1st digit indicates whether to disable album; the 2nd digit indicates whether to allow only inputting on mobile. "11": can not select album, and can only input on mobile
Increase	increase New auto-numbering
RelationWSColumn	Null or 00: Redundant values, 10: Show only
Location	Null or 0: Locate position on map, 1: Get the current latitude and longitude (APP)

# AdvancedSetting Description (adset)

key	Value Description (string)	Control-Default
filters	Associate filters	Relationship/Rollup
defsource	<div>[{"rcid":"Control ID in associated worksheet","cid":"Control ID","staticValue":"string"}]</div>	
hasdef	Whether or not the field has a default value when it is hidden. 1: Yes, 0: No	
getinput	Get value before filling form. 1: In advance, 0: Not in advance	
getsave	Submit form directly after getting. 1: Submit directly, 0: Not submit directly	
direction	Options arrangement	Selection
dismanual	Manual input. Null or 0: Allow, 1: Disable	
scantype	Code type. Null or 0, 1: only barcode, 2: only QR code	
distance	Distance, 100/200/500, in meters	Positioning
checkrange	Null or 0: No verification, 1: Verification	
min	Minimum	
max	Maximum	
summaryresult	Summary result. Null or 0: Value, 1: Percentage	Rollup
showtype	Display type. 1: Card, 2: List, 3: Dropdown box (tiled), 4: Dropdown tree, Null: Single-Card, Multiple-List	Relationship

key	Value Description (string)	Control-Default
showtype	Display type. 5:Year, 4:Year-Month, 3:Year-Month-Day, 2:Year-Month-Day-Hour, 1:Year-Month-Day-Hour-Minute, 6:Year-Month-Day-Hour-Minute-Second	Date
showtype	Display type. Null or 0: Checkbox, 1: Switch, 2: Yes/No	Check items
showtype	Display type. Null or 0: Value, 1: Percentage, 2: Progress	Numeric
showtype	Display type. 0: Dropdown, 1: Tile, 2: Progress	Choice
showformat	Format of display. Null or 0: ISO, 1: China, 2: US, 3: EU	Date
allowlink	Allow to view records. Null or 0: Not allow, 1: Allow	Relationship/Embedded
allpath	Show full path. Null or 0: Show last level, 1: Show full path	Cascading
anylevel	Allow to select any level. Null or 0: Select any level, 1: Select the last level	Cascading
hide	Hide or not. 1: Hide, Null or 0: Not hide	
ddset	Configure the drop-down box to display the cover. 1: Configure, Null or 0: Not configure	Relationship
dateformulatype	Calculation method. 1: Target date minus date of today, 2: Date of today minus target date	Formula
hiddeneg	Not show negative values. 1: Not show	Formula
increase	Autonumber configuration	Autonumber
showxy	Show latitude and longitude. 1: Show, Null or 0: Not show	Positioning

key	Value Description (string)	Control-Default
allowweek	Allowed combinations of days 1234567 of the week	Date
allowtime	Allowed interval 2:30-12:45 middle-split	Date
timeinterval	Preset interval to N minutes	Date
checktype	Type of multi-select box. 0 or null: Tile, 1: Dropdown	Selection
allowadd	Allow new options. 0: Not allowed, 1: Allowed	Selection-0/Subform-1
allowcancel	Allow cancelling. 0: Not allowed, 1: Allowed	Relationship /Subform-1
allowedit	Allow editing. 0: not allowed, 1: allowed	Sunform-1
userrange	Users allowed to be selected	Members
regex	Regular expression <code>{type="quick type on the left",regex="string",err="string"}</code>	Text
ocrmap	OCR field mapping <code>[{type="",name="",cid="",subId=""}]</code>	OCR
allowcountries	Countries allowed to be selected <code>[{iso2="cn",name="China",dialCode:"+86"}]</code>	Telephone
commcountries	Frequently selected countries <code>[{iso2="cn",name="China",dialCode:"+86"}]</code>	Telephone
defaultarea	Default area code	Telephone
suffix	Formula suffix	Numeric type
prefix	Formula prefix	Numeric type
searchcontrol	Associate search fields	Relationship

key	Value Description (string)	Control-Default
searchtype	0 or null: Fuzzy match, 1: Exact search	Relationship
clicksearch	0 or null: Show directly, 1: Show data after search	Relationship
searchfilters	<code>[{controlId:"string",filterType:int}]</code>	Relationship
sorts	<code>[{controlId:"string",isAsc:bool (false: Descending, true: Ascending)}]</code>	Relationship/Sunform
weekday	Weekday 1234567	Date formula
batchcids	<code>[""]</code> Bulk add fields	Subform
dynamicsrc	Default value for worksheet query []	
defaultfunc	Default value of function	
defaulttype	Type of default value. Null or 0: Base default, 1: Function 2: Worksheet query	
nullzero	When the formula is empty, calculated as 0. Null or 0: Not as 0, 1: As 0	Formula
usertype	User type. 1: Internal member, 2: External portal	Members-selection
watermark	Add watermark <code>["user","time","address","xy"]</code>	File
maxcount	Maximum number	File
filetype	File type <code>{type:int,values:[""]}</code>	File
compress	Upload compressed attachments. Null or 0: Not compressed, 1: Compressed	File
width	Width	

key	Value Description (string)	Control-Default
height	Height	
allowsingle	Allow single addition	Subform
numshow	Null or 0: Normal display, 1: Show percentage, 2: Show progress	Numeric type
thousandth	Null or 0: Show thousandths, 1: Not show thousandths	Numeric type
itemicon	Icon style	Level
itemcolor	Color <code>{type:int(1: Fixed, 2: Dynamic), color:"#000", colors: [{key:"1",value:"#000"}]}</code>	Level/Numeric
itemnames	Custom text <code>[{key:"1",value:"general"}]</code>	Level/Check items/Numeric
showvalue	Show results. 0 or null: Not show, 1: Show	Level
numinterval	Interval	Numeric
showinput	Show input box	0: Not show, 1: Show
faultrate	Error tolerance	Barcode
analysislink	Parse link. 1: Parse, Null or 0: Not parse	Text/Concat

# wsfilter

## Filter

```
{
  "_id": "ObjectID",
  "wsid": "string",    // Worksheet ID
  "type": "int",       // Type. 1: personal, 2: public
  "module": "int",     // Type. 0-1: worksheet, 2: statistics
  "name": "string",    // Name
  "status": "int",     // Status. 1: normal, 9: delete
  "items": [] ,       // Filter items. Refer to the filter parameters in the
worksheet
  "ctime": "datetime", // Date created
  "caid": "string",    // Creator account ID
}
```

# wslogs

## Operation Log in Worksheet

```
{
  "_id": "ObjectID",
  "entityid": "ObjectID", // Worksheet ID (wsid)
  "templateid": "string", // Log template ID
  "wsid": "string",       // Worksheet ID
  "viewid": "string",     // View ID
  "params": [],           // Data content
  "type": "int",          // Type. 1: add, 2: edit, 3: delete, 4: import, 5:
export, 6: restore
  "ctime": "datetime",    // Date recorded
  "caid": "string",       // Operator account ID
}
```



# wsrowindexconfigs

## Index

```
{
  "worksheetId":"string",           // Worksheet ID
  "projectId":"string",             // Organization ID
  "customeIndexName":"string",      // Custom index name
  "indexName":"string",             // Index name in the database
  "fields":[{"
    "fieldId":"string",
    "indexType":"string"
  }]                                // Index field
  "uniqueIndex":"bool",             // Unique index or not
  "wildcardIndex": "bool",          // Wildcard text index or not
  "backgroundIndex": "bool",        // Backend index or not
  "indexStateId":"string",          // Index status ID. -1: Creation failed,
0: In queue, 1: Normal
  "messageQueueId":"string",        // Message queue number
  "createAccountId":"string",        // Creator account ID
  "modifyAccountId":"string",        // Modifier account ID
  "createTime":"datetime",          // Date created
  "updateTime":"datetime",          // Date updated
  "delete":"bool",                  // Deleted or not
}
```

# wsshare

## Share

```
{
  "_id": "ObjectID",
  "wsid": "string",           // Worksheet ID
  "downloadnum": "long",     // Downloads
  "browsenum": "long",       // Views
  "type": "int",             // Sharing type. 1: list, 2: details
  "rowid": "string",         // Record ID
  "appid": "string",         // Application ID
  "viewid": "string",        // View ID
  "ctime": "datetime",       // Date created
  "caid": "string",          // Creator account ID
}
```

# wsview

## View in Worksheet

```
{
  "_id":"ObjectId",
  "name":"string",           // View name
  "status":"int",           // Status. 1: Normal; 9: Deleted
  "wsid":"string",          // Worksheet ID
  "unread":"bool",          // Unread or not
  "filters":[],             // Filter items. Refer to the filtering
parameters in the worksheet
  "fastfilters":[],         // Configuration for quick filtering
  "navgroup":[{"controlId":"","isAsc:bool,viewId:"string",filterType:"int"}], //
Navigation filter
  "sortcid":"string",        // Default sorting field
  "sorttype":"int",          // 1: Descending, 2: Ascending
  "controls":["string"],     // ID of hidde control
  "displaycontrols":["string"], // Fields to be displayed on the card
  "showcontrols":["string"], // Dispalyed fields
  "controlssort":["string"], // Displaying order
  "layersname":["string"],   // Level name
  "covercid":"string",       // Card cover ID
  "covertime":"int",         // Card display. 0: Filled 1: Full display
  "customdisplay":"bool",    // Customize the card display or not
  "viewtype":"int",          // 0: List, 1: Board, 2: Org, 3: Gallery,
4: Calendar, 5: Gantt
  "viewcontrol":"string",    // Dimension control ID
  "childtype":"int",         // 0 or 1: Single-worksheet hierarchy, 2:
Multi-worksheet hierarchy
  "viewcontrols":[{"
    "wsid":"string",
    "cid":"string",
    "covercid":"string",
    "covertime":"int",
    "scids":["string"]
  }],                        // Multi-worksheet dimension control IDs
  "showcn":"bool",          // Show control name or not
  "sorts":[],               // Filter items. Refer to the filter
parameters in the worksheet
  "rowheight":int ,         // Row height
  "caid":"string",          // Creator account ID
  "ctime":"datetime",       // Date created
  "adset":[{"
```

```

    "key": "string",           // Configuration key. More details are
    described below
    "value": "string"         // Value. More details are described below
  }]                           // Advanced settings
}
```

## Advanced Setting Description (adset)

key	Value Description (string)
hiddenone	Hide boards with no data. 1: Hide, 0 or null: Not hide
appshowtype	Layout 1, 2, 3 ...
abstract	ID of abstract control
showformat	Display format of fields <code>[{cid:xxxx,format:yyyyMMdd HH:mm:ss}]</code>
coverposition	Cover position
checkradioid	ID of checkbox control
customdisplay	Customize displaying order or not
begindate	ID of the start control in Calendar view
enddate	ID of the end control in Calendar view
unlunar	Show the lunar calendar or not. Null or 0: Show, 1: Not show
unweekday	Not displayed day of the week 1234567
colorid	ID of the color control
hidebtn	Hide the unavailable button. 1: hide, 0 or null: not hide
freezecol	IDs of frozen columns

key	Value Description (string)
colwidth	Column width setting
hour24	The 24-hour format. Null or 0: No, 1: Yes
sysids	Display of system fields
sysort	Displaying order of system fields
opencover	Click to view the cover. Null or 1: Allow, 2: Not allow
enablebtn	Query button. Null or 0: Disable, 1: Enable
clicksearch	Show data after query. Null or 0: Show directly, 1: Show after query
milepost	ID of the milestone control
navgroup	ID of the grouping control
calendartype	Calendar type (defined by the front-end)
refreshtime	Auto refresh time for the view
calendarcids	Date control in the view <code>[{begin:"ID of the start control",end:"ID of the end control",mark:"label"}]</code>
weekbegin	First day of every week 1234567
showall	Null or 0: not show, 1: show all calendars
nousenav	Null or 0: as default, 1: not as default

# ws[wsid]

## Row Record Data db.ws + WorksheetId

```
{
  "_id": "ObjectID",
  "rowid": "string",           // Row record ID
  "wsid": "string",           // Worksheet ID
  "autoid": "int",            // Autonumber
  "status": "int",            // Status. 1: Normal, 9: Deleted, 999: Completely
  deleted
  "cid[Control ID]": "object", // Column value. Store data in different formats
  depending on the control type
  "cid[Control ID]": "object", // Column value. Store data in different formats
  depending on the control type
  "relations": "[{
    "cid": "string",          // Control ID
    "wsid": "string",          // Associated worksheet ID
    "rows": "[string]",        // Associated row record IDs
  }]",                        // Old associative memory, no longer used in any
  edition after 2021
  "rc"+cid: ["rowid"],        // List of associated row records
  "ot"+cid: "{
    "idx": "int",             // Sort index
    "score": "double"         // Score
  }",                          // Index and score of options
  "sharerange": "int",        // Sharing scope. 1: Sharing disabled, 2: Visible to
  public, 3: Visible to members
  "ctime": "datetime",        // Date created
  "caid": "string",           // Creator account ID
  "ownerid": "string",         // Owner account ID
  "utime": "datetime",         // Date edited
  "dtime": "datetime",         // Date deleted
  "users": ["aid"],            // Redundant search for joiners
  "owners": ["aid"],           // Redundant search for owners
  "keywords": "string",        // Search for keywords
}
```

# wslogs

## Operation Log for Row Records

```
{
  "_id": "ObjectID",
  "entityid": "ObjectID", // Row record ID (rowid)
  "templateid": "string", // Log template ID
  "wsid": "string",       // Worksheet ID
  "viewid": "string",     // View ID
  "params": [],           // Data content
  "type": "int",          // Type. 1: add, 2: edit, 3: delete, 4: import, 5:
export, 6: restore
  "ctime": "datetime",    // Date recorded
  "caid": "string",       // Operator account ID
}
```

# Deployment Issues

## How to reinstall?

1. Stop the HAP service that may have been running, execute `bash . /service.sh stopall` (output stopped if successful) in root directory of manager;
2. Back up the files of the HAP service, `mv /data/hap/ /home/hapbak/` (you can customize the target location of the backup, usually not needed for the first deployment, it can be `rm -rf /data/hap/`);
3. Confirm again that it is cleaned up. Execute `docker ps | grep hap`, `netstat -ntpl | grep 38881`, and `ps -ef | grep 'hap\|service.sh' | grep -v grep` respectively to ensure that the output is empty;
4. Restart the manager with `bash . /service.sh start`, and visit `http://{server IP}:38881` to install again.

## Failed to initialize?

You can check the reason of failure by executing the command `cat /data/hap/script/hap.log`. In most cases, it is due to **insufficient space**, **mirror not exist**, **port occupied**, **iptables failed**, etc. If **iptables failed**, the reason is that closing the firewall will clear the iptables rules, so you need to restart Docker and regenerate default iptables rules, and [reinstall](#How to reinstall) after fixing the problem.

## Unable to access the Internet in the container?

Using firewalld as a firewall, if the container can not access the external network (**under normal circumstances, HAP services do not need to interoperate with the Internet, but inevitably some system features need the Internet, such as sending mail, and SMS**), you need to add the following firewall rules. Execute both 2 commands (**172.16.0.1/ 12 is the network segment in the container**):

Allow the network segment in the container to access the Internet (permanent effect)

```
firewall-cmd --zone=trusted --add-source=172.16.0.1/12 --permanent
```

Allow the network segment in the container to access the Internet (temporary effect)



```
firewall-cmd --zone=trusted --add-source=172.16.0.1/12
```

## How to set the startup (CentOS as an example)?

Make sure it has execution permissions `chmod +x /etc/rc.d/rc.local`

Modify `/etc/rc.d/rc.local` (make sure the file has executable permissions), adding the following script:

```
sleep 30
docker system prune -f
/bin/rm -f {manager absolute path}/service.pid
/bin/bash {manager absolute path}/service.sh startall
```

## HAP service does not start properly after restarting the server?

Execute `bash . /service.sh stopall` in the root directory of the manager, if `service.pid` still exists, delete `rm -f service.pid`, execute `bash . /service.sh startall` and wait for the command to finish.

## What if the key is lost and server id is not displayed?

1. Stop the service, and execute `bash . /service.sh stopall` in the root directory of manager;
2. Execute the command `ps -ef | grep 'hap\|service.sh' | grep -v grep` (if there is output, kill all corresponding pid);
3. Execute `bash . /service.sh startall` and wait for the command to finish.

## After restarting HAP service due to power failure and other unexpected circumstances, workflow, statistics and other features still can not be used as normal?

1. Stop the service, and execute `bash . /service.sh stopall` in the root directory of manager;
2. Back up `/data/hap/`;

3. Execute the command `rm -rf /data/hap/script/volume/data/{kafka,zookeeper}/*` to remove abnormal data from the message queue, which will not cause data loss in normal conditions **(unless there is an unfinished workflow)**;
4. Execute `bash . /service.sh startall` in the root directory of manager and wait for the command to finish.

## Documents cannot be previewed online?

Since the document preview service needs to read files from the microservice application, if the intranet environment cannot access the system's external access address, the document preview will fail. In this case, you need to add the environment variable `ENV_FILE_INNER_URI` to the doc configuration of `docker-compose.yml` to specify the system intranet access address. Refer to the following:

```
services:
  doc:
    environment:
      ENV_FILE_INNER_URI: "10.140.100.6:8880"
```

## Failed to export a worksheet as Excel?

Due to the amount of data in some worksheets, when exporting, there will be 504 Gateway Time-out, etc. This is generally due to the default timeout time and file size limits of the proxy layer, the following rules can be added (nginx as an example):

```
location ~ /excelapi {
  proxy_set_header Host $http_host;
  proxy_read_timeout 1800s;
  client_max_body_size 256m;
  proxy_pass http://hap; # Here to modify according to the actual upstream name
}
```

## Interface timed out for uploading attachments?

When uploading files over 4MB, it takes a slice upload mode by default. In some cases, the network may cause a timeout for uploading a particular slice, thus failed to upload the entire file. To solve this problem, add the following rules (nginx as an example)

---

```
location ~ /file {
    proxy_set_header    Host $http_host;
    proxy_read_timeout  3600s;
    client_max_body_size 20480m;
    proxy_pass http://hap; # Here to modify according to the actual upstream name
}
```

## Failed to load workflow list, or unable to download attachments?

In the case of adding proxy to HAP service, you need to ensure that the relevant environment variables in `/data/hap/script/docker-compose.yaml` are the same as the access address when using proxy address for access, and you also need to ensure that the proxy configuration contains all the recommended configuration. View more details in [set proxy](#).

## How to enable subpath deployment?

Adjust the environment variable `ENV_MINGDAO_SUBPATH` in `docker-compose.yaml` corresponding to the microservice application, as follows:

```
services:
  app:
    environment:
      ENV_MINGDAO_SUBPATH: "/hap"          # Add, e.g. /hap
```

## How to enable two access addresses?

Set environment variables `ENV_EXT_MINGDAO_PROTO`, `ENV_EXT_MINGDAO_HOST`, `ENV_EXT_MINGDAO_PORT` in `docker-compose.yaml` corresponding to the microservice application (\*a set of configuration corresponding to `ENV_MINGDAO_PROTO`, `ENV_MINGDAO_HOST`, `ENV_MINGDAO_PORT`). Expose port 18880 (the corresponding host port is customizable, here still use 18880), and resolve `http://hap1.domain.com` to port 18880 of the host (**if you use the host's internal and external IP directly, you can ignore the domain name resolution configuration**), as follows:

```
services:
  app:
    environment:
```

```
ENV_EXT_MINGDAO_PROTO: "http"
ENV_EXT_MINGDAO_HOST: "hap1.domain.com"
ENV_EXT_MINGDAO_PORT: "80"
ports:
  - 8880:8880
  - 18880:18880
```

## How to modify the default storage path?

### New installation

#### Version 3.6.0 and later

Before starting HAP manager (before executing `./service.sh start`), modify the `installDir` parameter value in `service.sh`.

#### Version before 3.6.0

Before starting HAP manager (before executing `./service.sh start`), create `/etc/pdcaptain.json`, and specify the `dataDir` property. The related files will be stored in the directory `/app/hap` after the startup, with the following configuration:

```
{
  "dataDir": "/app/hap"
}
```

### Migration

#### Version 3.6.0 and later

For the installed HAP, modify the `installDir` parameter value in `service.sh`. If the HAP service is stopped, move all the files in `/data/hap` to `installDir` and restart the service.

#### Version before 3.6.0

For the installed HAP, create `/etc/pdcaptain.json` and specify the data storage directory `dataDir`. If the HAP service is stopped, move all the files in the `/data/hap` to `dataDir` and restart the service.

## How to access each storage component externally?

In standalone deployment mode, each dependent storage component, including `mysql`, `mongodb`, `redis`, `kafka`, `file`, `elasticsearch`, is started in the container by default, and the ports of these components are not exposed to the external by default. If you need external connections, you can expose the corresponding ports by modifying the `ports` field in `docker-compose.yaml` corresponding to the microservice application, as follows:

Please read the documentation about [data security](#) carefully before operation ⚠⚠⚠

```
services:
  app:
    ports
      - 8880:8880
      - 3306:3306    # mysql
      - 27017:27017  # mongodb
      - 6379:6379    # redis
      - 9092:9092    # kafka
      - 9000:9000    # file
      - 9200:9200    # elasticsearch
```

## How to partially enable external storage components in standalone deployment mode?

Set the environment variable `ENV_STANDALONE_DISABLE_SERVICES` in `docker-compose.yaml` corresponding to the microservice application, which supports setting `mysql`, `mongodb`, `redis`, `kafka`, `file`, `elasticsearch`, as follows:

```
services:
  app:
    environment:
      ENV_STANDALONE_DISABLE_SERVICES: "redis,file" # If there is more than
      one, separate them with English commas
```

To enable the custom storage component, you need to configure the connection address of the corresponding service via environment variable, refer to [environment variable description](#).

## How to customize the maximum memory for mongodb?

Set the environment variable `ENV_MONGODB_CACHEGB` in `docker-compose.yml` corresponding to the microservice application. It defaults to **(physical memory-10)/2** (in G), as follows:

```
services:
  app:
    environment:
      ENV_MONGODB_CACHEGB: "3"
```

## How to be referenced by other systems via IFrame?

By default, it is supported to be used by IFrame embedded in the same domain. If you want to be introduced by other systems, you can set environment variable `ENV_FRAME_OPTIONS` by modifying `docker-compose.yml` corresponding to microservice application, supporting `ALLOWALL`, `SAMEORIGIN`, `DENY`, `ALLOW-FROM uri`.

```
services:
  app:
    environment:
      ENV_FRAME_OPTIONS: "ALLOWALL"
```

## ## Set IP whitelist access policy

To restrict access based on the source IP, you need to configure it based on the first layer of the client's accessing proxy. Below is the configuration example:

```
# Example 1
http {
  server {
    .....
    # Allow access from IP 192.168.0.1
    allow 192.168.0.1;
    # Allow access from IPs within the 192.168.0.1/32 subnet
    allow 192.168.0.1/32;
    # Deny access from all other IP addresses
    deny all;

    location / {
      .....
    }
  }
}
```

```

    }
  }
}

# Example 2
http {
  server {
    .....

    location / {
      # Allow access from IP 192.168.0.1
      allow 192.168.0.1;
      # Allow access from IPs within the 192.168.0.1/32 subnet
      allow 192.168.0.1/32;
      # Deny access from all other IP addresses
      deny all;
    }
  }
}

```

## How to customize the timeout for webhook execution in workflows?

Set the environment variable `ENV_WORKFLOW_WEBHOOK_TIMEOUT` in `docker-compose.yaml` corresponding to the microservice application, with a default of 10 (in seconds), as follows:

```

services:
  app:
    environment:
      ENV_WORKFLOW_WEBHOOK_TIMEOUT: "30"

```

## How to customize the timeout for code block execution in workflows?

Set the environment variable `ENV_WORKFLOW_COMMAND_TIMEOUT` in `docker-compose.yaml` corresponding to the microservice application, with a default of 10 (in seconds), as follows:

```

services:
  app:

```

```
environment:  
  ENV_WORKFLOW_COMMAND_TIMEOUT: "30"
```

## How to customize the maximum memory for code block execution in workflows?

Set the environment variable `ENV_WORKFLOW_COMMAND_MAXMEMORY` in `docker-compose.yml` corresponding to the microservice application, with a default of 64 (in M), as follows:

```
services:  
  app:  
    environment:  
      ENV_WORKFLOW_COMMAND_MAXMEMORY: "128"
```

## How to customize consumer threads of message queues in workflows?

Before modifying the consumer threads of message queues, you need to adjust the number of partitions of the related Topic (`Workflow`, `Workflow-Batch`, `Workflow-Button`, `Workflow-Process`, `Worksheet`, `Worksheet-Batch`) in Kafka, with a default of 3. The number of consumer threads set cannot be greater than the number of partitions.

1. Enter the container: `docker exec -it $(docker ps | grep '\-sc' | awk '{print $1}')`  
`bash`

2. Execute the following command to see the number of existing partitions and message stacking

```
/usr/local/kafka/bin/kafka-consumer-groups.sh --bootstrap-server  
${ENV_KAFKA_ENDPOINTS:=127.0.0.1:9092} --describe --group md-workflow-  
consumer
```

3. If necessary, the following commands can be executed separately to adjust the number of partitions (e.g. adjust to 10 partitions)

```
/usr/local/kafka/bin/kafka-topics.sh --alter --zookeeper localhost:2181 --  
partitions 10 --topic Workflow
```



```
/usr/local/kafka/bin/kafka-topics.sh --alter --zookeeper localhost:2181 --
partitions 10 --topic WorkFlow-Batch
/usr/local/kafka/bin/kafka-topics.sh --alter --zookeeper localhost:2181 --
partitions 10 --topic WorkFlow-Button
/usr/local/kafka/bin/kafka-topics.sh --alter --zookeeper localhost:2181 --
partitions 10 --topic WorkFlow-Process
/usr/local/kafka/bin/kafka-topics.sh --alter --zookeeper localhost:2181 --
partitions 10 --topic WorkSheet
/usr/local/kafka/bin/kafka-topics.sh --alter --zookeeper localhost:2181 --
partitions 10 --topic WorkSheet-Batch
```

4. Set the environment variable `ENV_WORKFLOW_CONSUMER_THREADS` in `docker-compose.yaml` corresponding to the microservice application, with a default of 3 (**do not set the value too large, making sure the server can handle it**), as follows:

```
services:
  app:
    environment:
      ENV_WORKFLOW_CONSUMER_THREADS: "10"
```

## How to customize the delays for workflows triggered by event from worksheet?

Set the environment variable `ENV_WORKFLOW_TRIGGER_DELAY_SECONDS` in `docker-compose.yaml` corresponding to the microservice application, with a default of 5 (in seconds), as follows:

```
services:
  app:
    environment:
      ENV_WORKFLOW_TRIGGER_DELAY_SECONDS: "1"
```

## How to customize consumer threads of message queues in worksheets?

Before modifying the consumer threads of message queues, you need to adjust the number of partitions of the related Topic (`ws-editcontrols`, `ws-passivereleation`) in Kafka, with a default of 3. Do not set the number of consumer threads larger than the number of partitions.

1. Enter the container: `docker exec -it $(docker ps | grep '\-sc' | awk '{print $1}')`

`bash`

2. Execute the following command to see the number of partitions and message stacking

```
/usr/local/kafka/bin/kafka-consumer-groups.sh --bootstrap-server  
${ENV_KAFKA_ENDPOINTS:=127.0.0.1:9092} --describe --group worksheet-  
passiverelation  
/usr/local/kafka/bin/kafka-consumer-groups.sh --bootstrap-server  
${ENV_KAFKA_ENDPOINTS:=127.0.0.1:9092} --describe --group worksheet-  
editcontrols
```

3. If necessary, the following commands can be executed separately to adjust the number of partitions (e.g. adjust to 10 partitions)

```
/usr/local/kafka/bin/kafka-topics.sh --alter --zookeeper localhost:2181 --  
partitions 10 --topic ws-editcontrols  
/usr/local/kafka/bin/kafka-topics.sh --alter --zookeeper localhost:2181 --  
partitions 10 --topic ws-passiverelation
```

4. Set the environment variable `ENV_WORKSHEET_CONSUMER_THREADS` in `docker-compose.yaml` corresponding to the microservice application, with a default of 2 (**do not set the value too large, making sure the server can handle it**), as follows:

```
services:  
  app:  
    environment:  
      ENV_WORKSHEET_CONSUMER_THREADS: "10"
```

## How to customize the style of function module?

1. Create custom style files (*like freestyle.css*), locate and get element identifiers (*like html tag, id, class*) through the function of reviewing elements on the web page, and set custom styles (*like hiding function modules, adjusting fonts or background colors*)

```
/* hide the message module */  
#chat {display:none!important }
```

2. In the `docker-compose.yaml` corresponding to the microservice application, mount the custom style file into the container (`/usr/local/MDPrivateDeployment/www/staticfiles/mdcss/freestyle.css`) and restart the service to take effect: ``

```
services:
  app:
    volumes:
      - freestyle.css host
path:/usr/local/MDPrivateDeployment/www/staticfiles/mdcss/freestyle.css
```

Go to <https://github.com/mingdaocom/pd-openweb> for source code to complete more custom development.

## How to add global Javascript scripts?

By adding a global Javascript script, you can introduce some data analysis engines such as Baidu Statistics, Google Statistics, GrowingIO, SENSORS, etc. to monitor the usage status of the system.

1. Create a file of custom extension scripts (*like freestyle.js*) and copy the third-party provided scripts into the file. Generally, the third-party scripts need to refer to basic user information such as user unique identifiers, which developers can obtain through the global Javascript object `md.global.Account` provided by the HAP system.
2. In the `docker-compose.yaml` corresponding to the microservice application, mount the file of the extension script into the container (`/usr/local/MDPrivateDeployment/www/staticfiles/mdjs/freestyle.js`) and restart the service to take effect: ```:

```
services:
  app:
    volumes:
      - freestyle.js host
path:/usr/local/MDPrivateDeployment/www/staticfiles/mdjs/freestyle.js
```

Go to <https://github.com/mingdaocom/pd-openweb> for source code to complete more custom development.

## How to customize the expiration time of login status?

Set the environment variable `ENV_SESSION_TIMEOUT_MINUTES` in `docker-compose.yaml` corresponding to the microservice application, with a default of **10080** (in minutes), as follows:

```
services:
  app:
    environment:
      ENV_SESSION_TIMEOUT_MINUTES: "30"
```

## How to customize whether or not verification code appears every time you log in?

Set the environment variable `ENV_LOGIN_CAPTCHA_LIMIT_COUNT` in `docker-compose.yaml` of the microservice application. If set to 0, the verification code will appear every time, as follows:

```
services:
  app:
    environment:
      ENV_LOGIN_CAPTCHA_LIMIT_COUNT: "0"
```

## How to customize the number of consecutive login failures, start locking, and lock duration

Add the environment variables `ENV_LOGIN_LOCK_LIMIT_COUNT` to the configuration file, with a default value of 5, and `ENV_LOGIN_LOCK_MINUTES`, with a default value of 20 (in minutes), as follows:

```
services:
  app:
    environment:
      ENV_LOGIN_LOCK_LIMIT_COUNT: "4"
      ENV_LOGIN_LOCK_MINUTES: "30"
```

## How to customize the number of consecutive login failures under the same IP and start locking and locking duration

Add the environment variables `ENV_LOGIN_IP_LOCK_LIMIT_COUNT` and `ENV_LOGIN_IP_LOCK_MINUTES` (in minutes) to the configuration file, as follows:

To enable, it is necessary to ensure that the proxy layer has added `X-Real-IP` ([Proxy Reference](#), which is used to determine the same IP address); In addition, special attention should be paid to the fact that if it is an internal network environment, the client IP may be the same, and once enabled, all personnel will be affected.

```
services:
  app:
    environment:
      ENV_LOGIN_IP_LOCK_LIMIT_COUNT: "10"
      ENV_LOGIN_IP_LOCK_MINUTES: "30"
```

## How to customize the black and white list for the format of uploaded files?

Set environment variable `ENV_FILEEXT_BLOCKLIST` (blacklist, default value: `.exe, .vbs, .bat, .cmd, .com, .sh`) and `ENV_FILEEXT_ALLOWLIST` (whitelist, default value: `.exe, .vbs, .bat, .cmd, .com, .sh`) in `docker-compose.yaml` corresponding to [standalone mode] of microservice application or [cluster mode] of file storage service to control the uploaded file formats. When `ENV_FILEEXT_ALLOWLIST` is set, `ENV_FILEEXT_BLOCKLIST` will be invalid automatically, as follows:

```
services:
  app:
    environment:
      ENV_FILEEXT_BLOCKLIST: ".exe, .sh, .html"
      ENV_FILEEXT_ALLOWLIST: ".docx, .txt, .png"
```

## How to customize the expiration time of token for attachment uploads or downloads?

Set the environment variable `ENV_FILE_UPLOAD_TOKEN_EXPIRE_MINUTES` in `docker-compose.yaml` corresponding to the microservice application. The expiration time of the token for file uploading defaults to 120 (in minutes, maximum: 5256000); set the environment variable

`ENV_FILE_DOWNLOAD_TOKEN_EXPIRE_MINUTES` and the expiration time of the token for file downloading defaults to 60 (in minutes, maximum: 5256000), as follows:

```
services:
  app:
    environment:
      ENV_FILE_UPLOAD_TOKEN_EXPIRE_MINUTES: "10"
      ENV_FILE_DOWNLOAD_TOKEN_EXPIRE_MINUTES: "10"
```

## How to configure Amap API Key

The default Amap API Key in the system is shared by all private deployment users and has a daily call usage limit. Exceeding the maximum daily call usage limit will cause related functions to not function properly. Therefore, it is recommended to configure your own API Key (which can be purchased if the usage exceeds the limit)

1. Through the Gaode Open Platform <https://lbs.amap.com> Register a developer account (both personal and corporate, it is recommended to authenticate the developer as an individual)
2. Create an application and set the key. Select **Web end (JS API)** from the service platform. Upon successful creation, a key and security key will be generated
3. In the HAP system [System Configuration] > [Integration] > [Map Services], select the type of Amap and configure the API Key and security key
4. When there is an abnormality when using the system positioning/map function, you can confirm whether it is due to insufficient usage of Amap. If the usage is insufficient, you can enter the Amap console to check the real-time usage. Insufficient usage can be purchased, please refer to: <https://lbs.amap.com/upgrade#price> Generally speaking, you only need to purchase a basic search service traffic package

## How to reference online translation resource files

By default, translation resources reference files carried within the microservice mirror version. Due to the unpredictable iteration cycle of versions, in order to quickly reference the latest translation resource files, the product supports configuring external translation resource addresses (with a relatively high update frequency).

Add the environment variable `ENV_TRANSLATION_SOURCE_HOST` to the yaml configuration file corresponding to the microservice application, as follows:

```
services:
  app:
    environment:
      ENV_TRANSLATION_SOURCE_HOST: "https://file.domain.com/pd"
```

Attention:

1. The deployment environment needs to support access `https://file.domain.com`
2. It is not recommended to use the front-end for secondary development (there may be copywriting alienation and translation mismatch)

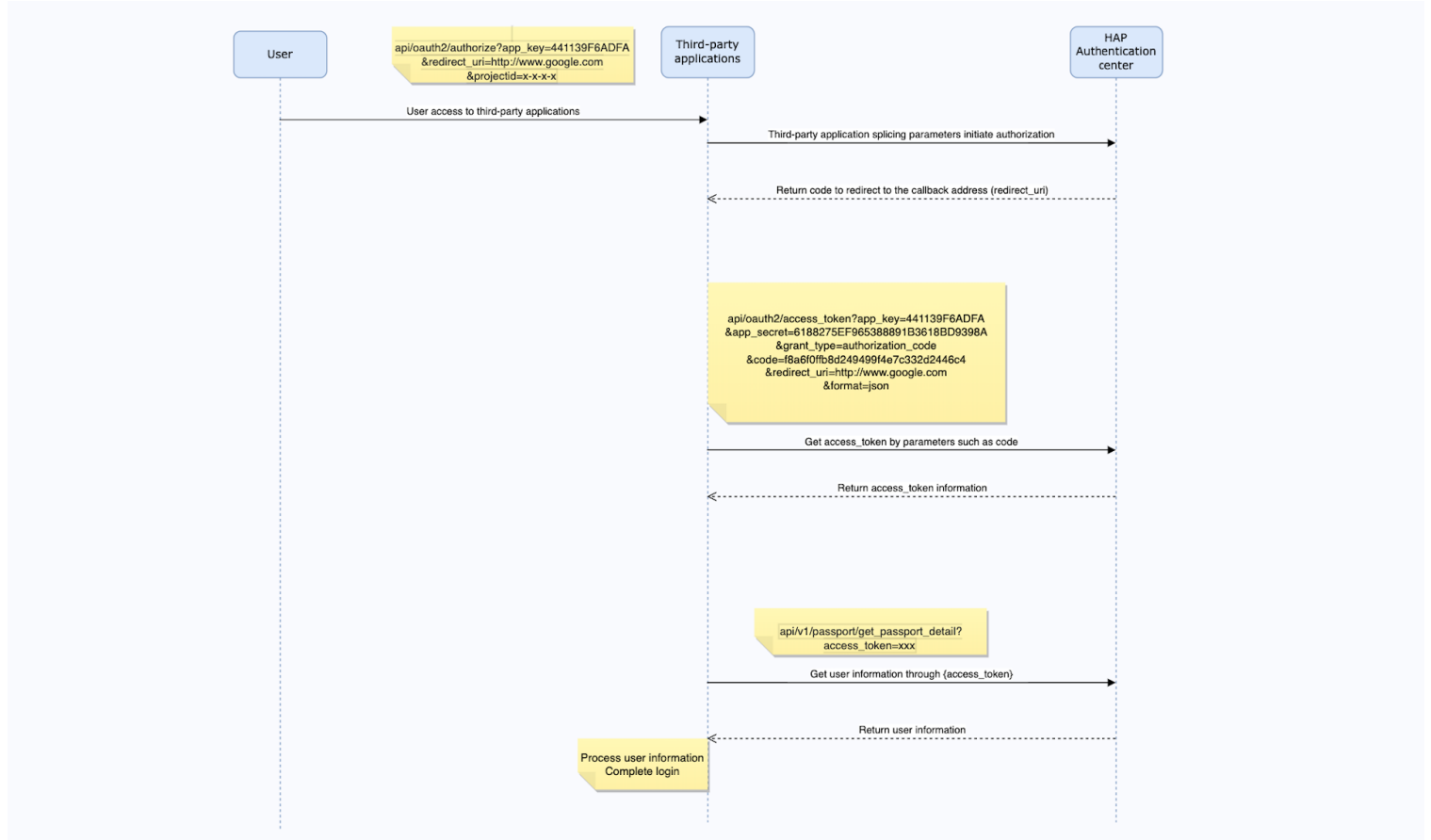
## How to change the Web front-end to refer to CDN resources?

1. Purchase CDN service from cloud market (e.g. Tencent Cloud, Alibaba Cloud). If you already have a CDN service, ignore it.
2. Set the acceleration domain name and configure back to the source site (the access address of HAP system)
3. Set the environment variable `ENV_CDN_URI` in `docker-compose.yaml` corresponding to the microservice application, such as `http://hapcdn.domain.com`

```
services:
  app:
    environment:
      ENV_CDN_URI: "acceleration domain name"
```

# Third-Party Applications

The use case for this feature is to use HAP as an identity authentication center, with third-party applications created as the entry point from HAP to external applications. The authentication process is based on the OAuth 2.0 protocol.



## 1. Create Application

Go to **Org Admin** » **Integration** » **3rd-Party Application** to create an application (you can also click the bottom left of the homepage to view the list of created applications).



Org Admin

APaaS 平台

Home

User

Organization

Apps

Security

Log

Integration

Third-party Platforms

Connected Apps

Misc.

Third-party application management

Application	Summary	App Key	App Secret	Application address	Callback address	Action
D 应用	-	9CB3BAEF7940	*****	http://161.mingdao.net	http://161.mingdao.n...	⋮
C 应用	-	93A0576B1185	*****	http://161.mingdao.net/	http://161.mingdao.net/	⋮
一起写	-	CEEC7A00F6BE	*****	http://161.mingdao.net/	http://161.mingdao.net/	⋮
药品执法系统	-	B075CCB01018	*****	http://161.mingdao.net/	http://161.mingdao.net/	⋮
A 应用	-	7BBCD5A23476	*****	https://www.mingdao...	https://www.mingdao...	⋮

New

## 2. Obtain Request Verification Code

Request Example (GET):

```
{HAP access address}/api/oauth2/authorize?app_key=441139F6ADFA&redirect_uri=http://www.baidu.com&projectid=f0dc6a74-854a-46cb-ab8c-909b7846d34c
```

Request Parameters:

- app\_key: Application App Key
- redirect\_uri: Application callback address
- projectid: Organization ID in HAP

https://.../api/oauth2/authorize?app\_key=441139F6ADFA&redirect\_uri=http://www.baidu.com&projectid=f0dc6a74-854a-46cb-ab8c-909b7846d34c

https://www.baidu.com/?code=2c5b92bf252b4c38bf0d13063a23e176

## 3.Obtain Access Token

Request Parameters:

- app\_key: App Key
- app\_secret: App Secret
- grant\_type: Fixed authorization\_code
- code: Request verification code obtained in the previous step

- redirect\_uri: Application callback URL
- format: json (by default it returns XML format)

Request Example (GET):

```
{HAP access address}/api/oauth2/access_token?
```

```
app_key=441139F6ADFA&app_secret=6188275EF965388891B3618BD9398A&grant_type=authoriza  
tion_code&code=f8a6f0ffb8d249499f4e7c332d2446c4&redirect_uri=http://www.baidu.com&&  
format=json
```

Request Result:

```
{  
  "access_token": "9h9kapa*****",  
  "expires_in": "604800",  
  "refresh_token": "00b100b5*****",  
  "success": true  
}
```

## 4. Obtain Info about Current Account

Request Example (GET):

```
{HAP access address}/api/v1/passport/get_passport_detail?
```

```
access_token=9h9kapa*****
```

Request Result (Key Fields):

```
{  
  "data": {  
    "user_id": "string", // User ID, different in different organizations  
for the same account  
    "full_name": "string", // Name  
    "account_id": "string", // Account ID, same in different organizations  
for the same account  
    "project_id": "string", // Organization ID  
    "email": "string", // Email  
    "mobile_phone": "string", // Phone number  
    "department": "string", // Department  
    "job": "string", // Position  
    "job_number": "string", // Job number  
    "work_site": "string", // Work location  
    "contact_phone": "string", // Work phone  
  }  
}
```

```
},  
  "success": true,  
  "error_code": 1  
}
```

# Forgot Login Password?

In a private deployment environment, if you forget the login password for HAP and have not integrated email or SMS services, you can reset the password by checking the verification code in the logs.

1. Click [Forgot password?] in the login page, enter your account and click [Get Code], if you have been prompted that the verification code failed to be sent, you can ignore this step first and check in the next step.
2. Log in to the server and copy the following command to the machine where the HAP microservice is running to view the verification code.

```
docker exec -it $(docker ps | grep community | awk '{print $1}') bash -c 'source /entrypoint.sh && log log | grep "verification code"'
```

3. After checking the verification code, copy the code to the page and continue to reset your password.

---

If you can't get the verification code in the microservice container by the above command, it is usually due to **incorrect account entered**.

If you forget your account, you can go to MySQL and check the account table:

1. Access to the microservice container

```
docker exec -it $(docker ps | grep community | awk '{print $1}') bash
```

2. Log in to MySQL

```
mysql -h sc -uroot -p123456
```

3. Check the account table

```
SELECT MobilePhone,Email,CreateTime,Status FROM MDProject.Account;
```

# Dependent Cloud Services

Feature	Require Internet Access	Need to be Accessed by Internet	Support for Forward Proxy	Destination Address
Integration-WeCom	√	√	√	<a href="https://qyapi.weixin.qq.com">https://qyapi.weixin.qq.com</a>
Integration-DingTalk	√	√	√	<a href="https://oapi.dingtalk.com">https://oapi.dingtalk.com</a>
Integration-Feishu	√	√	√	<a href="https://open.feishu.cn">https://open.feishu.cn</a>
Integration-Welink	√	√	√	<a href="https://open.welink.huaweicloud.com">https://open.welink.huaweicloud.com</a>
Integration-WeChat	√	√	×	<a href="https://api.weixin.qq.com">https://api.weixin.qq.com</a>

Feature	Require Internet Access	Need to be Accessed by Internet	Support for Forward Proxy	Destination Address
Official Account				
Integration-Email	√	×	×	<a href="https://qyapi.weixin.qq.com">https://qyapi.weixin.qq.com</a>
Integration-SMS	√	×	×	Tencent Cloud: <a href="https://yun.tim.qq.com">https://yun.tim.qq.com</a> Alibaba Cloud: <a href="https://dysmsapi.aliyuncs.com">https://dysmsapi.aliyuncs.com</a>
Workflow-Print Record Node (PDF)	√	√	×	<a href="https://dhs.open.wps.cn">https://dhs.open.wps.cn</a> <a href="https://openplatform-ks3.ks3-cn-beijing.ksyun.com">https://openplatform-ks3.ks3-cn-beijing.ksyun.com</a>
Worksheet-OCR Control	√	×	√	<a href="https://ocr.tencentcloudapi.com">https://ocr.tencentcloudapi.com</a>
Worksheet-Location	√	×	x	<a href="https://*.amap.com">https://*.amap.com</a>

Feature	Require Internet Access	Need to be Accessed by Internet	Support for Forward Proxy	Destination Address
field, Map view				
Integration Center-API Library	√	×	×	<a href="https://www.nocoly.com">https://www.nocoly.com</a>
APP Library	√	×	×	<a href="https://www.nocoly.com">https://www.nocoly.com</a>

# SSO

## How to enable single sign-on

HAP currently supports enabling single sign-on based on the following protocols:

### OAuth 2.0

1. Create configuration file

### CAS

1. Create configuration file

### OIDC

1. Create configuration file

### Ticket

1. Create configuration file

### SAML2

1. Create configuration file

## How to configure the authentication source



- **Microsoft ADFS Identity Authentication Source**
- **Google-SAML Authentication Source**
- **Google-OIDC Identity Authentication Source**

# OAuth 2.0

## 1. Create configuration file

The default data directory is `/data/hap/script/volume/sso/sso.json`, with the following contents:

**Note:** If 404 still appears after mounting, you can copy the content to [json.cn](https://json.cn) to verify whether the json format is legal

```
{
  "mode": "common-oauth2",
  "name": "oauth2",
  "oauth2": {
    "oauth2Url": "",
    "clientId": "",
    "clientSecret": "",
    "idParamName": "client_id",
    "secretParamName": "client_secret",
    "accessTokenParamName": "access_token",
    "redirectUri": "",
    "authorizePath": "/authorize",
    "tokenPath": "/access_token",
    "userInfoUrl": "",
    "params": {
      "UserId": "uid",
      "Name": "name",
      "Email": "email",
      "Mobile": "mobilePhone",
      "Positions": "positions",
      "Departments": "departments"
    },
    "autoRegister": true,
    "projectId": ""
  }
}
```

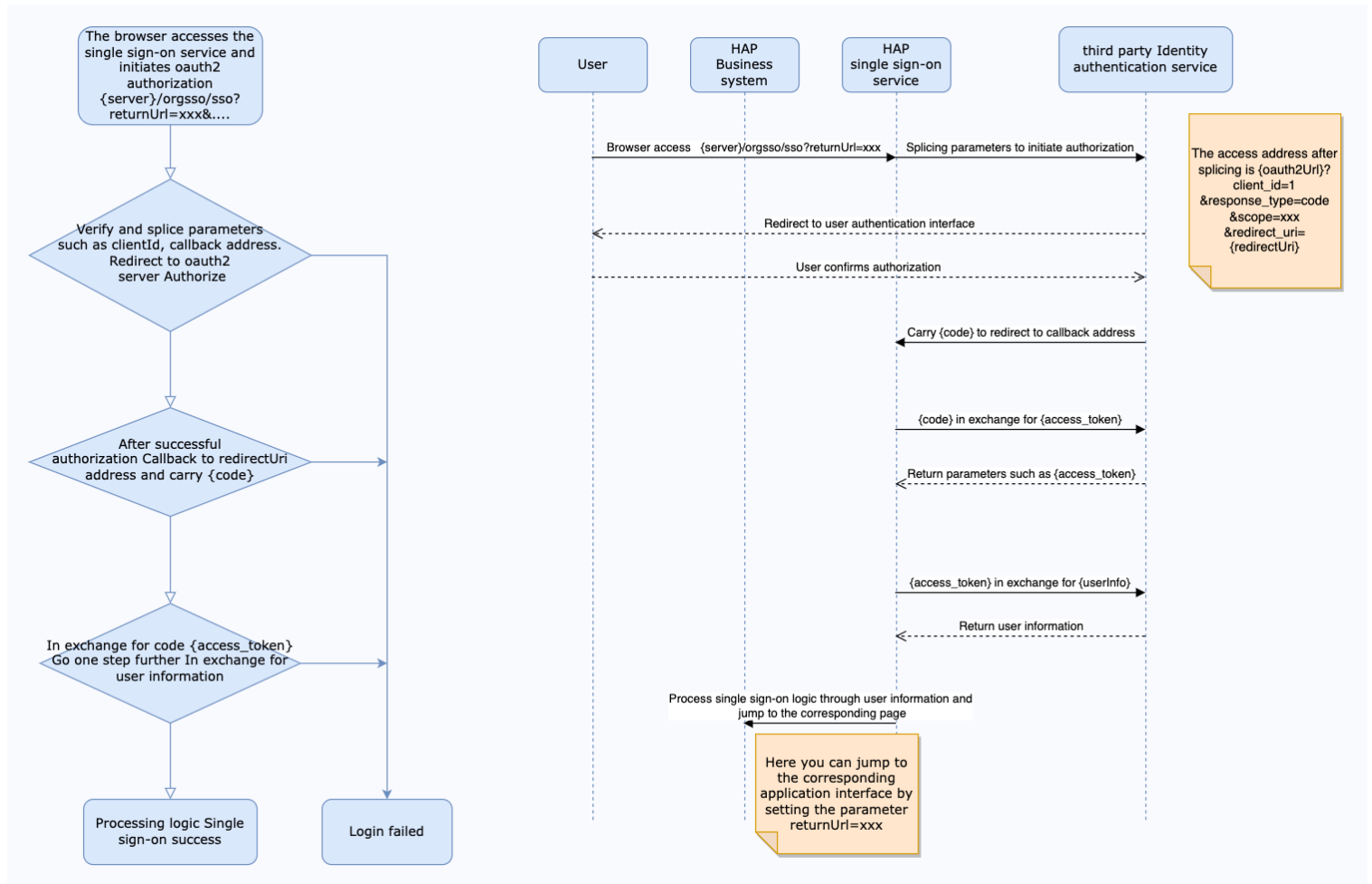
Some parameters and explanations

Parameters	Type	Required	Meaning
oauth2.oauth2Url	String	Yes	oauth2 identity service address; such as <a href="http://192.168.10.80/oauth2">http://192.168.10.80/oauth2</a>
oauth2.clientId	String	Yes	Client Id distributed to the application
oauth2.clientSecret	String	Yes	The client secret distributed to the application
oauth2.idParamName	String	Yes	The parameter name mapped by client_id, depending on the authorizePath and tokenPath interface request parameters; generally set to <code>client_id</code>
oauth2.secretParamName	String	Yes	The parameter name mapped by client_secret, depending on the authorizePath and tokenPath interface request parameters; generally set to <code>client_secret</code>
oauth2.accessTokenParamName	String	No	The parameter name of access_token mapping, depending on the parameters returned by the tokenPath interface; the default is <code>accessToken</code>
oauth2.redirectUri	String	Yes	Callback address; usually set to <code>{server}/orgsso/oauth2</code> ; such as <a href="http://192.168.10.20:8880/orgsso/oauth2">http://192.168.10.20:8880/orgsso/oauth2</a>
oauth2.authorizePath	String	Yes	authorize access path, spliced into <code>oauth2Url/authorizePath</code> ; generally set to <code>/authorize</code>
oauth2.tokenPath	String	Yes	Get the access token interface path, spliced into <code>oauth2Url/tokenPath</code> ;

Parameters	Type	Required	Meaning
			generally set to <code>/access_token</code> <a href="#">Return parameter requirements</a>
oauth2.userInfoUrl	String	Yes	Obtain user information interface <b>complete address</b> ; such as: <code>http://{host}/getUserInfo</code> , the final request url will automatically carry the token parameter; for example: <code>http://192.168.10.80/getUserInfo?{tokenKeyParamName}={access_token}</code> <a href="#">Return parameter requirements</a>
oauth2.tokenKeyParamName	String	No	Get the parameter name of access_token passed by user information, the default is <code>token</code>
oauth2.authorizationMethod	String	No	Parameter (client_id/client_secret) value transfer method; optional parameters are [header, body]; default <code>header</code>
oauth2.bodyFormat	String	No	Parameter request format content-type is application/json or application/x-www-form-urlencoded; optional parameters are [form,json]; default <code>form</code>
oauth2.params	Object	Yes	Returns the user information field mapping rules, the key is a fixed field and the value is configured according to the actual user information; <a href="#">Parameter configuration method</a>
oauth2.params.UserId	String	Yes	User unique identifier
oauth2.params.Name	String	No	User name, if the user already exists, it will be automatically overwritten

Parameters	Type	Required	Meaning
oauth2.params.Email	String	No	Email; this field must be set when searching or registering through email; <b>Either email or mobile phone number must be set; if a third-party relationship has been bound, Users can be found through relationships, and the email or mobile phone does not need to be set</b>
oauth2.params.Mobile	String	No	Mobile phone number; this field must be set to search or register by mobile phone number;
oauth2.params.Positions	Array	No	Position; automatically updates the user's position, there is no automatic creation
oauth2.params.Departments	Array	No	Department; automatically update the user's department, there is no automatic creation
autoRegister	Boolean	No	Whether to automatically create an account when the account does not exist; the default is <code>true</code>
projectId	String	Yes	HAP organization number; Organization Management (upper right corner) > Organization Information (page) > Organization Number ID; <b>(Multi-organization single sign-on does not require configuring this parameter, See step 3 for how to configure it)</b> ; such as 1x-2x-3x-4x-5x

## Interaction diagram



The interface for obtaining access tokens needs to meet the following requirements: -Support POST method call, as shown below:

POST /token HTTP/1.1

Host: server.example.com

Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

Content-Type: application/x-www-form-urlencoded

grant\_type=authorization\_code&code=Sp1xl0BeZQQYbYS6WxSbIA  
&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb

The return value is in JSON format and must meet the following format:

```
{
  "access_token":
  "wQ7kZ5iJ1lK9iU0mJ2oH4rN0uW8gI1vV4fR1jC7yG8yX4gP2qK6mH1iS7iX4zQ1w",
  "expires_in": 7200 // unit seconds
}
```

The interface for obtaining user information needs to meet the following **requirements**: -Supports GET method calling (refer to userInfoUrl parameter description) The return value is in JSON format and must meet the following format:

```
{
  "data": {
    "uid": "userid",
    "name": "name",
    "email": "email",
    "mobilePhone": "mobile phone number",
    "positions": ["position1", "position2"],
    "departments": ["department1", "department2"]
  }
}
```

As shown in the above return value, the configuration of **params** is:

```
"params": {
  "UserId": "uid",
  "Name": "name",
  "Email": "email",
  "Mobile": "mobilePhone",
  "Positions": "positions",
  "Departments": "departments"
}
```

## 2. Mount configuration file

Modify the `docker-compose.yaml` corresponding to the microservice application, the default path is `/data/hap/script/docker-compose.yaml`, add file mounting in `volumes` and restart the microservice application.

```
- ./volume/sso/sso.json:/usr/local/MDPrivateDeployment/sso/OptionFile/sso.json
```

### Enable relationship search

If you need to bind a third-party association ID, you need to create the file `extend.json`. The default path is `/data/hap/script/volume/sso/extend.json` and the content is as follows

```
{  
  "relation": true  
}
```

Add mounting files

```
- ./volume/sso/extend.json:/usr/local/MDPrivateDeployment/sso/extend.json
```

#### TIP

After the mounting configuration is completed, the microservice application needs to be restarted. After the restart is successful, you can access the `{server}/orgsso/checkssoconfig` interface through GET to check whether the configuration file is successfully mounted.

### 3. Single sign-on

**Single Organization** Browser access: `{server}/orgsso/sso?returnUrl={returnUrl}`

**Multiple Organizations** Browser access: `{server}/orgsso/sso?returnUrl={returnUrl}&appKey={appKey}&sign={sign}&timestamp={timestamp}&projectId={projectId}`

For **multi-organization** `projectId` needs to be passed through parameters, and enterprise authorization authentication parameters are also required;

For enterprise authentication and authorization signature algorithm, please refer to:

<https://www.showdoc.com.cn/mingdao/15539798>

#### NOTE

`{server}` is the HAP system address, for example, it can be replaced with:

`http://192.168.10.20:8880`

`{returnUrl}` is the jump address after successful login, **does not need to be filled in**; for example, if you need to jump to the application page, it can be replaced with:

`http://192.168.10.20:8880/app/cf595091-e3ac-4669 -a320-`

`068e55533c33/64477b37df36209b5f36f1cf/64477b4f61655012a90ed994?from=insite`



If an SSO Error prompt appears during the access process, you can log in to the HAP system through the administrator account, click the avatar in the upper right corner: System Configuration > Log; search for the **service name** as , and check the specific cause of the error.

# CAS

## 1. Create configuration file

The default data directory is `/data/hap/script/volume/sso/sso.json`.

**Note:** If you still encounter a 404 error after mounting, you can copy the content to [json.cn](https://json.cn) to check if the JSON format is valid.

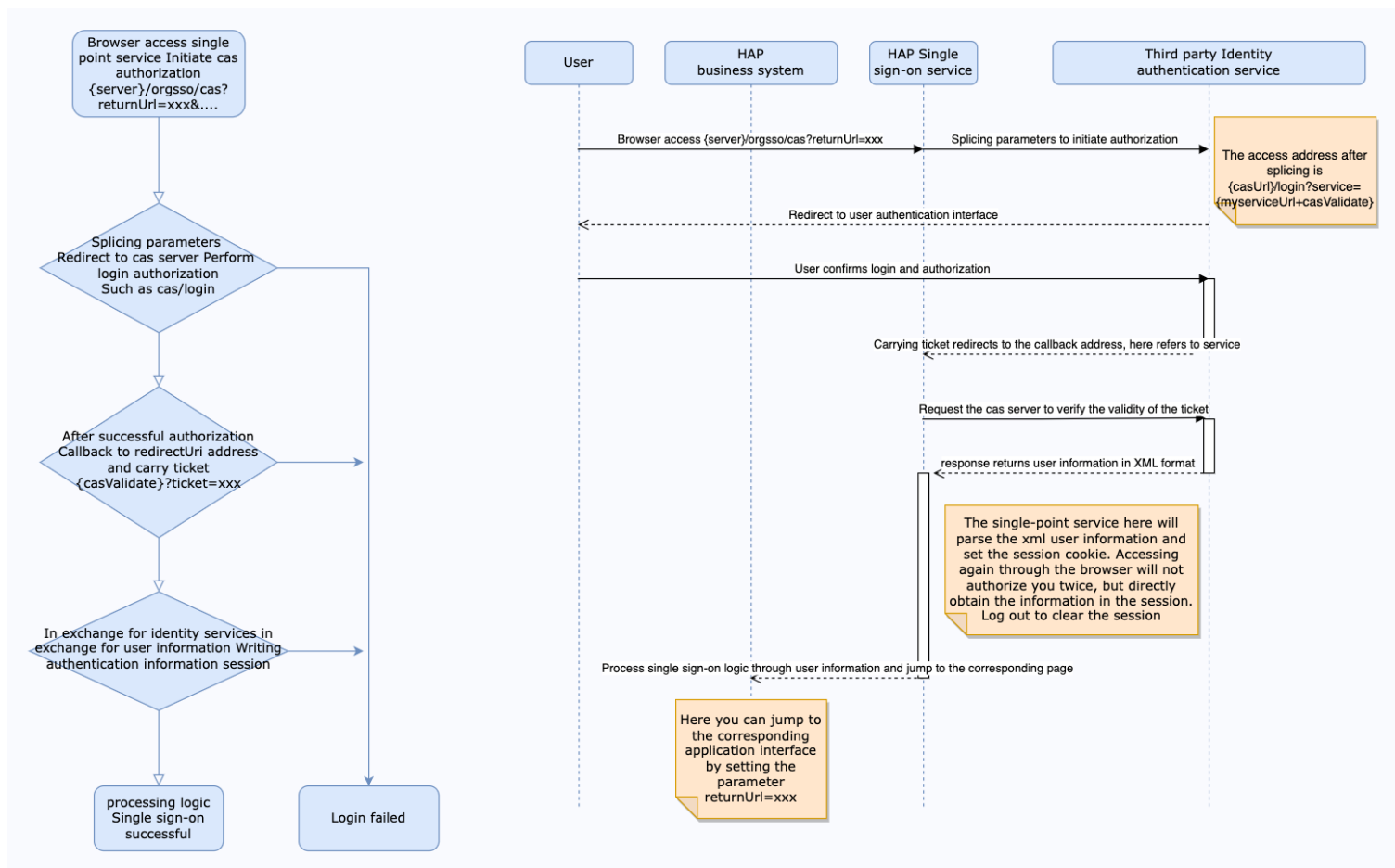
```
{
  "mode": "common-cas",
  "name": "cas",
  "cas": {
    "casUrl": "",
    "myserviceUrl": "",
    "serviceValidate": "/serviceValidate",
    "login": "/login",
    "logout": "/logout",
    "casParams": {
      "UserId": "{{&user}}",
      "Name": "{{&attributes.name}}",
      "Email": "{{&user}}@test.com"
    },
    "autoRegister": true,
    "projectId": ""
  }
}
```

### Parameters and Descriptions

Parameter	Type	Required	Description
cas.casUrl	String	Yes	The URL of the CAS identity server
cas.myserviceUrl	String	Yes	The URL of the CAS single sign-on service; set it as <code>{server}/orgsso</code>
cas.serviceValidate	String	Yes	The path for validating the ticket; access casUrl+serviceValidate for ticket authentication

Parameter	Type	Required	Description
cas.login	String	Yes	The path for logging in; access casUrl+login for authorization and authentication
cas.logout	String	Yes	The path for logging out; access casUrl+logout for CAS to exit the authorization state
cas.casParams	Object	Yes	Field mapping for returning user information, key as fixed fields; value configured based on actual user information; <a href="#">Parameter Configuration</a>
cas.params.UserId	String	Yes	Unique user identifier
cas.params.Name	String	No	Name, will overwrite if user already exists
cas.params.Email	String	No	Email; must be set if searching or registering by email; <b>Either email or mobile must be set; if already bound to a third-party relationship, users can be found through the relationship, email or mobile can be unset</b>
cas.params.Mobile	String	No	Mobile number; must be set if searching or registering by mobile
autoRegister	Boolean	No	Whether to automatically create an account if it does not exist; default to <code>true</code>
projectId	String	Yes	Organization ID in HAP; Org Admin > Organization > Basics > Org ID ( <b>This parameter is not needed if it is a multi-organization single sign-on, See Step 3</b> ), like 1x-2x-3x-4x-5x

Interaction Diagram



The result of user information returned by CAS must be in XML format, as other formats are not supported. Configure the mapping fields in `casParams` based on the returned result, for example:

```
<cas:serviceResponse xmlns:cas=\"http://www.yale.edu/tp/cas\">
  <cas:authenticationSuccess>
    <cas:user>test</cas:user>
    <cas:attributes>
      <cas:name>Jack</cas:name>
    </cas:attributes>
  </cas:authenticationSuccess>
</cas:serviceResponse>
```

The configuration for `casParams` is as follows:

```
"casParams": {
  "UserId": "{{&user}}",
  "Name": "{{&attributes.name}}",
  "Email": "{{&user}}@test.com"
}
```

## 2. Mount configuration file

Create a file and add CAS identity service synchronous logout configuration, `/data/hap/script/volume/wwwapi/appextensions.json` by default, with the following content:

```
{
  "AppSettings": {
    "EnableSsoLogoutRedirect": true
  }
}
```

Modify the `docker-compose.yaml` of the microservice application, and the default path is `/data/hap/script/docker-compose.yaml`. add file mounting in `volumes`, and restart the microservice application.

```
- ./volume/sso/sso.json:/usr/local/MDPrivateDeployment/sso/OptionFile/sso.json
-
./volume/wwwapi/appextensions.json:/usr/local/MDPrivateDeployment/wwwapi/appextensions
```

### Enable relationship search

If you need to bind a third-party association ID, you need to create the file `extend.json`. The default path is `/data/hap/script/volume/sso/extend.json` and the content is as follows

```
{
  "relation": true
}
```

Add mounting files

```
- ./volume/sso/extend.json:/usr/local/MDPrivateDeployment/sso/extend.json
```



**TIP**

After the mounting configuration is completed, the microservice application needs to be restarted. After the restart is successful, you can access the `{server}/orgsso/checkssoconfig` interface through GET to check whether the configuration file is successfully mounted.

### 3. Single sign-on

**Single Organization** Access in your browser: `{server}/orgsso/cas?returnUrl={returnUrl}`

**Multiple Organizations** Access in your browser: `{server}/orgsso/cas?returnUrl={returnUrl}&appKey={appKey}&sign={sign}&timestamp={timestamp}&projectId={projectId}`

For **multi-organization** `projectId` needs to be passed through parameters, and enterprise authorization authentication parameters are also required;

For enterprise authentication and authorization signature algorithm, please refer to:

<https://www.showdoc.com.cn/mingdao/15539798>

#### NOTE

`{server}` is the HAP system address, for example, it can be replaced with:

`http://192.168.10.20:8880`

`{returnUrl}` is the jump address after successful login, **does not need to be filled in**; for example, if you need to jump to the application page, it can be replaced with:

`http://192.168.10.20:8880/app/cf595091-e3ac-4669 -a320-`

`068e55533c33/64477b37df36209b5f36f1cf/64477b4f61655012a90ed994?from=insite`

If an SSO Error prompt appears during the access process, you can log in to the HAP system through the administrator account, click the avatar in the upper right corner: System Configuration > Log; search for the **service name** as `sso`, and check the specific cause of the error.

# OIDC

## 1. Create configuration file

The default data directory is `/data/hap/script/volume/sso/sso.json`, with the following contents:

**Note:** If 404 still appears after mounting, you can copy the content to [json.cn](https://json.cn) to verify whether the json format is legal

```
{
  "mode": "common-oidc",
  "name": "oidc",
  "oidc": {
    "oidcUrl": "",
    "clientId": "",
    "clientSecret": "",
    "redirectUrl": "",
    "responseTypes": "code",
    "scope": "openid email",
    "params": {
      "UserId": "sub",
      "Name": "name",
      "Email": "email",
      "Mobile": "phone_number"
    },
    "autoRegister": true,
    "projectId": ""
  }
}
```

Some parameters and explanations

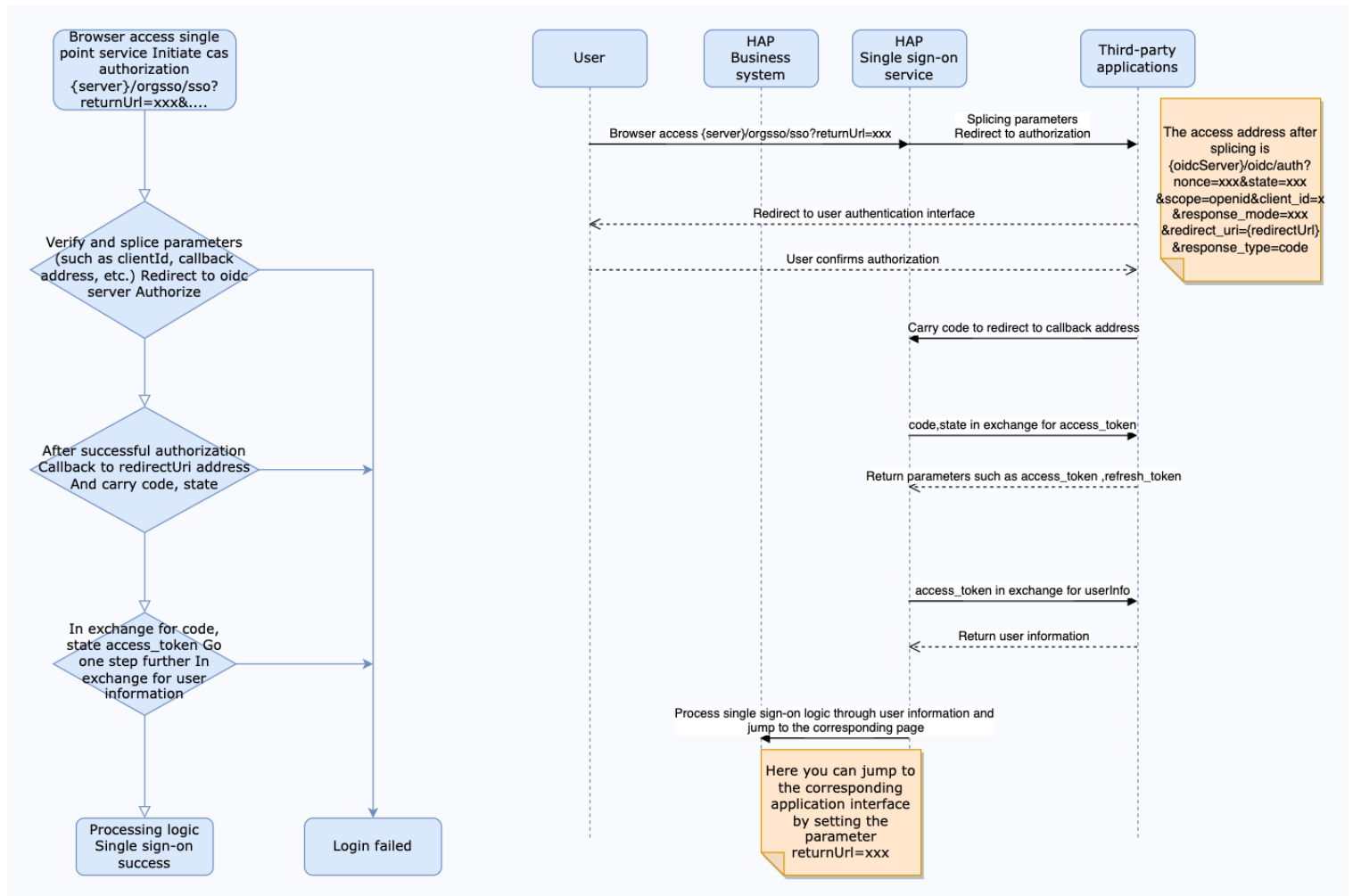
Parameters	Type	Required	Meaning
oidc.oidcUrl	String	Yes	Configure the oidc service discovery address; the return format can be set as follows: <a href="#">Configuration parameter example</a>
oidc.clientId	String	Yes	Client Id distributed to the application

Parameters	Type	Required	Meaning
oidc.clientSecret	String	Yes	The client secret distributed to the application
oidc.redirectUrl	String	Yes	Callback address; set to <code>{server}/orgsso/oidc-redirect</code>
oidc.responseTypes	String	Yes	Support authorization code mode; configured as <code>code</code>
oidc.scope	String	No	Get the user information scope; it can be filled in as any combination of openid, email, profile...; for example, set to <code>openid email</code> ; The default is <code>openid</code>
oidc.params	Object	Yes	Returns user information field mapping rules, key is a fixed field and value is configured according to actual user information; <a href="#">Parameter configuration method</a>
oidc.params.UserId	String	Yes	User unique identifier
oidc.params.Name	String	No	Name, if the user already exists, it will be overwritten
oidc.params.Email	String	No	Email; this field must be set to search or register by email; <b>Either email or mobile phone number must be set; if a third-party relationship has been bound, Users can be found through relationships, and the email or mobile phone does not need to be set</b>
oidc.params.Mobile	String	No	Mobile phone number; this field must be set to search or register by mobile phone number;
oidc.params.Positions	Array	No	Position; automatically updates the user's position, there is no automatic creation



Parameters	Type	Required	Meaning
oidc.params.Departments	Array	No	Department; automatically update the user's department, there is no automatic creation
autoRegister	Boolean	No	Whether to automatically create an account when the account does not exist; the default is <code>true</code>
projectId	String	Yes	HAP organization number; Organization Management (upper right corner) > Organization Information (page) > Organization Number ID; (Multi-organization single sign-on does not require configuring this parameter, See step 3); such as 1x-2x-3x-4x-5x

## Interaction diagram



The interface to obtain access token (access\_token) needs to meet the following requirements: -Support POST method calling (request parameters are passed through URL)

- Need to obtain server authorizePath authorization first
- The return value is in JSON format and must meet the following format:

```
{
  "access_token": "SlAV32hkKG",
  "token_type": "Bearer",
  "refresh_token": "8xL0xBtZp8",
  "expires_in": 3600,
  "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ..."
}
```

The interface for obtaining user information needs to meet the following **requirements**: -Supports GET method calling (refer to userInfoUrl parameter description)

- Need to obtain server authorizePath authorization first
- The return value is in JSON format and must meet the following format:

```
{
  "sub": "248289761001",
  "name": "Jane Doe",
  "email": "janedoe@example.com",
  "phone_number": "123"
}
```

As shown in the above return value, the configuration of params is:

```
"params": {
  "UserId": "sub",
  "Name": "name",
  "Email": "email",
  "Mobile": "phone_number"
}
```

oidcUrl reference response parameter example

```

{
  "issuer": "https://oidc-demo.domain.com/oidc",
  "authorization_endpoint": "https://oidc-demo.domain.com/oidc/auth",
  "token_endpoint": "https://oidc-demo.domain.com/oidc/token",
  "claims_parameter_supported": false,
  "claims_supported": [
    "sub",
    "username"
  ],
  "code_challenge_methods_supported": [
    "plain",
    "S256"
  ],
  "end_session_endpoint": "https://oidc-demo.domain.com/oidc/session/end",
  "grant_types_supported": [
    "authorization_code",
    "password",
    "refresh_token"
  ],
  "response_types_supported": [
    "code"
  ],
  "scopes_supported": [
    "openid",
    "offline_access",
    "username",
    "phone",
    "email",
    "address",
    "profile"
  ],
  "userinfo_endpoint": "https://oidc-demo.domain.com/oidc/me",
  .....
}

```

## 2. Mount configuration file

Modify the `docker-compose.yaml` corresponding to the microservice application, the default path is `/data/hap/script/docker-compose.yaml`, add file mounting in `volumes` and restart the microservice application.

```

- ./volume/sso/sso.json:/usr/local/MDPrivateDeployment/sso/OptionFile/sso.json

```

## Enable relationship search

If you need to bind a third-party association ID, you need to create the file `extend.json`. The default path is `/data/hap/script/volume/sso/extend.json` and the content is as follows

```
{
  "relation": true
}
```

Add mounting files

```
- ./volume/sso/extend.json:/usr/local/MDPrivateDeployment/sso/extend.json
```



**TIP**

After the mounting configuration is completed, the microservice application needs to be restarted. After the restart is successful, you can access the `{server}/orgsso/checkssoconfig` interface through GET to check whether the configuration file is successfully mounted.

## 3. Single sign-on

**Single Organization** Browser access: `{server}/orgsso/sso?returnUrl={returnUrl}`

**Multiple Organizations** Browser access: `{server}/orgsso/sso?returnUrl={returnUrl}&appKey={appKey}&sign={sign}&timestamp={timestamp}&projectId={projectId}`

For **multi-organization** `projectId` needs to be passed through parameters, and enterprise authorization authentication parameters are also required;

For enterprise authentication and authorization signature algorithm, please refer to:

<https://www.showdoc.com.cn/mingdao/15539798>

### NOTE

`{server}` is the HAP system address, for example, it can be replaced with:

`http://192.168.10.20:8880`

`{returnUrl}` is the jump address after successful login, **does not need to be filled in**; for example, if you need to jump to the application page, it can be replaced with:

`http://192.168.10.20:8880/app/cf595091-e3ac-4669 -a320-068e55533c33/64477b37df36209b5f36f1cf/64477b4f61655012a90ed994?from=insite`

If an SSO Error prompt appears during the access process, you can log in to the HAP system through the administrator account, click the avatar in the upper right corner: System Configuration > Log; search for the **service name** as `sso`, and check the specific cause of the error.

# Ticket

## 1. Create configuration file

The default data directory is `/data/hap/script/volume/sso/sso.json`, with the following contents:

**Note:** If 404 still appears after mounting, you can copy the content to [json.cn](https://json.cn) to verify whether the json format is legal

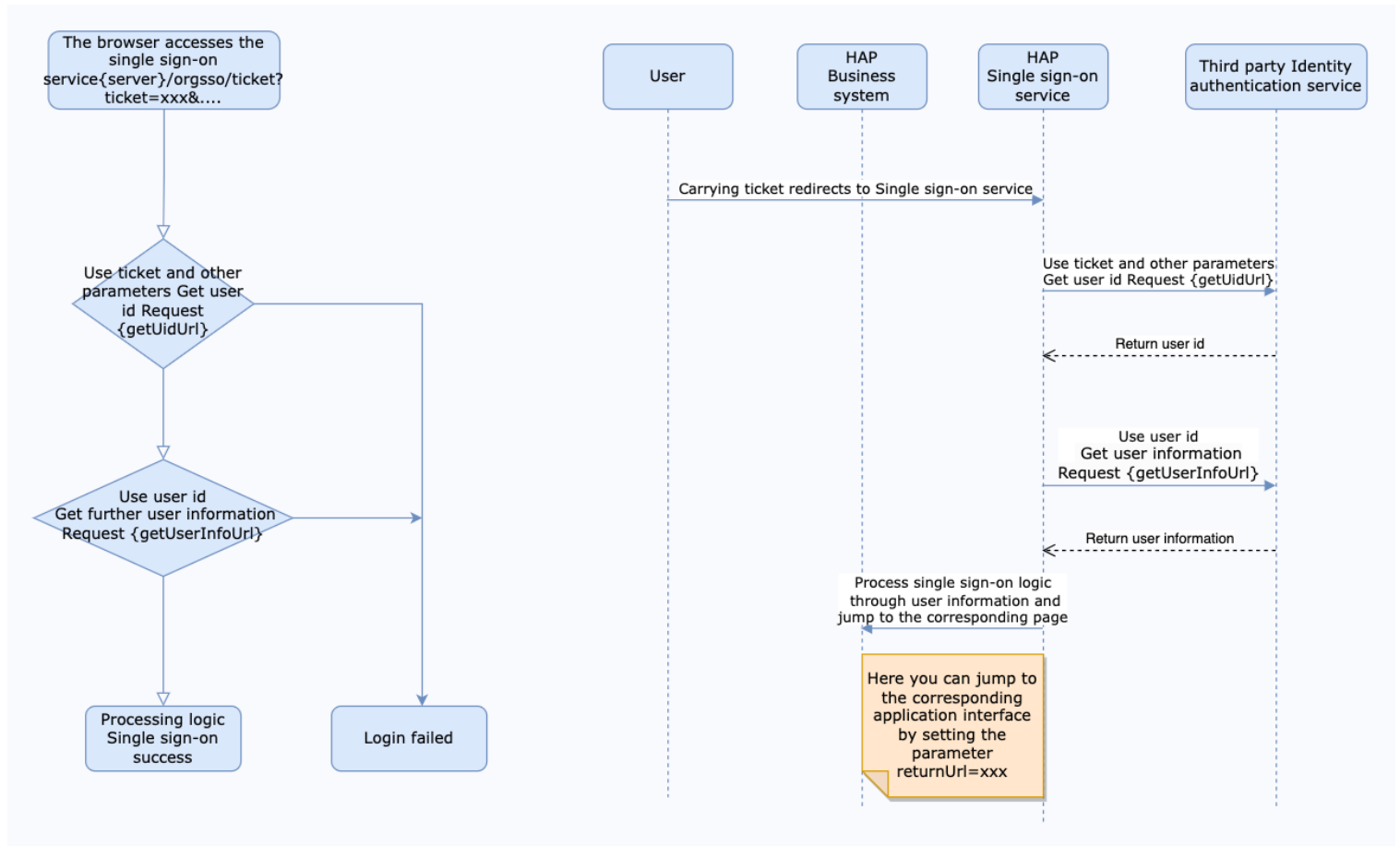
```
{
  "mode": "common-ticket",
  "name": "ticket",
  "ticket": {
    "getUidUrl": "",
    "ticketParams": {
      "ticketParam": "ticket"
    },
    "getUserInfoUrl": "",
    "params": {
      "UserId": "uid",
      "Name": "name",
      "Email": "email",
      "Mobile": "mobilePhone"
    },
    "autoRegister": true,
    "projectId": ""
  }
}
```

Some parameters and explanations

Parameters	Type	Required	Explanation
ticket.getUidUrl	String	Yes	Get the user ID interface address; the parameters configured in <code>ticketParams</code> will be sent to this interface in Post mode to obtain the user ID; <a href="#">Interface requirements</a>

Parameters	Type	Required	Explanation
ticket.ticketParams	Object	Yes	More custom parameters can be added to ticketParams as needed; this parameter is used to obtain the user id
ticket.ticketParams.ticketParam	String	Yes	The ticket parameter name carried in the URL; if configured as "ticketParam": "tk", the configured address needs to be /ticket?tk=xxx&.... Get the value correctly
ticket.getUserInfoUrl	String	Yes	The interface address for obtaining user information; <b>Interface requirement</b>
ticket.params	Object	Yes	Returns the user information field mapping rules, the key is a fixed field and the value is configured according to the actual user information; <b>Parameter configuration method</b>
autoRegister	Boolean	No	Whether to automatically create an account when the account does not exist; the default is true
projectId	String	Yes	HAP organization number; Organization Management (upper right corner) > Organization Information (page) > Organization Number ID; (Multi-organization single sign-on does not require configuring this parameter, see <b>Step 3</b> ); such as 1x-2x-3x-4x-5x

## Interaction diagram



The interface for obtaining user ID needs to meet the following **requirements**:

- Support POST (application/json) method call (the request body is the parameters configured in ticketParams)
- The return value is in JSON format and must meet the following format:

```
{
  "code": "1",
  "message": "success",
  "data": "248289761" // user id
}
```

The interface for obtaining user information needs to meet the following **requirements**:

- Support POST (application/json) method call, the request parameters are as follows

```
{
  "userId": "248289761"
```



```
}
```

- The return value is in JSON format and must meet the following format:

```
{
  "code": "1",
  "message": "success",
  "data": {
    "uid": "248289761001",
    "name": "Jane Doe",
    "email": "janedoe@example.com",
    "mobilePhone": "123"
  }
}
```

As shown in the above return value, the configuration of **params** is:

```
"params": {
  "UserId": "uid",
  "Name": "name",
  "Email": "email",
  "Mobile": "mobilePhone"
}
```

## 2. Mount configuration file

Modify the `docker-compose.yaml` corresponding to the microservice application, the default path is `/data/hap/script/docker-compose.yaml`, add file mounting in `volumes` and restart the microservice application.

```
- ./volume/sso/sso.json:/usr/local/MDPrivateDeployment/sso/OptionFile/sso.json
```

### Enable relationship search

If you need to bind a third-party association ID, you need to create the file `extend.json`. The default path is `/data/hap/script/volume/sso/extend.json` and the content is as follows

```
{
  "relation": true
}
```

```
}
```

Add mounting files

```
- ./volume/sso/extend.json:/usr/local/MDPrivateDeployment/sso/extend.json
```

#### TIP

After the mounting configuration is completed, the microservice application needs to be restarted. After the restart is successful, you can access the `{server}/orgsso/checkssoconfig` interface through GET to check whether the configuration file is successfully mounted.

### 3. Single sign-on

**Single Organization** Browser access: `{server}/orgsso/ticket?{ticketParams.ticketParam}={ticket}&returnUrl={returnUrl}`

**Multiple Organizations** Browser access: `{server}/orgsso/ticket?returnUrl={returnUrl}&appKey={appKey}&sign={sign}&timestamp={timestamp}&projectId={projectId}`

For **multi-organization** `projectId` needs to be passed through parameters, and enterprise authorization authentication parameters are also required;

For enterprise authentication and authorization signature algorithm, please refer to: <https://www.showdoc.com.cn/mingdao/15539798>

#### NOTE

`{server}` is the HAP system address, for example, it can be replaced with: `http://192.168.10.20:8880`

`{returnUrl}` is the jump address after successful login, **does not need to be filled in**; for example, if you need to jump to the application page, it can be replaced with: `http://192.168.10.20:8880/app/cf595091-e3ac-4669 -a320-068e55533c33/64477b37df36209b5f36f1cf/64477b4f61655012a90ed994?from=insite`

If an SSO Error prompt appears during the access process, you can log in to the HAP system through the administrator account, click the avatar in the upper right corner: System Configuration > Log;

search for the **service name** as `sso`, and check the specific cause of the error.

# SAML2

## 1. Create configuration file

The default data directory is `/data/hap/script/volume/sso/sso.json`, with the following contents:

**Note:** If 404 still appears after mounting, you can copy the content to [json.cn](https://json.cn) to verify whether the json format is legal

```
{
  "mode": "common-saml2",
  "name": "saml2",
  "saml2": {
    "entityId": "{server}/orgsso/metadata.xml",
    "assertUrl": "{server}/orgsso/assert",
    "params": {
      "UserId": "PrimarySid",
      "Name": "Name",
      "Email": "EmailAddress"
    },
    "autoRegister": true,
    "projectId": ""
  }
}
```

Some parameters and explanations

Parameters	Type	Required	Meaning
saml2.entityId	String	Yes	The unique identifier of the SP, usually set to the SP metadata address, <code>{server}/orgsso/metadata.xml</code> ; such as <a href="http://192.168.10.20:8880/orgsso/metadata.xml">http://192.168.10.20:8880/orgsso/metadata.xml</a>
saml2.assertUrl	String	Yes	SP assertion address, receives SAMLResponse; fixed setting is <code>{server}/orgsso/assert</code> , needs to be

Parameters	Type	Required	Meaning
			configured in IDP; such as <a href="http://192.168.10.20:8880/orgsso/assert">http://192.168.10.20:8880/orgsso/assert</a>
saml2.params	Object	Yes	Returns user information field mapping rules, key is a fixed field and value is configured according to actual user information; <a href="#">Parameter configuration method</a>
saml2.params.UserId	String	Yes	User unique identifier
saml2.params.Name	String	Yes	Name, if the user already exists, it will be updated and overwritten
saml2.params.Email	String	No	Email; this field must be set when searching or registering through email; <b>Either email or mobile phone number must be set; if a third-party relationship has been bound, Users can be found through relationships, and the email or mobile phone does not need to be set</b>
saml2.params.Mobile	String	No	Mobile phone number; this field must be set to search or register by mobile phone number;
saml2.params.Positions	Array	No	Position; automatically updates the user's position, there is no automatic creation
saml2.params.Departments	Array	No	Department; automatically update the user's department, there is no automatic creation
autoRegister	Boolean	No	Whether to automatically create an account when the account does not exist; the default is <code>true</code>
projectId	String	Yes	HAP organization number; Organization Management (upper right corner) > Organization Information (page) > Organization

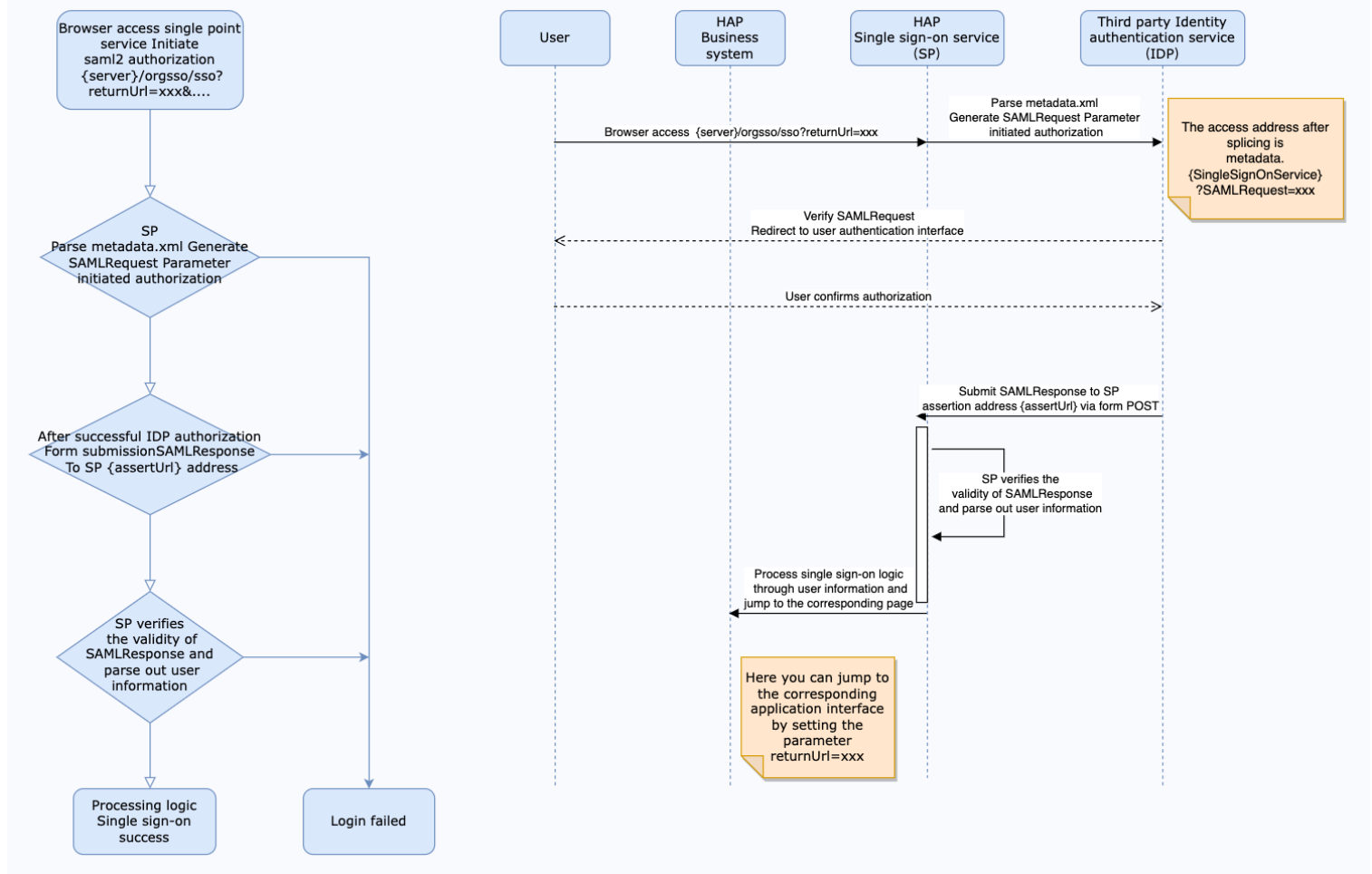
Parameters	Type	Required	Meaning
			Number ID; (Multi-organization single sign-on does not require configuring this parameter, See step 3); such as 1x-2x-3x-4x-5x

Create the IDP metadata file `idp.xml`. The default path is `/data/hap/script/volume/sso/metadata/idp.xml`. The reference content is as follows:

```
<EntityDescriptor entityID="https://saml2.domain.com" ID="pfxea2a0d2f-c296-4b4d-a108-53711984eee"
  xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <IDPSSODescriptor WantAuthnRequestsSigned="false"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <KeyDescriptor use="signing">
      <KeyInfo
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
          <X509Data>
            <X509Certificate>MIIDHjCCAgagAw....</X509Certificate>
          </X509Data>
        </KeyInfo>
      </KeyDescriptor>
      <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location="https://saml2.domain.com/65681be085c07db1c8136eee/logout"/>
      <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location="https://saml2.domain.com/65681be085c07db1c8136eee"/>
      <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://saml2.domain.com/65681be085c07db1c8136eee"/>
    </IDPSSODescriptor>
  </EntityDescriptor>
```

*This configuration is generally automatically generated by the IDP service*

Interaction diagram



SP obtains the SAMLResponse submitted by IDP

```
// The request body from IDP POST is roughly as follows
{
  SAMLResponse:
  'PHNhbwXw0lJlc3BvbmlIElEPSJwZngyNzQxM2ZiYi1mYTdmLTRjZWItYjYkY...'
  ...
}
```

Use **Base64 Decode + Inflate** to return user, timestamp, signature and other information. If the user information part corresponds to the following, then **params** Configuration:

```
...
<AttributeStatement>
  <Attribute Name="Name">
    <AttributeValue>Zhangsan</AttributeValue>
  </Attribute>
  <Attribute Name="PrimarySid">
    <AttributeValue>12345</AttributeValue>
  </Attribute>
</AttributeStatement>
```

```

</Attribute>
<Attribute Name="EmailAddress">
  <AttributeValue>test@domain.com</AttributeValue>
</Attribute>
...
</AttributeStatement>
...

```

According to the above 🙋 return, the params configuration:

```

"params": {
  "UserId": "PrimarySid",
  "Name": "Name",
  "Email": "EmailAddress"
}

```

Note: [Microsoft ADFS Identity Authentication Source](#), the following attribute declarations will be automatically converted to mapped values

```

{
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress": "email",
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname":
"given_name",
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name": "name",
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn": "upn",
  "http://schemas.xmlsoap.org/claims/CommonName": "common_name",
  "http://schemas.xmlsoap.org/claims/Group": "group",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/role": "role",
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname": "surname",

"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier":
"ppid",
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier":
"name_id",

"http://schemas.microsoft.com/ws/2008/06/identity/claims/authenticationmethod":
"authentication_method",
  "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/denonlysid":
"deny_only_group_sid",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/denonlyprimarysid":
"deny_only_primary_sid",

```



```
"http://schemas.microsoft.com/ws/2008/06/identity/claims/denonlyprimarygroupsid":
"deny_only_primary_group_sid",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/groupsid":
"group_sid",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarygroupsid":
"primary_group_sid",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/primarysid":
"primary_sid",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname":
"windows_account_name"
}
```

params can be configured as

```
"params": {
  "UserId": "name_id",
  "Name": "name", // Equivalent configuration
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name"
  "Email": "email" // Equivalent configuration
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress"
}
```

## 2. Mount configuration file

Modify the `docker-compose.yaml` corresponding to the microservice application, the default path is `/data/hap/script/docker-compose.yaml`, add file mounting in `volumes` and restart the microservice application.

```
- ./volume/sso/sso.json:/usr/local/MDPrivateDeployment/sso/OptionFile/sso.json
-
./volume/sso/metadata/idp.xml:/usr/local/MDPrivateDeployment/sso/OptionFile/metadata
```

SP's metadata.xml [Download address: `{server}/orgsso/metadata.xml`] The default is a self-signed certificate. If you need to customize the SP signing certificate, you need to mount the replacement private key and certificate files.

```
- ./volume/sso/ssl/key.pem:/usr/local/MDPrivateDeployment/sso/OptionFile/ssl/key.p
# Replace the private key
-
./volume/sso/ssl/cert.crt:/usr/local/MDPrivateDeployment/sso/OptionFile/ssl/cert.c
```

```
# Replace the certificate
```

Accessible via `{server}/orgsso/metadata.xml`

## Enable relationship search

If you need to bind a third-party association ID, you need to create the file `extend.json`. The default path is `/data/hap/script/volume/sso/extend.json` and the content is as follows

```
{
  "relation": true
}
```

Add mounting files

```
- ./volume/sso/extend.json:/usr/local/MDPrivateDeployment/sso/extend.json
```



**TIP**

After the mounting configuration is completed, the microservice application needs to be restarted. After the restart is successful, you can access the `{server}/orgsso/checkssoconfig` interface through GET to check whether the configuration file is successfully mounted.

## 3. Single sign-on

**Single Organization** Browser access: `{server}/orgsso/sso?returnUrl={returnUrl}`

**Multiple Organizations** Browser access: `{server}/orgsso/sso?returnUrl={returnUrl}&appKey={appKey}&sign={sign}&timestamp={timestamp}&projectId={projectId}`

For **multi-organization** `projectId` needs to be passed through parameters, and enterprise authorization authentication parameters are also required;

For enterprise authentication and authorization signature algorithm, please refer to:

<https://www.showdoc.com.cn/mingdao/15539798>

**i NOTE**

{server} is the HAP system address, for example, it can be replaced with:

`http://192.168.10.20:8880`

{returnUrl} is the jump address after successful login, **does not need to be filled in**; for example, if you need to jump to the application page, it can be replaced with:

`http://192.168.10.20:8880/app/cf595091-e3ac-4669 -a320-`

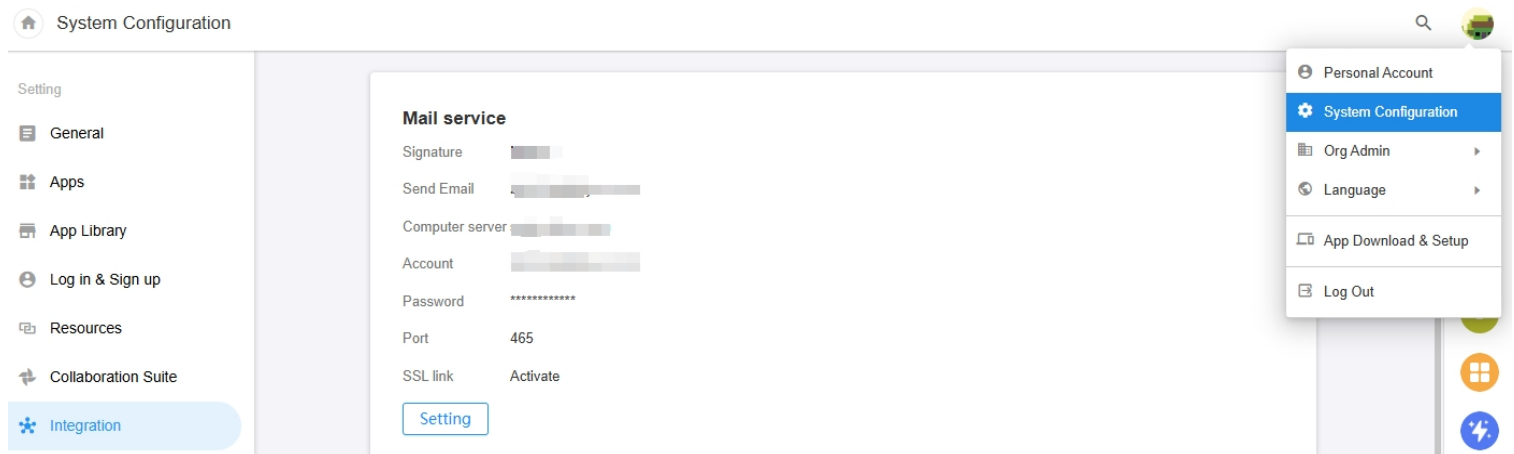
`068e55533c33/64477b37df36209b5f36f1cf/64477b4f61655012a90ed994?from=insite`

If an SSO Error prompt appears during the access process, you can log in to the HAP system through the administrator account, click the avatar in the upper right corner: System Configuration > Log; search for the **service name** as `sso`, and check the specific cause of the error.

# How to enable email service

## SMTP Server Configuration

Click the profile photo and go to **System Configuration > Integration > Email Service**:



## Configuration for Common SMTP Server

**Note: If using port 25, some servers may require unblocking port 25.**

### Tencent Exmail

The administrator needs to enable POP/SMTP/IMAP for configuring email sending. For more details, visit [https://work.weixin.qq.com/help?person\\_id=1&doc\\_id=277&helpType=exmail](https://work.weixin.qq.com/help?person_id=1&doc_id=277&helpType=exmail).

Use the Exmail account and password to configure the email service. The email and account should be the same.

Server	Port (non-SSL)	Port (SSL)
smtp.exmail.qq.com	25 or 587	465

### Alibaba Cloud Exmail

The administrator needs to enable POP/SMTP/IMAP for configuring email sending.

Use the Exmail account and password to configure the email service. The email and account should be the same.

Server	Port (non-SSL)	Port (SSL)
smtp.mxhichina.com	25 or 587	465

### Alibaba Cloud Mail

Enable POP/SMTP/IMAP for configuring email sending.

Use the email account and password to configure the email service. The email and account should be the same.

Server	Port (non-SSL)	Port (SSL)
smtp.aliyun.com	25	465

### Tencent QQ Mail

Enable POP/SMTP/IMAP for configuring email sending.

Use the email account and authorization code (**not email password**). to configure the email service. The email and account should be the same.

Server	Port (non-SSL)	Port (SSL)
smtp.qq.com	25 or 587	465

### 189 Mail

Enable POP/SMTP/IMAP for configuring email sending.

Use the email account and password to configure the email service. The email and account should be the same.

Server	Port (non-SSL)	Port (SSL)
smtp.189.cn	25 or 587	465

## 163 Mail

Enable POP/SMTP/IMAP for configuring email sending.

Use the email account (**only the part before "@"**) and authorization code (**not email password**). to configure the email service. The email and account should be the same.

Server	Port (non-SSL)	Port (SSL)
smtp.163.com	25	465

If the email sending fails, please check the connectivity using the following commands.

```
docker exec -it $(docker ps | grep community | awk '{print $1}') bash -c 'nc -vz server port'
```

## Self Integration

Additionally, it supports configuring a Webhook URL to receive emails and then implement email sending independently.

### 1. Create a configuration file

For example: `/data/hap/script/volume/email/appextensions.json`:

```
{
  "WebhookUrl": "API address to receive messages", // Required, for example:
https://api.domain.com/hooks/NjA0NzdjMDNjMGFjMTE3ZGUwMjRjN2Nl
  "WebhookHeaders": {} // Optional, custom request headers
}
```

The format is as follows: (POST application/json)

```
{
  "ToEmails": ["email1","email2"], // List of recipient emails
  "ReplyToEmails": ["email1","email2"], // List of emails to reply to by
default
  "CcEmails": ["email1","email2"], // List of emails to CC
}
```

```
"BccEmails": ["email1","email2"], // List of emails to BCC
"Signature": "Signature", // Sender's name
"Subject": "Subject",
"Body": "Body", // in HTML format
"Attachments": { "Attachment Name 1": "Download Link 1", "Attachment Name
2": "Download Link 1" }
}
```

## 2. Mount the configuration file

Add the following to `docker-compose.yaml` `volumes` of the microservices application:

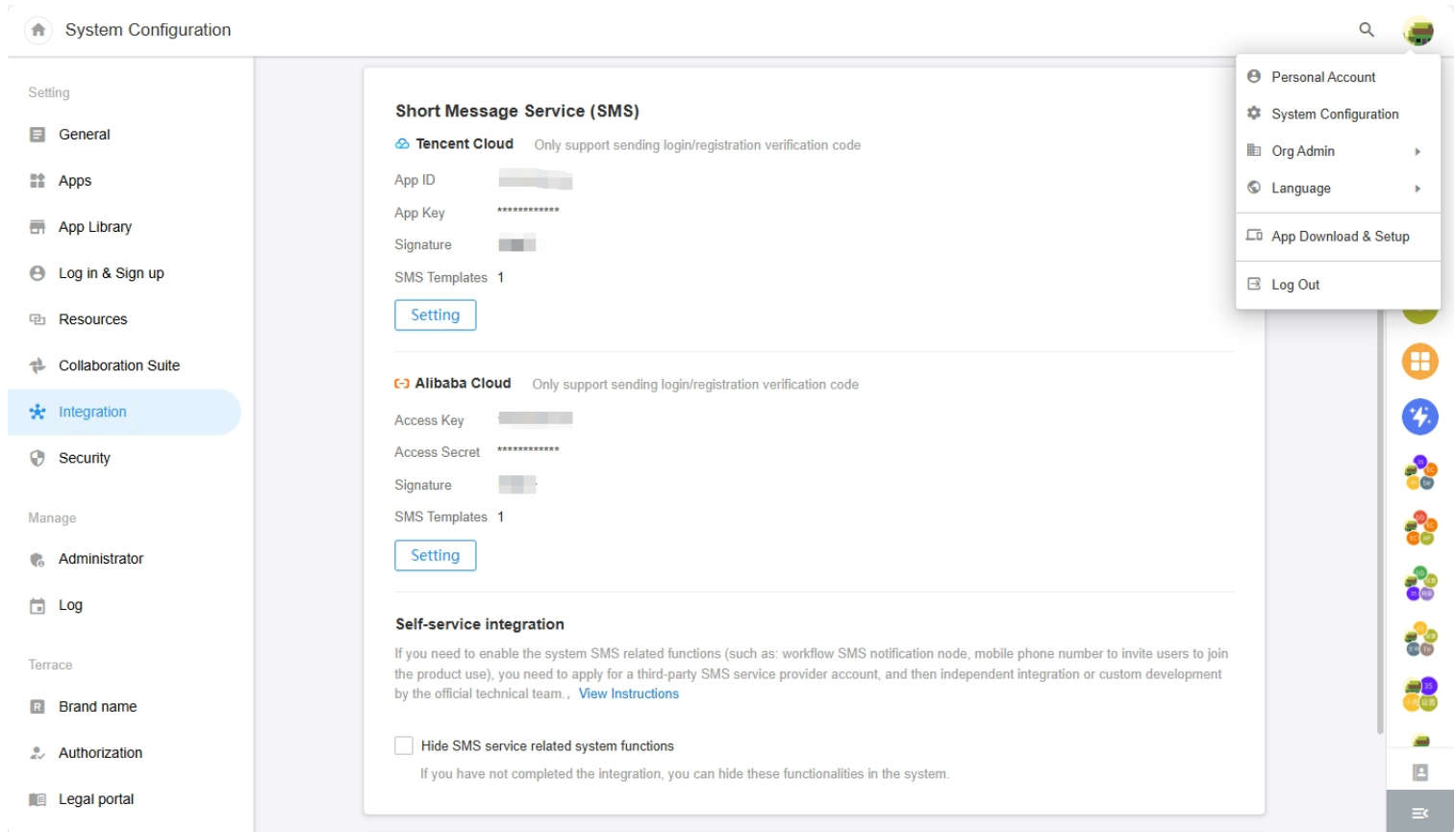
```
-
  ./volume/email/appextensions.json:/usr/local/MDPrivateDeployment/email/appexter
```

## 3. 1.Restart the service

# How to enable SMS service

## Configuration

Click the profile photo and go to **System Configuration > Integration > SMS Service**:



## Default Supported SMS Service Providers

### Tencent Cloud

1. Enable [Tencent Cloud SMS service](#)
2. Create domestic SMS templates, and configure them in **SMS Service Settings** after the templates are approved.



## Tencent Cloud SMS configuration services



App ID

App Key

Signature

SMS Templates

Template Type	Template code	Variable parameter	Edit	Delete
Verification Code		{1}		

[+ Add template](#)

Cancel

Save

## Alibaba Cloud

1. Enable [Alibaba Cloud SMS service](#)
2. Create domestic SMS templates, and configure them in **SMS Service Settings** after the templates are approved.

## Alibaba Cloud SMS configuration services



Access Key

Access Secret

Signature

SMS Templates

Template Type	Template code	Variable parameter	Edit	Delete
Verification Code	<input type="text"/>	{1}		

[+ Add template](#)

Cancel

Save

Currently, we only support SMS template configuration for verification codes, which satisfy needs such as registering an account via phone number, binding a phone number to an account, and resetting a password. You only need to configure one of the mentioned service providers. If configured all simultaneously, a provider will be randomly chosen when sending verification codes

Additionally, there are some functions that involve sending custom SMS messages, which cannot be standardized. If you need to enable such functions (e.g. the send SMS node in workflows, inviting users to register via phone number, and phone number verification in public forms), you will need to apply for an account with a third-party SMS service provider, and then integrate it independently or customize development by the technical team. **Once this mode is activated, the verification code sending channels of Tencent Cloud and Alibaba Cloud in the system configuration will automatically be disabled.**

## Self Integration

1. Create the configuration file

For example: `/data/hap/script/volume/sms/appextensions.json`:

```
{
  "WebhookUrl": "API address to receive messages", // Required, for example:
  https://api.domain.com/hooks/NjA0NzdjMDNjMGFjMTE3ZGUwMjRjN2Nl
  "WebhookHeaders": {} // Optional, custom request header parameters
}
```

The format for SMS messages is as follows: (POST application/json)

```
{
  "Type": 1,
  "Data": {
    "Mobiles": ["Phone number"], // List of phone numbers to receive
    messages
    "Message": "SMS content",
    "Signature": "Signature"
  }
}
```

The format for voice verification codes is as follows: (POST application/json)

```
{
  "Type": 2,
  "Data": {
    "Mobile": "Phone number"
    "Code": "Verification code"
  }
}
```

## 1. Mount the configuration file

Add the following to the `docker-compose.yaml` `volumes` of the microservice application:

```
-
  ./volume/sms/appextensions.json:/usr/local/MDPrivateDeployment/sms/appextension
```

## 2. Restart the service

# How to enable message integration

Assuming that single sign-on has been integrated (such as integration based on **WeCom**, **DingTalk**, **Welink**, **Feishu**, or **Organization Structure Sync API** which is equivalent to enabling single sign-on), the HAP system supports pushing application messages to external systems, and can also be customized by the HAP team to integrate with external systems [custom development will incur additional charges].

To actively receive application messages from HAP:

1. If not based on integration with WeCom, DingTalk, Welink, Feishu, or Organization Structure Sync API, you need to mount the following configuration file in advance:

`/data/hap/script/volume/sso/extend.json`:

```
{
  "relation": true
}
```

Add the following to `docker-compose.yaml` `volumes` corresponding microservice application:

```
- ./volume/sso/extend.json:/usr/local/MDPrivateDeployment/sso/extend.json
```

2. Create the configuration file required to receive application messages:

`/data/hap/script/volume/integrate/appextensions.json`:

```
{
  "WebhookUrl": "API address to receive messages", // Required, e.g.,
  https://api.domain.com/hooks/NjA0NzdjMDNjMGFjMTE3ZGUwMjRjN2Nl
  "FinishTodoWebhookUrl": "API address to receive to-do completion
  messages", // Optional, no need to synchronize pending status, no need to
  configure
  "WebhookHeaders": {} // Custom parameters for request headers, optional
}
```

Message format is as follows: (POST `application/json`)

```
{
  "messageId": "Message id",
  "message": "Message content",
  "messageType": 11,    // 11: Workheet system message, 12: Mentions in
                        // worksheet discussion, 13: Replies in worksheet discussion, 14: To-do message
                        // in workflow (fill in, approval, system message with sending record), 15:
                        // Workflow system message without sending record
  "messageTypeV2": 11,  // 11: Workheet system message, 12: Mentions in
                        // worksheet discussion, 13: Replies in worksheet discussion, 14: To-do message
                        // in workflow (fill in, approval), 15: Workflow system message without sending
                        // record, 16: Workflow system message with sending record
  "appId": "App id",
  "appName": "App name",
  "attachments": ["Attachment link|Attachment name"], // Images can be used
                        // directly, document types currently only serve as a reminder of the number of
                        // attachments (cannot be downloaded directly)
  "redirectUrl": "Redirect url", // Corresponding to the landing page for
                        // message processing in the HAP system
  "targets": ["User id"], // Third-party user id in single sign-on
  "projectId": "Organization Id",
  "createUserId": "Creator", // Third-party user id in single sign-on, system
                        // triggered message is empty
  "processId": "Process id" // Only workflow messages have value
}
```

The format of the completed to-do message is as follows: (POST `application/json`)

```
{
  "messageId": "Message id",
  "projectId": "Organization Id",
  "userId": "Creator", // Third-party user id in single sign-on
  "processId": "Process id"
}
```

### 3. Mount configuration file

Add the following to the `docker-compose.yaml` `volumes` of the microservice application:

```
-
  ./volume/integrate/appextensions.json:/usr/local/MDPrivateDeployment/integrate/
```

4. Restart the HAP microservice application

# How to enable custom file object storage

HAP supports using custom file object storage (**based on the S3 standard, such as Alibaba Cloud OSS, Tencent Cloud COS, Huawei Cloud OBS, Qiniu Cloud, etc.**) to replace the default storage service (based on MinIO). Follow the steps below to enable custom file object storage.

1. Create a configuration file `s3-config.json`, with the template below (take Alibaba Cloud OSS as an example):

```
{
  "mode": 1,
  "accessKeyID": "${Key}",
  "secretAccessKey": "${Secret}",
  "bucketEndPoint": "oss-cn-beijing.aliyuncs.com",
  "bucketName": {
    "mdmedia": "oss-mdtest",
    "mdpic": "oss-mdtest",
    "mdpub": "oss-mdtest",
    "mdoc": "oss-mdtest"
  },
  "region": "oss-cn-beijing"
}
```

Note: **Currently, HAP system uses 4 buckets, `mdmedia`, `mdpic`, `mdpub`, `mdoc`. You can map the actual buckets used in the `bucketName` field of the configuration file.**

2. Mount configuration

*In standalone mode, modify the `docker-compose.yaml` file corresponding to the microservice application; in cluster mode, modify the `docker-compose.yaml` file corresponding to the file storage service.*

```
volumes:
  - ${s3-config.json host path}:/usr/local/file/s3-config.json
```

3. Initialize the pre-made files

You can use the tools provided by various object storage vendors to upload the contents of the [pre-made file package](#) (if migrating file storage for an existing HAP system, use the resources under

`/data/hap/script/volume/data/storage/data/` **directory as pre-made files**) (data in the `mdmedia`, `mdpic`, `mdpub`, `mdoc` 4 folders) to the corresponding bucket in the cloud according to the bucketName mapping. For example: [HAP Server Alibaba Cloud OSS Initialization Instructions](#)

#### 4. Restart HAP microservices



# How to push session status to external system

If the user is active (there is the interface call that triggers the HAP system), in the case of configuring the receiving address, it will be pushed to users every 2 minutes, and the push content contains the account ID in the HAP system. If you need to get the corresponding third-party user ID, go to <https://www.showdoc.com.cn/mingdao/7237672824191664> for more details.

## 1. Create a configuration file

For example, `/data/hap/script/volume/wwwapi/appextensions.json`:

```
{
  "AppSettings": {
    "SessionWebhookUrl": "API address to receive push messages", //
    Required, e.g. https://api.domain.com/hooks/NjA0NzdjMDNjMGFjMTE3ZGUwMjRjN2Nl
    "SessionWebhookHeaders": {} // Custom request headers, optional
  }
}
```

The format is as follows: (POST `application/json`)

```
{
  "accountId": "Account ID in HAP"
}
```

## 1. Mount the configuration file

Modify the corresponding `docker-compose.yaml` for the microservice application, and add the following in `volumes`:

```
-
  ./volume/wwwapi/appextensions.json:/usr/local/MDPrivateDeployment/wwwapi/appext
```

## 2. Restart the HAP microservice application

# How to extend the dependencies of code blocks

## Persistence of Dependency Libraries

The following operations need to be performed while the service is running and only need to be completed once. There is no need to repeat these steps when installing additional extension libraries.

1. Create a mount directory for the dependency library (the actual directory can be customized), for example:

```
mkdir -p /data/hap/script/volume/command/package/python-3.6/
mkdir -p /data/hap/script/volume/command/package/nodejs-10.18.0/
```

2. Get the pre-installed dependency libraries

Microservice version $\geq$ 5.1.0

Microservice version $<$ 5.1.0

```
docker cp $(docker ps | grep command | awk '{print $1}'): /usr/local/lib/python3.6/site-packages/
/data/hap/script/volume/command/package/python-3.6/
docker cp $(docker ps | grep command | awk '{print $1}'): /usr/local/node-
10.18.0/lib/node_modules/ /data/hap/script/volume/command/package/nodejs-
10.18.0/
```

3. Modify the `docker-compose.yml` file to mount the host's dependency library directory into the container

Microservice version $\geq$ 5.1.0

Microservice version $<$ 5.1.0

```
services:
  command:
    volumes:
      - ./volume/command/package/python-3.6/site-
```

```
packages/::usr/local/lib/python3.6/site-packages/  
- ./volume/command/package/nodejs-10.18.0/node_modules/::usr/local/node-  
10.18.0/lib/node_modules/
```

4. Execute the following command in the root directory of the manager and restart the service

```
bash ./service.sh restartall
```

## Online Installation of Extension Library

### NOTE

- **You must complete the "Persistence of Dependency Libraries" steps above first**, mounting the directory of dependency libraries in the container to the host, otherwise the extension libraries will need to be reinstalled after container restart.
- Online installation of extension libraries requires ensuring that the server can access the internet.

## Python

Taking the installation of the `python-dateutil` extension library as an example:

Microservice version $\geq$ 5.1.0

Microservice version $<$ 5.1.0

1. Enter the command container

```
docker exec -it $(docker ps | grep command | awk '{print $1}') bash
```

2. Install the `python-dateutil` extension library in the command container

```
pip3 install --target=/usr/local/lib/python3.6/site-packages/ python-  
dateutil
```

After installation, you can find the newly installed extension library in the default path `/data/hap/script/volume/command/package/python-3.6/site-packages/` on the host.

## JavaScript (Based on Nodejs)

Taking the installation of the `dayjs` extension library as an example:

Microservice version  $\geq 5.1.0$

Microservice version  $< 5.1.0$

1. Enter the command container

```
docker exec -it $(docker ps | grep command | awk '{print $1}') bash
```

2. Install the `dayjs` extension library in the command container

```
/usr/local/node-10.18.0/bin/npm -g install dayjs
```

After installation, you can find the newly installed extension library in the default path

`/data/hap/script/volume/command/package/nodejs-10.18.0/node_modules/` on the host.

## Offline Installation of Extension Library

### NOTE

- **You must first complete the "Persistence of Dependency Libraries" steps above** to persistently mount the directory of dependency libraries in the container to the host. Otherwise, the extension library will need to be reinstalled after the container restarts.
- This process is applicable when the server cannot access the internet or when offline packages are provided for the extension library.

## Python

Taking the installation of the `pycryptodome` extension library as an example:

1. When internet access is available, get the offline file of the extension library

```
pip3 download pycryptodome -d "/package"
```

- An offline file for the pycryptodome dependency library will be generated in the /package directory.

2. Upload the offline file to the server where the HAP service is located

3. Copy the offline file into the container and install it

**Microservice version >= 5.1.0**

**Microservice version < 5.1.0**

- i. Copy the offline file of the pycryptodome dependency library to the /tmp path in the command container

```
docker cp pycryptodome-3.15.0-cp35-abi3-manylinux1_x86_64.whl $(docker ps | grep command | awk '{print $1}'): /tmp
```

- ii. Enter the command container

```
docker exec -it $(docker ps | grep command | awk '{print $1}') bash
```

- iii. Install the offline package of the pycryptodome extension library in the command container

```
pip3 install --target=/usr/local/lib/python3.6/site-packages/ /tmp/pycryptodome-3.15.0-cp35-abi3-manylinux1_x86_64.whl
```

## JavaScript (based on Nodejs)

Taking the installation of the `dayjs` extension library as an example:

1. When internet access is available, get the offline files for the extension library

```
npm install dayjs --save
```

- This will generate a directory `node_modules/dayjs` for the dependency library in the current directory

2. Package the directory for the dayjs extension library
-

```
cd node_modules  
tar czf dayjs.tar.gz dayjs
```

3. Upload the packaged file `dayjs.tar.gz` of the extension library to the server where the HAP service is running.
4. Unzip the offline files to the path where the dependency library is operated for persistence.

```
tar xf dayjs.tar.gz -C /data/hap/script/volume/command/package/nodejs-  
10.18.0/node_modules
```

- If the default data path in HAP has been modified, the unzip path should be based on the actual environment path.

# How to extend the execution environment of code blocks

By default, in HAP the execution environment for code blocks is Node.js (10.18) and Python (3.6). Starting from v5.1, HAP will support the extension of more execution environment.

Currently, in addition to providing the default mirror: `nocoly/hap-command:node1018-python36`

There is also a relatively newer mirror available: `nocoly/hap-command:node2011-python312`

Considering compatibility issues, it supports switching between multiple versions or completely switching to the new version.

For extending dependencies of code blocks, view [how to extend the dependencies of code blocks](#) for more details. Just adjust the corresponding paths: change `python3.6` to `python3.12`, and `node-10.18.0` to `node-20.11`.

**Multi-version Mode**

**New Version**

1. i. Download the mirror ([Offline Package](#))

```
docker pull nocoly/hap-command:node2011-python312
```

2. Modify the configuration file by adding the following configuration

```
commandv2:
  image: nocoly/hap-command:node2011-python312
  environment:
    <<: *app-environment
```

3. Create a configuration file for extending code block services, such as:



`/data/hap/script/volume/workflow/application-www-ext.properties`, with the following content:

```
md.grpc.client.MDCommandService[0].address=static://commandv2:9098
md.grpc.client.MDCommandService[0].nodeVersion=20.11
md.grpc.client.MDCommandService[0].pythonVersion=3.12
```


4. Mount the configuration file by adding the following configuration to the volumes of the app service:

```
volumes:
  - ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflow/application-www-
ext.properties
  - ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowconsumer/application-
www-ext.properties
  - ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowintegration/application-
www-ext.properties
  - ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowplugin/application-www-
ext.properties
```

5. Restart the service



 **JavaScript** 

With the JavaScript language

Node.js v10.18 

Define the input object


key

+ key-value pairs

Code block

Output example: output = {output: "hello world" };

 Code Snippet Library



# Format Conversion (WPS)

## NOTE

Network requirements: Please ensure that the server where the microservice is deployed can communicate with <https://solution.wps.cn>, and that the HAP system access address is accessible from the Internet.

1. Visit <https://solution.wps.cn/> (If you do not have an account, please sign up first).
2. Refer to the document at <https://solution.wps.cn/docs/console/flow.html> to confirm if the relevant preparation has been completed.

Choose **Format Conversion** for both test and formal application types.

3. Obtain the **AppID** and **AppSecret** for the formal application.



The screenshot shows the '应用信息' (Application Information) page in the WPS console. The breadcrumb trail is '应用管理 / 明道云正式应用 (格式转换) / 应用信息'. The page has two tabs: '应用信息' (selected) and '人员配置'. A blue notification bar at the top states: '格式转换详细接口说明和开发文档详见[文档中心](#)'. Under the '基本信息' (Basic Information) section, the following details are listed: Application Name: 明道云正式应用 (格式转换), Creation Time: 2023-08-10 16:54:16, Approval Time: 2023-08-10 16:59:06, AppID: [redacted], AppSecret: [redacted], Application Status: 正常 (Normal), Application Environment: 生产环境 (Production Environment), Billing Type: 按量计费 (Pay-as-you-go), and Application Type: 格式转换应用 (Format Conversion Application). The AppID and AppSecret fields are highlighted with red boxes.

4. Set the environment variables **ENV\_WPS\_CONVERT\_APPID** and **ENV\_WPS\_CONVERT\_APPSECRET** in the yaml configuration file corresponding to the microservice application

```
ENV_WPS_CONVERT_APPID: "AppID value"
ENV_WPS_CONVERT_APPSECRET: "AppSecret value"
```

5. Restart the service

# Format Conversion (Self Integration)

For some special scenarios, such as data security requirements, network policies, etc., HAP also provides a way for self integration.

## ❗ TIPS:

For the header parameters in the interface, if it is in an intranet environment, consider ignoring authentication logic.

Please ensure that the servers deploying microservices have normal connectivity with ENV\_WPS\_HOST.

After completing interface development, you need to add the following 5 environment variables in the configuration file of the HAP microservice:

```
ENV_WPS_HOST: "Host address, default to https://solution.wps.cn"
ENV_WPS_CONVERT_PDF_API_PATH: "Request path for converting documents to pdf,
default to /api/developer/v1/office/convert/to/pdf"
ENV_WPS_CONVERT_TASK_API_PATH: "Request path for querying format conversion
results, default to /api/developer/v1/tasks/:task_id"
ENV_WPS_CONVERT_APPID: "Any string"
ENV_WPS_CONVERT_APPSECRET: "Any string"
```

Finally, restart the HAP microservice.

# LibreOffice

## NOTE

The default document online preview service provided by HAP is based on OnlyOffice (hap-doc). Due to font copyright issues, some fonts may not render correctly or appear as garbled text. However, as there is no ideal free alternative product available, a mirror based on LibreOffice (hap-ldoc) is provided for optimizing Word, PDF, and PPT file previews, while Excel file preview still relies on OnlyOffice. Word, PDF, and PPT file previews are based on LibreOffice.

## Configuration in HAP Standalone Environment

1. Download the mirror ([Offline package](#))

```
docker pull nocoly/hap-ldoc:1.0.0
```

2. Edit the docker-compose.yaml file. Add environment variables and service configurations as shown below.

The default path of the docker-compose.yaml file is /data/hap/script/docker-compose.yaml.

```

version: '3'

services:
  app:
    image: registry.cn-hangzhou.aliyuncs.com/mdpublic/mingdaoyun-community:5.1.0
    environment: &app-environment
      ENV_ADDRESS_MAIN: "https://hap.domain.com"
      ENV_APP_VERSION: "5.1.0"
      ENV_APT_TOKEN: "275zDiFc1MigqgszwvD20jm5mWHFT1SQ0jEWvRK1IEiYf7fV"
      ENV_DOCPREVIRE_EXT_ENDPOINTS: "ldoc:8000"
    ports:
      - 8880:8880
    volumes:
      - ./volume/data:/data/
      - ../data:/data/mingdao/data

  sc:
    image: registry.cn-hangzhou.aliyuncs.com/mdpublic/mingdaoyun-sc:1.0.0
    environment:
      <<: *app-environment
    volumes:
      - ./volume/data:/data/

  command:
    image: registry.cn-hangzhou.aliyuncs.com/mdpublic/mingdaoyun-command:node1018-python36
    environment:
      <<: *app-environment

  doc:
    image: registry.cn-hangzhou.aliyuncs.com/mdpublic/mingdaoyun-doc:1.2.0
    environment:
      ENV_FILE_INNER_URI: "app:8880"

  ldoc:
    image: registry.cn-hangzhou.aliyuncs.com/mdpublic/mingdaoyun-ldoc:1.0.0
    environment:
      ENV_FILE_INNER_URI: "app:8880"
      ENV_DOCPREVIRE_SUBPATH: "/ldocpreview"

```

```

services:
  app:
    image: nocoly/hap-community:6.0.2
    environment:
      ENV_DOCPREVIRE_EXT_ENDPOINTS: "ldoc:8000" # Add environment variable

  # Add service configuration
  ldoc:
    image: nocoly/hap-ldoc:1.0.0
    environment:

```

```
ENV_FILE_INNER_URI: "app:8880"  
ENV_DOCPREVIRE_SUBPATH: "/ldocpreview"
```

3. Restart the HAP microservice.

## Standalone deployment of ldoc services based on Docker

1. Download the mirror ([Offline package](#))

```
docker pull nocoly/hap-ldoc:1.0.0
```

2. Initialize Docker Swarm (skip if already initialized)

```
docker swarm init
```

3. Create a `ldoc.yaml` file in any directory with the following content:

```
version: '3'  
  
services:  
  ldoc:  
    image: nocoly/hap-ldoc:1.0.0  
    ports:  
      - 8881:8000  
    environment:  
      ENV_FILE_INNER_URI: "Internal address of microservice application,  
e.g., 172.27.0.13:8880"  
      ENV_DOCPREVIRE_SUBPATH: "/ldocpreview"
```

4. Start the ldoc document preview service. Execute the following command in the directory where `ldoc.yaml` is located:

```
docker stack deploy -c ldoc.yaml ldoc
```

To remove the service, you can run `docker stack rm ldoc`.

5. Add environment variable to the corresponding `docker-compose.yml` file of the microservice (default path is `/data/hap/script/docker-compose.yml`).

```
ENV_DOCKERPREVIEW_EXT_ENDPOINTS: "Internal address of ldoc service, e.g.,  
172.27.0.13:8881"
```

6. Restart the HAP microservice.

## Deployment of ldoc services based on Kubernetes

1. The cluster environment for HAP microservices is based on Kubernetes by default. To integrate ldoc service, you can deploy it using the following configuration file:

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: ldoc  
  namespace: default  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      app: ldoc  
  template:  
    metadata:  
      labels:  
        app: ldoc  
    spec:  
      containers:  
        - name: ldoc  
          image: nocoly/hap-ldoc:1.0.0  
          env:  
            - name: ENV_FILE_INNER_URI  
              value: "www:8880"  
            - name: ENV_DOCKERPREVIEW_SUBPATH  
              value: "/ldocpreview"  
          resources:  
            limits:  
              cpu: "4"  
              memory: 4096Mi  
            requests:
```

```

      cpu: "0.01"
      memory: 64Mi
    readinessProbe:
      tcpSocket:
        port: 8000
      initialDelaySeconds: 10
      periodSeconds: 10
    livenessProbe:
      tcpSocket:
        port: 8000
      initialDelaySeconds: 180
      periodSeconds: 10

---

apiVersion: v1
kind: Service
metadata:
  name: ldoc
  namespace: default
spec:
  selector:
    app: ldoc
  ports:
    - name: internal
      port: 8001
      targetPort: 8000

```

- In the above configuration, the value of the ENV\_FILE\_INNER\_URI variable is www:8880, which accesses the www service through internal container communication. If the deployment environment of ldoc service is not in the same namespace or environment as the HAP microservice, this variable value needs to be changed to the internal network address of the HAP system.
- Other services to access ldoc, defined here in service it is also through the container internal communication. Similarly, if they are not in the same namespace or environment, change it to the nodePort method and configure the microservice to connect to the external address of ldoc.

2. Add variables in config.yaml.

```
ENV_DOCPREVIRE_EXT_ENDPOINTS: "ldoc:8001"
```

3. Restart the HAP microservice.



## NOTE

Different from LibreOffice, if the WPS WebOffice preview service is enabled, all Office document previews will be based on WPS WebOffice.

Starting from v6.0.0, online document editing capability has been supported, using the same AppID and AppSecret.

HAP also supports integration with WPS private deployment edition to achieve online editing and preview of documents. This can be discussed with the HAP team offline.

**Network requirements: If the cloud version is docked, Please ensure that the HAP system access address is accessible from the Internet.**

1. Visit <https://solution.wps.cn/> (If you do not have an account, please sign up first).
2. Refer to the document at <https://solution.wps.cn/docs/console/flow.html> to confirm if the relevant preparation has been completed.

Choose **Online Edit Preview** for both test and formal application types.

3. Configure the callback gateway for the formal application in the format of `HAP system access address/wps`, for example:

The screenshot shows the '回调配置' (Callback Configuration) tab in the WPS WebOffice console. At the top, there are tabs for '应用信息', '回调配置', '服务配置', '文档配置', '人员配置', and '日志查询'. Below the tabs, a blue notification bar states: '回调接口会在用户触发预览、编辑在线文档时被调用。了解详情' (The callback interface will be called when the user triggers preview, edit online documents. Learn more). Under the '回调网关' (Callback Gateway) section, there is a label '\* 配置网关:' followed by a text input field containing 'https://mdy.domain.com/wps' and a blue '确定' (Confirm) button. Below this, the '文档预览' (Document Preview) section is visible, showing a table with columns: '配置' (Configuration), '是否必配' (Whether required), '接口' (Interface), '说明' (Description), and '调试' (Debug).

4. Obtain the `AppID` and `AppSecret` for the formal application.

应用信息 回调配置 服务配置 文档配置 人员配置 日志查询

### 基本信息

应用名称: 明道云正式应用

创建时间: 2023-07-20 18:34:46

审核通过时间: 2023-07-20 18:36:18

AppID: [REDACTED]

AppSecret: [REDACTED]

应用环境: 生产环境

应用状态: 正常

计费类型: 按量计费

应用类型: 在线编辑预览应用

- Set the environment variables `ENV_WPS_PREVIEW_APPID` and `ENV_WPS_PREVIEW_APPSECRET` in the yaml configuration file corresponding to the microservice application.

```
ENV_WPS_PREVIEW_APPID: "AppID value"
ENV_WPS_PREVIEW_APPSECRET: "AppSecret value"
```

- Restart the service

# Based on OpenAI

Basic AI functions include: worksheet field suggestions, worksheet custom fields, automatic generation of workflow code blocks, and application of multilingual intelligent translation

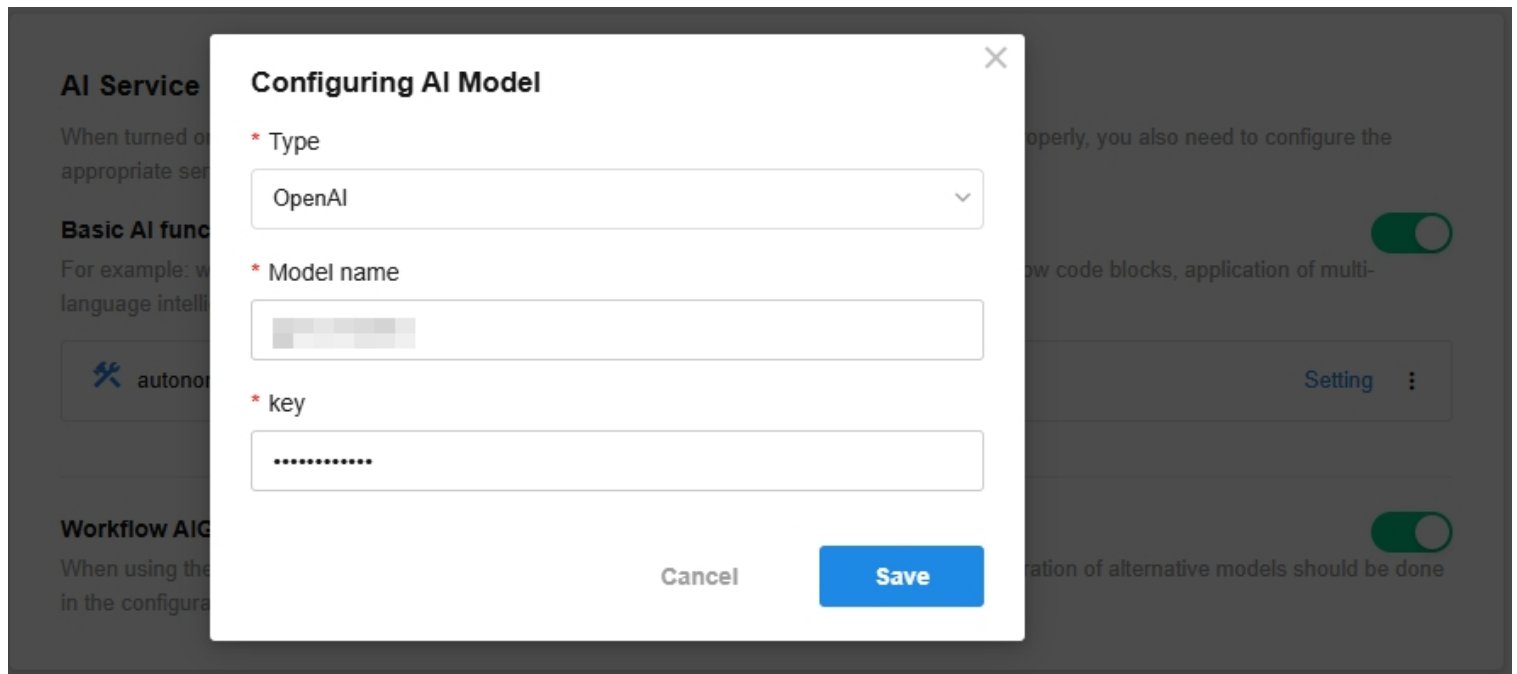
The basic AI functions in HAP products are implemented based on OpenAI by default. If you have the conditions to use OpenAI, you can follow the steps below to complete the configuration.

## Register OpenAI account

Enter the API Keys management in the personal center and generate Secret Keys

## Configure AI model

Go to HAP **System Configuration>Integration>AI services>Basic AI Functions**, select OpenAI as the type and configure the relevant keys, as shown in the following figure:



The screenshot shows a 'Configuring AI Model' dialog box with the following fields:

- Type:** A dropdown menu with 'OpenAI' selected.
- Model name:** A text input field containing a blurred placeholder.
- key:** A text input field containing a series of dots, representing a masked API key.

At the bottom of the dialog are 'Cancel' and 'Save' buttons. The background shows the 'Basic AI Functions' configuration page with a toggle switch turned on and a 'Setting' link.

# Autonomous integration

Basic AI functions include: worksheet field suggestions, worksheet custom fields, automatic generation of workflow code blocks, and application of multilingual intelligent translation

The basic AI functions in HAP are based on OpenAI by default

<https://api.openai.com/v1/chat/completions> implementation, complete interface definition reference: <https://platform.openai.com/docs/api-reference/chat/create>

HAP sends request header parameters to OpenAI:

```
Authorization: Bearer $API_KEY
```

The request body sent by HAP to OpenAI only contains the following parameters:

```
{
  "model": "gpt-4o-mini",           // model
  "messages": [
    {
      "role": "developer",          // role
      "content": "You are a helpful assistant." // content
    }
  ],
  "temperature": 1,                // control the
  randomness of generated text. The higher the value, the more random the
  generated text will be
  "top_p": 1,                      // core probability
  threshold. control the diversity of generated text
  "frequency_penalty": 0,          // punish repeated words
  and reduce repetition
  "presence_penalty": 0,           // punish words that
  have not appeared before and increase the probability of new words appearing
  "stream": false                  // whether to enable
  streaming transmission
}
```

HAP only uses the following parameters to receive OpenAI return values:

When the stream is false

```
{
  "choices": [
    {
      "message": {
        "content": "Full content"
      }
    }
  ]
}
```

When the stream is true

```
{
  "choices": [
    {
      "delta": {
        "content": "Chunk content"
      }
    }
  ]
}
```

So in the autonomous integration mode, developers need to encapsulate interfaces and meet the above input and output formats. It can be understood that in HAP products, the original call to OpenAI interface is now changed to call an independently implemented interface, and the input and output of the interface are completely consistent with OpenAI. For example: [Example code](#) based on Baidu AI autonomous integration

After the interface development is completed and deployed, go to HAP **System**

**Configuration>Integration>AI services>Basic AI Functions**, select the type of autonomous integration, configure the service address (baseUrl), model name, and key, as shown in the following figure:

## Configuring AI Model



\* Type

autonomous integration



Autonomous integration according to interface specifications [View](#)

\* baseURL

[Redacted]

\* Model name

[Redacted]

\* key

[Redacted]

Cancel

Save

# Based on OpenAI

## Register OpenAI account

Enter the API Keys management in the personal center and generate Secret Keys

## Configure AI model

1. Create or modify the configuration file `/data/hap/script/volume/workflow/application-www-ext.properties`, and add the following content:

```
md.aigc.configs=[{\n  'model': 'GPT-3.5-Turbo', \n  'resource': 'https://api.openai.com/v1/chat/completions', \n  'deployment': 'gpt-3.5-turbo', \n  'apiVersion': '', \n  'apiKey': 'Secret Key', \n  'price': '0.015' \n}]
```



Parameter	Remark
model	Interface display title, fully customizable
resource	Fixed value <a href="https://api.openai.com/v1/chat/completions">https://api.openai.com/v1/chat/completions</a> , cannot be modified
deployment	Model name
apiKey	OpenAI Secret Key
price	Non HAP platform version is invalid; HAP Platform version is valid (deducted from organizational balance, can be priced by oneself)

2. Mount the configuration file and add the following configuration to the volumes of the app service:


```
volumes:\n  - ./volume/workflow/application-www-ext.properties:/usr/local/MDPrivateDeployment/workflow/application-www-
```


```
ext.properties
- ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowconsumer/application-
www-ext.properties
- ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowintegration/application-
www-ext.properties
- ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowplugin/application-www-
ext.properties
```

### 3. Restart the service


 **AI Text Generation** 

**Selecting AI Models**

GPT-3.5-Turbo(default) 

 Search

GPT-3.5-Turbo(default)





# Based on the Alibaba Cloud Bailian platform

## Log in to Alibaba Cloud

Select the product **Alibaba Cloud Model Studio**.

×

阿里云

Workbench

Products and Services

My Favorites

Favorite list is empty, you can add products to Favorites from the right.

Popular Products

Elastic Compute Service

Domain Names

Alibaba Cloud DNS

Resource Access Management

Server Load Balancer

CDN

Object Storage Service

Products and Services

Q Enter a keyword

ApsaraDB for MyBase

Database Autonomy Service

Analytics Computing

Data Computing & Analytics

MaxCompute

Hologres

Elasticsearch

Realtime Compute for Apache Flink

GraphCompute

Vector Retrieval Service for Milvus **Public BETA**

Data Lakes

E-MapReduce

Data Lake Formation

Databricks DataInsight

Cloudera Data Platform

Data Application and Visualization

Quick BI

DataV

Quick Audience

Quick Tracking

Data Development & Services

DataWorks

Dataphin

Data Integration

DataHub

Data Resource Platform

Bigdata Expert Service

AI & Machine Learning

Machine Learning Platform for AI

Platform for AI

Intelligent Computing LINGJUN

DashScope

Modelling Platforms and Services

Vector Retrieval Service **NEW**

**Alibaba Cloud Model Studio**

AI Search

Open Search

AIRec

Vision

Visual Intelligence API

image search

Visual Compute Service

Industrial Visual Intelligence

HoloWatcher

OCR

NLP

Natural Language Processing

Address Purification

Machine Translation

Decision Intelligence

Optimization Solver

AI Application

Analytical Insight of Earth

3D space reconstruction

digitized virtual human

Industry intelligence

Autodrive Cloud Develop Platform

genomics

Intelligent customer service

Cloud Contact Center

## Enable the service

Default business space

Home

Model Center - Spiritual Product

Model Square

Model experience

Model tools

App Center

Application Square

My Apps

Application components

Data Centers

Data management

data processing

Data Applications

System administration

System tools

Permissions management

Hello, welcome to Alibaba Cloud Bailian

Selected Tongyi Qianwen - Max Enter your question to start the model experience

Please enter the question you would like to ask

0/6000

### Model calls

Based on business requirements, select a suitable large model and support the API or SDK invocation mode

Call now →

Check out the tutorial

### App creation

Based on the Assistant API, it supports multiple capabilities such as calling prompts, plug-ins, and RAGs to complete the creation of large model applications.

Create it now →

Check out the tutorial

## Get the API-KEY

Default business space

Home

Model Center - Spiritual Product

Model Square

Model experience

Model tools

App Center

Application Square

My Apps

Application components

Data Centers

Data management

data processing

Data Applications

Hello, welcome to Alibaba Cloud Bailian

Selected Tongyi Qianwen - Max Enter your question to start the model experience

Please enter the question you would like to ask

0/6000

### Model calls

Based on business requirements, select a suitable large model and support the API or SDK invocation mode

Call now →

Check out the tutorial

### App creation

Based on the Assistant API, it supports multiple capabilities such as calling prompts, plug-ins, and RAGs to complete the creation of large model applications.

Create it now →

Check out the tutorial

Root account management

API-KEY

Service Agreement

Sign out

If haven't created an API-KEY yet, click **Create my API-KEY** to create a new one.

[Return to the console](#)

[MY API-KEY](#)

[All API-KEYs](#)

MY API-KEY

[Create my API-KEY \( 1/10\)](#)

API KEY	The business space to which it belongs	description	Creation time	operate
GB-9****8132	Default business space(llm-jmzd73ugxSMW0ZJN)		2024-08-05 16:43:51	<a href="#">View the deletion</a>

## Configure AI models

1. Create or modify the configuration file `/data/hap/script/volume/workflow/application-www-ext.properties`, and add the following content:

**apiKey**: Replace it with the API-KEY created above

**price**: Take effect in HAP Application Platform Core only (fees deducted from organization's balance, can be self-priced)

```
md.aigc.configs=[{\n
  'model': 'Qwen-Turbo', \n
  'resource': '', \n
  'deployment': 'qwen-turbo', \n
  'apiVersion': '', \n
  'apiKey': 'API-KEY', \n
  'price': '0.006' \n
},{\n
  'model': 'Qwen-Plus', \n
  'resource': '', \n
  'deployment': 'qwen-plus', \n
  'apiVersion': '', \n
  'apiKey': 'API-KEY', \n
  'price': '0.012' \n
},{\n
  'model': 'Qwen-Max', \n
  'resource': '', \n
  'deployment': 'qwen-max', \n
  'apiVersion': '', \n
  'apiKey': 'API-KEY', \n
  'price': '0.12' \n
}]
```

Parameter	Remark
model	Interface display title, fully customizable
deployment	Model name
apiKey	Replace it with the API-KEY created above
price	Non HAP platform version is invalid; HAP Platform version is valid (deducted from organizational balance, can be priced by oneself)

2. Mount the configuration file and add the following configuration to the volumes of the app service:

```
volumes:
  - ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflow/application-www-
ext.properties
  - ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowconsumer/application-
www-ext.properties
  - ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowintegration/application-
www-ext.properties
  - ./volume/workflow/application-www-
ext.properties:/usr/local/MDPrivateDeployment/workflowplugin/application-www-
ext.properties
```

3. Restart the service



Selecting AI Models

QWen-Turbo(default)



🔍 Search

QWen-Turbo(default)

QWen-Plus

QWen-Max

# Based on Tencent Cloud

The default implementation of worksheet OCR fields in HAP products is based on Tencent Cloud's OCR service. Currently, three types of APIs are mainly used: General OCR, ID card OCR (FRONT), and Invoice OCR. If you have the conditions to use Tencent Cloud services, you can complete the configuration according to the following steps.

1. Open Tencent Cloud OCR service, <https://cloud.tencent.com/product/ocr>
2. From the menu under the username in the upper right corner, go to **Access Management>Access Keys>API Key Management** Create or retrieve existing keys
3. Go to HAP **System Configuration>Integration>OCR**, select Tencent Cloud as the type, and configure the relevant keys, as shown in the following figure:

The screenshot shows a modal dialog titled "Configuring the OCR Service" with a close button (X) in the top right corner. The dialog contains three required fields, each marked with a red asterisk:

- Type:** A dropdown menu with "Tencent Cloud" selected.
- ID:** A text input field containing a blurred, masked string.
- key:** A text input field containing a series of dots, indicating a masked key.

At the bottom of the dialog are two buttons: "Cancel" and "Save".

The background shows the "OCR" configuration page in the HAP system. It includes a toggle switch for enabling OCR, a description of the functionality, and a list of configured services, with "Tencent Cloud" currently selected. The "Setting" button next to the service name is visible.

# Autonomous integration

The default implementation of worksheet OCR fields in HAP products is based on Tencent Cloud's OCR service. Currently, three types of APIs are mainly used: General OCR, ID card OCR (FRONT), and Invoice OCR.

The corresponding interfaces are as follows:

- General OCR:<https://cloud.tencent.com/document/product/866/33526>
- ID card OCR:<https://cloud.tencent.com/document/product/866/33524>
- Invoice OCR:<https://cloud.tencent.com/document/product/866/36210>

The key parameters of the above three interfaces are mainly the `X-TC-Action` in the request header (corresponding to the API type) and the `ImageURL` in the request body (corresponding to the relevant image resource addresses in HAP products).

In the autonomous integration mode, developers need to encapsulate interfaces and meet the input and output formats of the corresponding interfaces. It can be understood that the HAP product originally called the Tencent Cloud interface, but now it calls an independently implemented interface, and the input and output of the interface are completely consistent with Tencent Cloud. For example: [Example code](#) based on Baidu OCR autonomous integration

After the interface development is completed and deployed, go to HAP **System Configuration>Integration>OCR**, select the type of autonomous integration, configure the service address (baseUrl), and theoretically fill in any value for ID and key, as shown in the following figure:

### Configuring the OCR Service

\* Type

autonomous integration

Autonomous integration according to interface specifications [View](#)

\* baseURL

[Redacted]

\* ID

[Redacted]

\* key

.....

Cancel


Save

## AI Service

When turned on, the appropriate services will be displayed.

### Basic AI function

For example: when using the language intelligence function, you also need to configure the appropriate services.


 autonomous integration


### Workflow AI function

When using the workflow AI function, you also need to configure the appropriate services in the configuration page.

## OCR

Once enabled, the text recognition related functionality entry (e.g. worksheet text recognition fields) will be displayed. To ensure that the function works properly, you also need to configure the appropriate services below.

 Tencent Cloud

Setting 



# How to enable data pipeline

Data pipeline is an extension module in the HAP system that users can choose to enable independently. Before enabling, please make sure that **Docker is of version 20.10.16+ and the server has at least 8GB of available physical memory**.

For quick installation in standalone deployment mode, follow this article: (in cluster deployment mode it is more complex and may require assistance from the HAP team).

1. i. Download the mirror ([Offline Package](#))

```
docker pull nocoly/hap-flink:1.17.1.530
```

2. Modify docker-compose.yaml (default path is /data/hap/script/docker-compose.yaml).

- i. Add the flink service

```
flink:
  image: nocoly/hap-flink:1.17.1.530
  entrypoint: ["/bin/bash"]
  command: ["/run.sh"]
  volumes:
    - ./volume/data/:/data/
```

Note: If `docker logs $(docker ps | grep '\-sc' | awk '{print $1}') | grep minio` outputs results, then the `ENV_FLINK_S3_ENDPOINT` environment variable needs to be added in the flink service

```
flink:
  image: nocoly/hap-flink:1.17.1.530
  entrypoint: ["/bin/bash"]
  command: ["/run.sh"]
  environment:
    ENV_FLINK_S3_ENDPOINT: "sc:9010"
  volumes:
    - ./volume/data/:/data/
```

- ii. Add the `ENV_FLINK_URL` environment variable in the app service

```
app:  
  environment:  
    ENV_FLINK_URL: "http://flink:8081"
```

▶ Example

3. Restart the HAP microservices

# How to independently deploy data pipeline service

Data pipeline is an extension module in the HAP system, and users can choose whether to enable it independently. [Enable data pipeline](#).

Quick deployment involves deploying the data pipeline service on the same server as the HAP microservices, which requires high hardware resources. If a single server cannot meet the requirements, follow this article to independently deploy the data pipeline service on a new server. [More details on server configuration](#).

## Install Docker

To install Docker, check the official [installation instructions](#) for different Linux versions or view the [Docker Installation Section](#) in the deployment examples.

## Microservices Adjustment

The data pipeline service requires file storage and Kafka components, so it is necessary to map the access points of these two components in the sc service in standalone mode.

If your HAP Server environment is in cluster mode, no adjustment is needed, and you can directly connect the data pipeline service to file storage and Kafka components for configuring.

For standalone mode, to map the ports of file storage and Kafka components, you need to modify the docker-compose.yaml file by adding environment variables and port mappings as shown below.

```
app:
  environment:
    ENV_FLINK_URL: http://192.168.10.30:58081 # Add, this is the Host resolution
    of the Flink data pipeline service, make sure to modify it to the actual IP
    address

sc:
  ports:
    - 9000:9000
    - 9092:9092
```

## ► docker-compose.yaml Configuration File Example

After modifications, execute `bash service.sh restartall` in the manager directory to restart the microservices.

## Deploy Data Pipeline Service

### 1. Initialize the swarm environment

```
docker swarm init
```

### 2. Create a directory

```
mkdir -p /data/hap/script/volume/data
```

### 3. Create a configuration file

```
cat > /data/hap/script/flink.yaml <<EOF
version: '3'
services:
  flink:
    image: nocoly/hap-flink:1.17.1.530
    entrypoint: ["/bin/bash"]
    command: ["/run.sh"]
    environment:
      ENV_FLINK_S3_ACCESSKEY: "mdstorage"
      ENV_FLINK_S3_SECRETKEY: "eBxExGQJNhGosgv5FQJiVNqH"
      ENV_FLINK_S3_SSL: "false"
      ENV_FLINK_S3_PATH_STYLE_ACCESS: "true"
      ENV_FLINK_S3_ENDPOINT: "sc:9000" # For versions before 5.1.0
      (excluding 5.1.0), fill in "app"; for versions 5.1.0+ (including 5.1.0),
      fill in "sc"
      ENV_FLINK_S3_BUCKET: "mdoc"
      ENV_FLINK_LOG_LEVEL: "INFO"
      ENV_FLINK_JOBMANAGER_MEMORY: "2000m"
      ENV_FLINK_TASKMANAGER_MEMORY: "10000m"
      ENV_FLINK_TASKMANAGER_SLOTS: "50"
      ENV_KAFKA_ENDPOINTS: "sc:9092" # For versions before 5.1.0 (excluding
      5.1.0), fill in "app"; for versions 5.1.0+ (including 5.1.0), fill in "sc";
```

```
if Kafka is using external components, fill in the actual IP of Kafka.
ports:
  - 58081:8081
volumes:
  - ./volume/data:/data/
extra_hosts:
  - "sc:192.168.10.28" # This is the host resolution for the sc service
    (corresponding to the value filled in ENV_KAFKA_ENDPOINTS as "sc:9092"),
    make sure to modify it to the actual IP address
  #- "app:192.168.10.28" # This is the host resolution for the sc
    service (corresponding to the value filled in ENV_KAFKA_ENDPOINTS as
    "sc:9092"), make sure to modify it to the actual IP address
EOF
```

#### 4. Configure the startup script

```
cat > /data/hap/script/startflink.sh <<-EOF
docker stack deploy -c /data/hap/script/flink.yaml flink
EOF
chmod +x /data/hap/script/startflink.sh
```

#### 5. Start the data pipeline service

```
bash /data/hap/script/startflink.sh
```

- It takes about 5 minutes for the data pipeline service container to fully start after startup.
- Stop command: `docker stack rm flink`

## Other Considerations

The data pipeline service needs to create two directories, `checkpoints` and `recovery`, under the bucket of the file storage service to store relevant data.

If **external object storage** is enabled, the file storage will switch to S3 mode, causing issues with the data pipeline, as the data pipeline service currently does not support direct use of the S3 protocol for object storage.

Therefore, if **external object storage** is enabled, a new file storage service needs to be deployed for the data pipeline service.

# Deploy File Storage Service

## 1. Create `file-flink.yaml`

```
version: '3'
services:
  file-flink:
    image: nocoly/hap-file:1.6.0
    volumes:
      - /usr/share/zoneinfo/Etc/GMT-8:/etc/localtime
      - ./volume/data/file-flink/volume:/data/storage
    environment:
      MINIO_ACCESS_KEY: storage
      MINIO_SECRET_KEY: ITwWPDGvSLxxxxxxM46XiSEmEdF4 # customize the authentication key
    command: ["/main", "server", "/data/storage/data"]
```

## 2. Download the mirror for the file service

```
docker pull nocoly/hap-file:1.6.0
```

## 3. Create persistent storage directories for the file-flink service

```
mkdir -p /data/hap/script/volume/data/file-flink/volume
```

## 4. Start the file-flink file storage service

```
docker stack deploy -c file-flink.yaml file-flink
```

## 5. Enter the file-flink container to create the required bucket

```
docker exec -it xxx bash
```

- Replace xxx with the container id of file-flink

## 6. Create buckets

```
# mc command configuration
mc config host add file-flink http://127.0.0.1:9000 storage
ITwWPDGvSLxxxxxxM46XiSEmEdF4 # modify it to your custom authentication key

# Create the required bucket: mdoc
mc mb file-flink/mdoc
```

7. Modify relevant variables in the data pipeline service to specify connection to the file-flink service

```
ENV_FLINK_S3_ACCESSKEY: "storage"
ENV_FLINK_S3_SECRETKEY: "ITwWPDGvSLxxxxxxM46XiSEmEdF4" # modify it to your
custom authentication key
ENV_FLINK_S3_ENDPOINT: "192.168.10.30:9000" # replace with the actual IP of
the file-flink service
ENV_FLINK_S3_BUCKET: "mdoc"
```

8. Restart the flink service

```
docker stack rm flink
sleep 30
bash /data/hap/script/startflink.sh
```

# How to enable aggregate table

**⚠ THE AGGREGATE TABLE (V5.6.0+) RELIES ON THE **MongoDB CDC** CONNECTOR OF **Flink** TO FUNCTION PROPERLY. BEFORE YOU CAN USE THIS FEATURE, ENSURE THAT THE FOLLOWING PREREQUISITES ARE MET:**

1. Deploy the data pipeline service. [How to enable data pipeline](#)
2. MongoDB needs to be upgraded to v4.4+. [Upgrade standalone mode MongoDB to 4.4](#) or [Upgrade cluster mode MongoDB to 4.4](#)
3. MongoDB needs to be configured to a replica set mode (if already a replica set, this step can be skipped). [Convert MongoDB single node to replica set](#)

**Built-in MongoDB  
with authentication  
disabled**

**Built-in MongoDB  
with authentication  
enabled**

**External MongoDB  
with authentication  
disabled**

**External MongoDB  
with authentication  
enabled**

Built-in MongoDB refers to the MongoDB component included in the hap-sc:3.0.0 mirror.

Once the base conditions are met, no other configuration is required to use the aggregate table. 🙌 🙌 🙌



# Microsoft ADFS-SAML Instructions

## Operational Scenarios

Active Directory Federation Services (ADFS) is Microsoft's Windows Server Active Directory Federation Services (ADFS). ADFS is a new technology that can be used to authenticate multiple web application users during a single session. You can use User SSO to integrate ADFS with HAP to enable ADFS account console management resources

## Prerequisites

1. Have a Windows Server server(This is Window Server 2012 R2, Window Server 2016/2019 configuration is slightly different).
2. Perform the following setup work within the server.
  - DNS server: resolves authentication requests to the correct Federation Service.
  - Active Directory Domain Services (AD DS): Provides functions such as creating, querying, and modifying objects such as domain users and domain devices.
  - Active Directory Federation Service (AD FS): Provides the function of configuring SSO relying parties and provides SSO authentication for the configured relying parties.

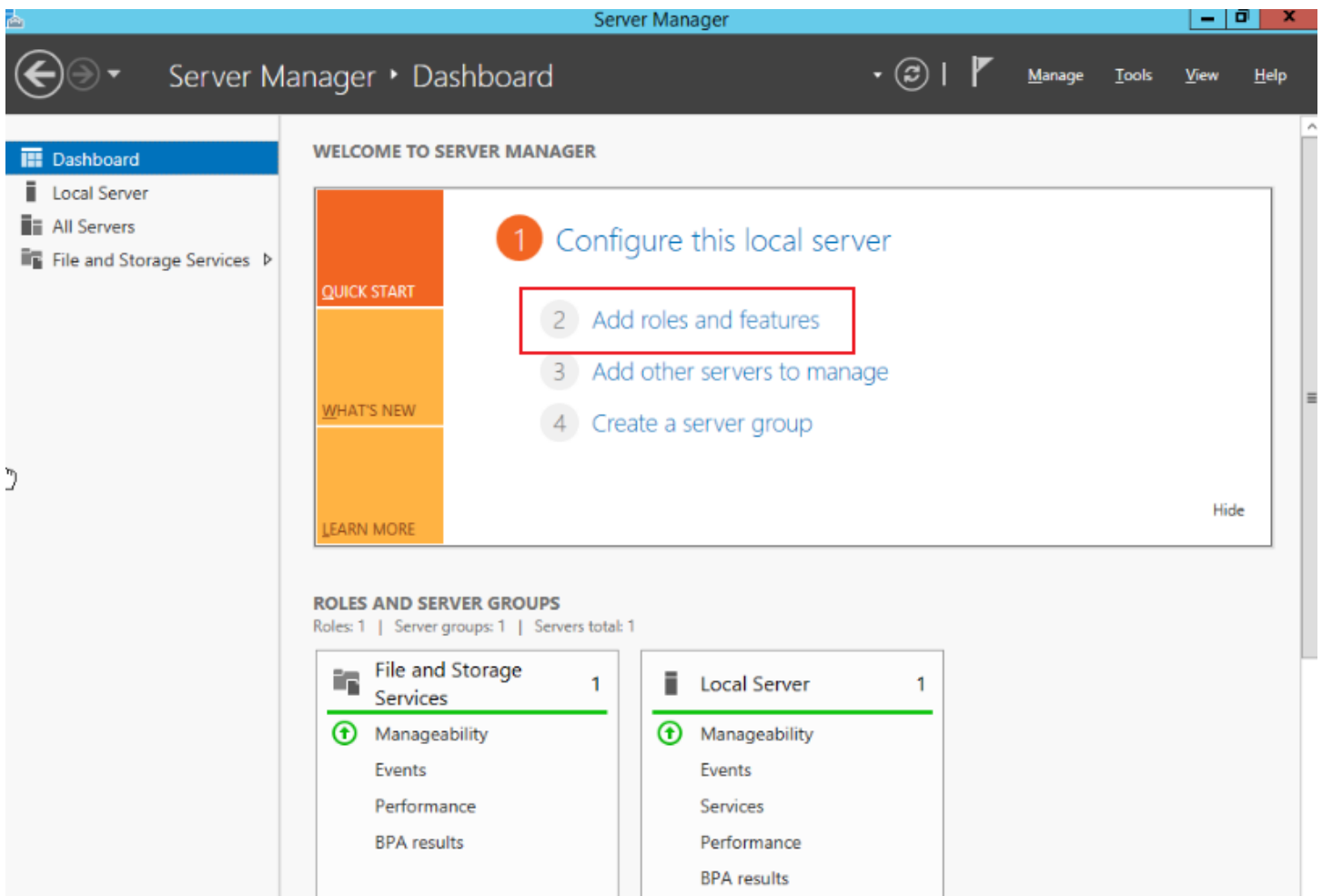
## Steps

### Install and deploy Microsoft AD

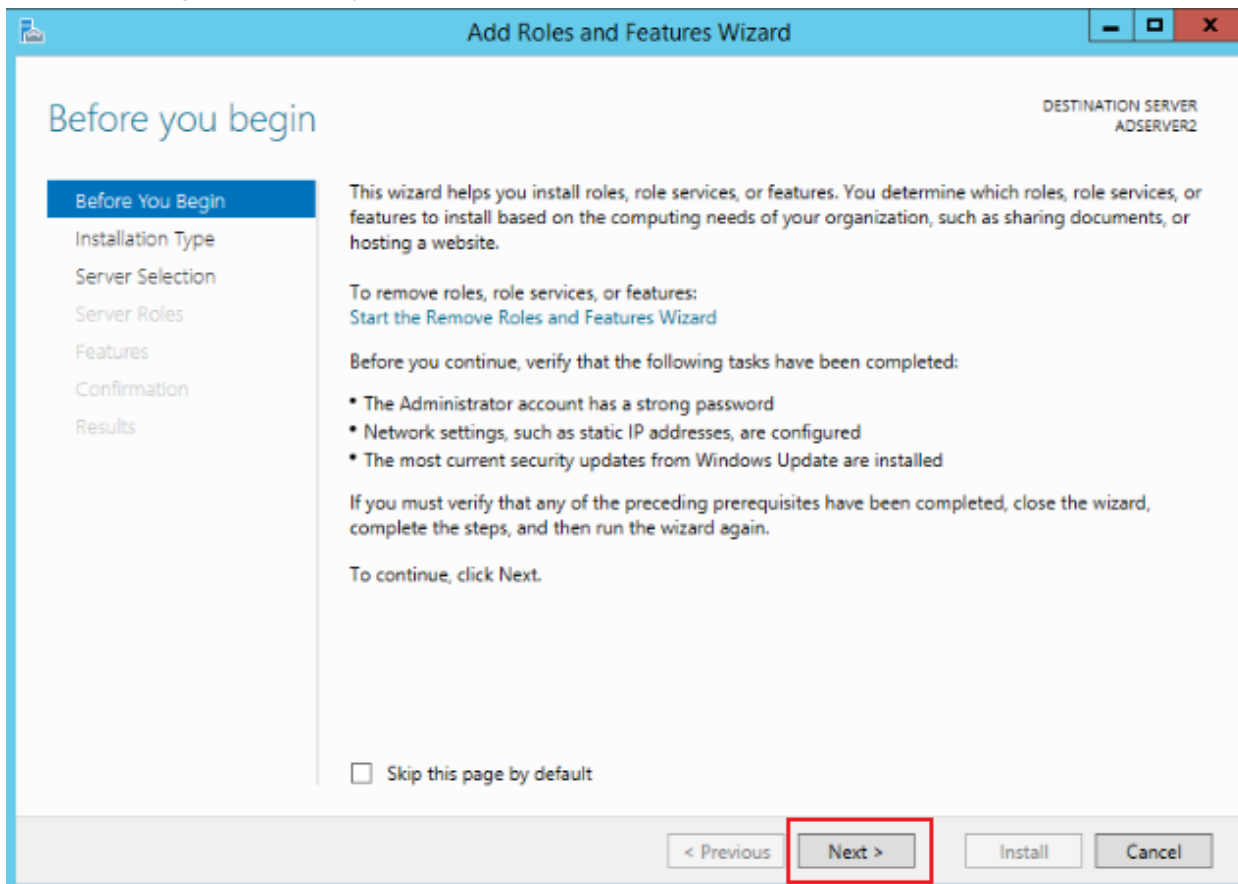
#### Explanation

If you have already installed and deployed Microsoft AD, you can ignore steps 1-5.

1. In the server, go to **Server Manager > Dashboard** and click **Add roles and features**, as shown in the figure below:



2. Keep clicking **Next** until you click **Install** to complete the installation, as shown in the figure below :



Add Roles and Features Wizard

Select installation type

DESTINATION SERVER  
ADSERVER2

Before You Begin

Installation Type

Server Selection

Server Roles

Features

Confirmation

Results

Select the installation type. You can install roles and features on a running physical computer or virtual machine, or on an offline virtual hard disk (VHD).

☒ **Role-based or feature-based installation**  
Configure a single server by adding roles, role services, and features.

☐ **Remote Desktop Services installation**  
Install required role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment.

< Previous

Next >

Install

Cancel

Add Roles and Features Wizard

Select destination server

DESTINATION SERVER  
ADSERVER2

Before You Begin

Installation Type

Server Selection

Server Roles

Features

Confirmation

Results

Select a server or a virtual hard disk on which to install roles and features.

☒ Select a server from the server pool

☐ Select a virtual hard disk

Server Pool

Filter:

Name	IP Address	Operating System
ADSERVER2	192.168.243.52	Microsoft Windows Server 2012 R2 Datacenter

1 Computer(s) found

This page shows servers that are running Windows Server 2012, and that have been added by using the Add Servers command in Server Manager. Offline servers and newly-added servers from which data collection is still incomplete are not shown.

< Previous

Next >

Install

Cancel

**Add Roles and Features Wizard**

DESTINATION SERVER  
ADSERVER2

## Select server roles

Before You Begin  
Installation Type  
Server Selection  
**Server Roles**  
Features  
AD DS  
Confirmation  
Results

Select one or more roles to install on the selected server.

Roles	Description
<input type="checkbox"/> Active Directory Certificate Services	Active Directory Domain Services (AD DS) stores information about objects on the network and makes this information available to users and network administrators. AD DS uses domain controllers to give network users access to permitted resources anywhere on the network through a single logon process.
<input checked="" type="checkbox"/> <b>Active Directory Domain Services</b>	
<input type="checkbox"/> Active Directory Federation Services	
<input type="checkbox"/> Active Directory Lightweight Directory Services	
<input type="checkbox"/> Active Directory Rights Management Services	
<input type="checkbox"/> Application Server	
<input type="checkbox"/> DHCP Server	
<input type="checkbox"/> DNS Server	
<input type="checkbox"/> Fax Server	
▸ <input checked="" type="checkbox"/> <b>File and Storage Services (1 of 12 installed)</b>	
<input type="checkbox"/> Hyper-V	
<input type="checkbox"/> Network Policy and Access Services	
<input type="checkbox"/> Print and Document Services	
<input type="checkbox"/> Remote Access	
<input type="checkbox"/> Remote Desktop Services	

< Previous **Next >** Install Cancel

**Add Roles and Features Wizard**

DESTINATION SERVER  
ADSERVER2

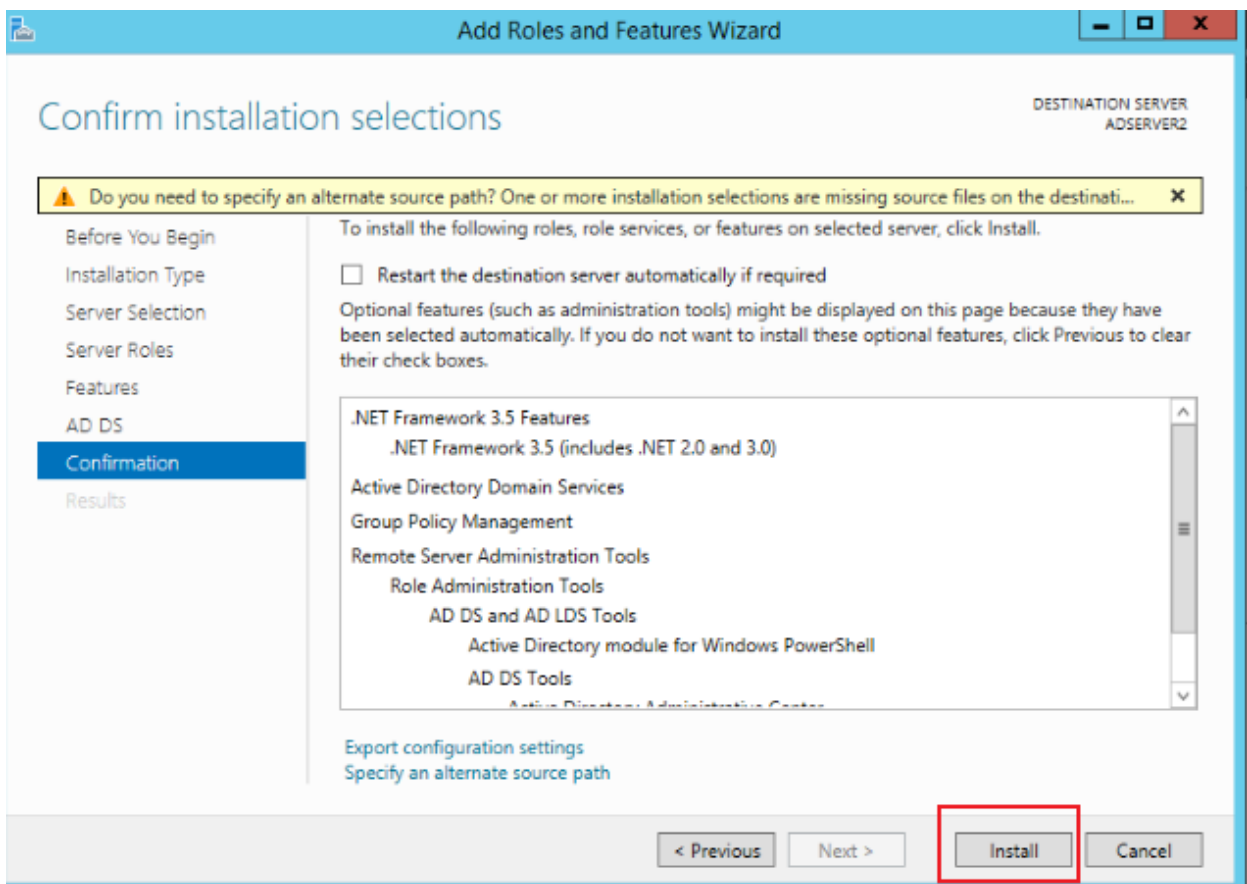
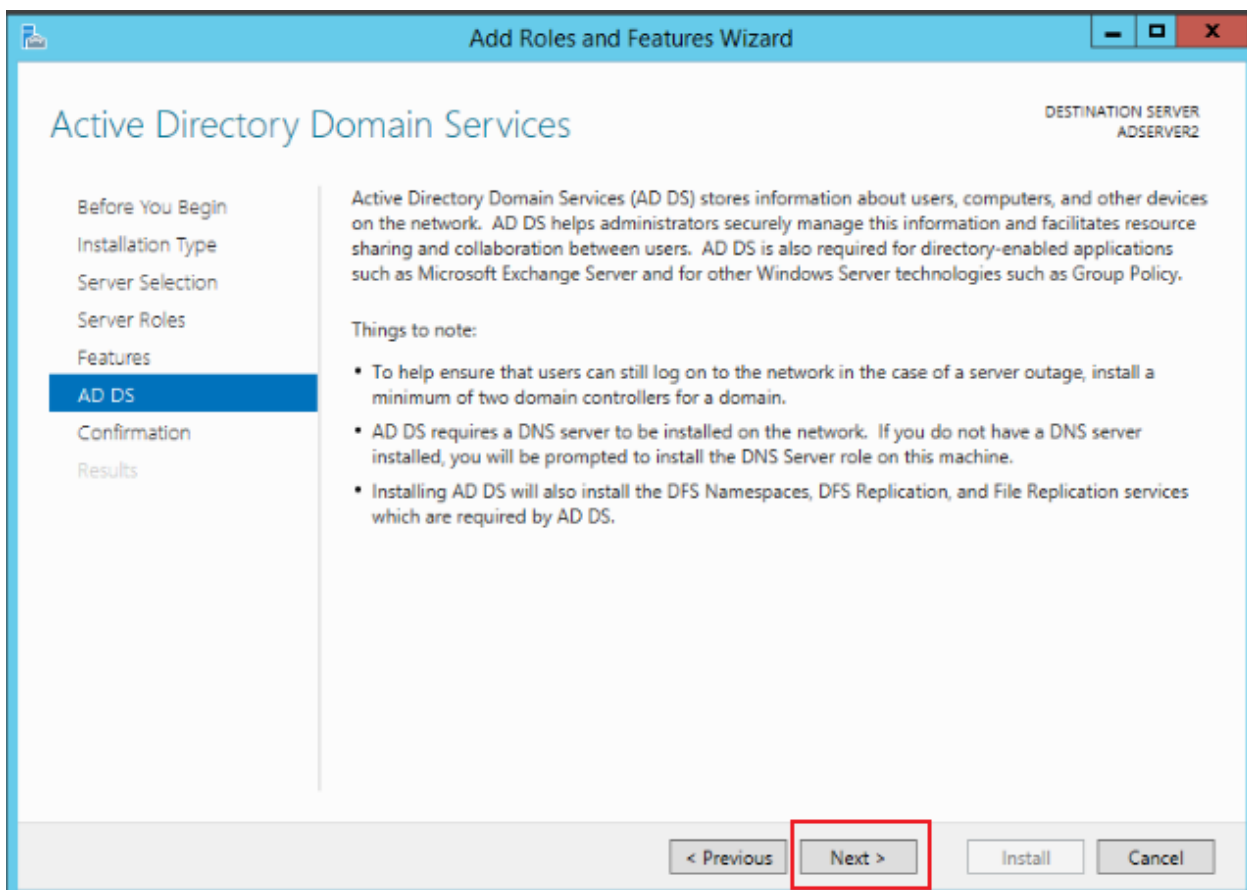
## Select features

Before You Begin  
Installation Type  
Server Selection  
Server Roles  
**Features**  
AD DS  
Confirmation  
Results

Select one or more features to install on the selected server.

Features	Description
▸ <input checked="" type="checkbox"/> <b>.NET Framework 3.5 Features</b>	.NET Framework 3.5 combines the power of the .NET Framework 2.0 APIs with new technologies for building applications that offer appealing user interfaces, protect your customers' personal identity information, enable seamless and secure communication, and provide the ability to model a range of business processes.
▸ <input checked="" type="checkbox"/> <b>.NET Framework 4.5 Features (2 of 7 installed)</b>	
▸ <input type="checkbox"/> Background Intelligent Transfer Service (BITS)	
<input type="checkbox"/> BitLocker Drive Encryption	
<input type="checkbox"/> BitLocker Network Unlock	
<input type="checkbox"/> BranchCache	
<input type="checkbox"/> Client for NFS	
<input type="checkbox"/> Data Center Bridging	
<input type="checkbox"/> Direct Play	
<input type="checkbox"/> Enhanced Storage	
<input type="checkbox"/> Failover Clustering	
<input checked="" type="checkbox"/> Group Policy Management	
<input type="checkbox"/> IIS Hostable Web Core	
<input type="checkbox"/> Ink and Handwriting Services	
< ▢ >	

< Previous **Next >** Install Cancel



3. After the installation is complete, click **Promote this server to a domain controller**, as shown in the

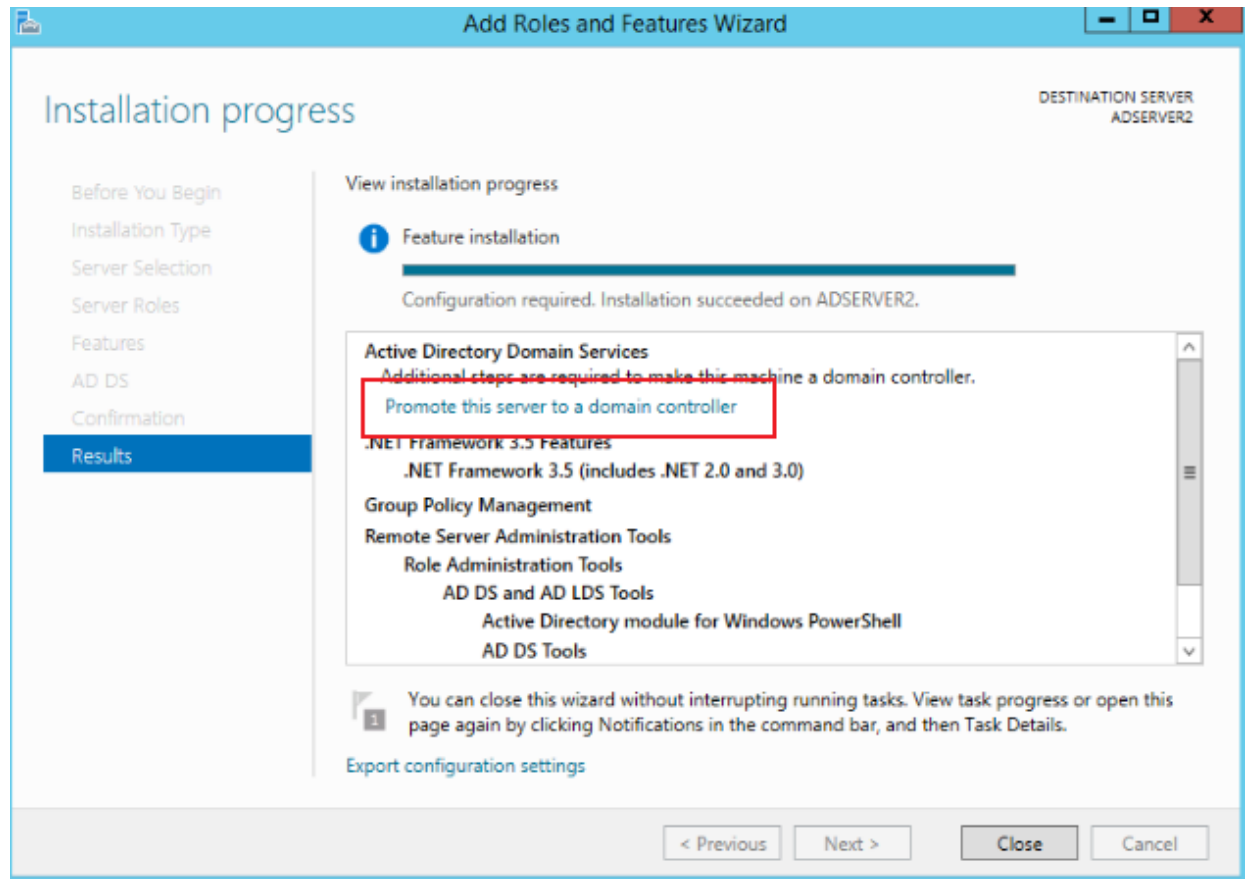
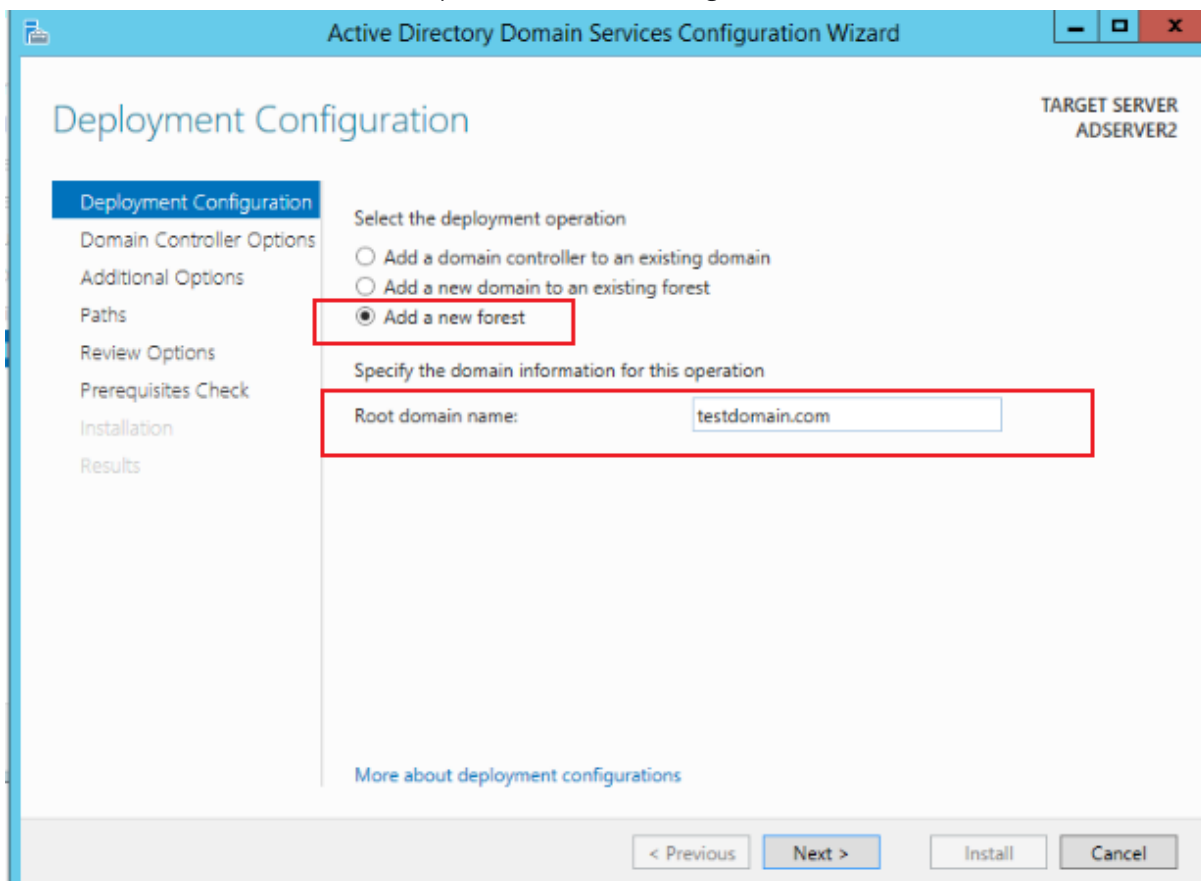


figure below:

4. Select **Add a new forest** on the Deployment Configuration page and add the Root domain name information to testdomain.com, as shown in the figure below:



5. Supplement the Password information in Domain Controller Options, as shown in the figure below. After completion, click **Next** and click **Install** to complete the installation:

The screenshot shows the 'Active Directory Domain Services Configuration Wizard' window. The title bar includes standard Windows window controls. The main title is 'Domain Controller Options'. In the top right corner, it says 'TARGET SERVER ADSERVER2'. On the left, a navigation pane lists the steps: 'Deployment Configuration', 'Domain Controller Options' (highlighted), 'DNS Options', 'Additional Options', 'Paths', 'Review Options', 'Prerequisites Check', 'Installation', and 'Results'. The main content area is titled 'Select functional level of the new forest and root domain'. It contains two dropdown menus: 'Forest functional level:' and 'Domain functional level:', both set to 'Windows Server 2012 R2'. Below these, under 'Specify domain controller capabilities', there are three checkboxes: 'Domain Name System (DNS) server' (checked), 'Global Catalog (GC)' (checked), and 'Read only domain controller (RODC)' (unchecked). A red rectangular box highlights the 'Type the Directory Services Restore Mode (DSRM) password' section, which includes 'Password:' and 'Confirm password:' fields, both masked with dots. At the bottom, there are four buttons: '< Previous', 'Next >', 'Install', and 'Cancel'.

The screenshot shows the 'Active Directory Domain Services Configuration Wizard' window at the 'Additional Options' step. The title bar and window controls are the same. The main title is 'Additional Options'. The 'TARGET SERVER' is still 'ADSERVER2'. The navigation pane on the left now highlights 'Additional Options'. The main content area is titled 'Verify the NetBIOS name assigned to the domain and change it if necessary'. It contains a text box labeled 'The NetBIOS domain name:' with the value 'TESTDOMAIN' entered. At the bottom, the 'Next >' button is highlighted with a red rectangular box. The other buttons ('< Previous', 'Install', 'Cancel') are also visible.

Active Directory Domain Services Configuration Wizard

TARGET SERVER  
ADSERVER2

## Paths

Deployment Configuration  
Domain Controller Options  
DNS Options  
Additional Options  
**Paths**  
Review Options  
Prerequisites Check  
Installation  
Results

Specify the location of the AD DS database, log files, and SYSVOL

Database folder: C:\Windows\NTDS  
Log files folder: C:\Windows\NTDS  
SYSVOL folder: C:\Windows\SYSVOL

[More about Active Directory paths](#)

< Previous **Next >** Install Cancel

Active Directory Domain Services Configuration Wizard

TARGET SERVER  
ADSERVER2

## Prerequisites Check

Deployment Configuration  
Domain Controller Options  
DNS Options  
Additional Options  
Paths  
Review Options  
**Prerequisites Check**  
Installation  
Results

✓ All prerequisite checks passed successfully. Click 'Install' to begin installation. [Show more](#) ✕

Prerequisites need to be validated before Active Directory Domain Services is installed on this computer  
[Rerun prerequisites check](#)

^ View results

⚠ Windows Server 2012 R2 domain controllers have a default for the security setting named "Allow cryptography algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms when establishing security channel sessions.  
For more information about this setting, see Knowledge Base article 942564 (<http://go.microsoft.com/fwlink/?LinkId=104751>).

⚠ A delegation for this DNS server cannot be created because the authoritative parent zone cannot be found or it does not run Windows DNS server. If you are integrating with an existing DNS infrastructure, you should manually create a delegation to this DNS server in the parent zone to ensure reliable name resolution from outside the domain "testdomain.com". Otherwise, no action is required.

⚠ If you click Install, the server automatically reboots at the end of the promotion operation.

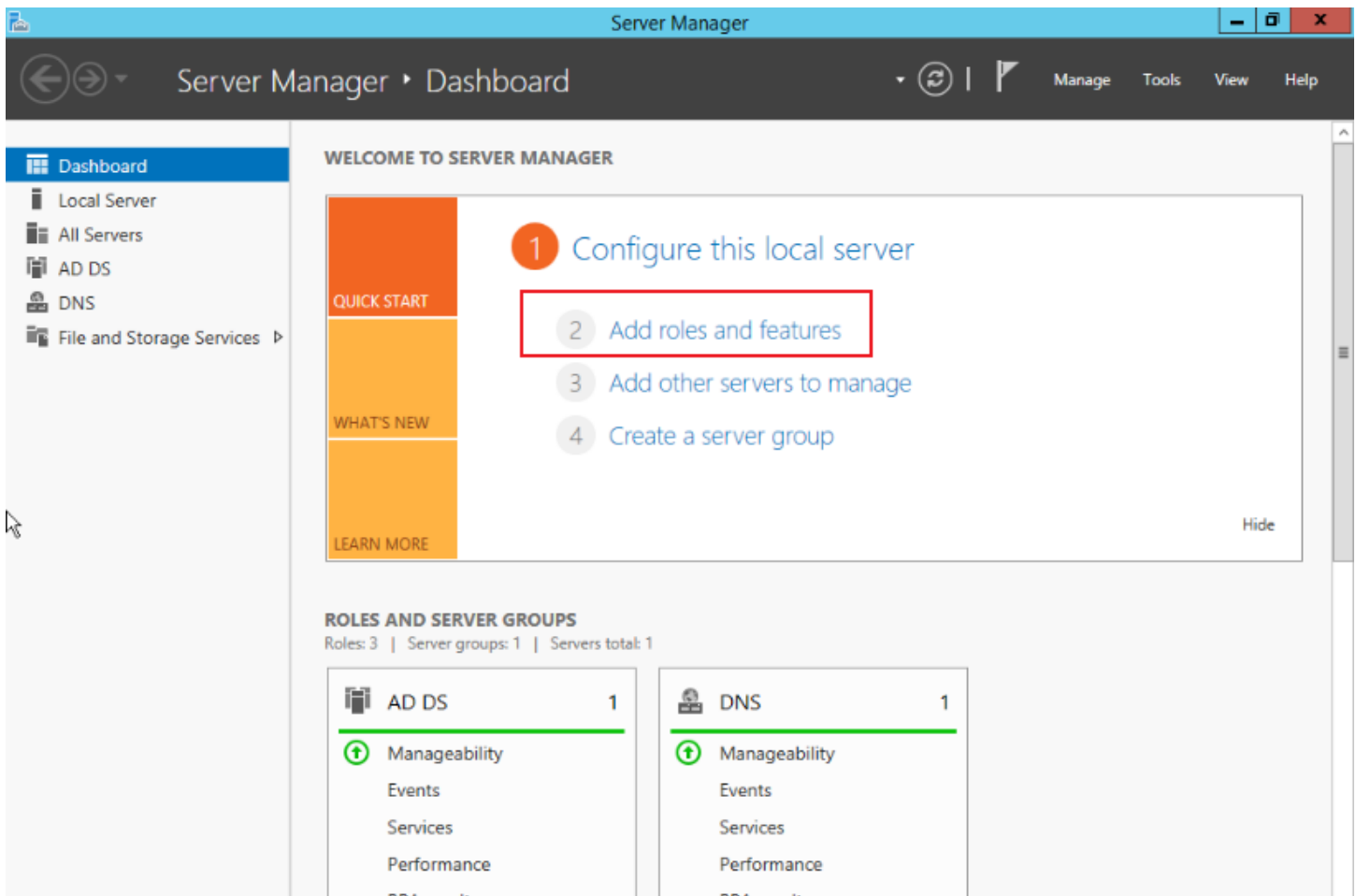
[More about prerequisites](#)

< Previous Next > **Install** Cancel

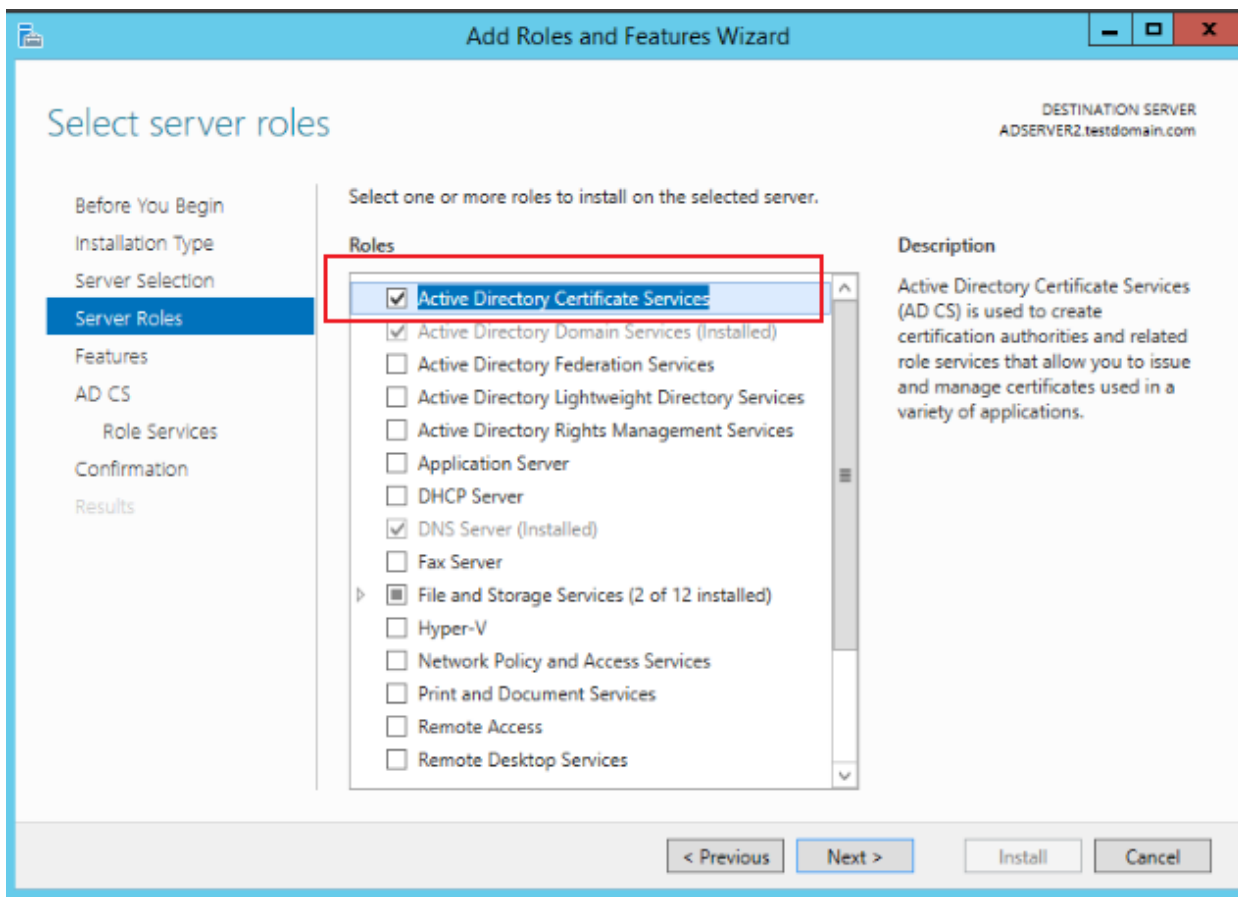
Install CA



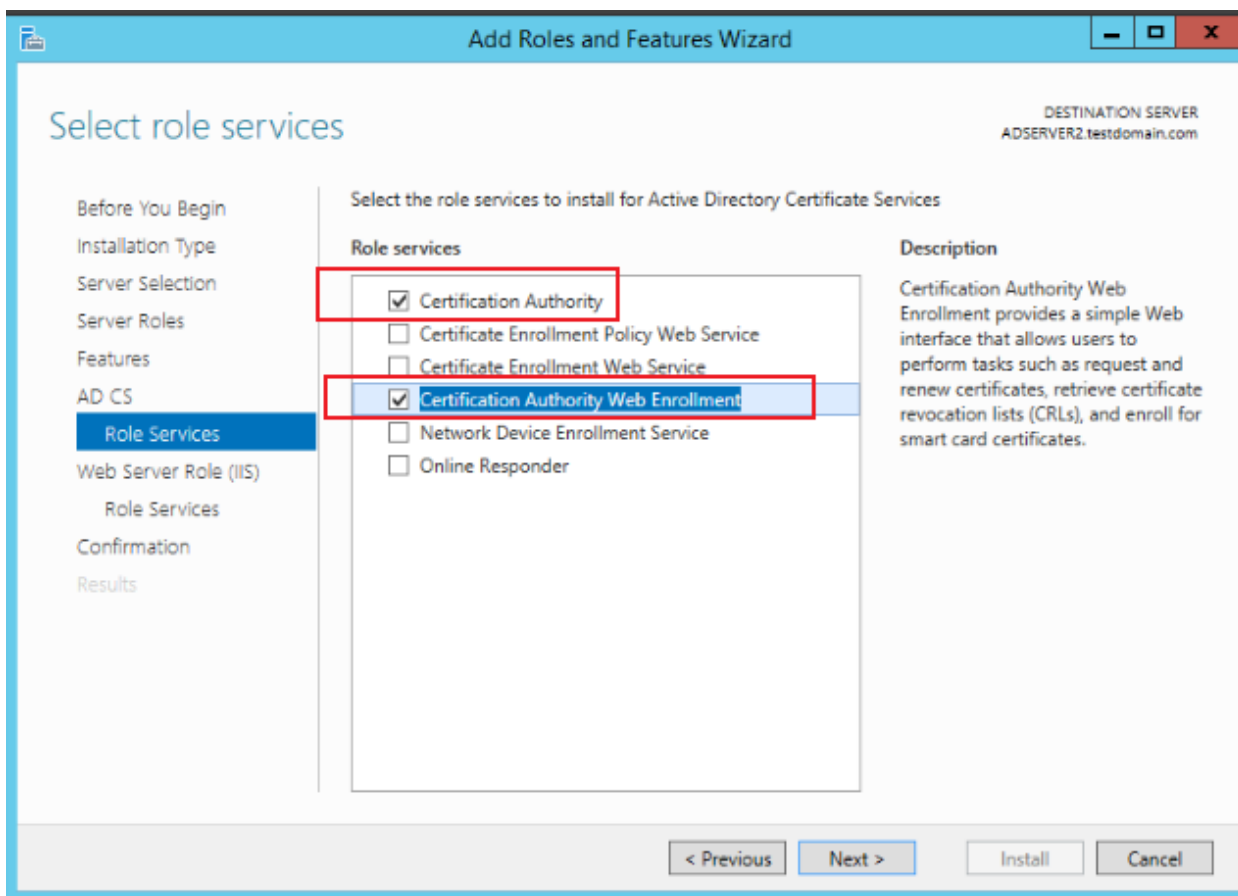
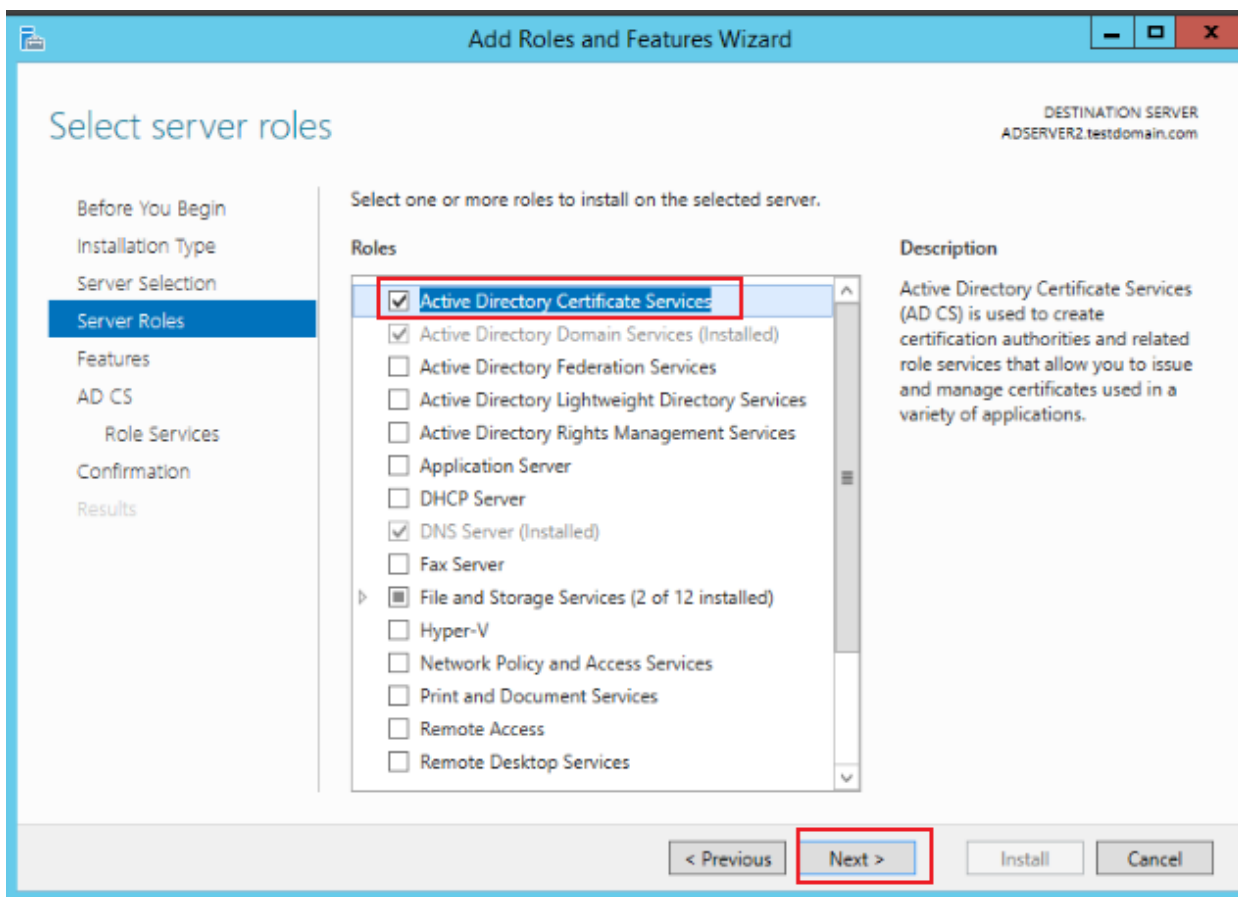
1. In the cloud server, go to **Server Manager** > **Dashboard** and click **Add roles and features**, as shown below:



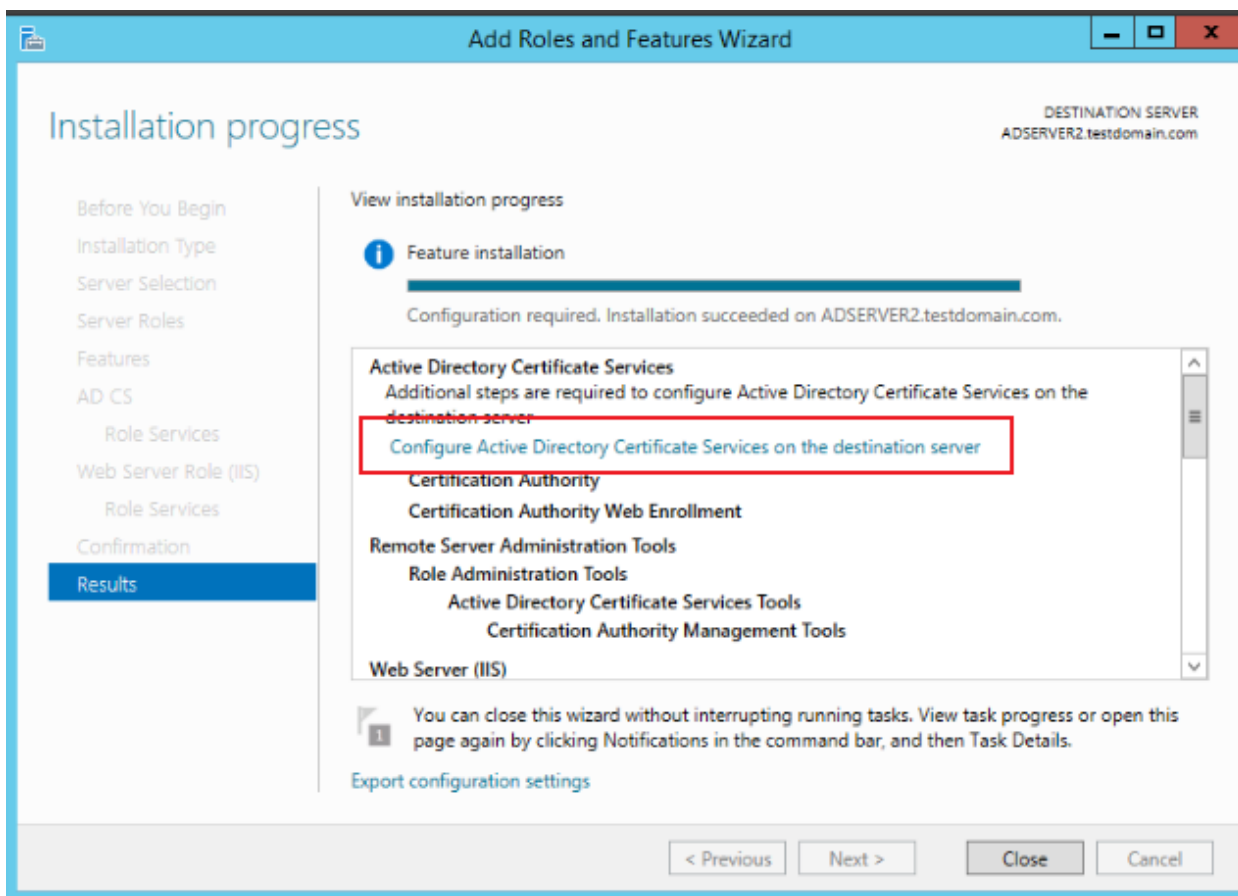
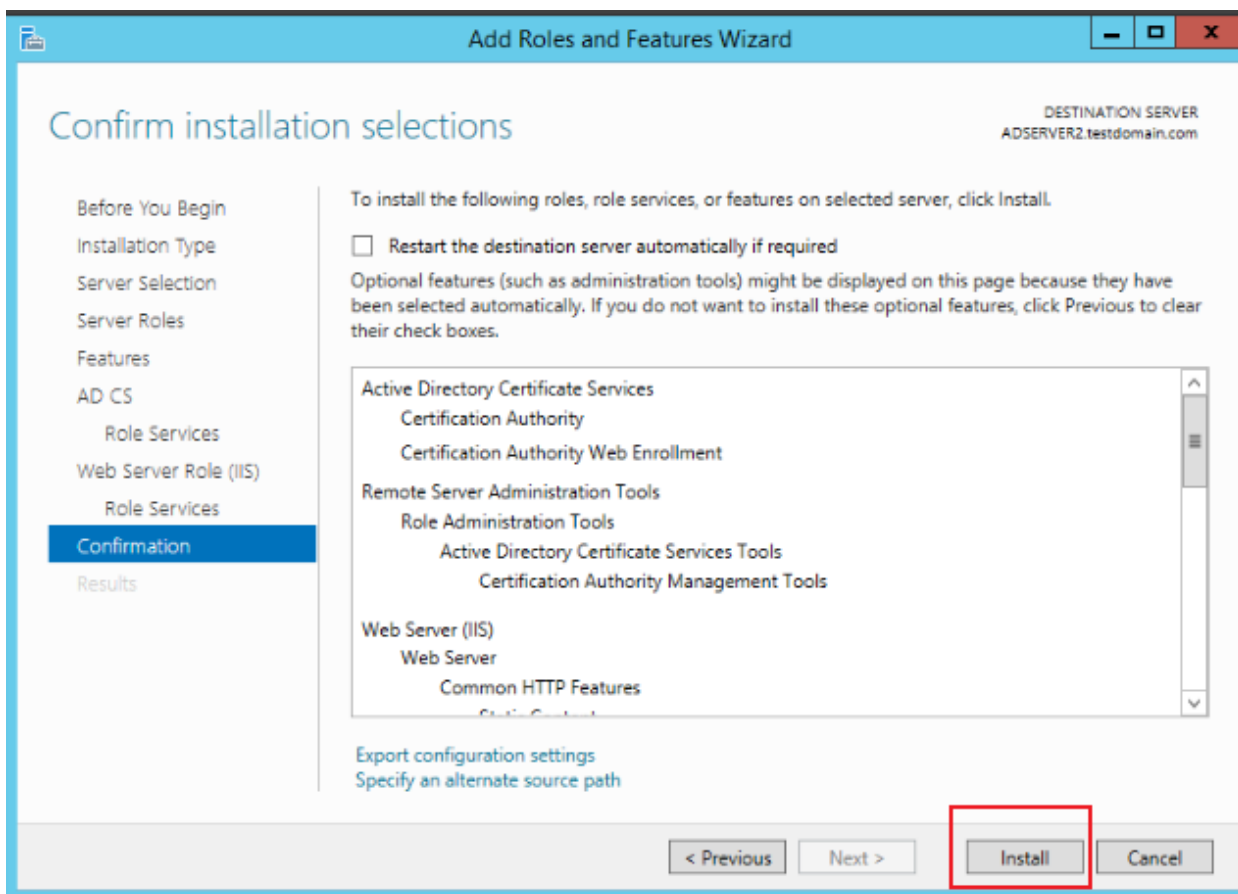
2. Keep clicking **Next** until the Server Roles page, and select Active Directory Certificate Services for Roles, as shown below:

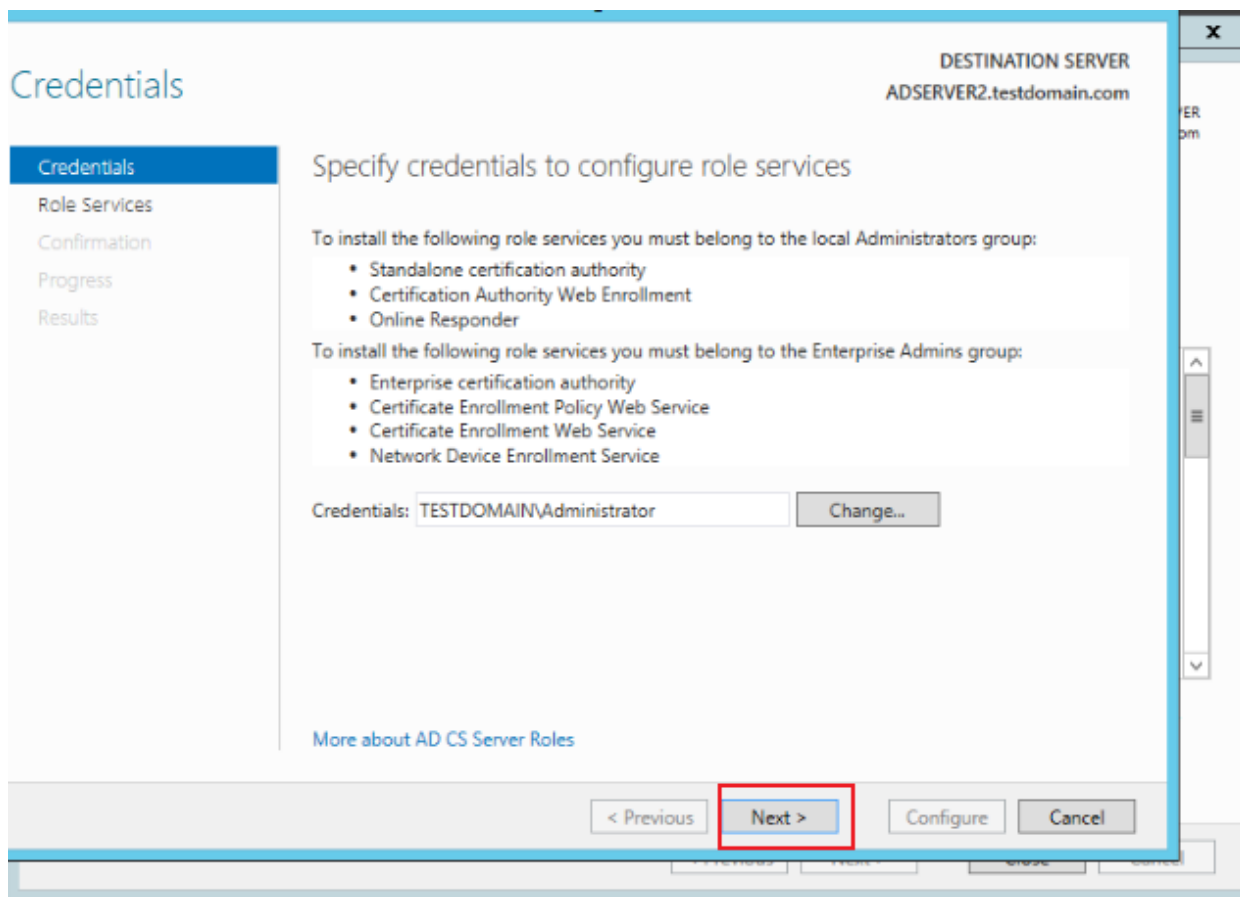


3. Keep clicking **Next** until you reach the AD CS-Server Roles page. For Server Roles, select Certification Authority, Certification Authority Web Enrollment, as shown in the figure below:

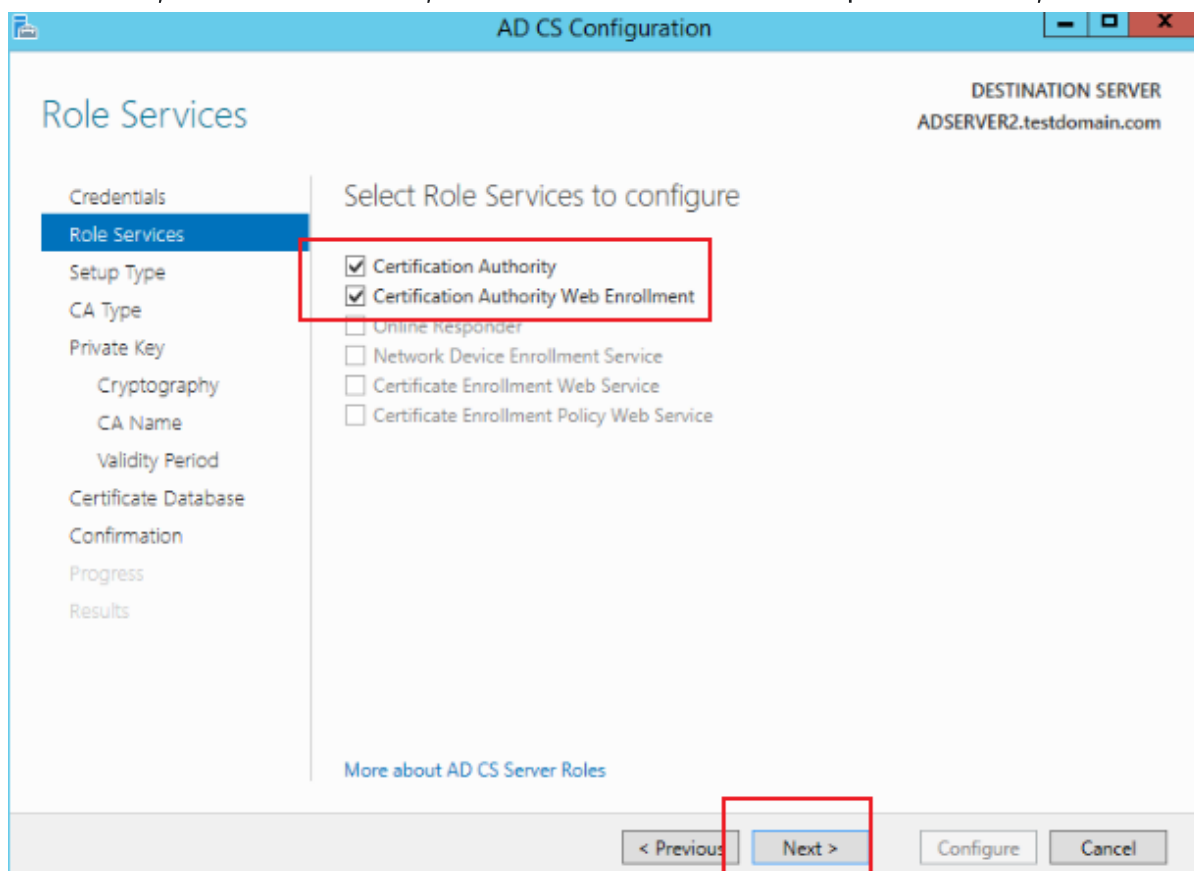


4. Keep clicking **Next** until the Results page, click the information below to configure AD CS Configuration, as shown below:

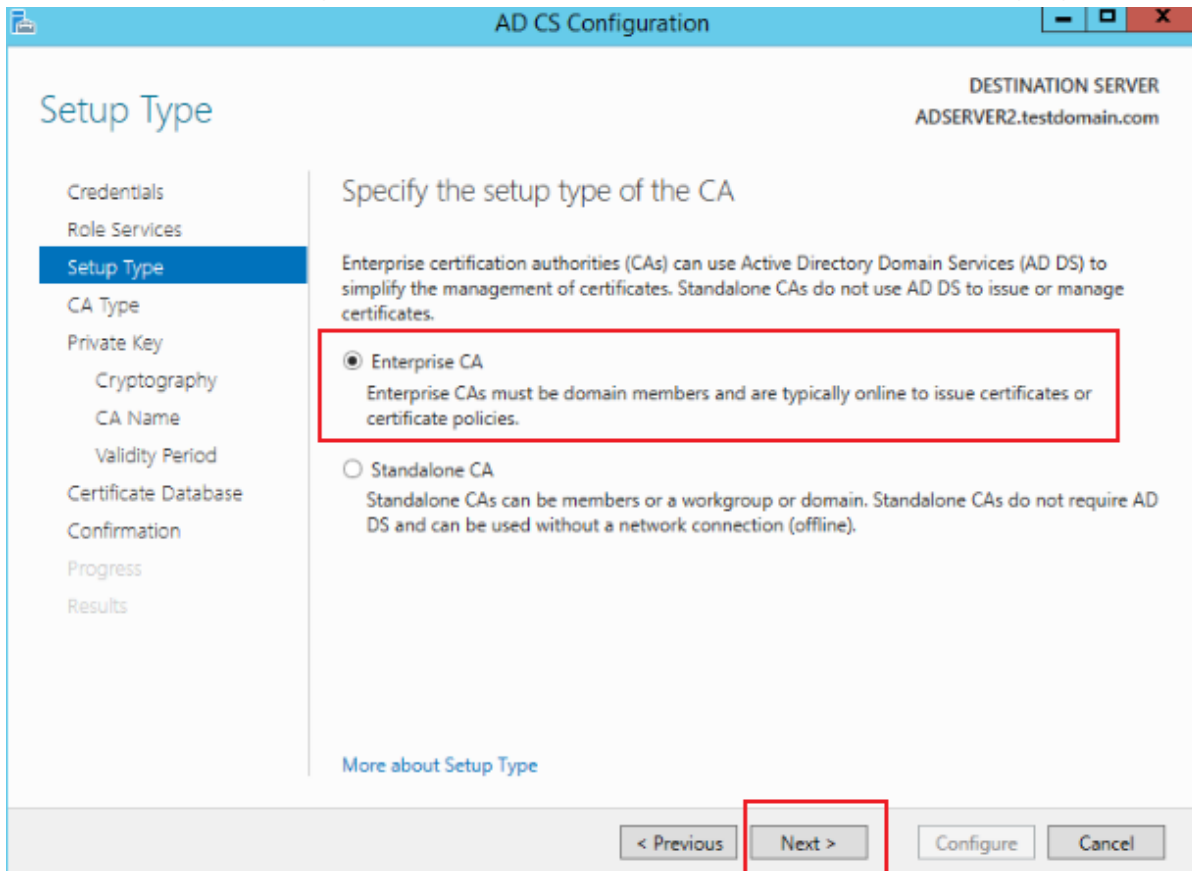




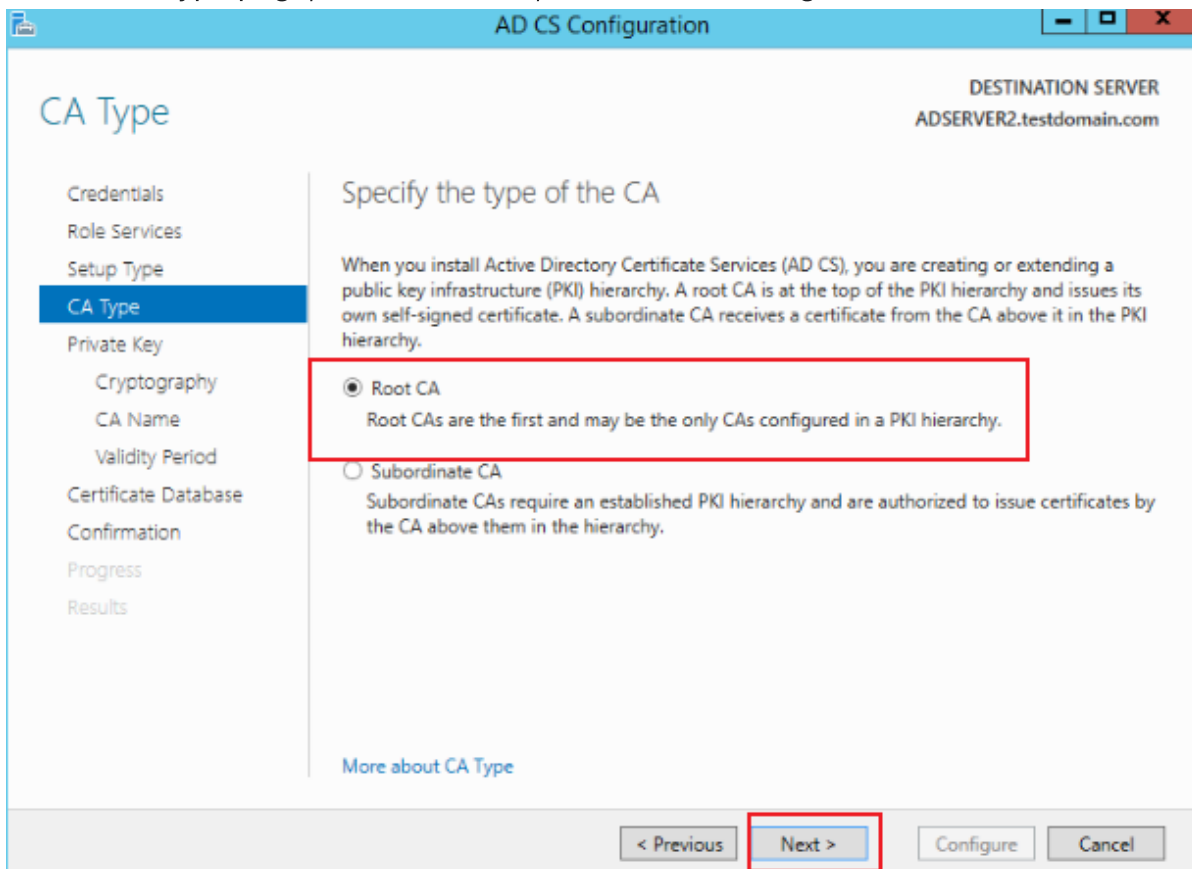
5. Click **Next**, in Role Services, check the information in the picture below, and click **Next**



6. On the Setup Type page, select Enterprise CA, as shown in the following figure:



7. On the CA Type page, select Root CA, as shown in the figure below:



8. On the Private Key page, select Create a new private key, as shown in the figure below:

AD CS Configuration

DESTINATION SERVER  
ADSERVER2.testdomain.com

### Private Key

Credentials  
Role Services  
Setup Type  
CA Type  
**Private Key**  
Cryptography  
CA Name  
Validity Period  
Certificate Database  
Confirmation  
Progress  
Results

Specify the type of the private key

To generate and issue certificates to clients, a certification authority (CA) must have a private key.

- ☒ **Create a new private key**  
Use this option if you do not have a private key or want to create a new private key.
- ☐ Use existing private key  
Use this option to ensure continuity with previously issued certificates when reinstalling a CA.
  - ☐ Select a certificate and use its associated private key  
Select this option if you have an existing certificate on this computer or if you want to import a certificate and use its associated private key.
  - ☐ Select an existing private key on this computer  
Select this option if you have retained private keys from a previous installation or want to use a private key from an alternate source.

[More about Private Key](#)

< Previous **Next >** Configure Cancel

AD CS Configuration

DESTINATION SERVER  
ADSERVER2.testdomain.com

### Cryptography for CA

Credentials  
Role Services  
Setup Type  
CA Type  
Private Key  
**Cryptography**  
CA Name  
Validity Period  
Certificate Database  
Confirmation  
Progress  
Results

Specify the cryptographic options

Select a cryptographic provider: RSA#Microsoft Software Key Storage Provider Key length: 2048

Select the hash algorithm for signing certificates issued by this CA:

- SHA256
- SHA384
- SHA512
- SHA1**
- MD5

☐ Allow administrator interaction when the private key is accessed by the CA.

[More about Cryptography](#)

< Previous **Next >** Configure Cancel

AD CS Configuration

CA Name

DESTINATION SERVER  
ADSERVER2.testdomain.com

Credentials  
Role Services  
Setup Type  
CA Type  
Private Key  
Cryptography  
**CA Name**  
Validity Period  
Certificate Database  
Confirmation  
Progress  
Results

Specify the name of the CA

Type a common name to identify this certification authority (CA). This name is added to all certificates issued by the CA. Distinguished name suffix values are automatically generated but can be modified.

Common name for this CA:  
testdomain-ADSERVER2-CA

Distinguished name suffix:  
DC=testdomain,DC=com

Preview of distinguished name:  
CN=testdomain-ADSERVER2-CA,DC=testdomain,DC=com

[More about CA Name](#)

< Previous

**Next >**

Configure

Cancel

AD CS Configuration

Validity Period

DESTINATION SERVER  
ADSERVER2.testdomain.com

Credentials  
Role Services  
Setup Type  
CA Type  
Private Key  
Cryptography  
CA Name  
**Validity Period**  
Certificate Database  
Confirmation  
Progress  
Results

Specify the validity period

Select the validity period for the certificate generated for this certification authority (CA):  
5 Years

CA expiration Date: 5/14/2029 6:30:00 PM

The validity period configured for this CA certificate should exceed the validity period for the certificates it will issue.

[More about Validity Period](#)

< Previous

**Next >**

Configure

Cancel



9. On the Certificate Database page, add information and click **Next**, as shown below:

The screenshot displays the 'AD CS Configuration' window, specifically the 'CA Database' tab. On the left, a navigation pane lists steps: Credentials, Role Services, Setup Type, CA Type, Private Key, Cryptography, CA Name, Validity Period, **Certificate Database** (selected), Confirmation, Progress, and Results. The main area is titled 'Specify the database locations'. It contains two text boxes: 'Certificate database location:' and 'Certificate database log location:', both containing the path 'C:\Windows\system32\CertLog'. In the top right corner, it says 'DESTINATION SERVER: ADSERVER2.testdomain.com'. At the bottom, there are four buttons: '< Previous', 'Next >' (highlighted with a red box), 'Configure', and 'Cancel'. A link 'More about CA Database' is also visible.

AD CS Configuration

Confirmation

DESTINATION SERVER  
ADSERVER2.testdomain.com

Credentials

Role Services

Setup Type

CA Type

Private Key

Cryptography

CA Name

Validity Period

Certificate Database

Confirmation

Progress

Results

To configure the following roles, role services, or features, click Configure.

^ Active Directory Certificate Services

Certification Authority

CA Type: Enterprise Root

Cryptographic provider: RSA#Microsoft Software Key Storage Provider

Hash Algorithm: SHA1

Key Length: 2048

Allow Administrator Interaction: Disabled

Certificate Validity Period: 5/14/2029 6:30:00 PM

Distinguished Name: CN=testdomain-ADSERVER2-CA,DC=testdomain,DC=com

Certificate Database Location: C:\Windows\system32\CertLog

Certificate Database Log Location: C:\Windows\system32\CertLog

Certification Authority Web Enrollment

< Previous

Next >

Configure

Cancel

AD CS Configuration

Results

DESTINATION SERVER  
ADSERVER2.testdomain.com

Credentials

Role Services

Setup Type

CA Type

Private Key

Cryptography

CA Name

Validity Period

Certificate Database

Confirmation

Progress

Results

The following roles, role services, or features were configured:

^ Active Directory Certificate Services

Certification Authority

More about CA Configuration

✓ Configuration succeeded

Certification Authority Web Enrollment

More about Web Enrollment Configuration

✓ Configuration succeeded

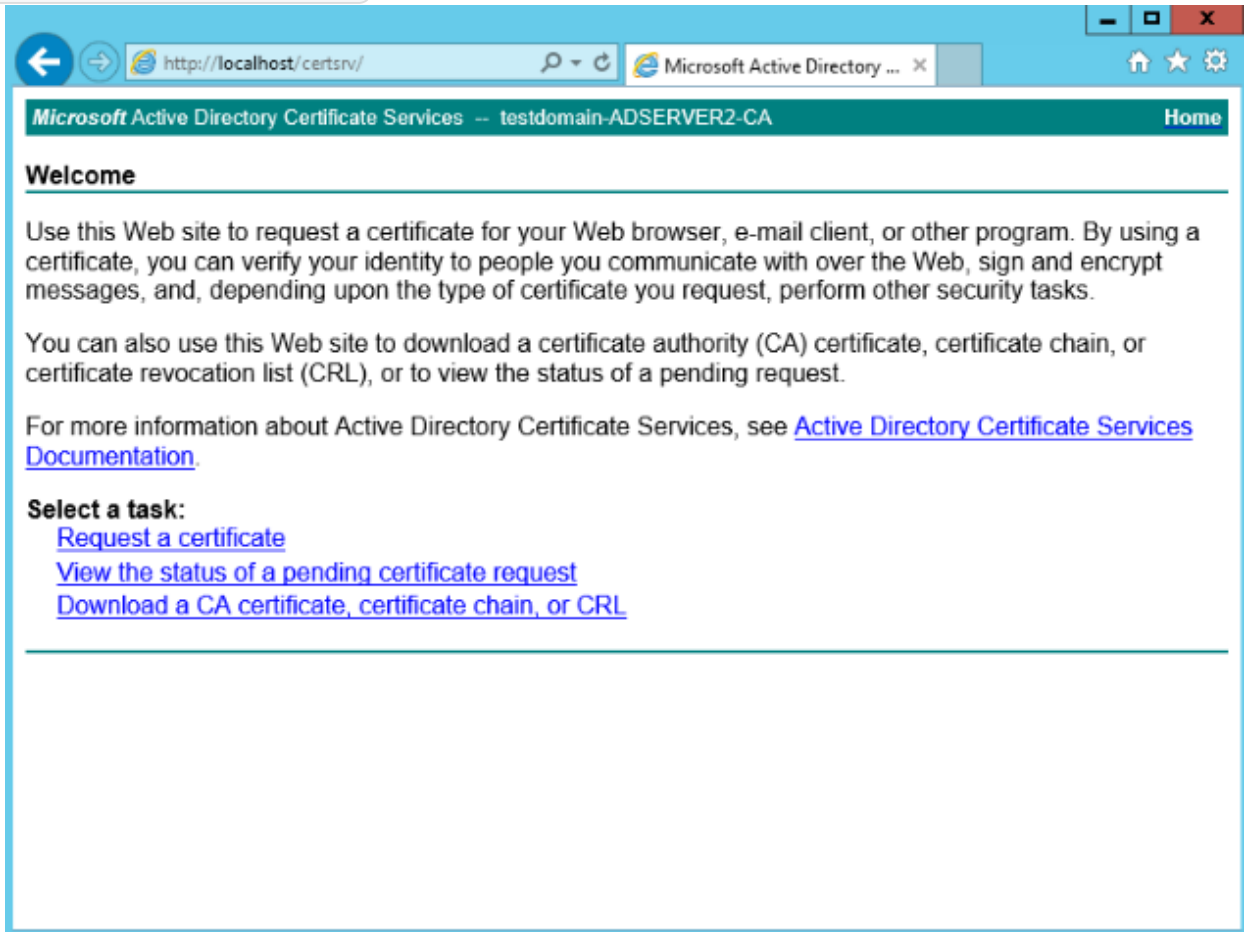
< Previous

Next >

Close

Cancel

10. Visit `http://localhost/certsrv` to ensure that the CA is installed successfully, as shown in the

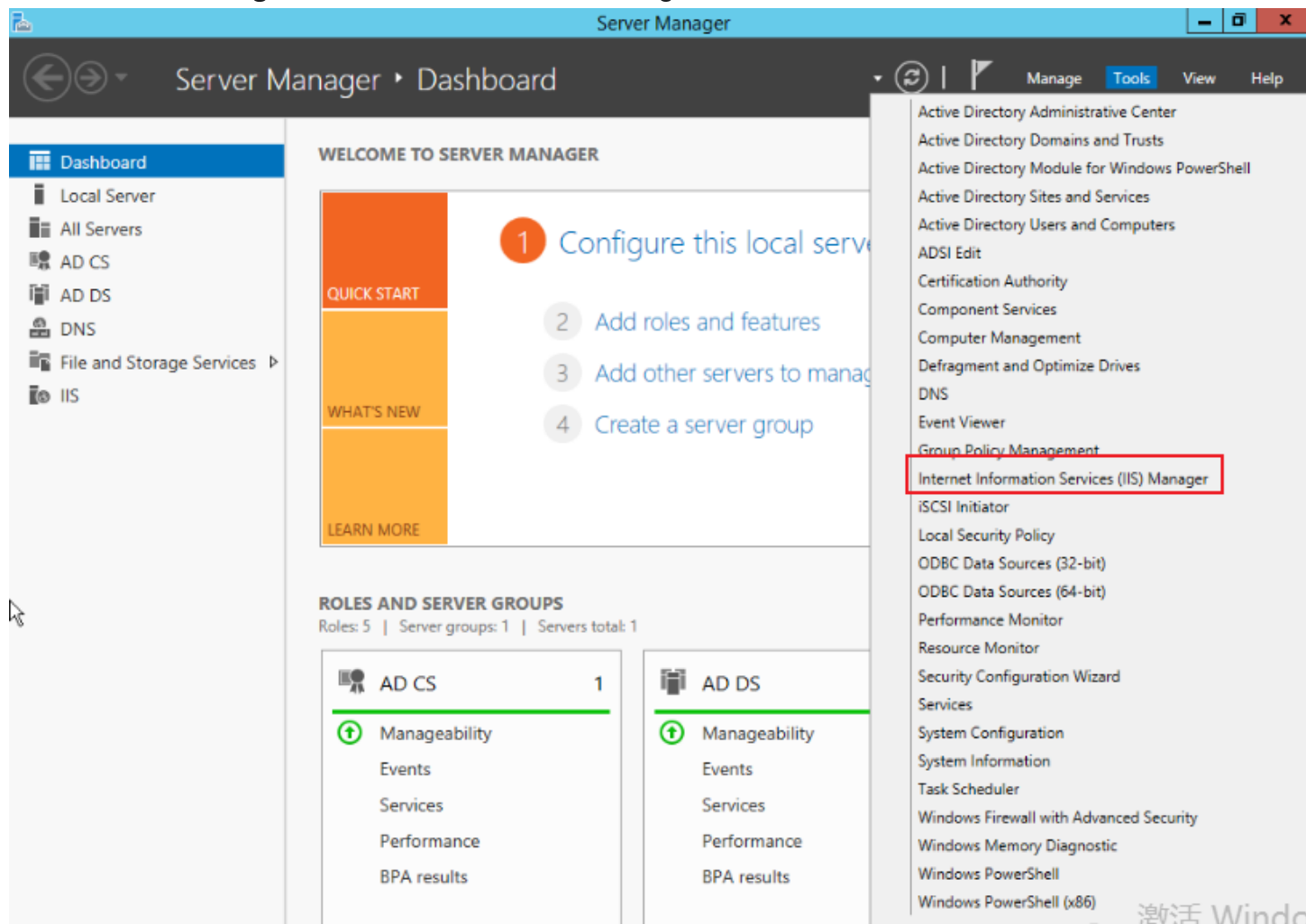


following:

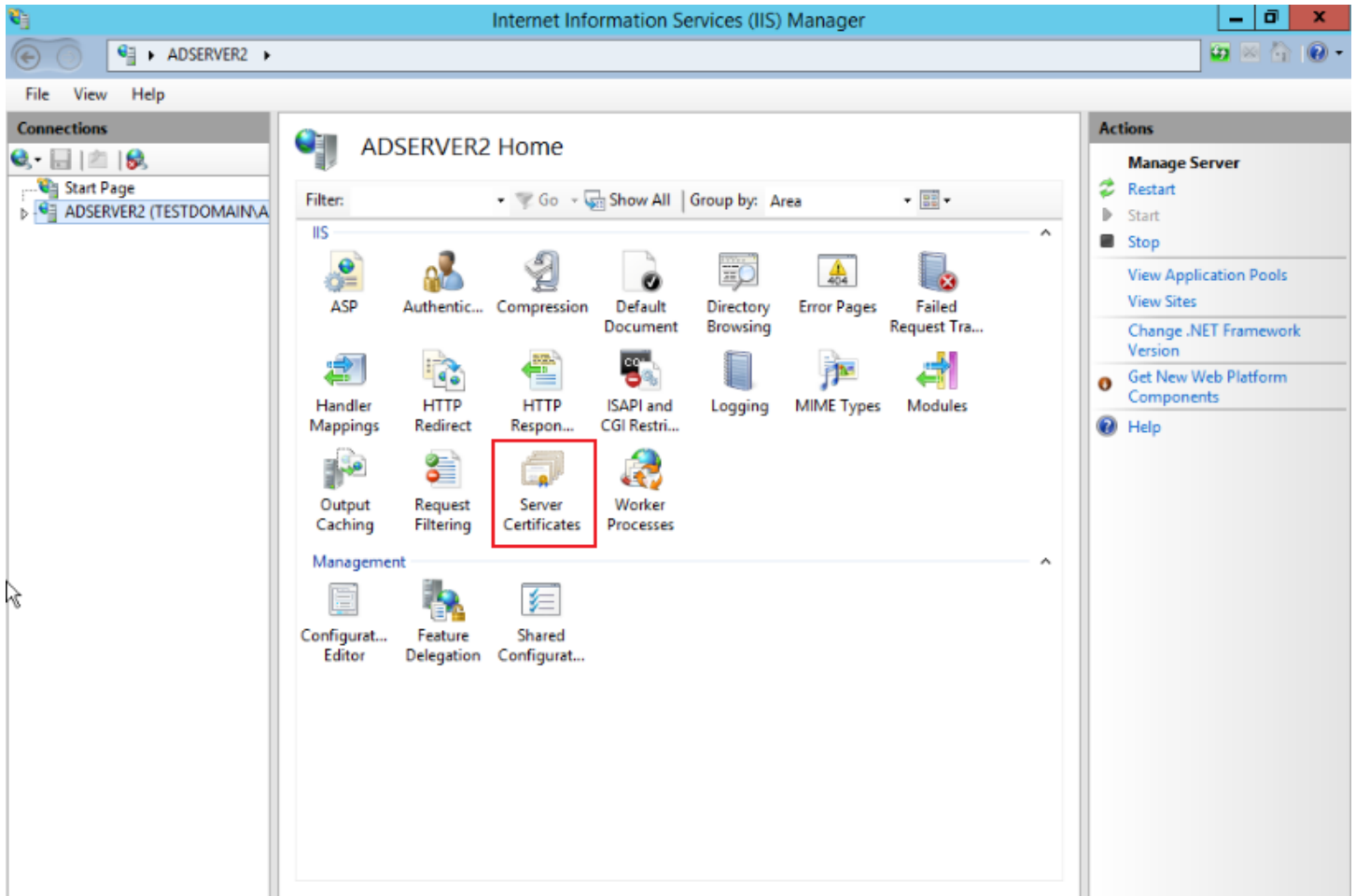
## Install ADFS service

Before configuring, you need to issue an authorization certificate to the computer or designated user or computer. Before installing ADFS, you need to create and configure a certificate. In this article, you apply for the certificate through IIS.

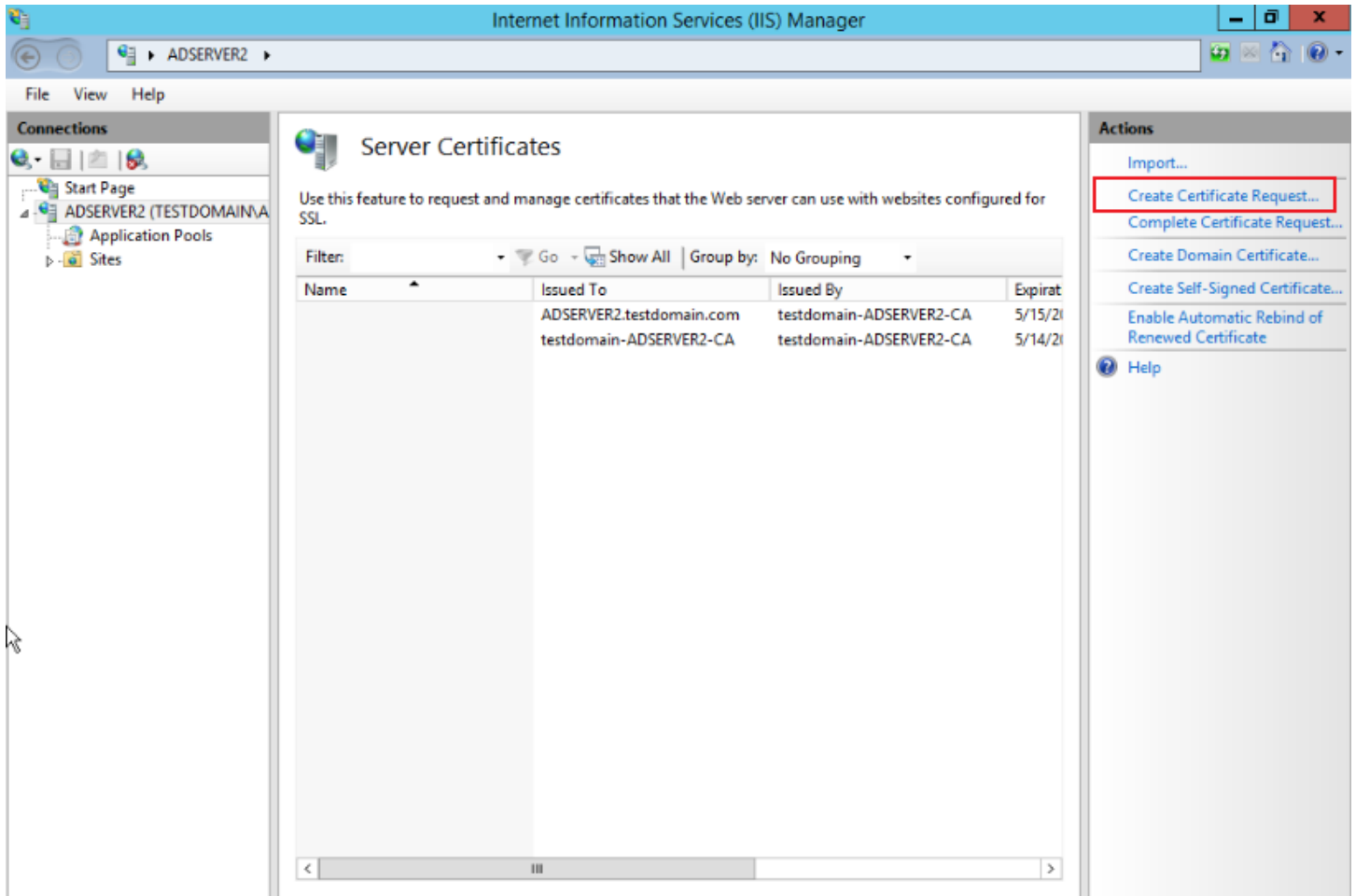
1. Click **Server Manager** > **Tools** and select IIS Manager.

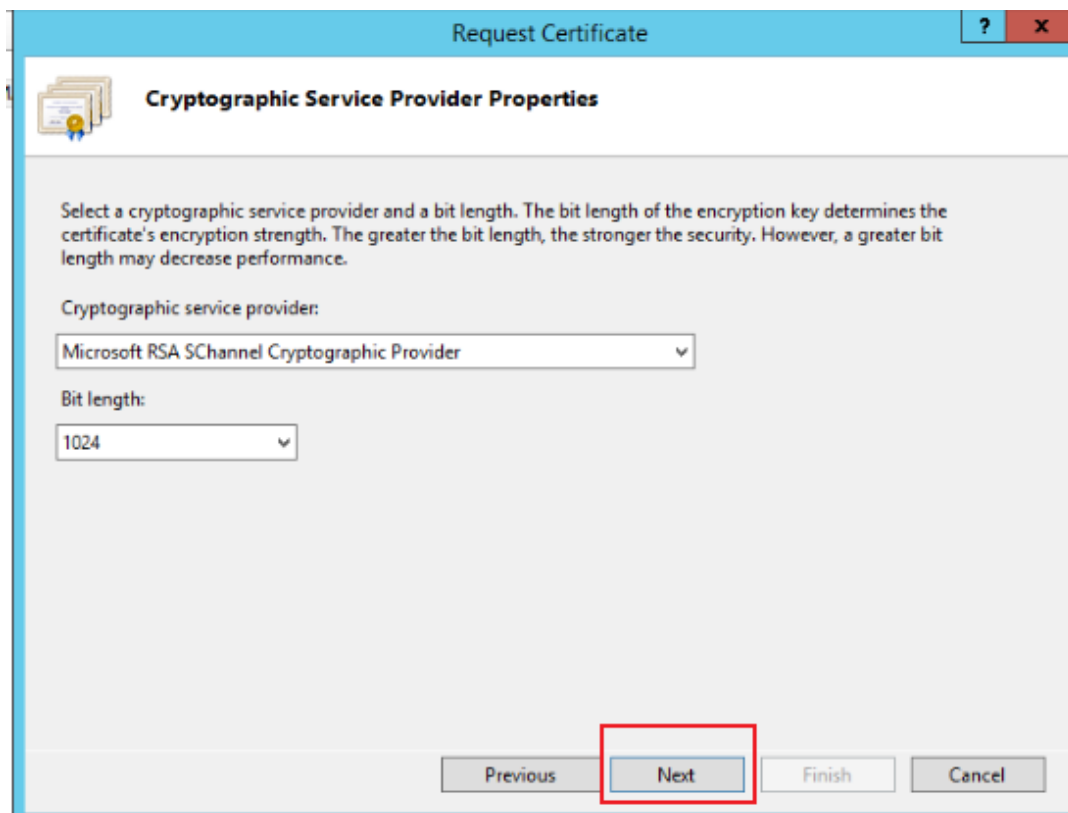
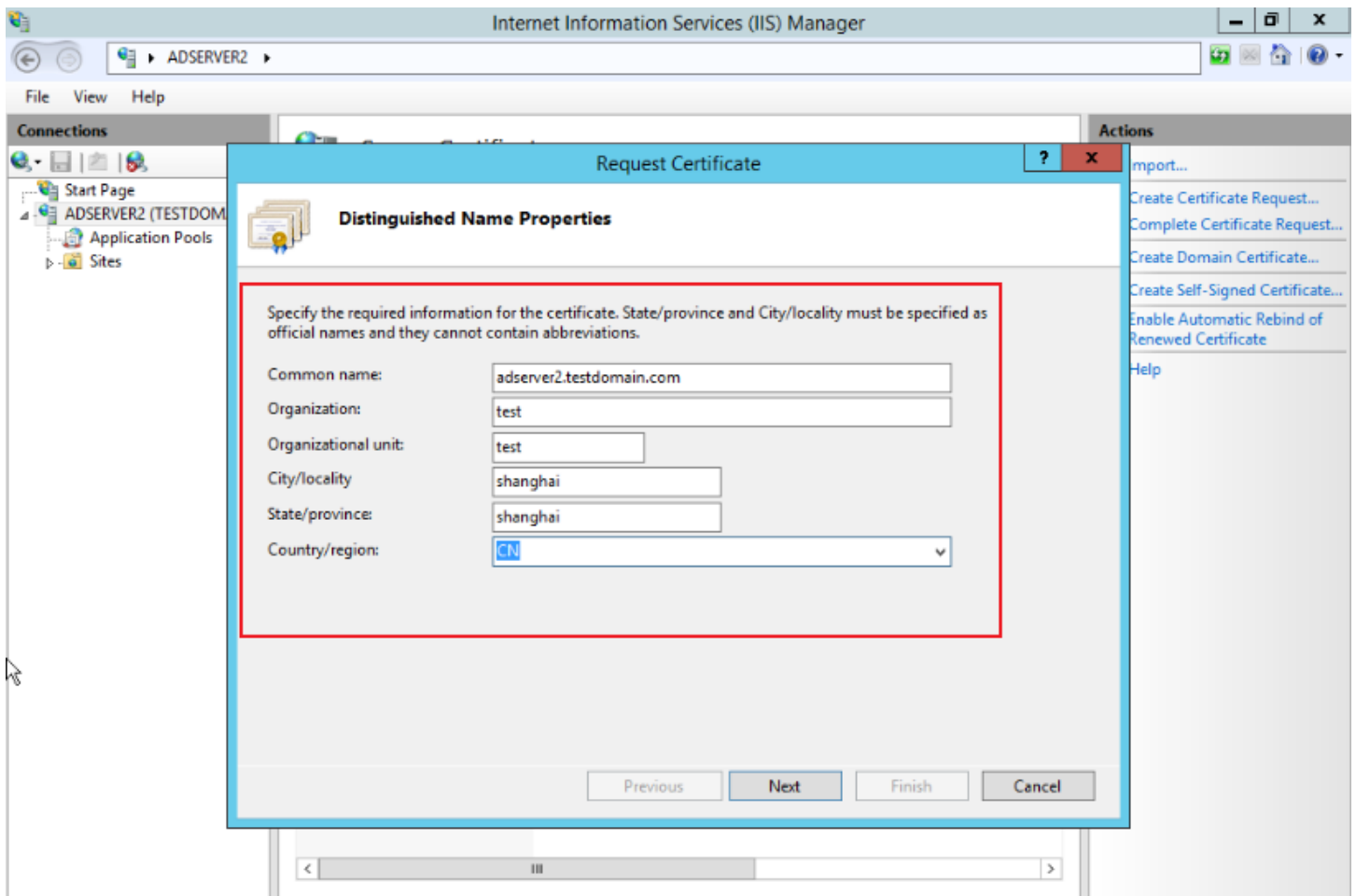


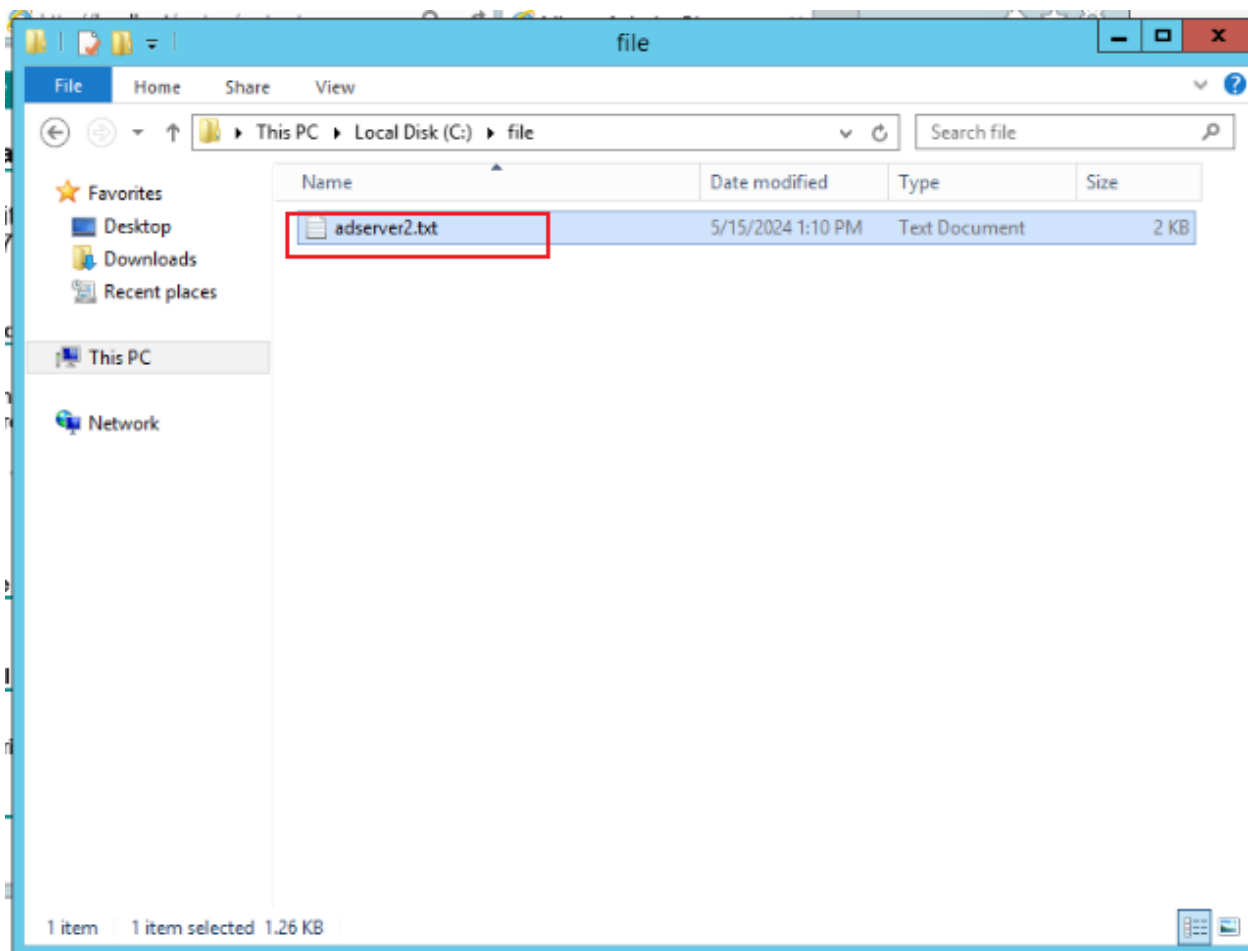
2. In IIS Manager, click **Server Certificates**, as shown below:



3. Enter the **Server Certificates** page and click **Create Certificate Request**, as shown below:

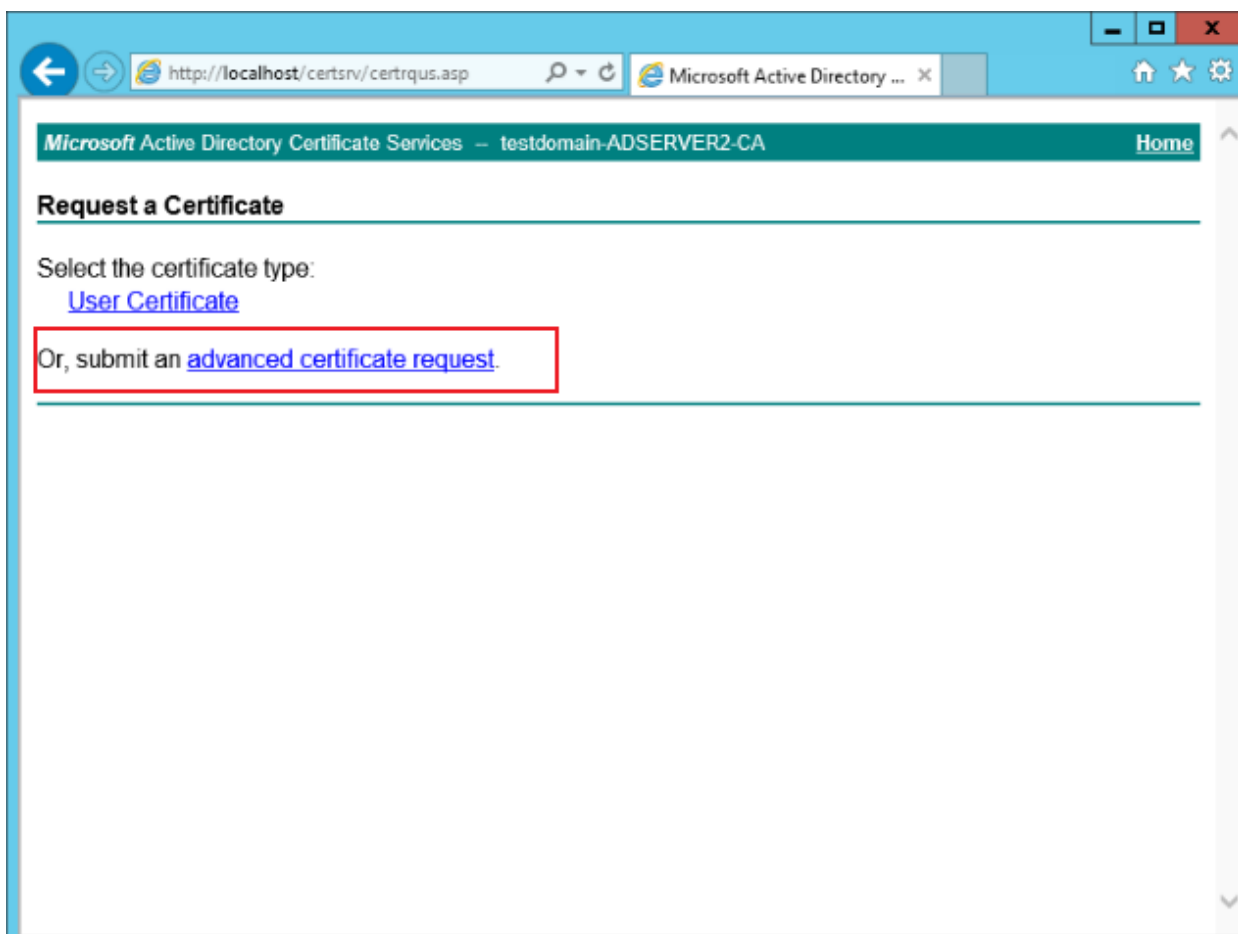
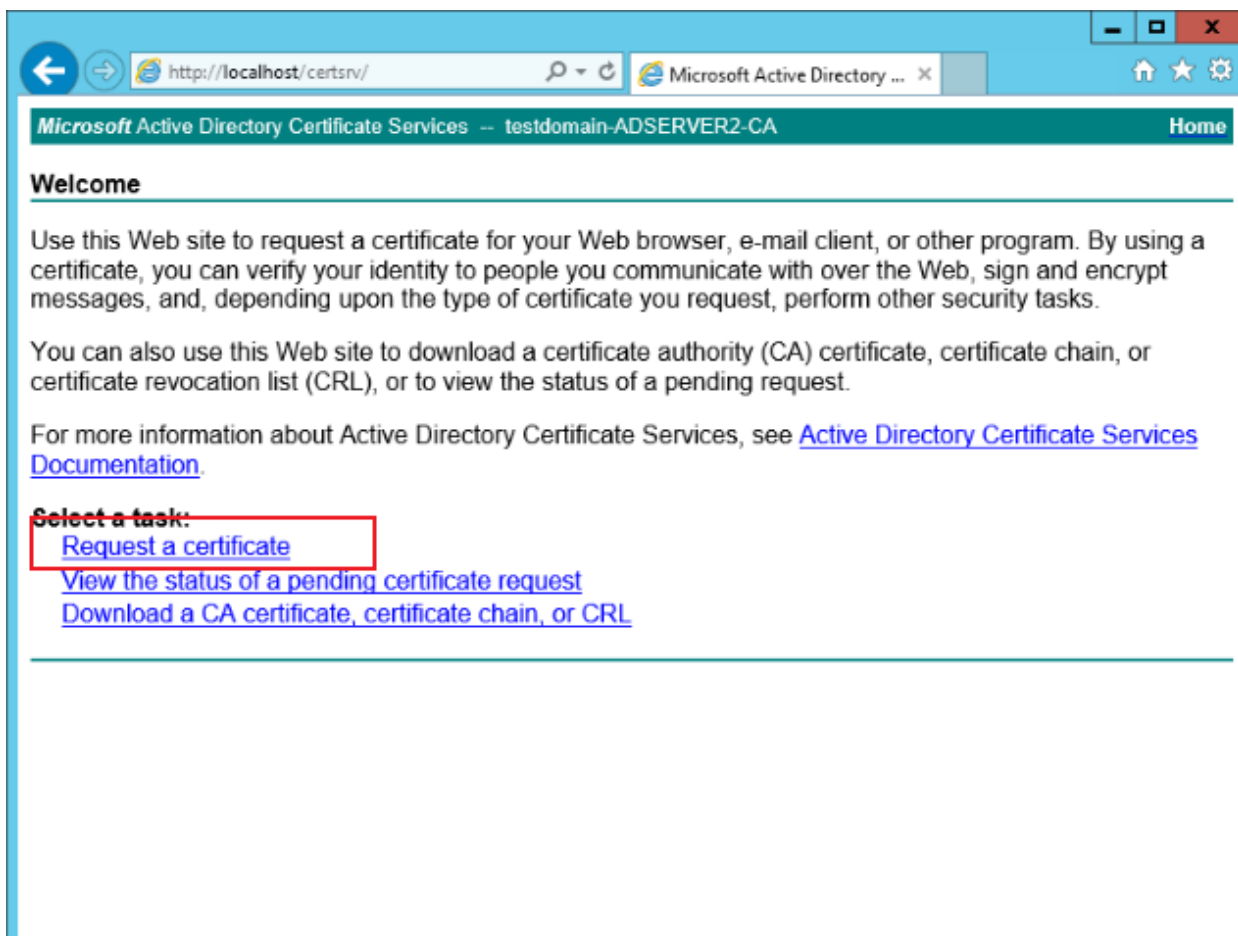


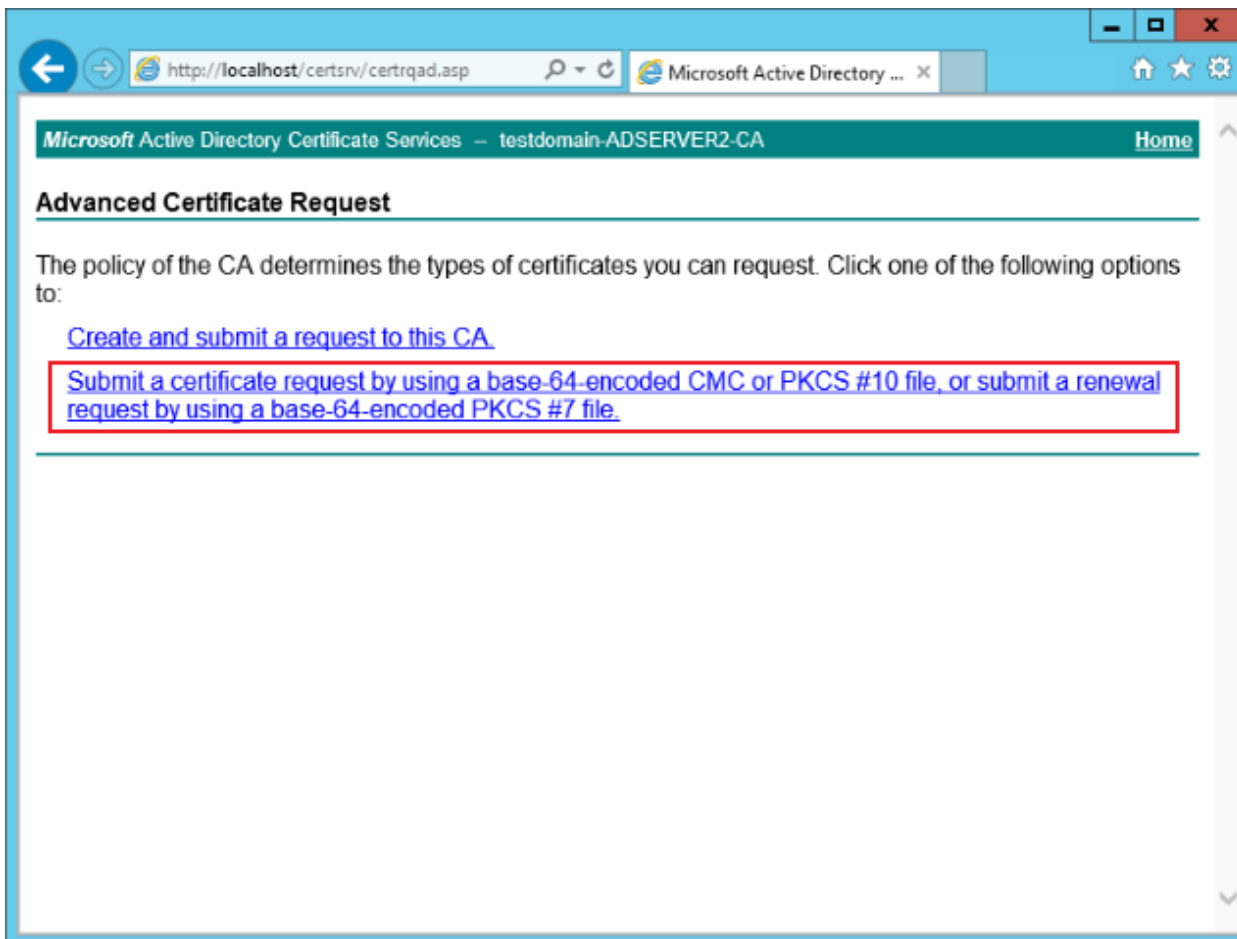




4. Visit <http://localhost/certsrv>, click **Request a certificate > advanced certificate request > using a base-64-encoded**, as shown below:







5. In the pop-up submit certificate request page, copy the contents of the certificate file saved in the certificate request (adserver2.txt in the step) and add it to the following input box. Select the Web

server as the certificate template and click **Submit**. As shown in the figure below:

Microsoft Active Directory Certificate Services -- testdomain-ADSERVER2-CA Home

### Submit a Certificate Request or Renewal Request

To submit a saved request to the CA, paste a base-64-encoded CMC or PKCS #10 certificate request or PKCS #7 renewal request generated by an external source (such as a Web server) in the Saved Request box.

**Saved Request:**

Base-64-encoded certificate request (CMC or PKCS #10 or PKCS #7):

```
vbIvdY9B7pr2NF8uOrIwDQYJKoZIhvcNAQEFBQAD^  
qVBlaE7IF21Na1KhRVEIKviPbXZh8vywp3EnZOK7;  
o/Huatcv2RKtTgUfXZazvyNkSCOQ1LiFguyGzGez'  
PP111DDn5s3XAtnNrMdkasI=  
-----END NEW CERTIFICATE REQUEST-----
```

**Certificate Template:**

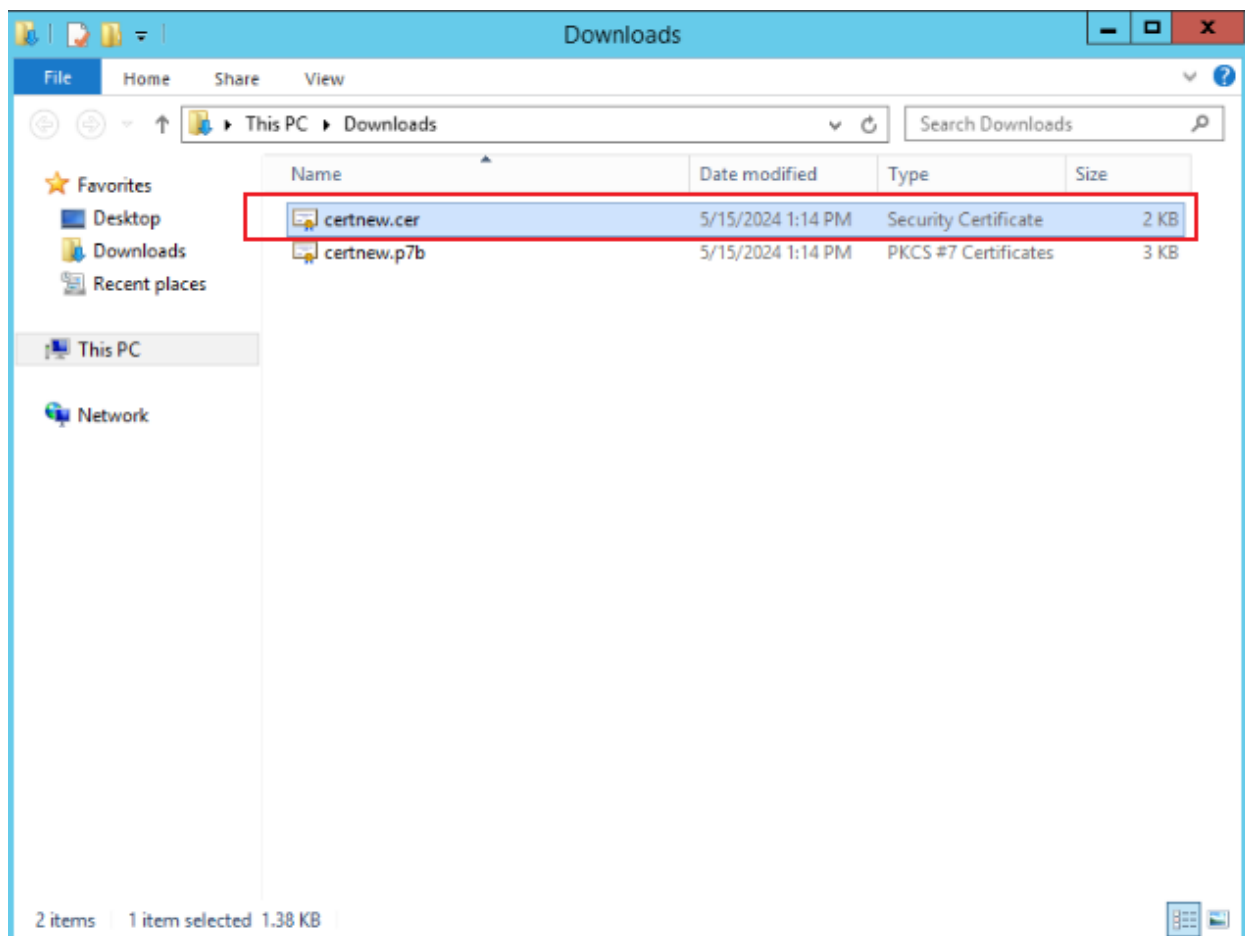
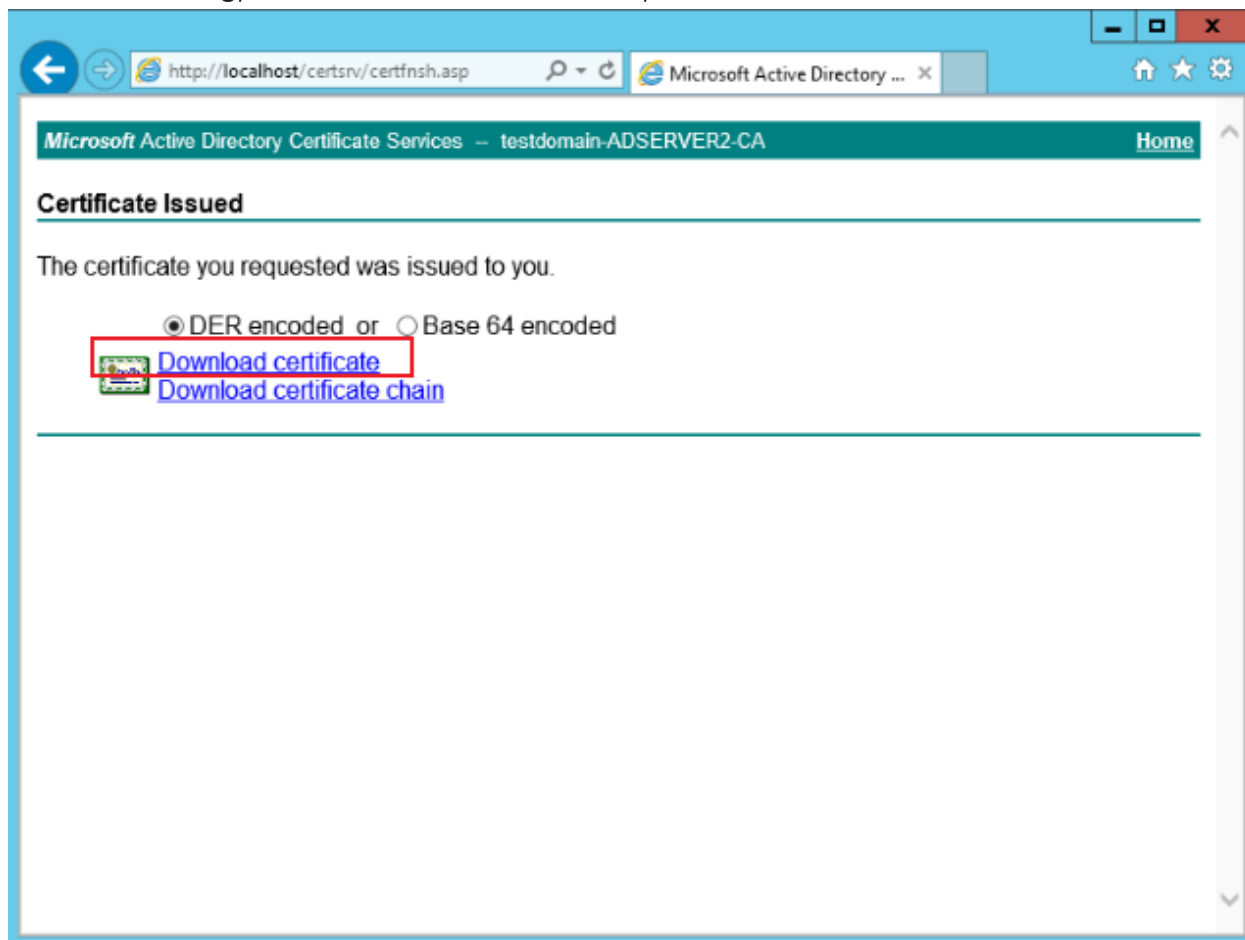
Web Server

**Additional Attributes:**

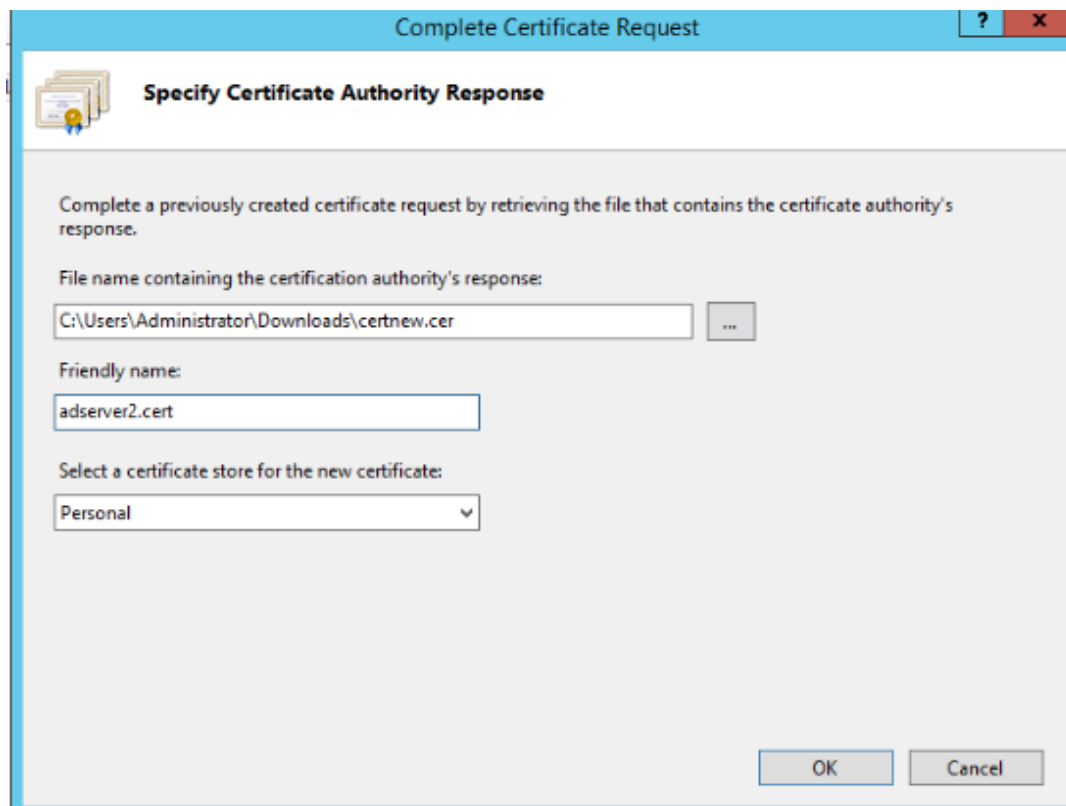
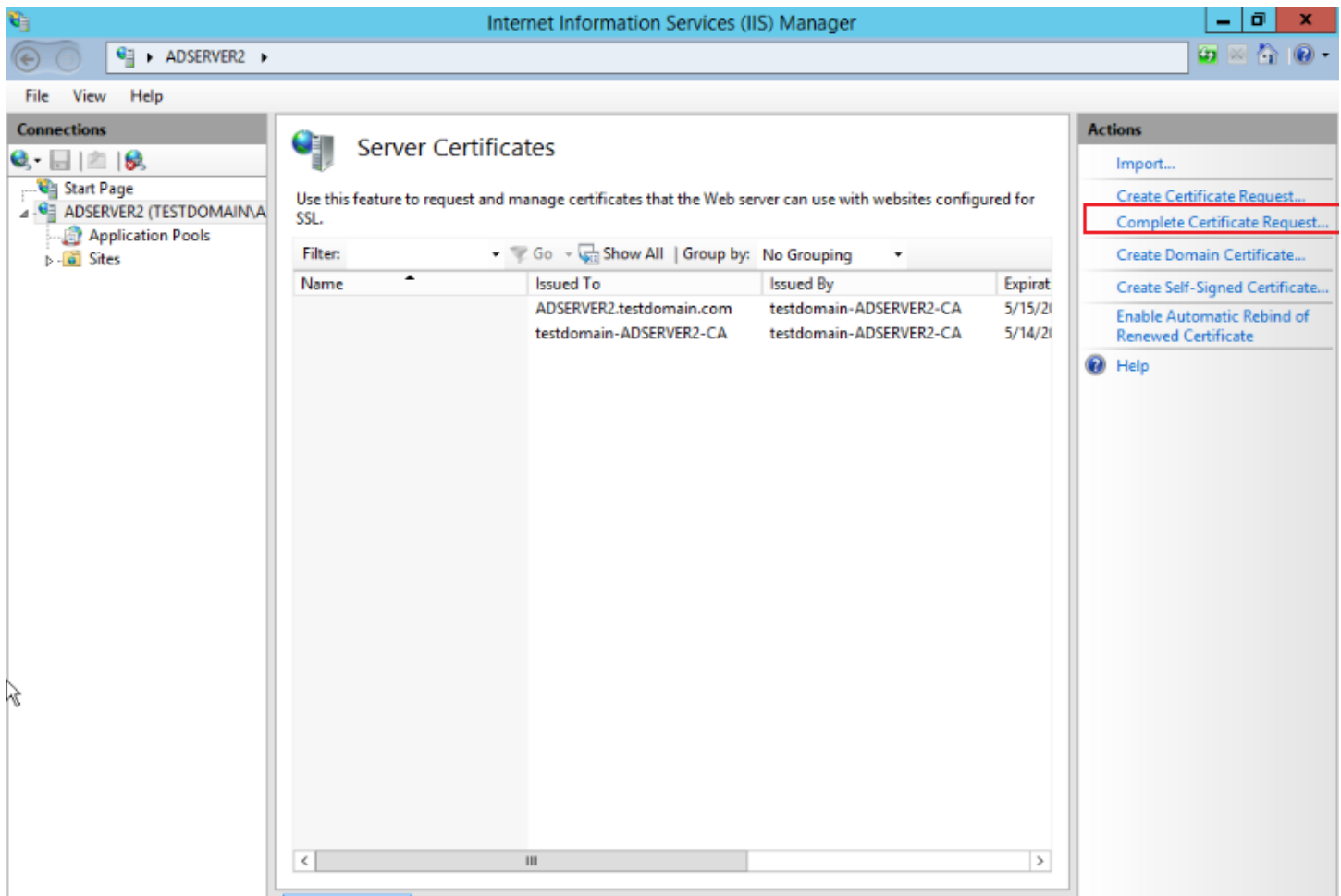
Attributes:

Submit >

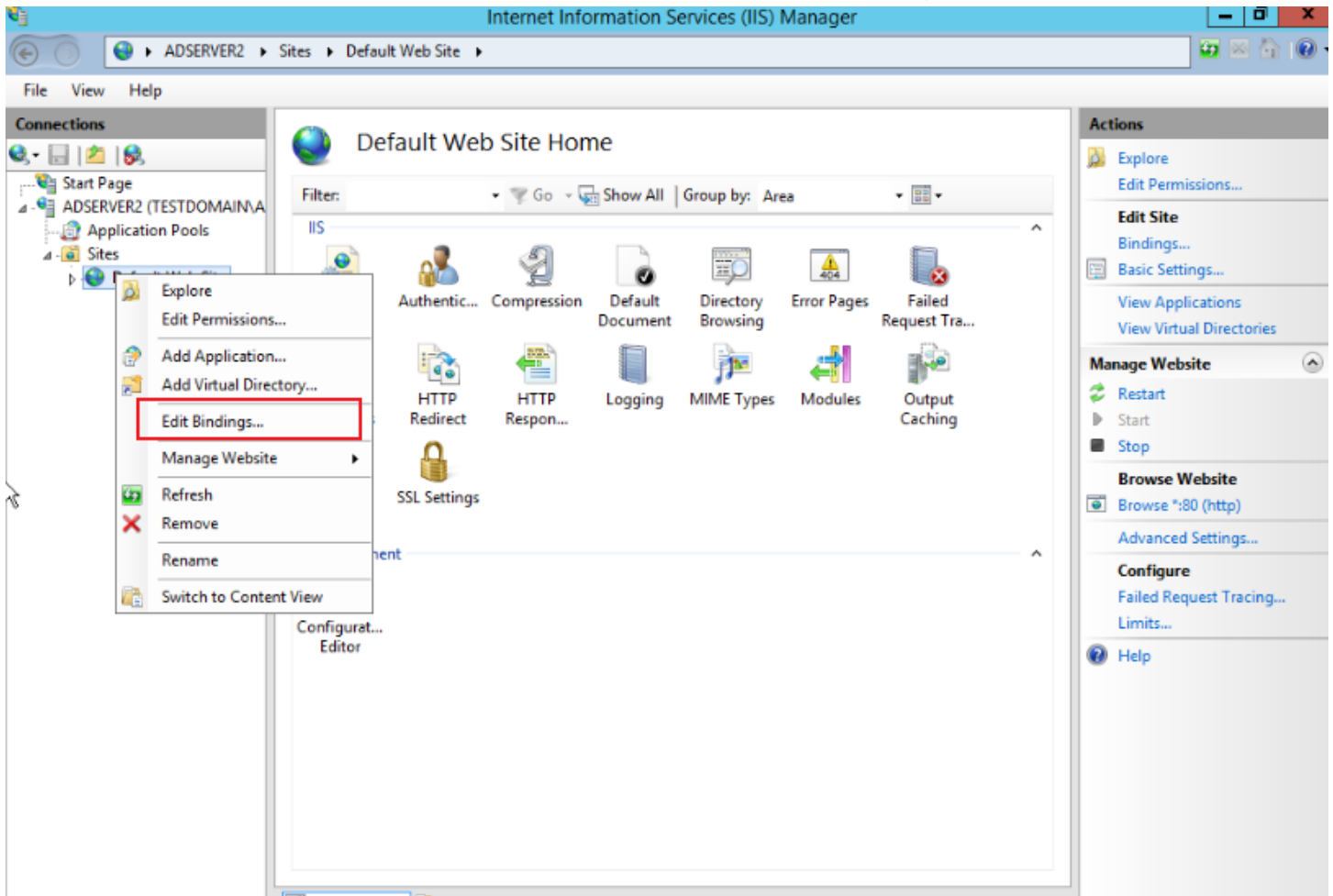
6. After submitting, click **Download certificate**, as shown below:



7. On the server certificate page, click **Complete Certificate Request**, and select the certificate downloaded in step 6 on the pop-up page, as shown below:

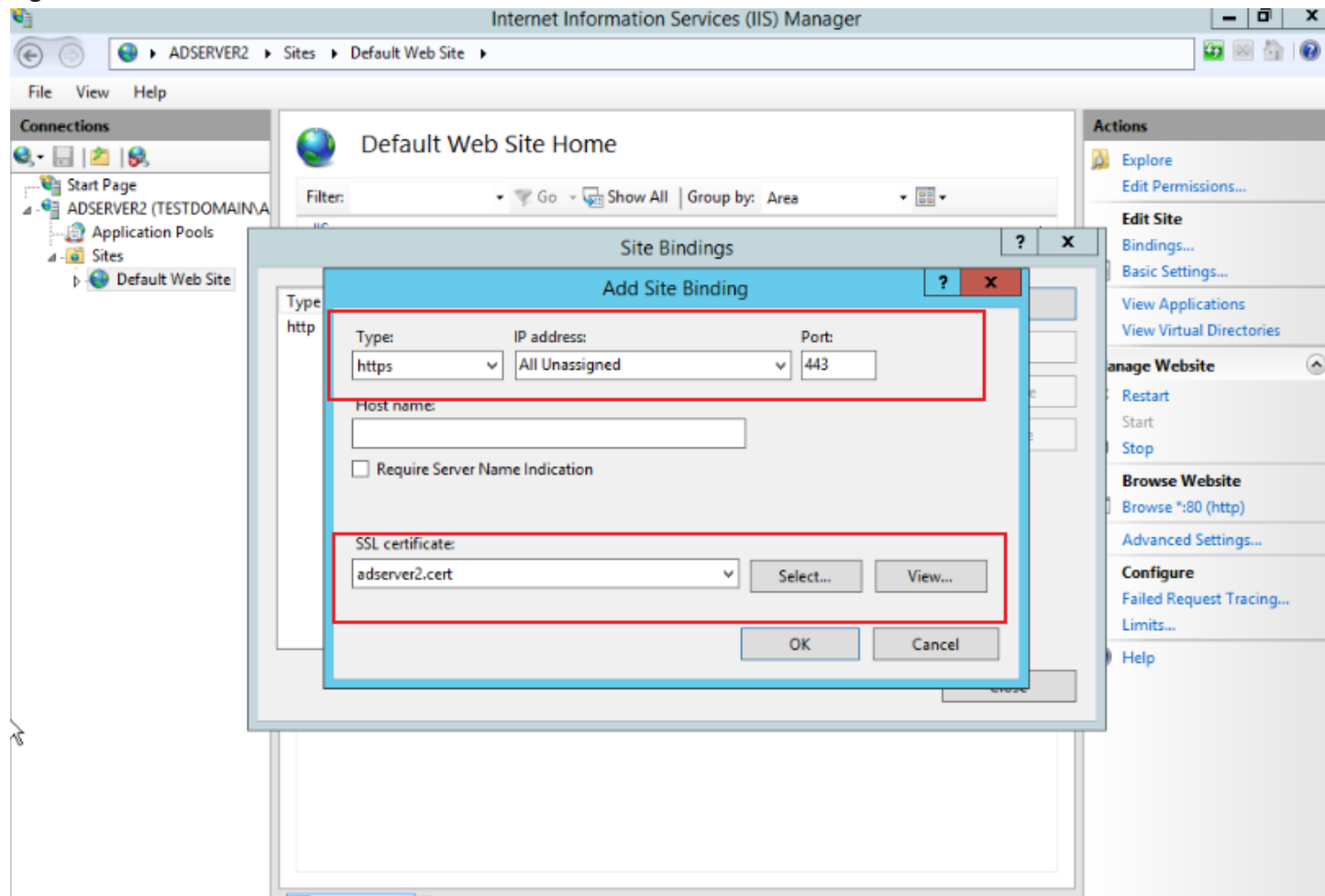


8. On the Website > Default Web Site Home page, right-click **Edit Bindings**, as shown below:



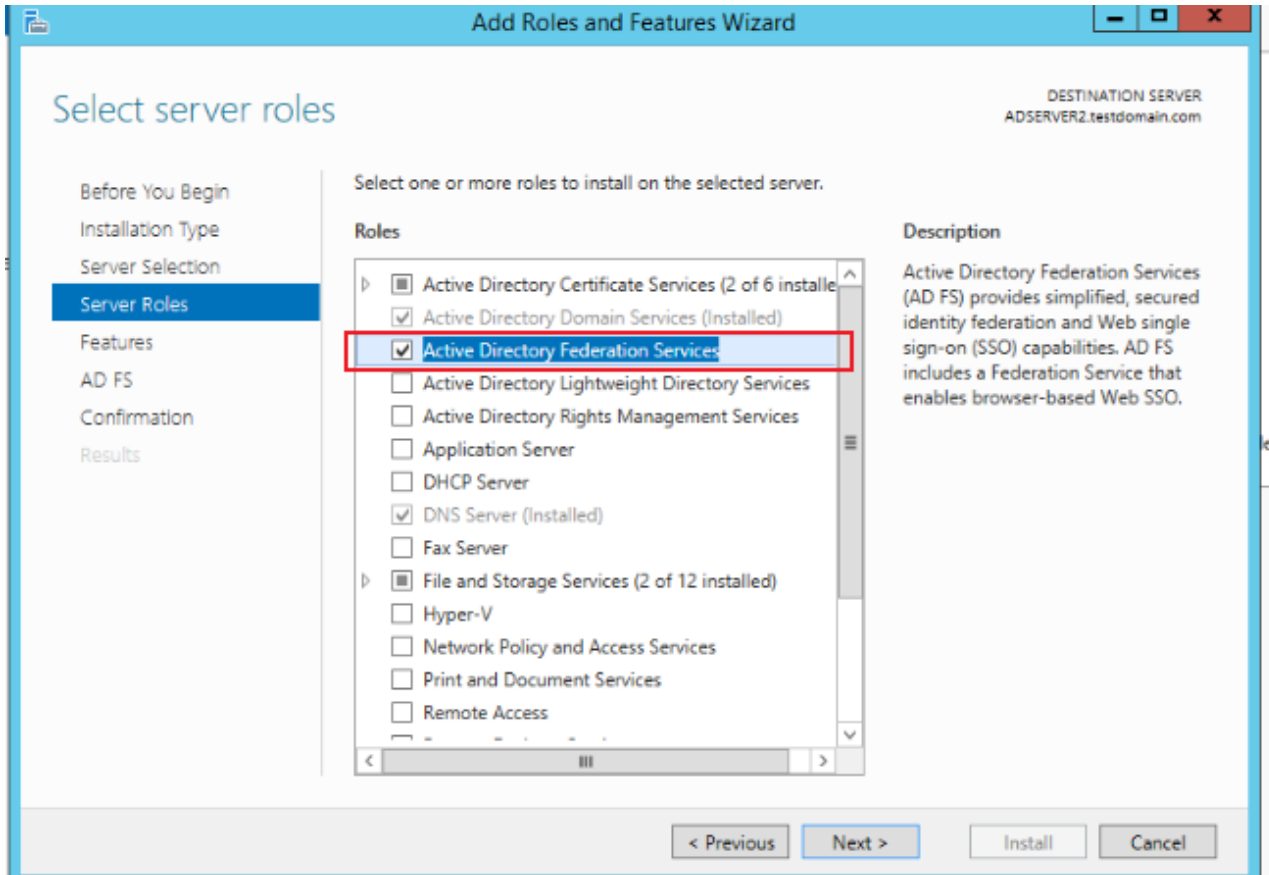
9. On the website binding page that pops up, click **Add**, select the type as https, the IP address as all unassigned, the port as 443, and the SSL certificate as adserver2.cert, as shown in the following

figure:

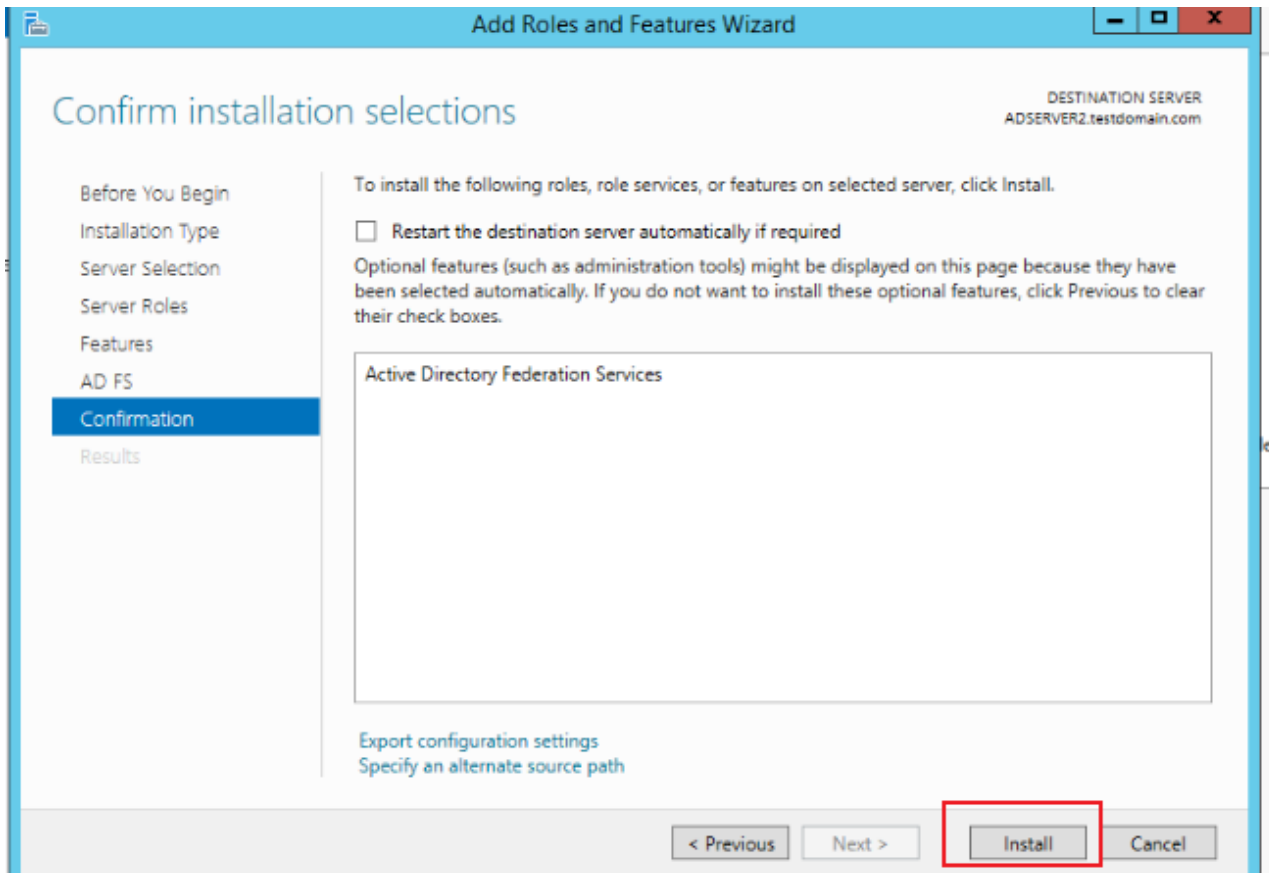


10. In the server, go to **Server Manager** > **Dashboard**, click **Add roles and features**, and keep clicking **Next** according to the default selection.

11. Go to the Select server roles page, check Active Directory Federation Services, and click **Next**

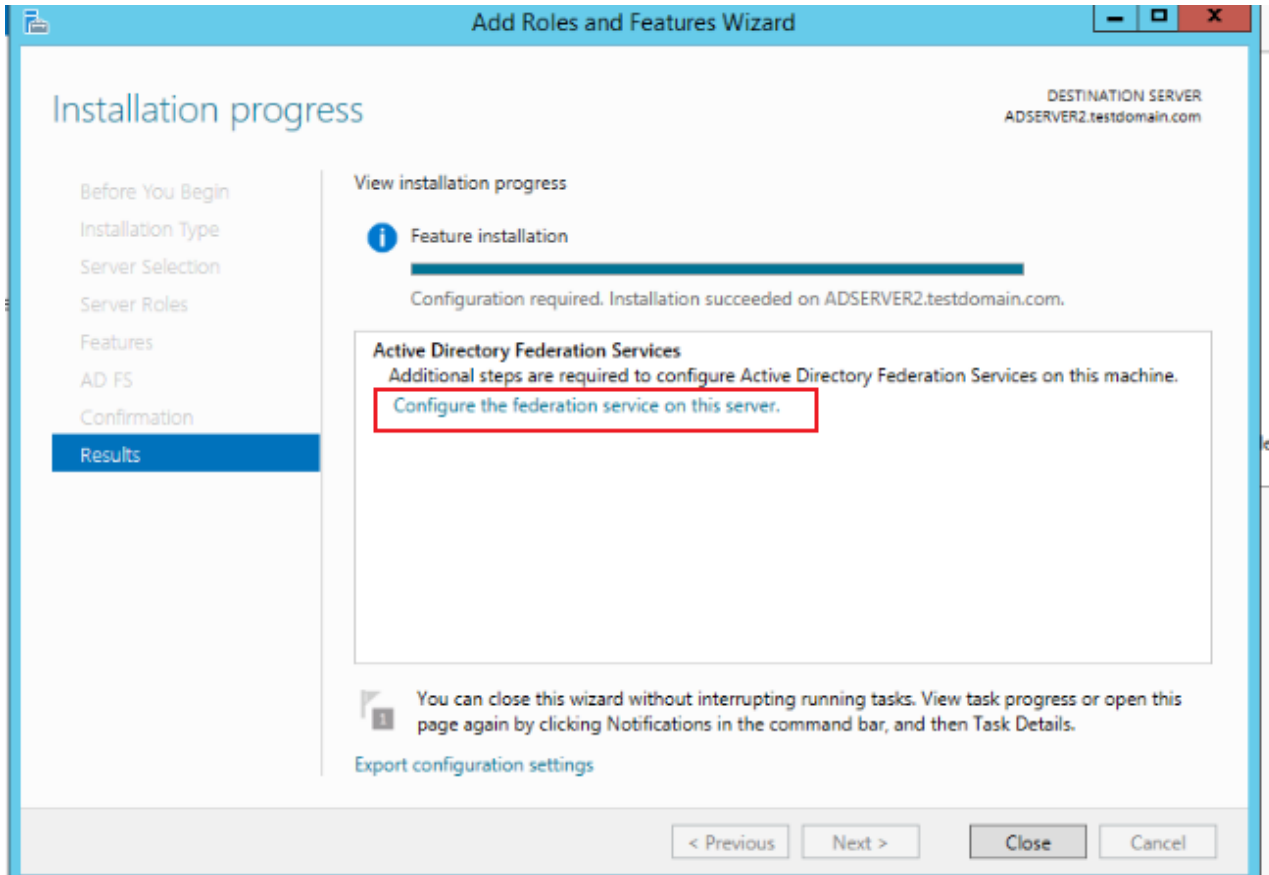


12. On the pop-up wizard page, click **Install**.

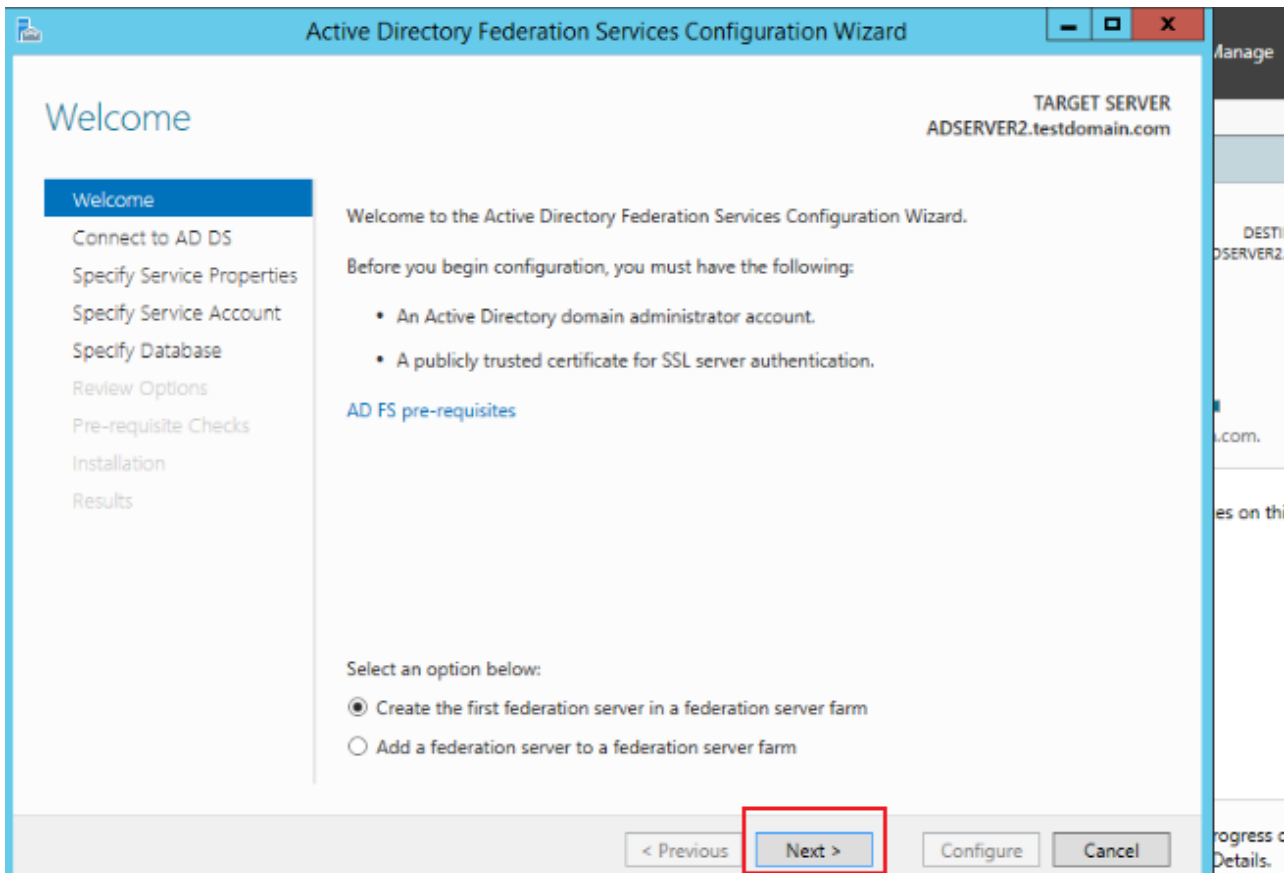


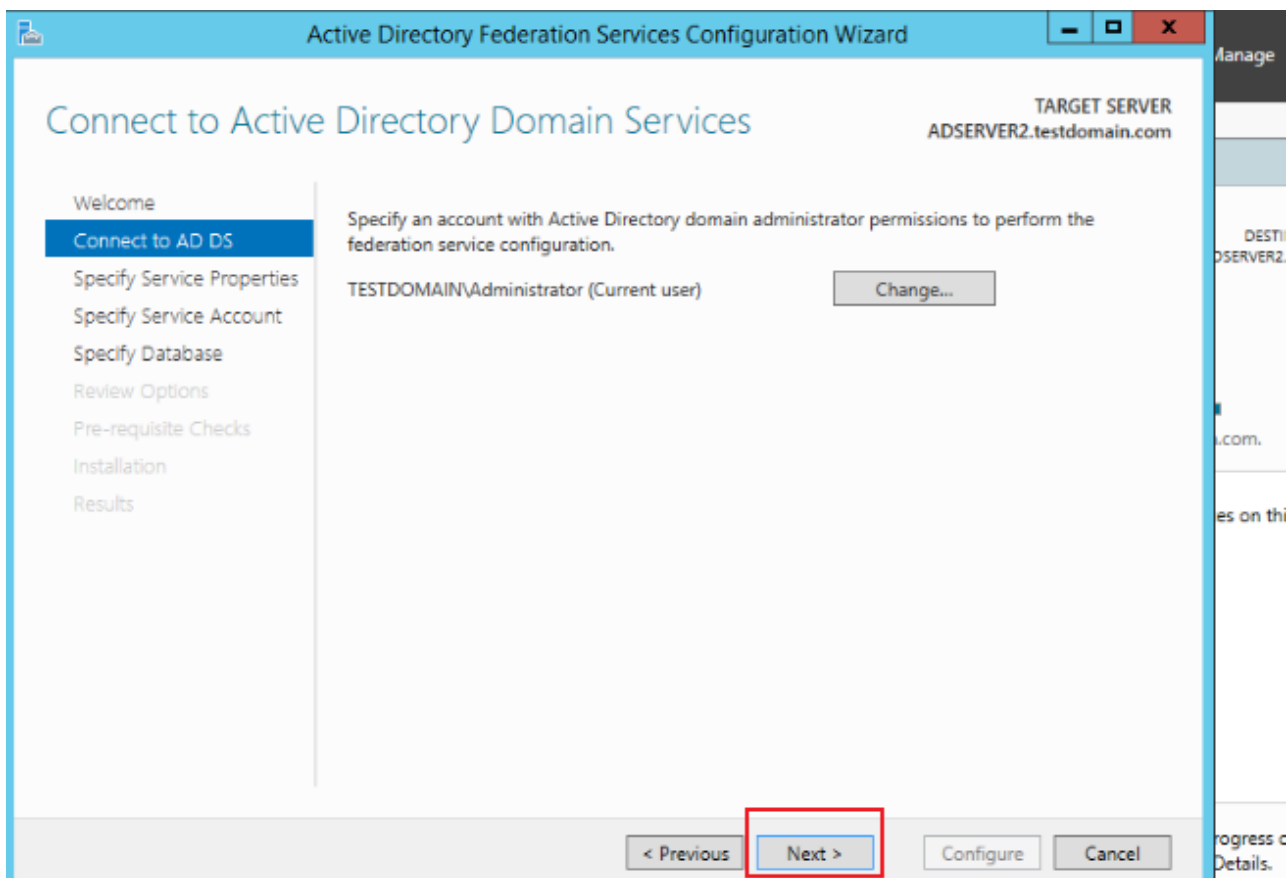


13. On the installation progress page, click **Configure the federation service on this server**.

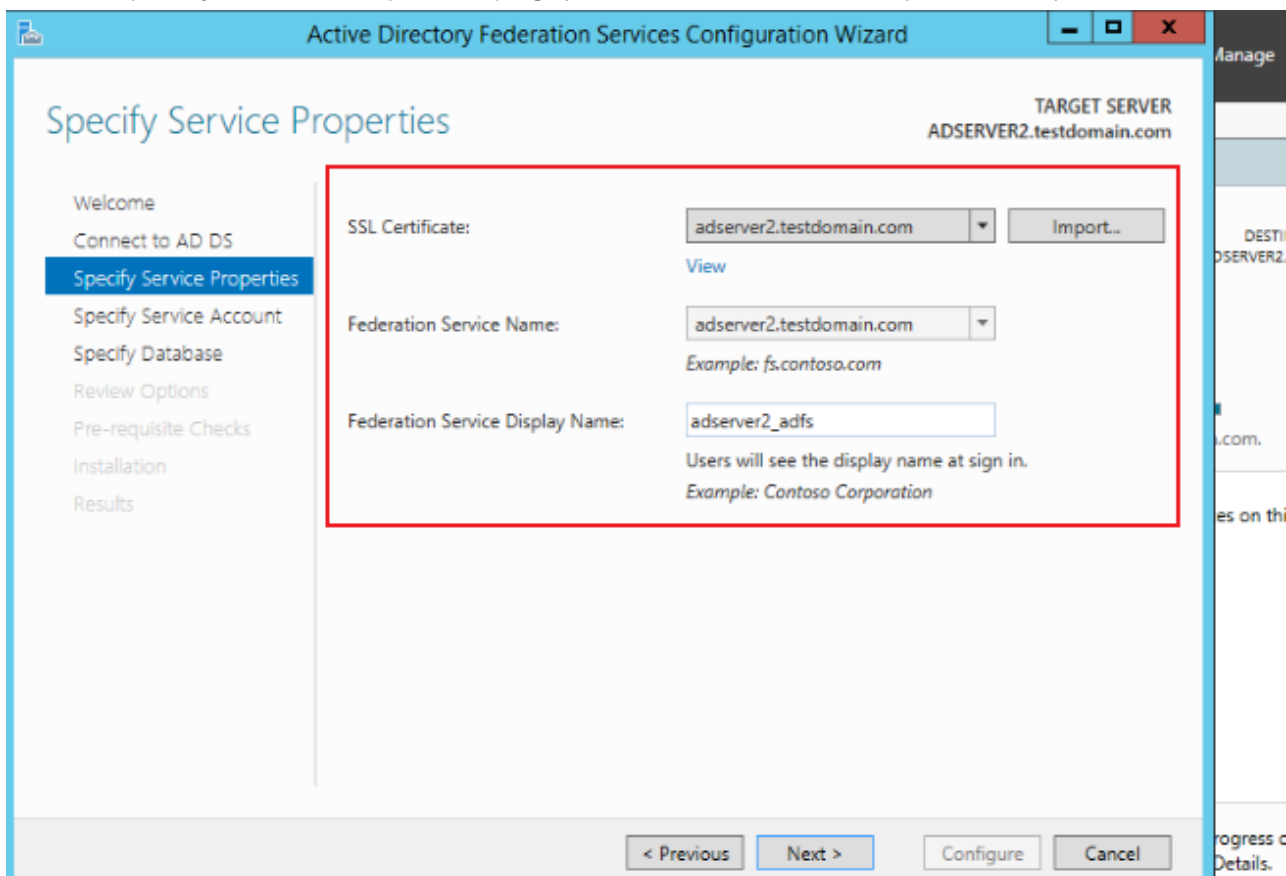


14. On the pop-up wizard page, click **Next**.





15. On the Specify Service Properties page, select and fill in the required data, and click **Next**.



16. On the Specify Service Account page, choose to use an existing domain user account or group managed service account, click **Select**.

The screenshot shows the 'Specify Service Account' page of the 'Active Directory Federation Services Configuration Wizard'. The window title is 'Active Directory Federation Services Configuration Wizard'. The page title is 'Specify Service Account'. In the top right corner, it says 'TARGET SERVER: ADSERVER2.testdomain.com'. A yellow warning banner at the top states: 'Group Managed Service Accounts are not available because the KDS Root Key has not been set. Use the foll... Show more X'. On the left, a navigation pane lists steps: 'Welcome', 'Connect to AD DS', 'Specify Service Properties', 'Specify Service Account' (highlighted), 'Specify Database', 'Review Options', 'Pre-requisite Checks', 'Installation', and 'Results'. The main area is titled 'Specify a domain user account or group Managed Service Account.' and contains two radio buttons: 'Create a Group Managed Service Account' (unselected) and 'Use an existing domain user account or group Managed Service Account' (selected). Below the selected option, the 'Account Name:' field shows 'TESTDOMAIN\' followed by a text box containing 'ADSERVER2\'. Below this, the 'Account Name:' field shows '<Not provided>' with a red asterisk. To the right of this field is a 'Select...' button, which is highlighted with a red rectangle. At the bottom, there are four buttons: '< Previous', 'Next >', 'Configure', and 'Cancel'.

Active Directory Federation Services Configuration Wizard

Specify Service Account

TARGET SERVER  
ADSERVER2.testdomain.com

Group Managed Service Accounts are not available because the KDS Root Key has not been set. Use the foll... Show more X

Welcome  
Connect to AD DS  
Specify Service Properties  
Specify Service Account  
Specify Database  
Review Options  
Pre-requisite Checks  
Installation  
Results

Specify a domain user account or group Managed Service Account.

☐ Create a Group Managed Service Account

Account Name: TESTDOMAIN\

☒ Use an existing domain user account or group Managed Service Account

Account Name: \* <Not provided> Select...

< Previous Next > Configure Cancel

Active Directory Federation Services Configuration Wizard

### Select User or Service Account

Select this object type:

User or Service Account

From this location:

testdomain.com

Enter the object name to select (examples):

Administrator

Advanced...

OK

Cancel

TARGET SERVER

ADSERVER2.testdomain.com

has not been set. Use the foll... [Show more](#)

d Service Account.

MAIN\

o Managed Service Account

provided>

Select...

< Previous Next > Configure Cancel

Active Directory Federation Services Configuration Wizard

## Specify Service Account

TARGET SERVER

ADSERVER2.testdomain.com

Group Managed Service Accounts are not available because the KDS Root Key has not been set. Use the foll... [Show more](#)

Welcome

Connect to AD DS

Specify Service Properties

**Specify Service Account**

Specify Database

Review Options

Pre-requisite Checks

Installation

Results

Specify a domain user account or group Managed Service Account.

☐ Create a Group Managed Service Account

Account Name: TESTDOMAIN\

☒ Use an existing domain user account or group Managed Service Account

Account Name: TESTDOMAIN\Adm... Clear Select...

Account Password: .....

< Previous Next > Configure Cancel

17. Specify database, and click **Next**.

Active Directory Federation Services Configuration Wizard

Specify Configuration Database

TARGET SERVER  
ADSERVER2.testdomain.com

Welcome  
Connect to AD DS  
Specify Service Properties  
Specify Service Account  
**Specify Database**  
Review Options  
Pre-requisite Checks  
Installation  
Results

Specify a database to store the Active Directory Federation Service configuration data.

☒ Create a database on this server using Windows Internal Database.  
☐ Specify the location of a SQL Server database.

Database Host Name:   
Database Instance:   
*To use the default instance, leave this field blank.*

< Previous **Next >** Configure Cancel

Active Directory Federation Services Configuration Wizard

Results

TARGET SERVER  
ADSERVER2.testdomain.com

✓ This server was successfully configured [Show more](#) ✕

Welcome  
Connect to AD DS  
Specify Service Properties  
Specify Service Account  
Specify Database  
Review Options  
Pre-requisite Checks  
Installation  
**Results**

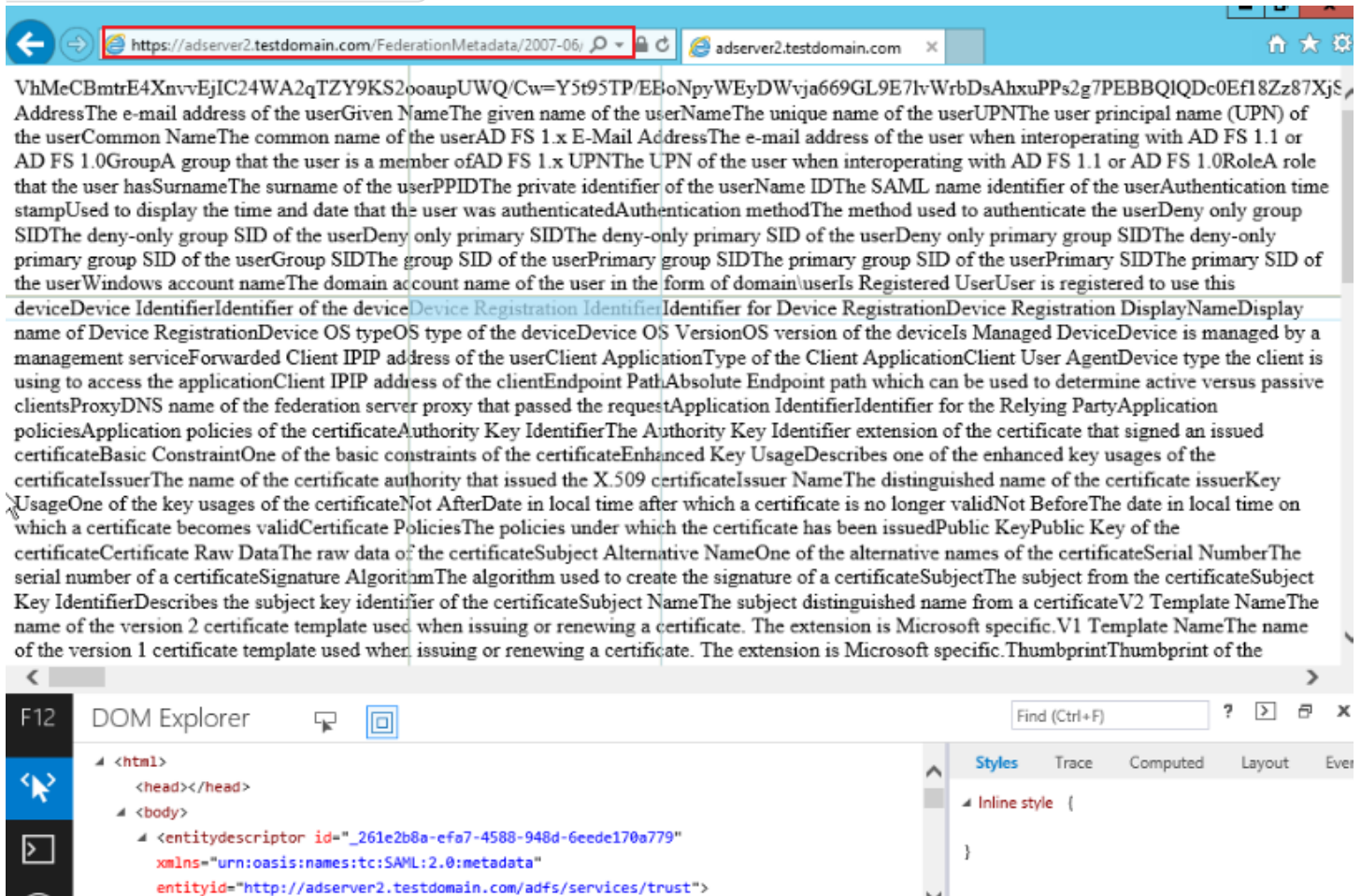
^ View detailed operation results

⚠ An error occurred during an attempt to set the SPN for the specified service account. Set the SPN for the service account manually. For more information about setting the SPN of the service account manually, see the AD FS Deployment Guide. Error message: The SPN required for this Federation Service is already set on another Active Directory account. Choose a different Federation Service name and try again.

Next steps required for completing your federation service deployment

< Previous Next > **Close** Cancel

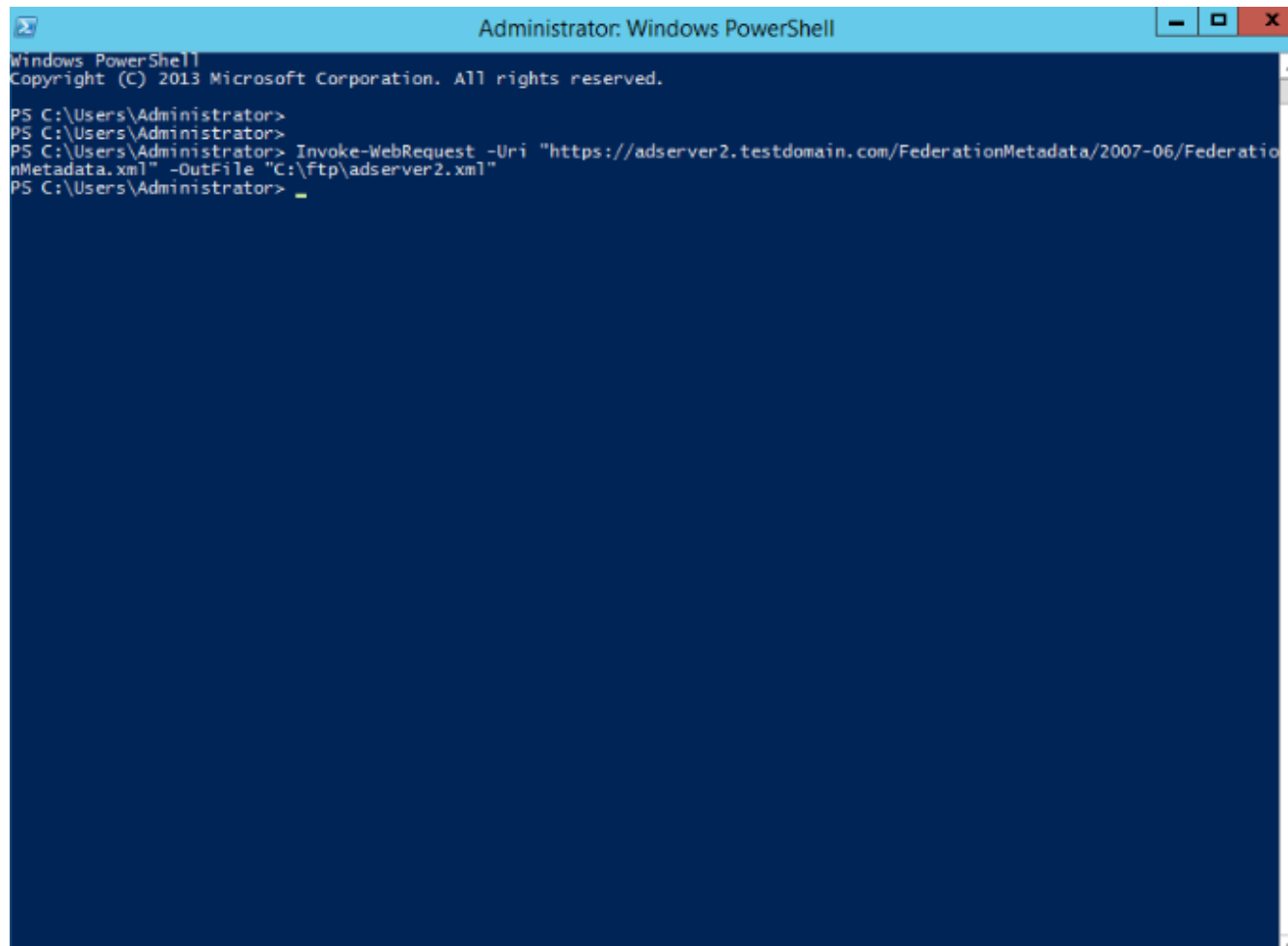
18. Visit <https://adserver2.testdomain.com/FederationMetadata/2007-06/FederationMetadata.xml> with the browser to check whether the installation is successful.



## User SSO configuration

1. Access <https://adserver2.testdomain.com/FederationMetadata/2007-06/FederationMetadata.xml> in the server browser and download the source data XML to the

local. Here, use the command line to export the data.

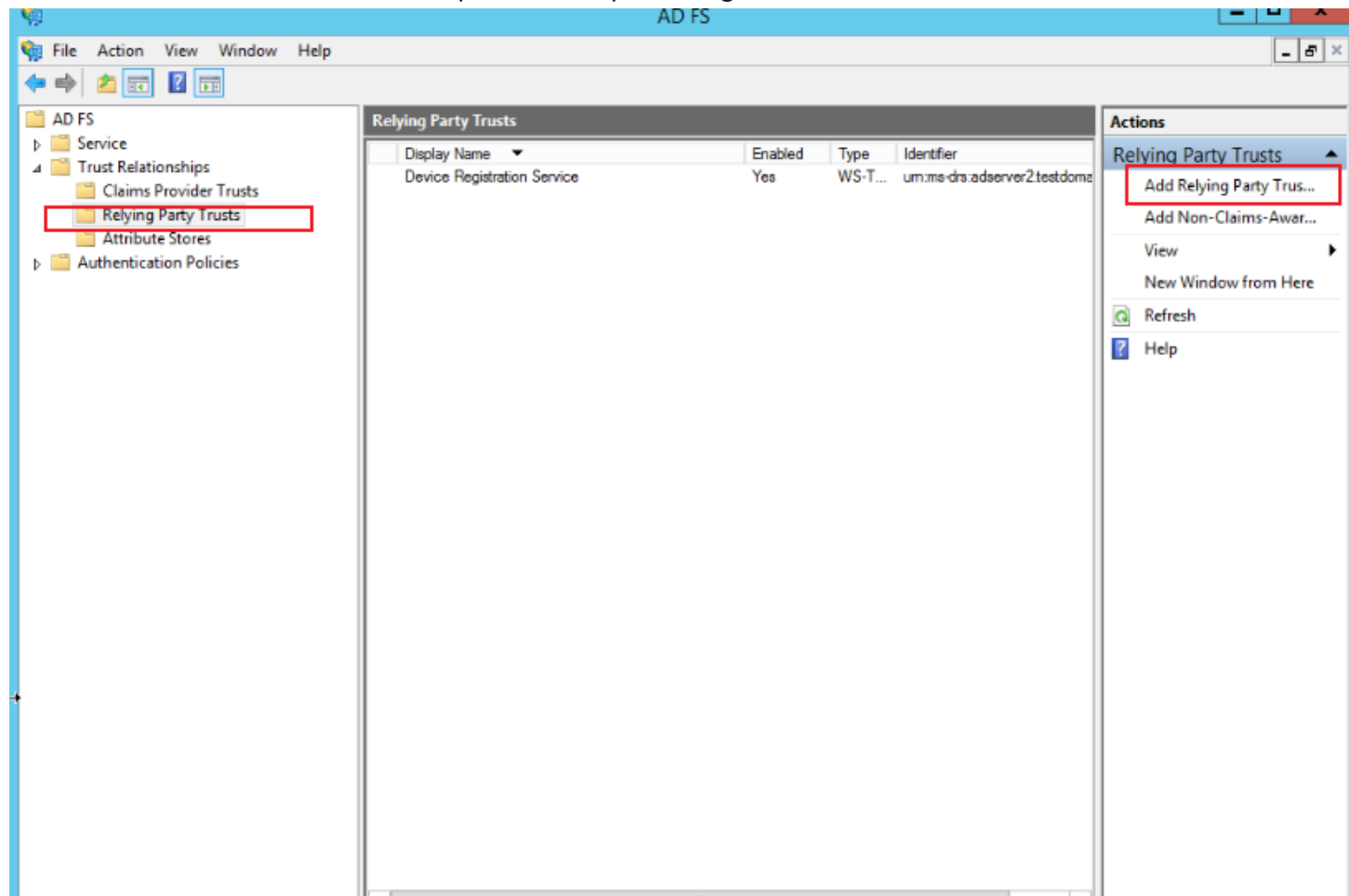


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator>
PS C:\Users\Administrator>
PS C:\Users\Administrator> Invoke-WebRequest -Uri "https://adserver2.testdomain.com/FederationMetadata/2007-06/FederationMetadata.xml" -OutFile "C:\ftp\adserver2.xml"
PS C:\Users\Administrator>
```

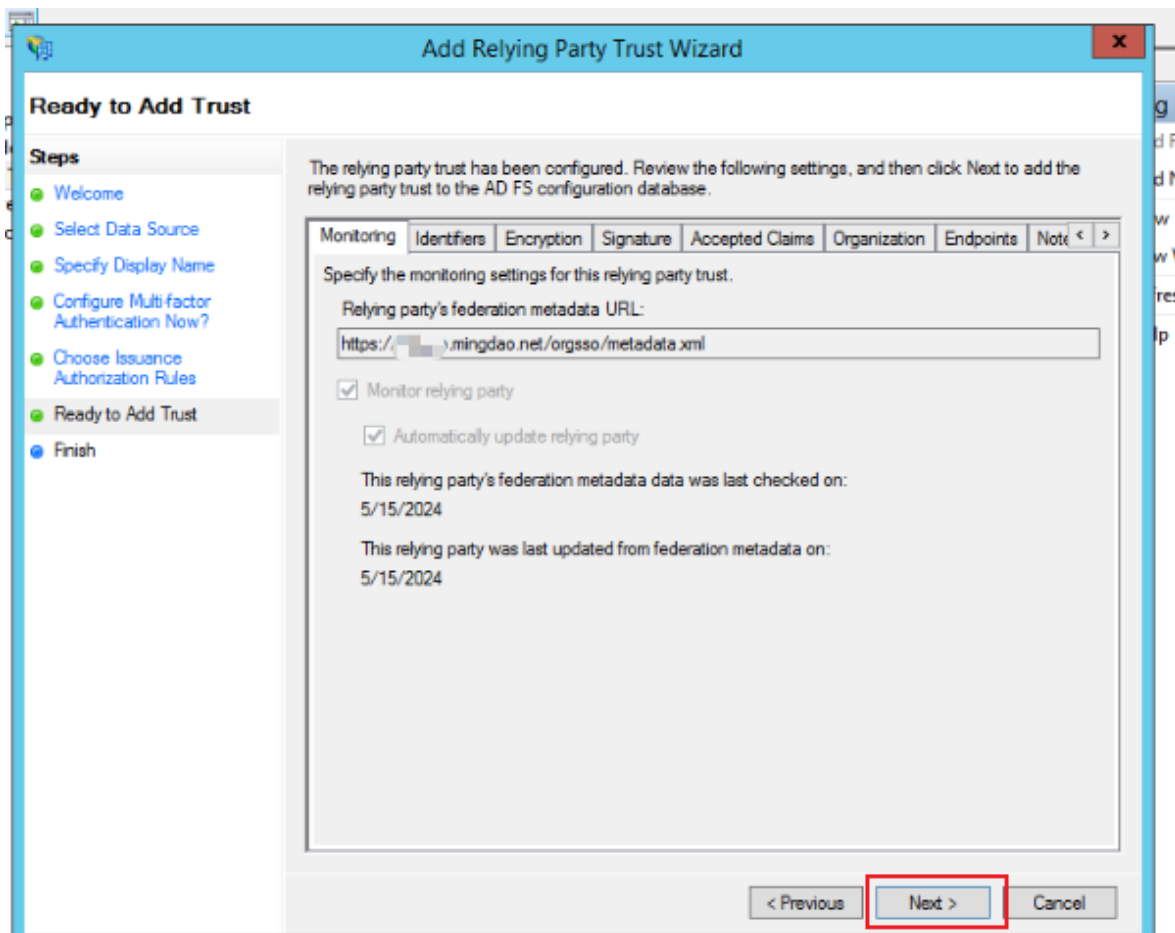
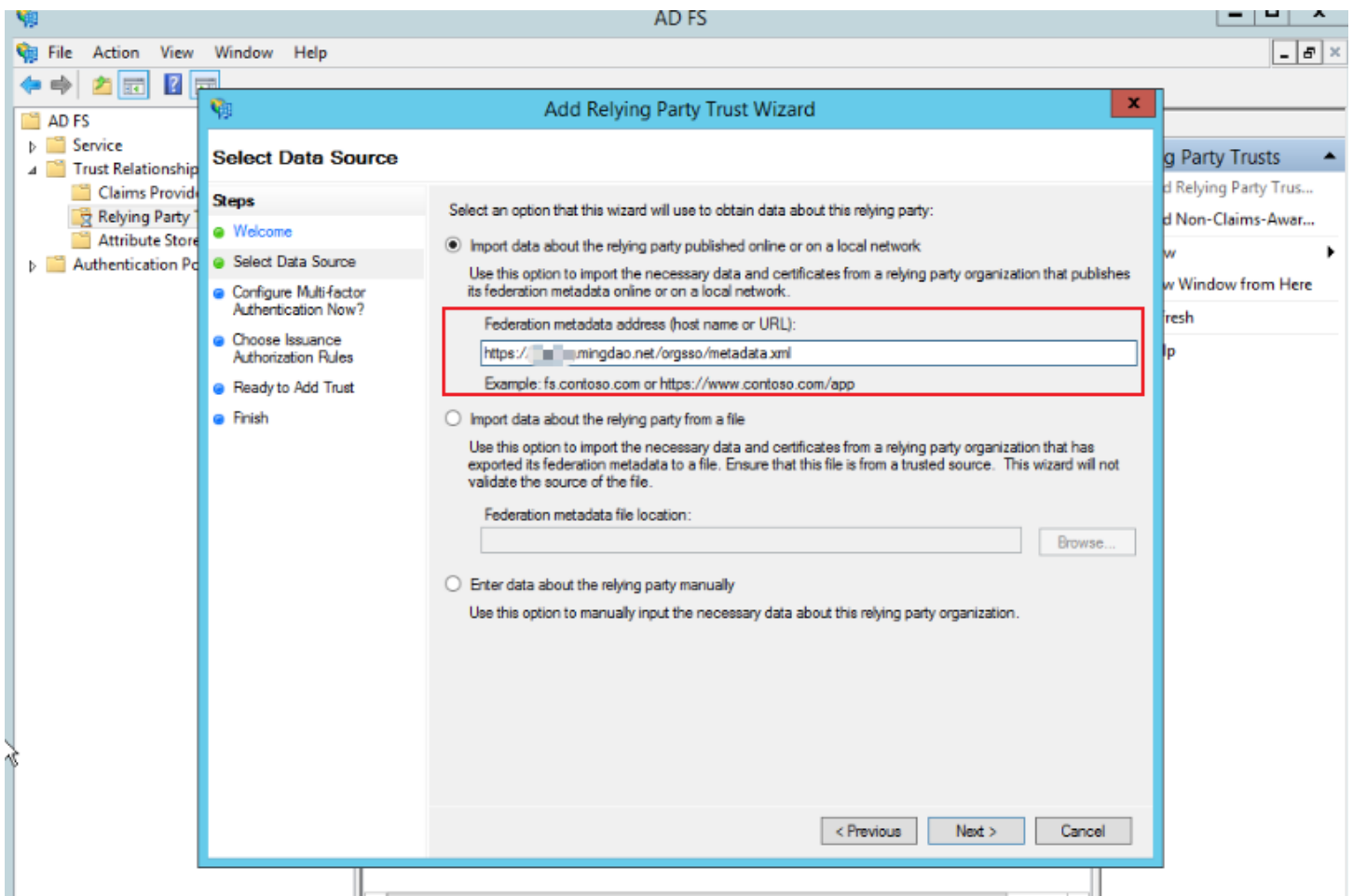
2. Enable [SAML2 docking configuration](#), configure and mount the XML file saved in step 1 according to the steps (named `idp.xml` in the docking document).
3. Enter the ADFS management page in the server, select **Trust Relationships > Relying Party Trusts**, right-click and select Add Relying Party Trust, click **Start**, and add the joint metadata address and

metadata address Take it from step 2 and keep clicking **Next** as shown below:

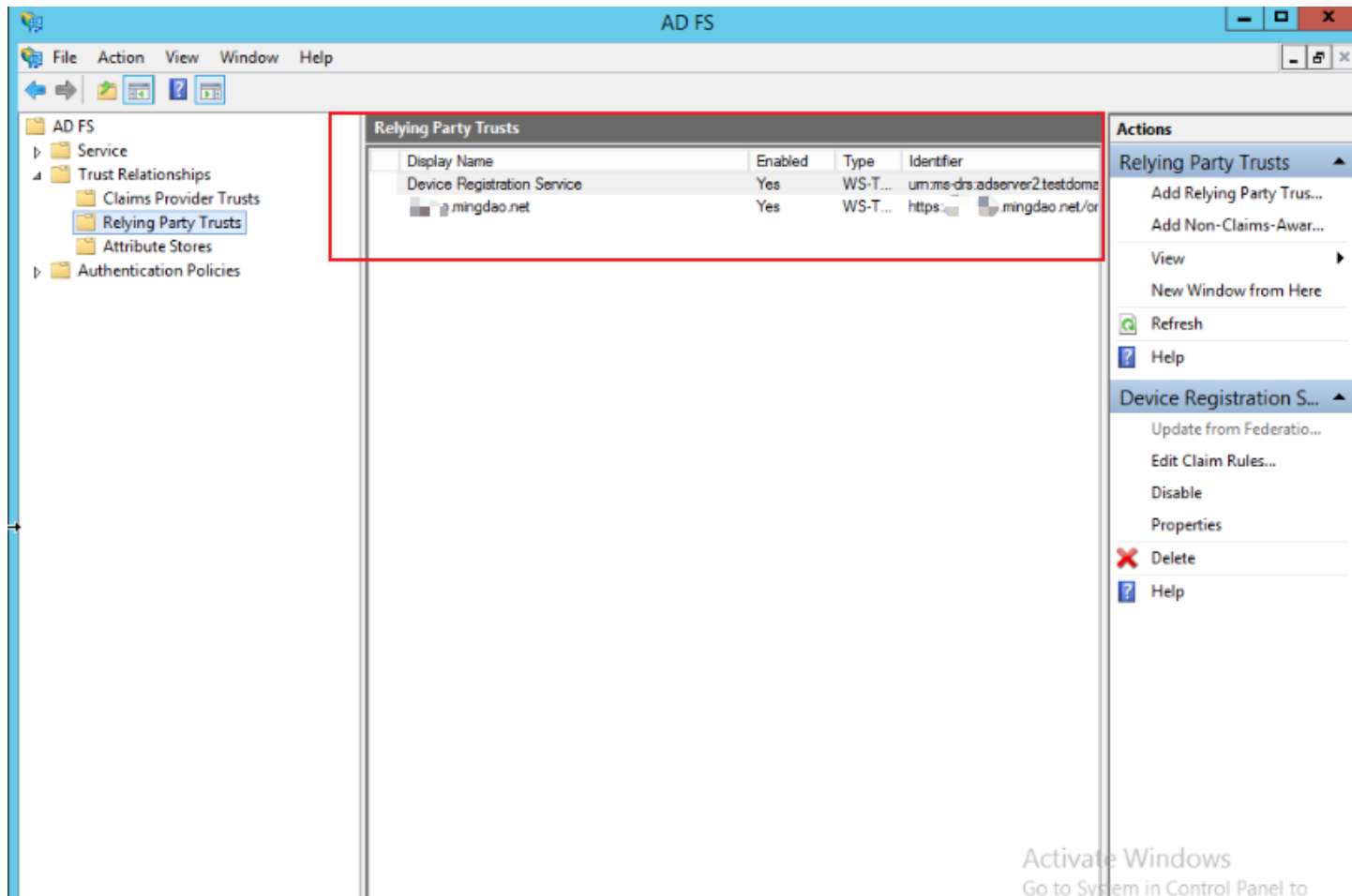


4. Here you choose to set the remote xml address. The HAP system is generally `{server}/orgsso/metadata.xml`.

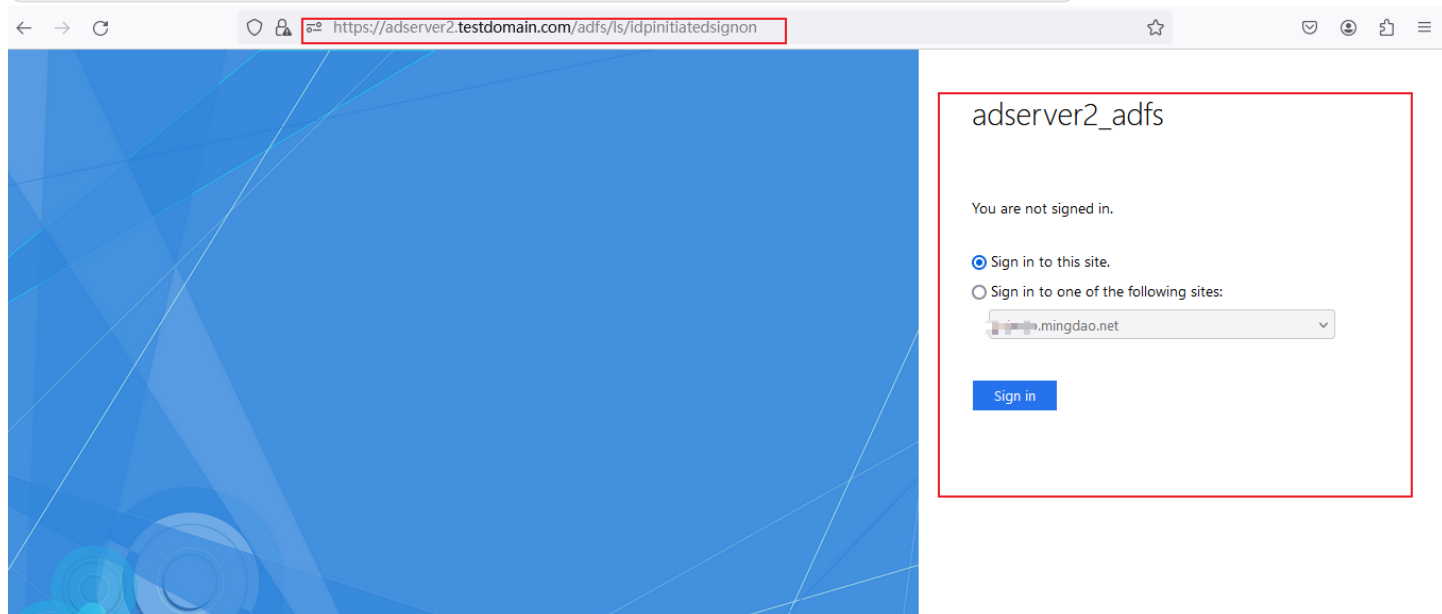




5. Keep clicking **Next**.

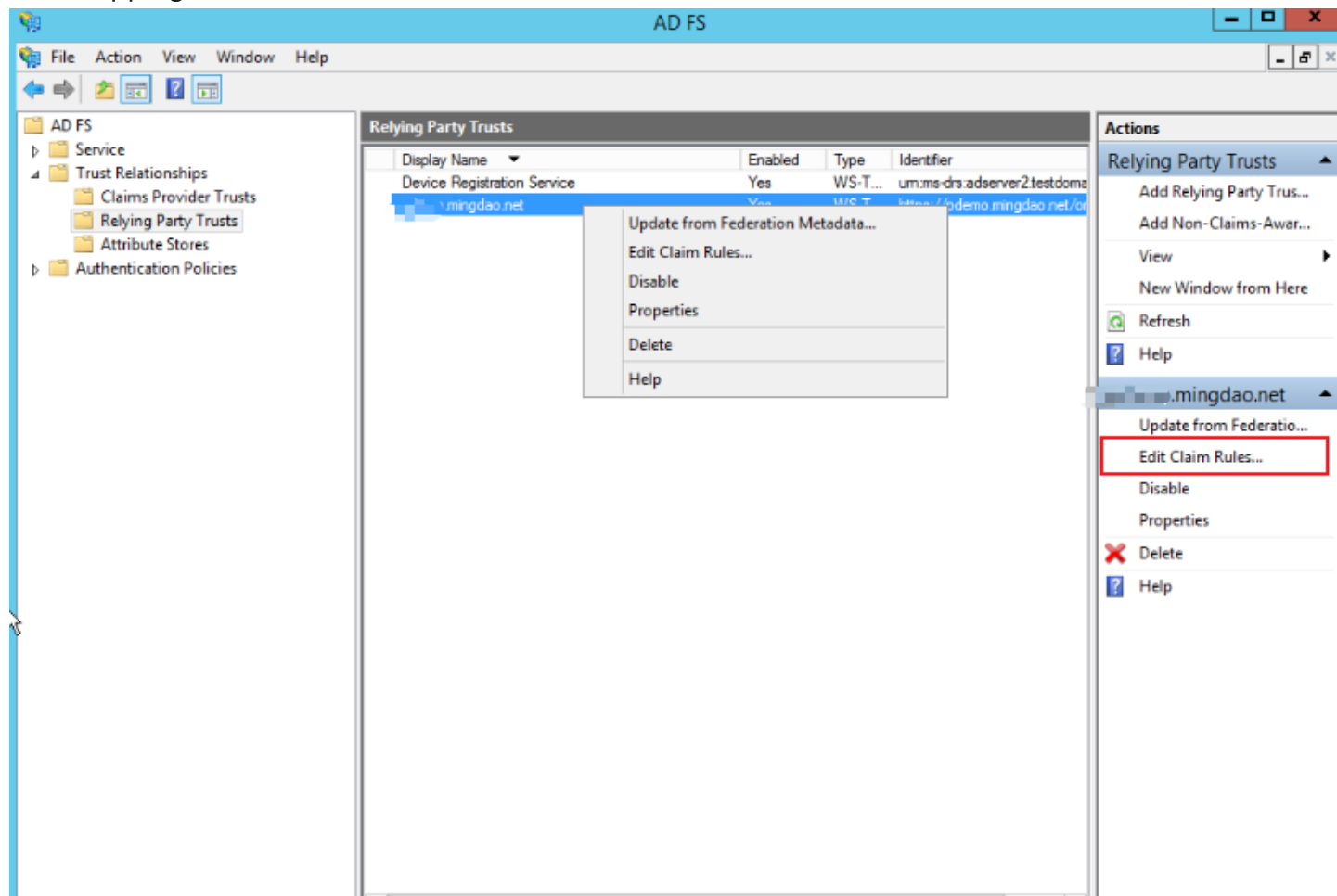


6. After the configuration is complete, you can visit <https://adserver2.testdomain.com/adfs/ls/idpinitiatedsignon> to view trusted sites.

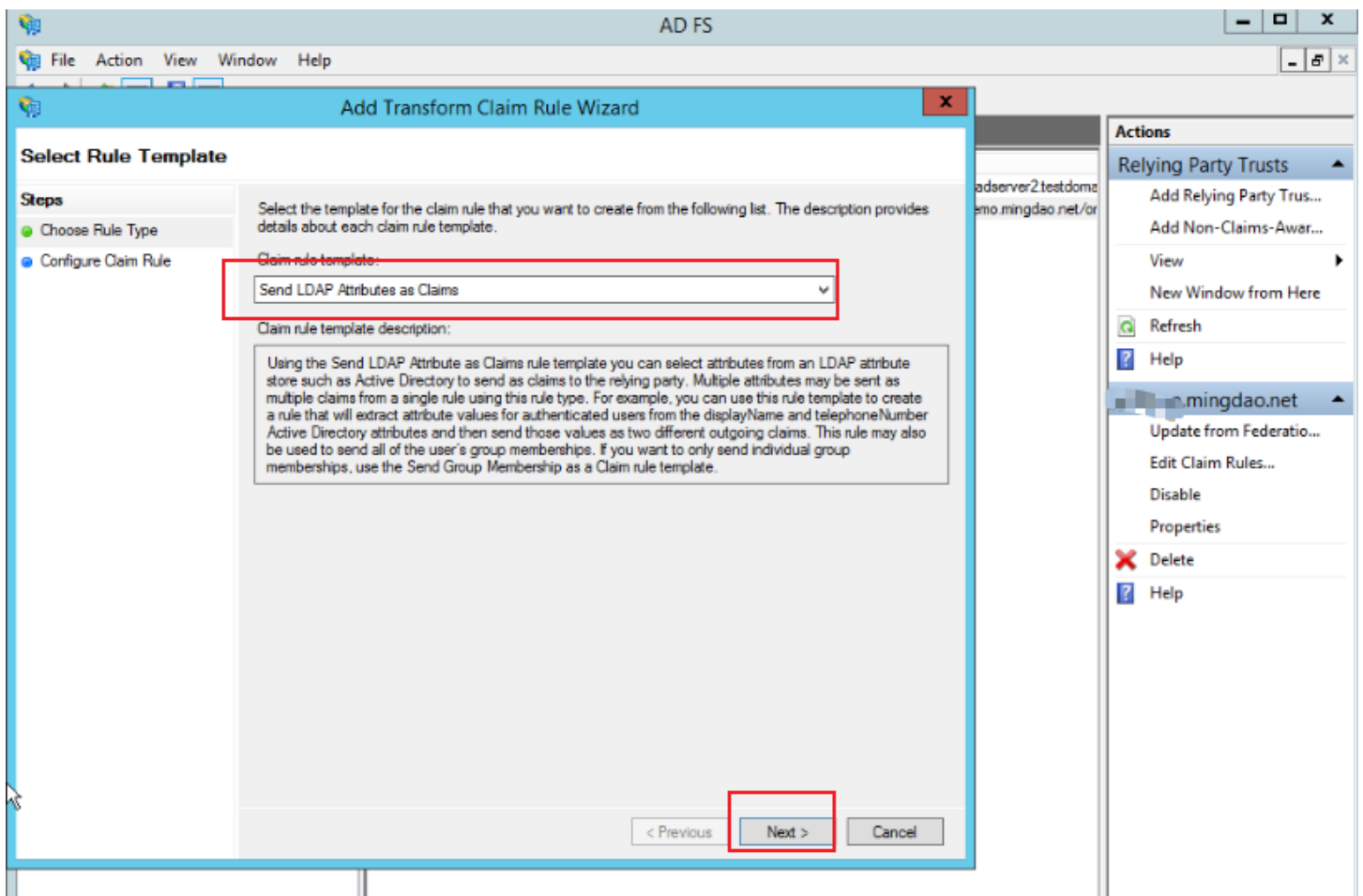


**Configure SAML assertion attributes for SP**

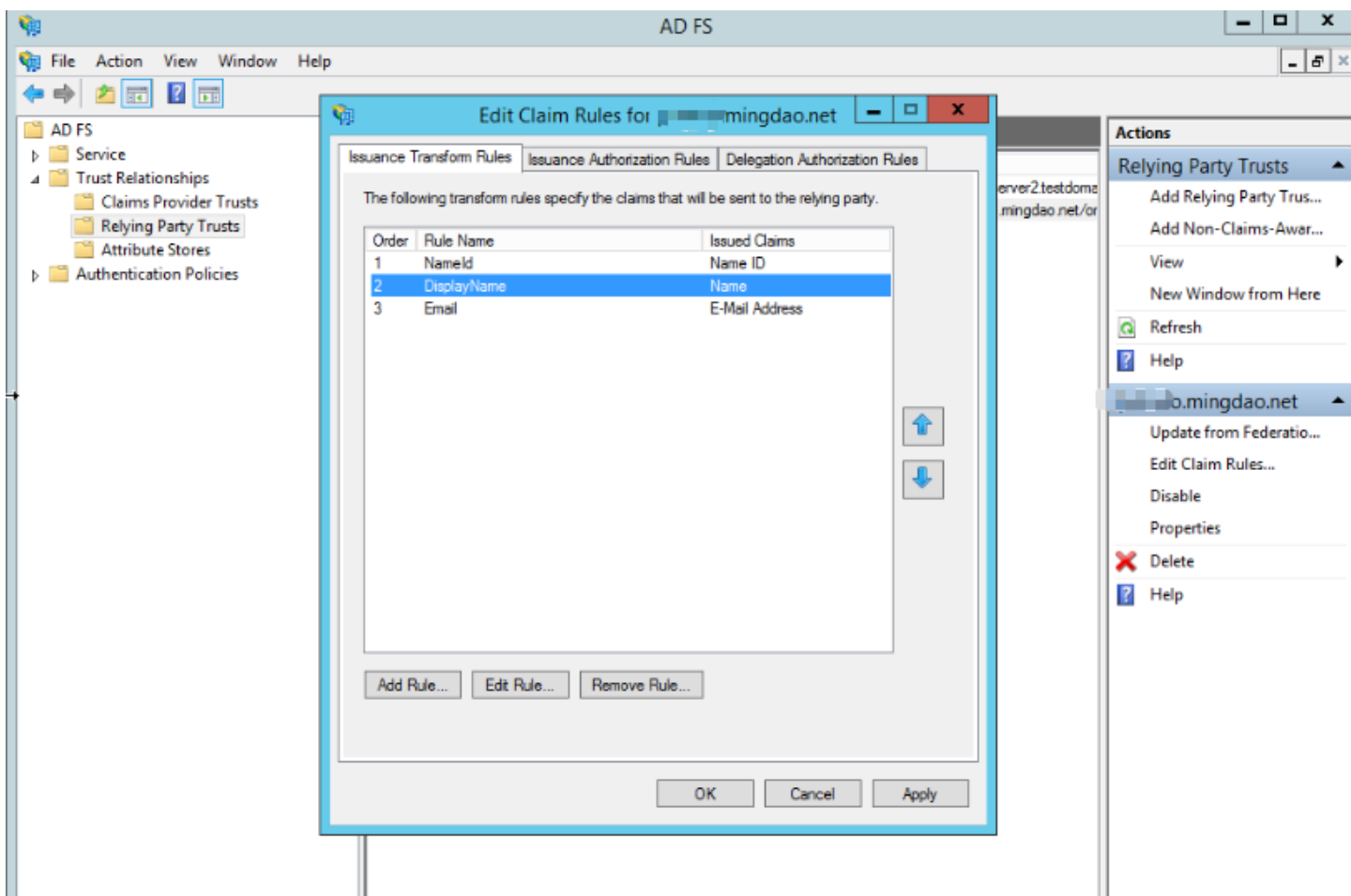
1. Add mapping rules:



2. Select the rule template, which can be set according to the actual situation. You can customize the template or select an existing rule.



3. Here is an example of submitting three existing rules, corresponding to user ID, Name, and Email corresponding to user attributes, for account creation.



jack

TASKS ▼ SECT

Account

Organization

Member Of

Password Settings

Profile

Policy

Silo

Extensions

Account

First name: jack

Middle initials:

Last name: lee

Full name: \* jack lee

User UPN login: jack @ testdomain.com

User SamAccountName l... testdomain \\* jack

☒ Protect from accidental deletion

Account expires: ☒ Never ☐ End of

Password options:

- ☐ User must change password at next log on
- ☒ Other password options
  - ☐ Smart card is required for interactive log on
  - ☒ Password never expires
  - ☐ User cannot change password

Encryption options:

Other options:

Log on hours... Log on to...

Organization

Display name: jack lee

Office:

E-mail: jack@md.com

Web page:

Phone numbers:

Job title:

Department:

Company:

Manager: Edit

Direct reports:

More Information

OK

## User SSO login

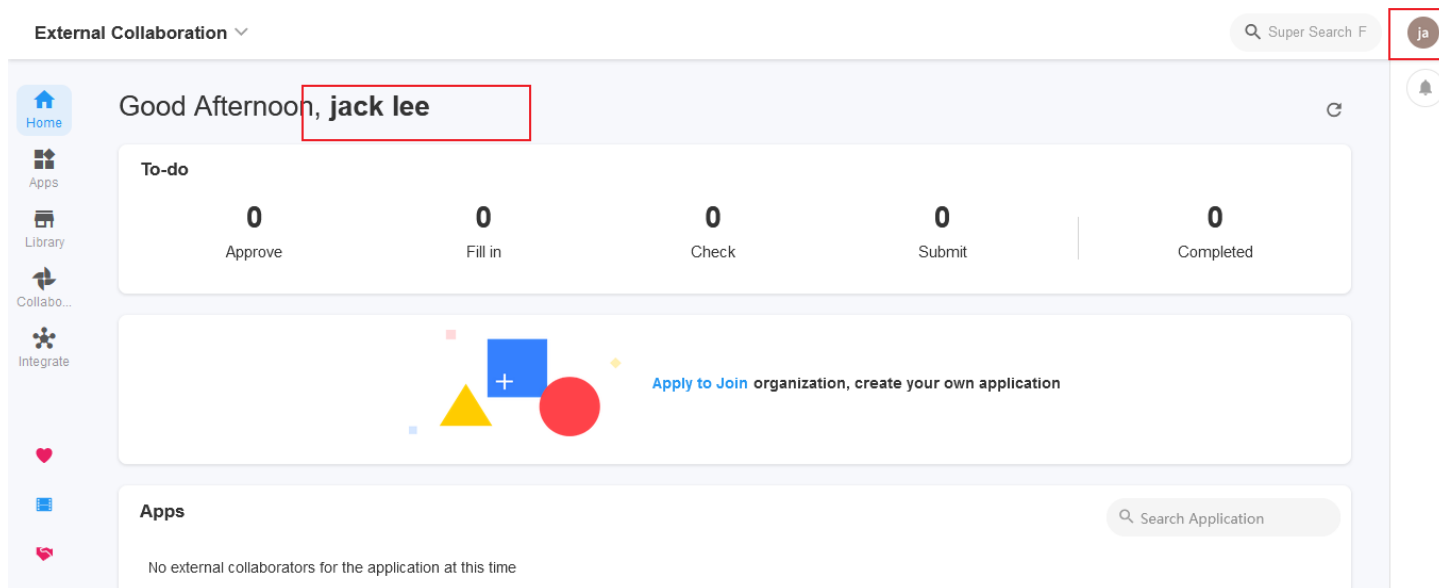
1. Enter `{server}/orgsso/sso` in the browser.



2. Redirect SAMLRequest to `https://adserver2.testdomain.com/adfs/ls/?SAMLRequest=xxx`



3. Enter adserver2 username and password information to complete system login, as shown in the figure below:



# Google-SAML Integration Guide

## Operational Scenarios

Google Workspace's SAML-based Single Sign-On (SSO) provides a secure authentication method, allowing users to access multiple services through a single Identity Provider (IdP).

## Prerequisites

1. A Google Workspace account with administrative privileges.
2. Ensure that your domain has been verified within Google Workspace.

## Operation Steps

### Configure SAML Application

1. Log in to [Google Admin Console](#)
2. Access security settings; navigate to **Apps > Web & Mobile Apps**
3. **Add App > Add Custom SAML App**

admin.google.com/ac/apps/unified

Admin

Search for users, groups or settings

Apps > Web and mobile apps

Apps (1)

Add app Settings

Search for apps

Add private Android app

Add private Android web app

Add custom SAML app

Overview

Google Workspace

Additional Google services

Web and mobile apps

Google Workspace Marketplace apps

Generative AI

Security

Data NEW

Reporting

Billing

Account

Rules

Name	User access	Details
HAP	ON for 1 organisational unit	Certificate expires or

4. Fill in the application parameters and click continue

Add custom SAML app

App details

Enter details for your custom SAML app. This information is shared with app users. [Learn more](#)

App name

HAP1

Description

App icon

Attach an app icon. Maximum upload file size: 4 MB

HAP

5. Review the IdP parameters and click continue



6. Fill in the parameters ACS URL, Entity ID. Here, enter the service address with a fixed suffix `{HAP}/orgsso/assert`, `{HAP}/orgsso/metadata.xml`. These addresses will be configured to take effect later. Click continue.

7. Fill in attribute mapping, set the user information attributes to be returned, such as name, email, etc., and click complete.

×

Add custom SAML app

Add and select user fields in the Google Directory, then map them to service provider attributes. Attributes marked with \* are mandatory. [Learn more](#)

Google directory attributes

Basic Information >  
First name

Basic Information >  
Last name

Basic Information >  
Primary email

App attributes

→ 

FirstName

 ×

→ 

LastName

 ×

→ 

PrimaryEmail

 ×

ADD MAPPING

Group membership (optional)

Group membership information can be sent in the SAML response if the user belongs to any of the groups that you add here.

Google Groups

Search for a group

→ Groups

App attribute

## 8. Configuration complete

Apps > Web and mobile apps > HAP1

SAML

HAP1

TEST SAML LOGIN

DOWNLOAD METADATA

EDIT DETAILS

DELETE APP

User access

To make the managed app available to selected users, choose a group or organisational unit. [Learn more](#)

[View details](#)

OFF for everyone

Service provider details

Certificate

Google\_2029-11-24-1223\_SAML2\_0  
(Expires 24 Nov 2029)

ACS URL

[https://](https://mingdao.net/orgsso/ass) [mingdao.net/orgsso/ass](https://mingdao.net/orgsso/ass)  
ert

Entity ID

[https://](https://mingdao.net/orgsso/metadata.xml) [mingdao.net/orgsso/met](https://mingdao.net/orgsso/metadata.xml)  
adata.xml

SAML attribute mapping

Map Google directory user profile fields to SAML service provider attributes.

FirstName

Basic Information > First name

LastName

Basic Information > Last name

PrimaryEmail

Basic Information > Primary email

## 9. Set user access permissions

SAML



HAP1

TEST SAML LOGIN

DOWNLOAD METADATA

EDIT DETAILS

DELETE APP

### User access

To make the managed app available to selected users, choose a group or organisational unit. [Learn more](#)

[View details](#)

OFF for everyone

### Service provider details

**Certificate**

Google\_2029-11-24-1223\_SAML2\_0  
(Expires 24 Nov 2029)

**ACS URL**

https://[redacted]ningdao.net/orgsso/ass  
ert

**Entity ID**

https://[redacted]ningdao.net/orgsso/met  
adata.xml

### SAML attribute mapping

Map Google directory user profile fields to SAML service provider attributes.

**FirstName**

Basic Information > First name

**LastName**

Basic Information > Last name

**PrimaryEmail**

Basic Information > Primary email



HAP1

All users in this account

Groups



Organisational units



Search for organisational units

▼ Nocoly.com Limited ●

Nocoly R&amp;D

Nocoly Sales

Showing settings for users in Nocoly R&amp;D

#### Service status:

**Service status:**

Inherited

☒ ON☐ OFF


Override will overrule the settings inherited from the parent org unit.  
Most changes take effect within a few minutes. [Learn more](#)

1 unsaved change

CANCEL

OVERRIDE

SAML

 HAP1

TEST SAML LOGIN

DOWNLOAD METADATA

EDIT DETAILS

DELETE APP





**User access**

To make the managed app available to selected users, choose a group or organisational unit. [Learn more](#)

[View details](#)

ON for 1 organisational unit  
Nocoly R&D

**Service provider details**

<b>Certificate</b> Google_2029-11-24-1223_SAML2_0 (Expires 24 Nov 2029)	<b>ACS URL</b> https://   ningdao.net/orgsso/ass ert	<b>Entity ID</b> https://   ningdao.net/orgsso/met adata.xml
---	---	--


**SAML attribute mapping**

Map Google directory user profile fields to SAML service provider attributes.

<b>FirstName</b> Basic Information > First name	<b>LastName</b> Basic Information > Last name	<b>PrimaryEmail</b> Basic Information > Primary email
--	--	--

10. Download metadata, which will be used later to configure the identity authentication source (IdP) for the **HAP** service.

SAML

 HAP1

TEST SAML LOGIN

DOWNLOAD METADATA

EDIT DETAILS

DELETE APP





**User access**

To make the managed app available to selected users, choose a group or organisational unit. [Learn more](#)

[View details](#)

ON for 1 organisational unit  
Nocoly R&D

**Service provider details**

<b>Certificate</b> Google_2029-11-24-1223_SAML2_0 (Expires 24 Nov 2029)	<b>ACS URL</b> https://   mingdao.net/orgsso/ass ert	<b>Entity ID</b> https://   mingdao.net/orgsso/met adata.xml
---	---	--

**SAML attribute mapping**

Map Google directory user profile fields to SAML service provider attributes.

<b>FirstName</b> Basic Information > First name	<b>LastName</b> Basic Information > Last name	<b>PrimaryEmail</b> Basic Information > Primary email
--	--	--

## HAP Integration with SAML Application Single Sign-On

**TIP**

For detailed steps, refer to the private deployment documentation [How to Integrate Single Sign-On - SAML2](#).

1. Configure `sso.json`, content as follows

```
{
  "mode": "common-saml2",
  "name": "saml2",
  "saml2": {
    "entityId": "{HAP}/orgsso/metadata.xml",
    "assertUrl": "{HAP}/orgsso/assert",
    "params": {
      "UserId": "name_id", // Fill in the user's unique attribute,
                           // the default name_id
      "Name": "FirstName", // Fill in the FirstName attribute of the
                           // attribute map
      "Email": "PrimaryEmail" // Fill in the PrimaryEmail attribute of
                           // the attribute map
    },
    "autoRegister": true,
    "projectId": ""
  }
}
```

2. Configure `idp.xml`, upload the metadata downloaded in the previous steps to the corresponding directory on the server.

3. Modify `docker-compose.yaml`, mount the configuration files (there are differences in cluster mode), and restart the service.

```
- ./volume/sso/sso.json:/usr/local/MDPrivateDeployment/sso/OptionFile/sso.json
- ./volume/sso/metadata/idp.xml:/usr/local/MDPrivateDeployment/sso/OptionFile/met
```

4. After the restart is complete, visit `{HAP}/orgsso/metadata.xml` to see the corresponding `xml` output.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" entityID="https:
validUntil="2024-12-06T09:18:03.795Z">
  ▼<md:SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:1.1:protocol urn:oasis:names:tc:SAML:2.0:protocol">
    ▼<md:KeyDescriptor use="signing">
      ▼<ds:KeyInfo>
        ▼<ds:X509Data>
          <ds:X509Certificate>MIIDVzCCAj8CFDpfgw/fmEphpqRjnZN0dQBpxRWEMA0GCSqGSIb3DQEBCwUAMGcxCzAJBgNVBAYTAKNOMREwDwYDVQQHDAhTaGFuZ2h
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    ▼<md:KeyDescriptor use="encryption">
      ▼<ds:KeyInfo>
        ▼<ds:X509Data>
          <ds:X509Certificate>MIIDVzCCAj8CFDpfgw/fmEphpqRjnZN0dQBpxRWEMA0GCSqGSIb3DQEBCwUAMGcxCzAJBgNVBAYTAKNOMREwDwYDVQQHDAhTaGFuZ2h
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
  </md:SPSSODescriptor>
</md:EntityDescriptor>
```

## User SSO Login

1. Enter the single sign-on entry address `{HAP}/orgsso/sso` in the browser.
2. Log in to the organizational account.

All organisations <

☐ Users from all organisational units

☒ Users from selected organisational units

Search for organisational units

Select multiple

▼ Nocoly.com Limited

Nocoly R&D


Nocoly Sales

MANAGE ORGANISATIONAL UNITS

✕ Users | Showing users from Nocoly R&D


Add new u... Bulk update u... Download us... More options ▼

User status: "Active" ✕ + Add a filter CLEAR FILTERS

<input type="checkbox"/>	Name ↑	Email	Status	Last sign-in ⚙
<input type="checkbox"/>	 user5 hap	user5@nocoly.com	Active (created 5 days ago)	2 days ago

Rows per page: 20 ▼

Page 1 of 1 < >



# Sign in

Use your Google Account

Email or phone

user5@nocoly.com

[Forgot email?](#)

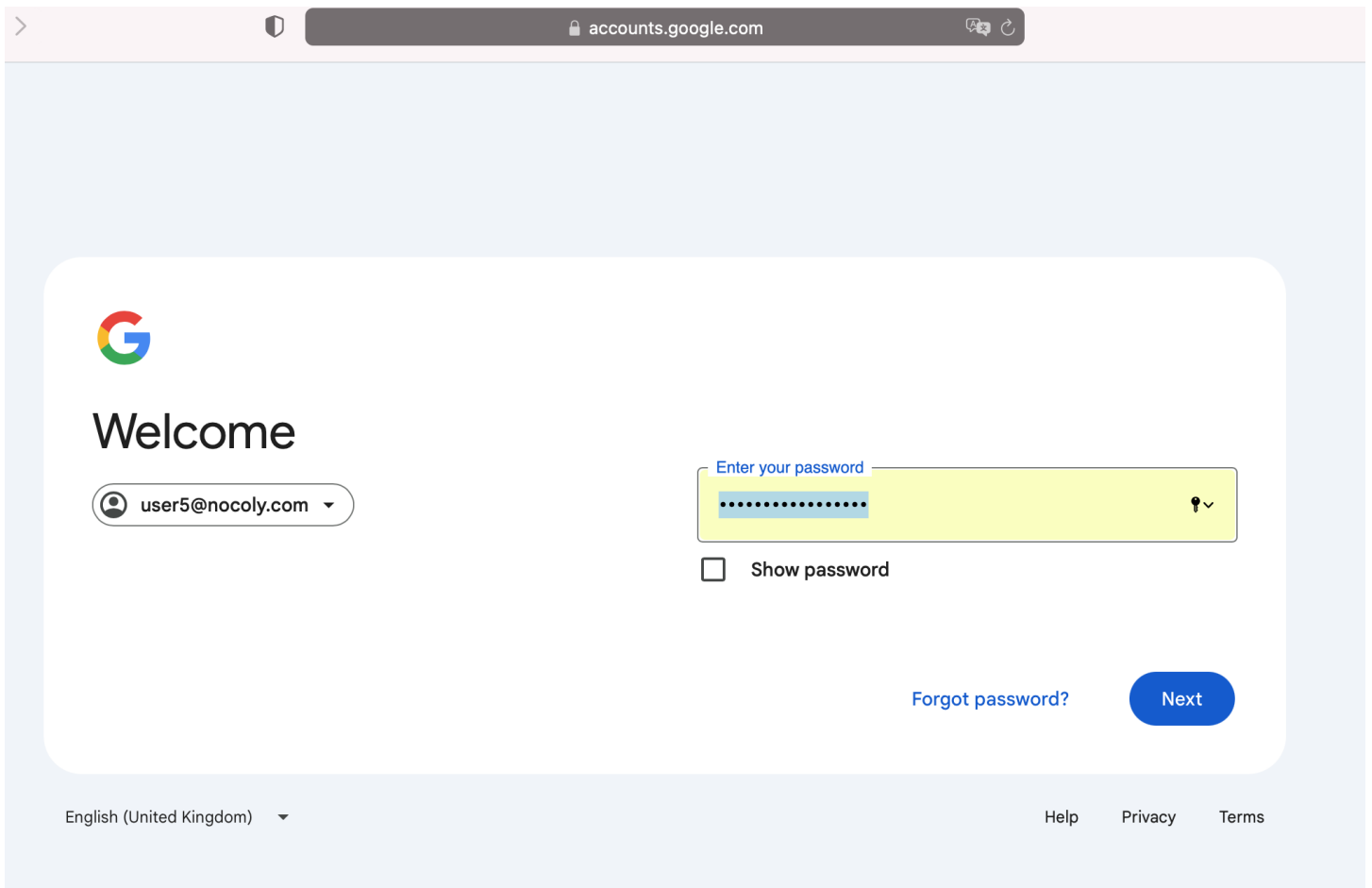
Not your computer? Use Private Browsing windows to sign in.  
[Learn more about using Guest mode](#)

[Create account](#)

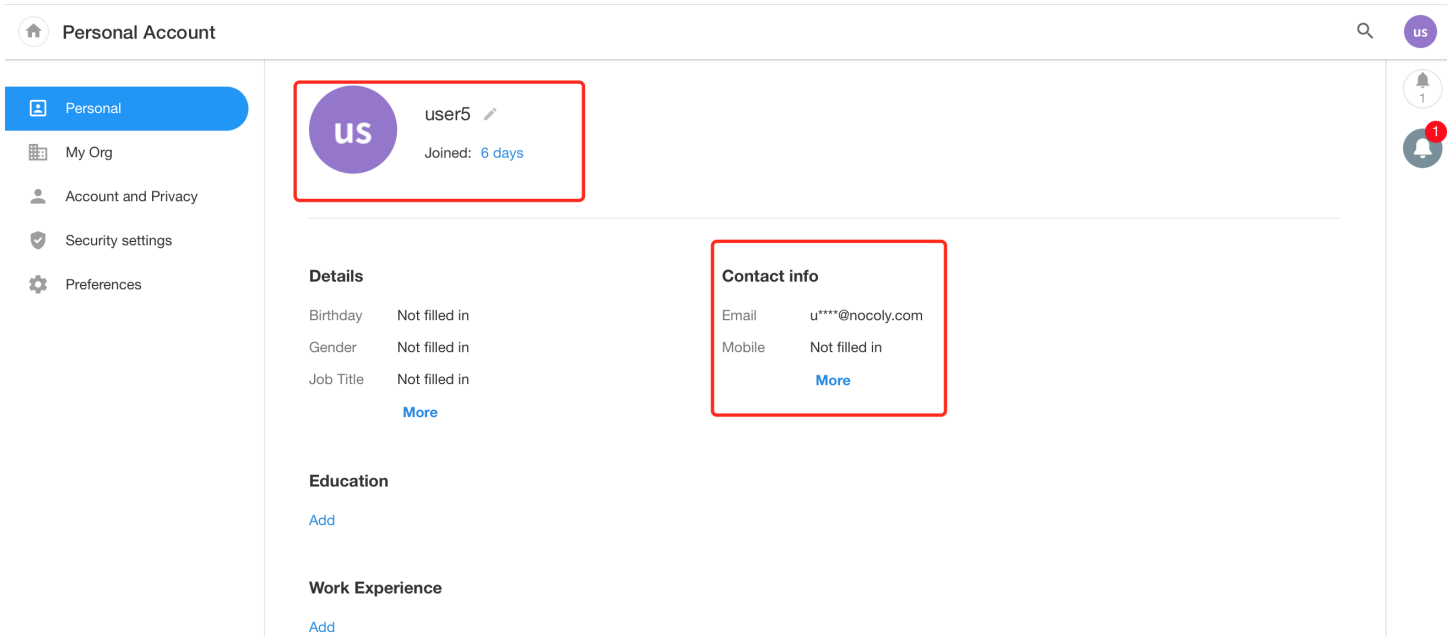
Next

English (United Kingdom) ▼

[Help](#) [Privacy](#) [Terms](#)



3. Successfully log in to HAP.



4. Change user organisational unit, prompt **403**.



All organisations <

☐ Users from all organisational units

☒ Users from selected organisational units

Search for organisational units

Select multiple

▼ Nocoly.com Limited

Nocoly R&D

Nocoly Sales

MANAGE ORGANISATIONAL UNITS

✕ Users | Showing users from Nocoly Sales

[Add new u...](#) [Bulk update u...](#) [Download u...](#) [More options](#) ▼

+ Add a filter

<input type="checkbox"/>	Name ↑	Email	Status	Last sign-in
<input type="checkbox"/>	user5 hap	user5@nocoly.com	Active (created 5 days ago)	2 days ago

Rows per page: 20 ▼

Page 1 of 1 < >



403. That's an error.

Error: app\_not\_enabled\_for\_user

Service is not enabled for this user.



# Google-OIDC Integration Guide

## Operational Scenarios

Google Cloud's OIDC-based Single Sign-On (SSO) provides a secure authentication method that allows users to access multiple services with a single identity.

Google OIDC supports both **internal** and **external** user access control (if not a Google Workspace user, the application can only be provided to external users).

## Prerequisites

- Have a Google account

## Operation Steps

### Configure OAuth2.0 Client Application

1. Log in to [Google Cloud Console](#)
2. Access security settings; navigate to **APIs & Services** > **OAuth consent screen**.

Google Cloud

My Project 99276

Search (/) for resources, docs, products and more

API APIs and services

Enabled APIs and services

Library

Credentials

OAuth consent screen

Page usage agreements

OAuth consent screen

OAuth consent screen management is changing. This page has been replaced with a new, easier to use experience. The current pages will only be available for a few more days.

GO TO NEW EXPERIENCE

HAP2

EDIT APP

User type

Internal

MAKE EXTERNAL

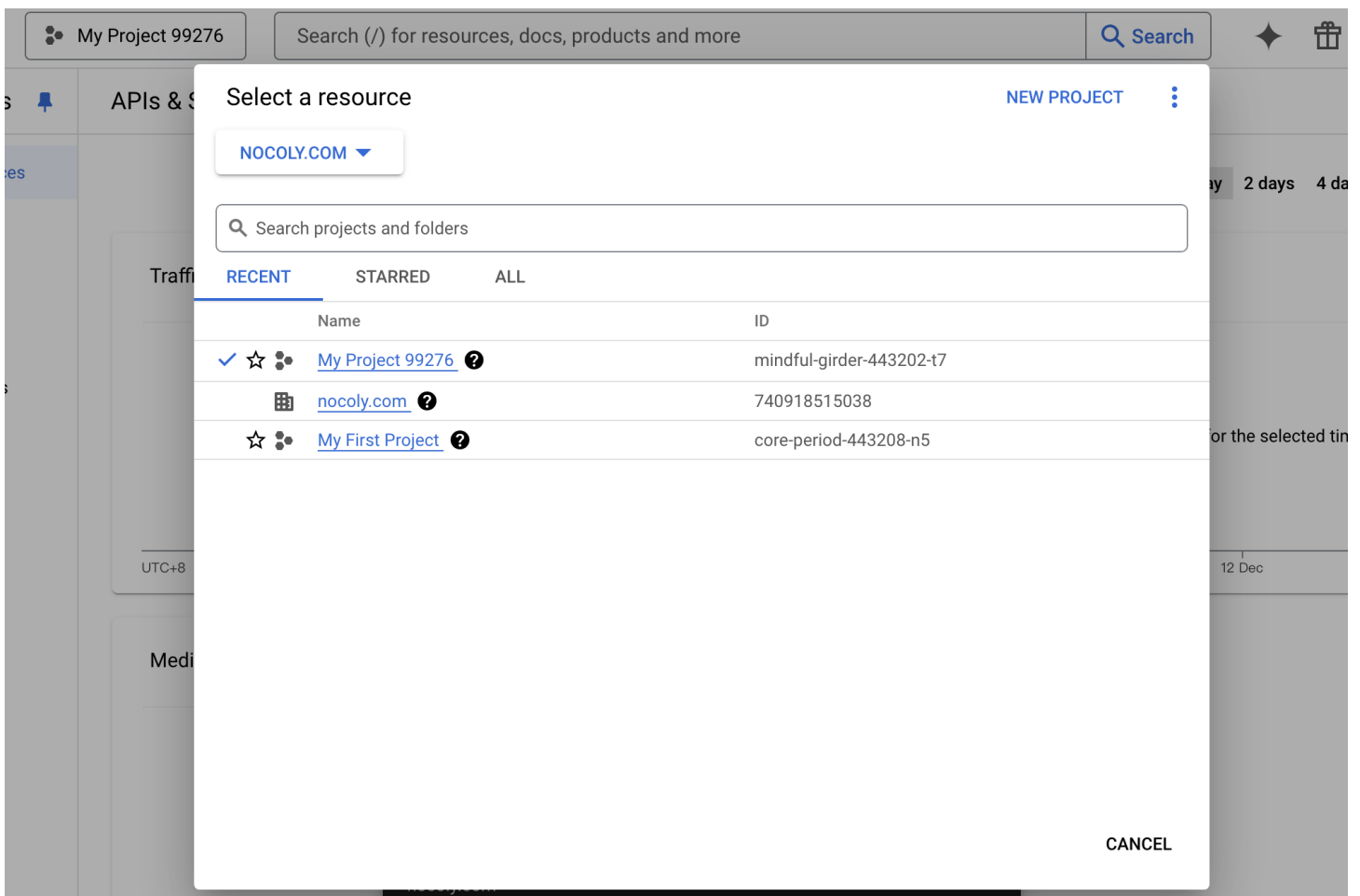
OAuth rate limits

OAuth token grant rate

Token grant rates limit how quickly your application can get new users.

Your current per-day token grant rate limit is 10,000 grants per day. The per-day token grant rate resets every day.[Raise daily token limit](#)

3. Choose to create a project or use an existing project



Google Cloud

Search (/) for resources, docs, products and more

Search

### New Project

Project name \*  
My Project 88297 ?

Project ID: durable-destiny-444503-p2. It cannot be changed later. [EDIT](#)

Organisation \*  
nocoly.com ?

Select an organisation to attach it to a project. This selection can't be changed later.

Location \*  
nocoly.com [BROWSE](#)

Parent organisation or folder

[CREATE](#) [CANCEL](#)

4. Create an application under the project, Fill in the branding and audience informatio



## App information

This shows in the consent screen, and helps end users know who you are and contact you

App name \*

HAP2

The name of the app asking for consent

User support email \*

ops@nocoly.com

For users to contact you with questions about their consent. [Learn more](#)

## App logo

This is your logo. It helps people to recognise your app and is displayed on the OAuth consent screen.

After you upload a logo, you will need to submit your app for verification unless the app is configured for internal use only or has a publishing status of 'Testing'. [Learn more](#)



Google Cloud

My Project 99276

Search (/) for resources, docs, products and more

Google Auth Platform

Overview

Branding

Audience

Clients

Data access

Verification centre

Audience

User type

Internal

In this mode, your app is limited to Google Workspace users within your organisation. [Learn more about user type](#)

5. Create an `OAuth2.0` client, Configure redirect address

Google Auth Platform

Clients

+ CREATE CLIENT

DELETE

RESTORE DELETED OAUTH CLIENTS

Overview

Branding

Audience

Clients

Data access

Verification centre

OAuth 2.0 Client IDs

Name

Creation date

Type

No OAuth clients to display

Google Cloud

My Project 99276

Search (/) for resources, docs, products and more

Search

Google Auth Platform

Create OAuth client ID

Overview

Branding

Audience

Clients

Data access

Verification centre

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type \*

Web application

Name \*

Web client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorised domains](#).

Authorised JavaScript origins

For use with requests from a browser

+ ADD URI

Overview

Branding

Audience

Clients

Data access

Verification centre

The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorised domains](#).

Authorised JavaScript origins

For use with requests from a browser

+ ADD URI

Authorised redirect URIs

For use with requests from a web server

URIs 1 \*

http://localhost:3000/oidc-redirect

+ ADD URI

Note: It may take five minutes to a few hours for settings to take effect

SAVE

CANCEL

Creation date

Client secret

Creation date

Status

+ ADD SECRET

6. After creation, download the **OIDC** related configuration for later **HAP** configuration of **OIDC** Single Sign-On

Google Cloud

My Project 99276

Search (/) for resources, docs, products and more

Search

Google Auth Platform

Clients

+ CREATE CLIENT

DELETE

RESTORE DELETED OAUTH CLIENTS

Overview

Branding

Audience

Clients

Data access

Verification centre

OAuth 2.0 Client IDs

☐ Name

☐ Web Client 1

Web Client 1

Client ID

88172c...-nhb4mdcupbrp...e6e0sfrje53jf75ai.apps...com

Client secret

GO...X-...SDL17ZpNF48Y4YJ652

Creation date

29 November 2024 at 11:15:00 GMT+8

Status

Enabled

DOWNLOAD JSON

CLOSE

Client ID

881725109444-nhb4



# HAP Integration with OIDC Single Sign-On

For detailed steps, refer to the HAP private deployment documentation [How to Integrate Single Sign-On - OIDC](#).

1. Configure `sso.json`, content as follows:

```
1  {
2    "web": {
3      "client_id": "444-nhb4mdc rp0nbe6e0sfrje53jf75ai.apps.googleusercontent.com",
4      "project_id": "mindful-girder-443202-t7",
5      "auth_uri": "https://accounts.google.com/o/oauth2/auth",
6      "token_uri": "https://oauth2.googleapis.com/token",
7      "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
8      "client_secret": "LSDL1 NF48Y4YJ652",
9      "redirect_uris": [
10       "http://localhost:3000/oidc-redirect"
11     ]
12   }
13 }
```

```
{
  "mode": "common-oidc",
  "name": "oidc",
  "oidc": {
    "clientId": "x-x.apps.googleusxx",
    "clientSecret": "x-xxxyZwQgUtLSDL17Zpxxx",
    "oidcUrl": "https://accounts.google.com/.well-known/openid-configuration",
    "redirectUrl": "http://localhost:3000/oidc-redirect",
    "responseTypes": "code",
    "scope": "openid email profile",
    "params": {
      "UserId": "sub",
      "Name": "name",
      "Email": "email"
    },
    "autoRegister": true,
    "projectId": ""
  }
}
```

2. Mount the `OIDC` configuration file (as follows), then restart the service.

```
./volume/sso/sso.json:/usr/local/MDPrivateDeployment/sso/OptionFile/sso.json
```

3. Restart completed.

## User SSO Login

1. Enter `{HAP}/orgsso/oidc` or `{HAP}/orgsso/sso` in the browser to log in.
2. Log in to the organizational account and authorize.

accounts.google.com

Sign in with Google

Sign in

to continue to HAP2

Email or phone

user5@nocoly.com

[Forgot email?](#)

Create account

Next

English (United States)

Help

Privacy

Terms

Sign in with Google

Sign in to HAP2

u

user5@nocoly.com

By continuing, Google will share your name, email address, language preference, and profile picture with HAP2. See HAP2's Privacy Policy and Terms of Service.

You can manage Sign in with Google in your [Google Account](#).

Cancel

Continue

English (United States)

Help

Privacy

Terms

Personal Account

Search icon

Help icon

US

Personal

My Org

Account and Privacy

Security settings

Preferences

Product key

US

user5 hap

Joined: 3 days

Account info

EditionFree

Details

BirthdayNot filled in

GenderNot filled in

Job TitleNot filled in

More

Contact info

Emailu\*\*\*\*@nocoly.com

MobileNot filled in

More

3. Switch to an external user.

Sign in with Google

Sign in

to continue to HAP2

Email or phone

yar[REDACTED]@gmail.com

Forgot email?

Create account

Next

English (United States)

Help

Privacy

Terms

4. Access is prohibited.

# Access blocked: HAP2 can only be used within its organization

 yangcharix@gmail.com

HAP2 is restricted to users within its organization. If you think you should have access, you can contact the [developer](#).

[Learn more about this error](#)

If you are a developer of HAP2, see [error details](#).

Error 403: org\_internal

# Workflow

## How to clear serial clogged workflows?

Query the MongoDB database for the workflow's `storeId` by workflow ID.

Example with workflow ID of `5f12855302740ff3c82f`.

```
> use mdworkflow  
> db.app_consumerSequence.find({_id:ObjectId("5f12855302740ff3c82f")})
```

Execute the clear statement after getting the `storeId` corresponding to the workflow.

Example of `32e8d0fdsa32948d449` for `storeId`.

```
> db.app_consumerSequence.remove({storeId:"32e8d0fdsa32948d449"})  
>  
db.app_consumerActivity.remove({processId:"5f12855302740ff3c82f",storeId:"32e8d0fd
```

## How to view workflows executed during a specified time period?

Get the organization ID in advance: click the profile photo, navigating to [Org Admin] > [Organization] > [Basics].

DemoA <<

Home

User

Organization

**Basics**

Administrator

Misc.

Apps

Security

Log

Integration

**Basics**

Logo

Recommended size 400\*180 px for display on workbench, print, share and corporate domain pages, recommended size 512KB or less

Organization Name DemoA [Change](#)

Org door number ZY2813


Members can join the organization by entering the organization's door number

ID a08b0cf4-e300-487a-83b8-3f91ee57bae9

Unique number for every org, used for communication and feedback

---

Extended Information Customizable organization alias and login background image [Setting](#)



Job Position [Setting](#)

Location [Setting](#)

## Query in MongoDB

```
> use mdworkflow
> db.wf_instance.aggregate([{"$match": {"companyId":"xxxxxx","createDate": {"$gt": ISODate("2023-05-06T06:10:00.000Z"), "$lt": ISODate("2023-05-06T06:20:00.000Z")}}}, {"$group": {"_id": "$processId", "count": {"$sum": 1}}}, {"$sort": {"count": -1}}])
```

- Note that replace `companyId` in the statement with the actual organization ID.
- The time in the query condition is in UTC time zone, 06:10:00.000Z corresponds to 14:10:00 in East 8 time zone.

## Example of query result:

```
{ "_id" : "6455b144ab61ff25944abd32", "count" : 5890 }
{ "_id" : "645359f4bf0d053e6dec5f43", "count" : 1618 }
{ "_id" : "6455dc569eb0167f785c0692", "count" : 1514 }
{ "_id" : "6455db87690c6744298e6aa4", "count" : 603 }
{ "_id" : "645359f3bf0d053e6dec5f1b", "count" : 332 }
{ "_id" : "645497ca77b2ac62d41c227a", "count" : 294 }
{ "_id" : "645359f4bf0d053e6dec5f5e", "count" : 113 }
```

- `_id` is the workflow ID
- `count` is the number of executions during the period

Visit the following in your browser: System address/workflowedit/workflow ID, to access the corresponding workflow

Example: <https://hap.domain.com/workflowedit/6455b144ab61ff25944abd32>

If you are prompted that you do not have permission to access the workflow, you can [get the workflow name and the application it belongs to by workflow ID](#).

## How to get the workflow name and the application it belongs to by workflow ID?

Query in MongoDB

```
> use mdworkflow
> db.wf_process.find({"_id":ObjectId("6455b144ab61ff25944abd32")})
```

- The output contains the name field, which corresponds to the **workflow name**.

After you get the workflow name, click the profile photo, going to [Org Admin] > [App Management] > [Workflow] to search for the application to which the workflow belongs.

Workflow (2)

Monitor

Short Message Template

This month 47,364 / 1,000,000,000 (Contains 999,600,000 monthly quota) Remaining 99.99%

All Apps

All Status

All Types

Initiate an approval

Workflow Name	This month	Status	Type	Create Date	Creator
<div><div></div><div>Initiate an approval 访客/班车预约管理</div></div>	0	<div>Off</div>	Workflow	2024-01-19	<div></div>
<div><div></div><div>Initiate an approval DAYU DEV TEST</div></div>	0	<div>On</div>	Workflow	2023-12-07	<div></div>



# File

## How to locate the record to which the file is attached by file name in a worksheet?

Query by file name saved in object store

Query by file name when uploaded

Take the filename saved in the object store as example

bR2HbXfudAfo9JafdveE4K0C9J2g519Gae3d4I84820k8a9Bee3t5I914q5J4p6e.pdf

Go to MongoDB's `mdattachment` to query the corresponding `docid`.

```
> use mdattachment
> db.attachment.find({"name":
"52a9dF0c71fscg6P9Lahfc5B41d4c0dv5FasdM5a5LaQdoaE9V1x64aK04en3Ydu.pdf"}).pretty()
{
  "_id" : ObjectId("6413d880840db440e5f5a7be"),
  "docid" : "a234f6ca-6bf2-4491-9261-e24758376b91",
  "fid" : "a234f6ca-6bf2-4491-9261-e24758376b91",
  "server" : "https://hap.domain.com/file/mdoc/",
  "oFilename" : "APaaS测试文件",
  "path" : "doc/20230317/",
  "name" :
"52a9dF0c71fscg6P9Lahfc5B41d4c0dv5FasdM5a5LaQdoaE9V1x64aK04en3Ydu.pdf",
  .....
}
```

- Get the `docid`: a234f6ca-6bf2-4491-9261-e24758376b91
- The `oFilename` here is also the actual name of the file saved in the record of the worksheet.

Query the control ID in the relationship table by `docid`.

```
> db.attachmentrelation.find({"docid" : "a234f6ca-6bf2-4491-9261-
e24758376b91"}).pretty()
{
  "_id" : ObjectId("6448eb672d71e11d04aae297"),
  "docid" : "a234f6ca-6bf2-4491-9261-e24758376b91",
  "fid" : "a234f6ca-6bf2-4491-9261-e24758376b91",
```

```

    "fType" : 9,
    "pids" : [
      "eaf823e5-4205-44fd-8521-109d3ec29a92"
    ],
    "ctime" : ISODate("2023-04-26T09:14:15.289Z"),
    "sourceid" : "bac739dc-9f8f-4ea5-a1f8-bdea56d27b81",
    "commentid" : "6448eb5c12053cf665be2131"
  }

```

- If `fType = 9`, then `commentid` is the control id, `sourceid` is the record id.
- The control id obtained `commentid` is `6448eb5c12053cf665be2131`
- **The record id obtained `sourceid` is `bac739dc-9f8f-4ea5-a1f8-bdea56d27b81`**
  - `sourceid` corresponds to `rowid` of the corresponding worksheet in `mdwsrows`.
  - `commentid` corresponds to `cid` in the `wscontrols` in `mdworksheet`.

Query worksheet ID by `commentid` in `wscontrols` of `mdworksheet`.

```

> use mdworksheet
switched to db mdworksheet
> db.wscontrols.find({cid:"6448eb5c12053cf665be2131"}).pretty()
{
  "_id" : ObjectId("6448eb5c12053cf665be2136"),
  "cid" : "6448eb5c12053cf665be2131",
  "wsid" : "6448eb5c12053cf665be212e",
  .....
}

```

- `wsid` corresponds to the worksheet id. So the worksheet id obtained is `6448eb5c12053cf665be212e`

Visit the following in your browser: System address/worksheet/worksheet ID, to access the worksheet where the queried file is attached.

- Example: <https://hap.domain.com/worksheet/6448eb5c12053cf665be212e>

Visit the following in your browser: System address/worksheet/worksheet ID/row/record ID, to access the record where the queried file is attached.

- Example: <https://hap.domain.com/worksheet/6448eb5c12053cf665be212e/row/bac739dc-9f8f-4ea5-a1f8-bdea56d27b81>

# Continuous Queuing of Workflows

After a workflow event is triggered, it enters the Kafka message queue in the form of a message, and is then consumed by the workflow consumption service for processing.

When encountering a persistent workflow queue within the platform, the initial step is to differentiate between a scenario where the queue is large but items are being processed gradually and one where the queue remains stagnant with no consumption.

Consider the following two scenarios:

## Workflow queued and consumed

## Workflow queued but not consumed



## Workflow queued and consumed

Check the monitoring page of the workflow to see if there are workflows with a very large queue, such as tens of thousands or millions of workflows. This may be due to misconfigured trigger logic by business personnel or loops causing a high volume of workflow triggers.

Consumption capacity may be affected by various factors. For example, while the system can normally process ten thousand workflows per minute, there may be instances where hundreds of thousands of workflows are triggered, resulting in a large number of workflows being queued up without being quickly consumed.

For example, when a few workflows have a queue of tens of thousands or more, causing severe queuing for all workflows, **if it is confirmed that these workflows with large queues do not need to be**

**processed again, they should be directly closed in a non-paused state.**

- When a workflow is closed directly, the workflows in the queue will be quickly consumed (without going through the nodes logic in the workflow), allowing them to be processed as soon as possible.
- If the workflows with a large queue need to be consumed, they can be paused first and then re-started during business downtime.

---

If there is no unintended triggering of workflows with high queue volumes, please log in to the server and check resource usage. You can use the `top` command to view the real-time CPU and memory usage of the server and its processes.

- If the process consuming a significant amount of CPU is the `mongod` process, it is usually caused by slow queries. For resolution, refer to the [slow query optimization](#).
- If the process consuming a significant amount of CPU is a `dotnet` or `java` process, it is typically due to complex logic in workflow nodes, which can also increase resource usage for related services.
  - If the server resources are fully utilized, you may choose to scale out or postpone running workflows with complex logic to periods of low business activity.
  - If it is a Kubernetes-based cluster deployment, and there is some resource redundancy, you can dynamically scale up the service instances with high resource usage.

## Workflow queued but not consumed

When the workflow is continuously queued but not consumed, it is usually due to a full server disk or an issue with the Kafka service.

- Use the `df -Th` command on the server to check the usage of the system disk and data disk.
- Check if the Kafka service is running properly:

**Standalone Mode**

**Cluster Mode**

Check the health check logs of the storage component container.

```
docker logs $(docker ps | grep hap-sc | awk '{print $1}')
```

- If the log output is normal, it will be all INFO. If the Kafka service keeps restarting continuously, it means the current Kafka service is abnormal. You can try restarting the service as a whole first. If Kafka still cannot start, [clear the Kafka error data](#).

If the triggered workflows were unable to be written to the Kafka queue due to reasons such as a full disk or Kafka service issues, the history "queued" workflows triggered will no longer be consumed.

---

An extremely rare possibility is that the Kafka consumer group is in a rebalance state. This is mainly caused by slow processes resulting in timeouts, or consumer service instances being affected by resource issues and restarting, which in turn triggers consumer group rebalancing.

You can use the following command to check the actual message accumulation status in each topic partition and whether the system is currently in a rebalancing state:

#### Standalone Mode

#### Kubernetes Cluster Mode

1. Enter the container of the storage component

```
docker exec -it $(docker ps | grep hap-sc | awk '{print $1}') bash
```

2. Execute the command to view the consumption of the md-workflow-consumer consumer group

```
/usr/local/kafka/bin/kafka-consumer-groups.sh --bootstrap-server  
${ENV_KAFKA_ENDPOINTS:=127.0.0.1:9092} --describe --group md-workflow-  
consumer
```

- If prompted with `Error: Executing consumer group command failed due to null`, you can click to [download the Kafka installation package](#), then upload the installation package to the deployment server and copy it into the hap-sc container. After that, unzip the file and use the `bin/kafka-consumer-groups.sh` from the new installation package to execute the above command.

The normal output is as follow. If prompted with `Warning: Consumer group 'md-workflow-consumer' is rebalancing.`, it means the consumer group is currently rebalancing.

## Normal State

## Rebalancing

```
[root@nj1-private-cluster002 ~]# /usr/local/kafka/bin/kafka-consumer-groups.sh --bootstrap-server 10.206.0.6:9092 --describe --group md-workflow-consumer
```

GROUP	TOPIC	PARTITION	CURRENT-OFFSET	LOG-END-OFFSET	LAG	CONSUMER-ID	HOST	CLIENT-ID
md-workflow-consumer	basic-user	1	16818	16818	0	consumer-md-workflow-consumer-18-c424fe1e-5730-41f3-8b22-483c3a9041aa	/10.206.0.5	consumer-md-workflow-consumer-18
md-workflow-consumer	Worksheet	9	83798	83798	0	consumer-md-workflow-consumer-18-c424fe1e-5730-41f3-8b22-483c3a9041aa	/10.206.0.5	consumer-md-workflow-consumer-18
md-workflow-consumer	Worksheet	7	83033	83033	0	consumer-md-workflow-consumer-18-c424fe1e-5730-41f3-8b22-483c3a9041aa	/10.206.0.5	consumer-md-workflow-consumer-18
md-workflow-consumer	Worksheet	10	84396	84396	0	consumer-md-workflow-consumer-18-c424fe1e-5730-41f3-8b22-483c3a9041aa	/10.206.0.5	consumer-md-workflow-consumer-18
md-workflow-consumer	Worksheet	8	84032	84032	0	consumer-md-workflow-consumer-18-c424fe1e-5730-41f3-8b22-483c3a9041aa	/10.206.0.5	consumer-md-workflow-consumer-18
md-workflow-consumer	Worksheet	12	83100	83100	0	consumer-md-workflow-consumer-18-c424fe1e-5730-41f3-8b22-483c3a9041aa	/10.206.0.5	consumer-md-workflow-consumer-18
md-workflow-consumer	Worksheet	13	83557	83557	0	consumer-md-workflow-consumer-18-c424fe1e-5730-41f3-8b22-483c3a9041aa	/10.206.0.5	consumer-md-workflow-consumer-18
md-workflow-consumer	Worksheet	11	83727	83727	0	consumer-md-workflow-consumer-18-c424fe1e-5730-41f3-8b22-483c3a9041aa	/10.206.0.5	consumer-md-workflow-consumer-18
md-workflow-consumer	basic-user	0	17732	17732	0	consumer-md-workflow-consumer-17-01f193c4-316f-4e32-b196-ee220f236655	/10.206.0.5	consumer-md-workflow-consumer-17
md-workflow-consumer	Worksheet	4	81723	81723	0	consumer-md-workflow-consumer-17-01f193c4-316f-4e32-b196-ee220f236655	/10.206.0.5	consumer-md-workflow-consumer-17
md-workflow-consumer	Worksheet	6	82188	82188	0	consumer-md-workflow-consumer-17-01f193c4-316f-4e32-b196-ee220f236655	/10.206.0.5	consumer-md-workflow-consumer-17
md-workflow-consumer	Worksheet	0	85723	85723	0	consumer-md-workflow-consumer-17-01f193c4-316f-4e32-b196-ee220f236655	/10.206.0.5	consumer-md-workflow-consumer-17
md-workflow-consumer	Worksheet	3	86617	86617	0	consumer-md-workflow-consumer-17-01f193c4-316f-4e32-b196-ee220f236655	/10.206.0.5	consumer-md-workflow-consumer-17
md-workflow-consumer	Worksheet	2	84172	84172	0	consumer-md-workflow-consumer-17-01f193c4-316f-4e32-b196-ee220f236655	/10.206.0.5	consumer-md-workflow-consumer-17
md-workflow-consumer	Worksheet	1	82270	82270	0	consumer-md-workflow-consumer-17-01f193c4-316f-4e32-b196-ee220f236655	/10.206.0.5	consumer-md-workflow-consumer-17
md-workflow-consumer	Worksheet	5	83376	83376	0	consumer-md-workflow-consumer-17-01f193c4-316f-4e32-b196-ee220f236655	/10.206.0.5	consumer-md-workflow-consumer-17
md-workflow-consumer	Workflow-Batch	5	14262	14262	0	consumer-md-workflow-consumer-41-fb6e8a7-c406-423a-82b8-ea942174bfe7	/10.206.0.5	consumer-md-workflow-consumer-41
md-workflow-consumer	Workflow-Batch	6	14075	14075	0	consumer-md-workflow-consumer-41-fb6e8a7-c406-423a-82b8-ea942174bfe7	/10.206.0.5	consumer-md-workflow-consumer-41
md-workflow-consumer	Workflow-Batch	4	14203	14203	0	consumer-md-workflow-consumer-41-fb6e8a7-c406-423a-82b8-ea942174bfe7	/10.206.0.5	consumer-md-workflow-consumer-41
md-workflow-consumer	Workflow-Batch	1	14118	14118	0	consumer-md-workflow-consumer-41-fb6e8a7-c406-423a-82b8-ea942174bfe7	/10.206.0.5	consumer-md-workflow-consumer-41
md-workflow-consumer	Workflow-Batch	2	13917	13917	0	consumer-md-workflow-consumer-41-fb6e8a7-c406-423a-82b8-ea942174bfe7	/10.206.0.5	consumer-md-workflow-consumer-41
md-workflow-consumer	Workflow-Batch	3	13713	13713	0	consumer-md-workflow-consumer-41-fb6e8a7-c406-423a-82b8-ea942174bfe7	/10.206.0.5	consumer-md-workflow-consumer-41
md-workflow-consumer	Workflow-Batch	0	13824	13824	0	consumer-md-workflow-consumer-41-fb6e8a7-c406-423a-82b8-ea942174bfe7	/10.206.0.5	consumer-md-workflow-consumer-41

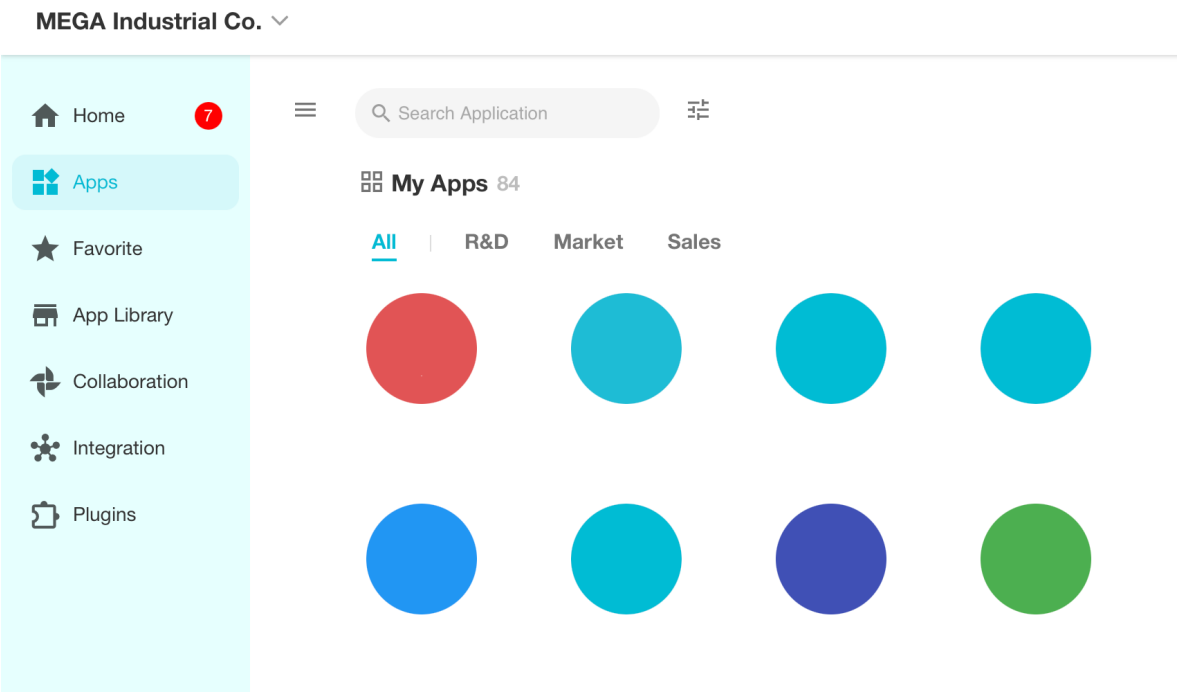
- The LAG column represents the number of messages currently accumulated in the Topic partition.
- Commonly used Topic names in workflows:
  - Workflow: Main workflow execution
  - Workflow-Process: Sub-workflow execution
  - Workflow-Router: Slow queue for workflow execution
  - Workflow-Batch: Bulk workflow execution
  - Workflow-Button: Button-triggered workflow execution
  - Worksheet: Row record validation for triggering workflows
  - Worksheet-Router: Slow queue row record validation for triggering workflows

# Icon Not Showing

Due to the nature of the HAP service itself, the address used to access the HAP system in the browser needs to match what is specified in the `docker-compose.yaml` configuration file.

If they do not match, the following issues may occur:

- Application icons not displaying



- Workflow page loading

Automation

Recycle Bin

New Workflow

Worksheet9

Time2

Personnel

Webhook1

Loop

Subflow3

Custom Action23

Approval Flow

External Workflow

PBP2

- File upload failures

Create Record

Failed to upload

Name

Description

Attachment

Add

Add

Drag here or paste the clipboard file

LocalKnowledge

☐ Keep as default

Save as draft

Submit and Next

Submit

Solution:



1. Log in to the server.
2. Edit the `/data/hap/script/docker-compose.yml` file.
3. Modify the value of the variable `ENV_ADDRESS_MAIN`.

In the file, locate the variable named `ENV_ADDRESS_MAIN` and change its value to the same address you use when accessing the system in the browser.

```
app:
  image: registry.cn-hangzhou.aliyuncs.com/mdpublic/mingdaoyun-community
  environment:
    ENV_ADDRESS_MAIN: "http://mdy.domain.com:80"
```

4. Execute `bash service.sh` in the manager directory to restart the HAP service

If you want to keep the original access address working normally, view [Multiple Address Configuration](#) for more details.

# Unable to Access Page

When a page is not accessible, it may be caused by various reasons. This article lists common scenarios and diagnostic methods to help quickly locate and resolve the issue.

## Basic Checks

Before diving into analyzing the issue, please confirm the following basics are correct:

### 1. URL accuracy

- Ensure that the entered page URL is correct, including the protocol (http:// or https://) and whether the path is complete.

### 2. Network environment

- Try changing the network (such as using a mobile hotspot) or device to rule out local network or device issues.

### 3. Service status

- Check the container's running status: Use the `docker ps -a` command to confirm if the container is running.
- View microservice container logs: Use the `docker logs $(docker ps | grep community | awk '{print $1}')` command to check the microservice logs, which should normally be INFO level log outputs.
- If the container is not running or there are many errors in the health check logs, you can try restarting the HAP service by running the `bash service.sh restartall` command in the directory where the manager is located.

## Common Issues and Solutions

### Network Issues

#### Common Prompts

- Unable to access this website.
- Prompt to check network connection, proxy server, or firewall.

- Error message `ERR_CONNECTION_TIMED_OUT`.

## Possible Causes

- Cloud host security group has not opened specific ports.
- Enterprise network firewall restricts external access.
- The server's local firewall (such as `firewalld`) has not opened necessary ports.

## Solutions

1. Check cloud service's network security group (applicable to cloud servers)
  - Log in to the cloud service management platform and check the security group rules bound to the instance.
  - Check the inbound rules to ensure the required ports for services are open and allow access.
2. Check enterprise network firewall
  - Contact the IT department to confirm firewall rules and ensure external traffic is not restricted.
3. Check server firewall configuration
  - Ensure `firewalld` is in a closed state.

## SSL/TLS Certificate Issues

### Common Prompts

- This website cannot provide a secure connection.
- Error message: `ERR_SSL_PROTOCOL_ERROR`.

## Possible Causes

- SSL certificate has expired.
- SSL certificate does not match the accessed domain.
- Using a self-signed certificate but did not install the root certificate.

## Solutions

1. Check the SSL certificate validity period, and if expired, update it immediately.
2. Confirm that the certificate's domain (Common Name or SAN) matches the accessed URL.
3. Ensure the client has installed the corresponding root certificate.

## DNS Configuration Issues

## Common Prompts

- Unable to find the IP address of the server.
- Check for any spelling errors.
- Error messages: `ERR_NAME_NOT_RESOLVED` or `DNS_PROBE_FINISHED_NXDOMAIN`.

## Possible Causes

- DNS records are not set up correctly or resolved.
- Domain name has not been registered or resolved.

## Solutions

1. Use DNS tools (such as nslookup or dig) to check the domain resolution status.
2. Ensure DNS records are configured correctly (such as A record or CNAME record).
3. If DNS configuration has been recently changed, wait a few minutes to ensure DNS cache has been updated.

# Changelog

## NOTES

HAP Server is continuously evolving. In addition to aligning with the feature updates of the SaaS edition, it undergoes its own architectural enhancements and optimizations. By default, only the latest three major versions are maintained (the third digit in the version number indicates the release type: 0 for a major version, and a number greater than 0 for a bug-fix release based on the major version. **For any major version, it's recommended to upgrade to the bug-fix release with the highest third digit.** Additionally, we may extend support for older versions depending on the release date of the latest version).

For versions no longer under maintenance, resources remain available for download, and existing environments will continue to operate normally. However, if bugs arise, users will need to upgrade to a supported version before reporting issues. In the case of critical bugs in maintained versions, we will prioritize fixing them and releasing an updated version.

### ▼ Version under maintenance

Version	Release Date	Detail	Additional Operations	AMD64	ARM64
v6.0.2	2025-01-21	<a href="#">View</a>		✓	
v6.0.1	2025-01-16	<a href="#">View</a>		✓	
v6.0.0	2025-01-13	<a href="#">View</a>	✓	✓	
v5.8.3	2024-12-05	<a href="#">View</a>		✓	
v5.8.2	2024-11-28	<a href="#">View</a>		✓	
v5.8.1	2024-11-19	<a href="#">View</a>		✓	
v5.8.0	2024-11-05	<a href="#">View</a>	✓	✓	
v5.7.1	2024-10-22	<a href="#">View</a>		✓	
v5.7.0	2024-09-29	<a href="#">View</a>	✓	✓	
v5.6.4	2024-09-12	<a href="#">View</a>	✓	✓	
v5.6.3	2024-08-29	<a href="#">View</a>		✓	
v5.6.2	2024-08-15	<a href="#">View</a>		✓	
v5.6.1	2024-08-07	<a href="#">View</a>		✓	
v5.6.0	2024-07-24	<a href="#">View</a>	✓	✓	
v5.5.0	2024-06-24	<a href="#">View</a>	✓	✓	

► Versions not maintained

# SBOM

HAP products integrate multiple open-source components. To ensure product security, we conduct comprehensive security checks using mainstream vulnerability scanning tools before each version release. For components found with medium to high-level security risks, we promptly upgrade or replace them. Below is a list of all open-source components used in the product, including their protocols and version information (some component details may be slightly outdated due to the time lag in statistics).

Component Name	Licenses
@babel/code-frame	MIT
@babel/helper-validator-identifier	MIT
@elastic/elasticsearch	Apache-2.0
@elastic/elasticsearch	Apache-2.0
@elastic/elasticsearch	Apache-2.0
@elastic/transport	Apache-2.0
@elastic/transport	Apache-2.0
@grpc/grpc-js	Apache-2.0
@grpc/proto-loader	Apache-2.0
@hapi/bourne	BSD-3-Clause
@isaacs/cliui	ISC
@isaacs/string-locale-compare	0BSD,ISC
@jimp/bmp	MIT

Component Name	Licenses
@jimp/core	MIT
@jimp/custom	MIT
@jimp/gif	MIT
@jimp/jpeg	MIT
@jimp/plugin-blit	MIT
@jimp/plugin-blur	MIT
@jimp/plugin-circle	MIT
@jimp/plugin-color	MIT
@jimp/plugin-contain	MIT
@jimp/plugin-cover	MIT
@jimp/plugin-crop	MIT
@jimp/plugin-displace	MIT
@jimp/plugin-dither	MIT
@jimp/plugin-fisheye	MIT
@jimp/plugin-flip	MIT
@jimp/plugin-gaussian	MIT
@jimp/plugin-invert	MIT
@jimp/plugin-mask	MIT



Component Name	Licenses
@jimp/plugin-normalize	MIT
@jimp/plugin-print	MIT
@jimp/plugin-resize	MIT
@jimp/plugin-rotate	MIT
@jimp/plugins	MIT
@jimp/plugin-scale	MIT
@jimp/plugin-shadow	MIT
@jimp/plugin-threshold	MIT
@jimp/png	MIT
@jimp/tiff	MIT
@jimp/types	MIT
@jimp/utils	MIT
@js-sdsl/ordered-map	MIT
@npmcli/agent	ISC
@npmcli/arborist	ISC
@npmcli/config	0BSD,ISC
@npmcli/fs	ISC
@npmcli/git	ISC

Component Name	Licenses
@npmcli/installed-package-contents	0BSD,ISC
@npmcli/map-workspaces	ISC
@npmcli/metavuln-calculator	0BSD,ISC
@npmcli/name-from-folder	ISC
@npmcli/node-gyp	ISC
@npmcli/package-json	ISC
@npmcli/promise-spawn	ISC
@npmcli/query	ISC
@npmcli/redact	ISC,MIT
@npmcli/run-script	0BSD,ISC
@one-ini/wasm	MIT
@opentelemetry/api	Apache-2.0
@pkgjs/parseargs	Apache-2.0,MIT
@protobufjs/aspromise	BSD-3-Clause
@protobufjs/base64	BSD-3-Clause
@protobufjs/codegen	BSD-3-Clause
@protobufjs/eventemitter	BSD-3-Clause
@protobufjs/fetch	BSD-3-Clause

Component Name	Licenses
@protobufjs/float	BSD-3-Clause
@protobufjs/inquire	BSD-3-Clause
@protobufjs/path	BSD-3-Clause
@protobufjs/pool	BSD-3-Clause
@protobufjs/utf8	BSD-3-Clause
@puppeteer/browsers	Apache-2.0
@redis/bloom	MIT
@redis/client	MIT
@redis/client	MIT
@redis/graph	MIT
@redis/json	MIT
@redis/search	MIT
@redis/time-series	MIT
@sigstore/bundle	Apache-2.0
@sigstore/core	Apache-2.0
@sigstore/protobuf-specs	Apache-2.0
@sigstore/sign	Apache-2.0
@sigstore/tuf	Apache-2.0

Component Name	Licenses
@sigstore/verify	Apache-2.0
@tokenizer/token	MIT
@tootallnate/quickjs-emscripten	MIT
@tufjs/canonical-json	MIT
@tufjs/models	MIT
@types/bson	MIT
@types/mongodb	MIT
@types/node	MIT
@types/node	MIT
@types/node	MIT
@types/node	MIT
@types/webidl-conversions	MIT
@types/whatwg-url	MIT
@types/yauzl	MIT
1to2	MIT
abbrev	ISC,MIT
accepts	MIT
aead.dev/minisign	MIT

Component Name	Licenses
agent-base	MIT
agentkeepalive	MIT
agentkeepalive	MIT
aggregate-error	MIT
alibabacloud.endpointutil	Unknown
aliyun.credentials	Apache-2.0
ansi-regex	MIT
ansi-regex	MIT
ansi-regex	MIT
ansi-regex	MIT
ansi-regex	MIT
ansi-styles	MIT
ansi-styles	MIT
ansi-styles	MIT
ansi-styles	MIT
any-base	MIT
any-promise	MIT
aproba	0BSD,ISC

Component Name	Licenses
aproba	0BSD,ISC
archy	MIT
are-we-there-yet	ISC
argparse	Python-2.0,Python-2.0.1
array-flatten	MIT
asap	MIT
aspectcore.extensions.reflection	MIT
ast-types	MIT
async	MIT
asynckit	MIT
autofac.engine	Unknown
autofac.engine	Unknown
autofac.extras.dynamicproxy	MIT
automapper	MIT
axios	MIT
b4a	Apache-2.0
backport-util-concurrent:backport-util-concurrent	CC-PDDC,LicenseRef-jfrog-public
balanced-match	MIT

Component Name	Licenses
bare-events	Apache-2.0
bare-fs	Apache-2.0
bare-os	Apache-2.0
bare-path	Apache-2.0
bare-stream	Apache-2.0
base64-js	MIT
basic-auth	MIT
basic-ftp	MIT
beep-boop	Unknown
before	MIT
benchmarks	MIT
binary	MIT
binary-extensions	MIT
bindings	MIT
bin-links	0BSD,ISC
bisection	MIT
bl	MIT
bl	LGPL-3.0,MIT

Component Name	Licenses
block-stream2	MIT
bluebird	MIT
bluebird	MIT
bmp-js	MIT
body-parser	MIT
boolbase	ISC
brace-expansion	MIT
brace-expansion	MIT
bson	Apache-2.0
buffer	MIT
buffer-alloc	MIT
buffer-alloc-unsafe	MIT
buffer-crc32	MIT
buffer-equal	MIT
buffer-fill	MIT
buffermaker	MIT
buffers	Unknown
builtins	MIT



Component Name	Licenses
busboy	MIT
bytes	MIT
cacache	ISC
cache-content-type	MIT
caching.csredis	MIT
call-bind	MIT
callsites	MIT
camelcase	MIT
castle.core	Apache-2.0
castle.windsor	Apache-2.0
centra	MIT
ch.qos.logback:logback-classic	EPL-1.0,GPL-2.0,LGPL-2.1
ch.qos.logback:logback-classic	EPL-1.0,GPL-2.0,LGPL-2.1
ch.qos.logback:logback-core	EPL-1.0,LGPL-2.1
ch.qos.logback:logback-core	EPL-1.0,LGPL-2.1
ch.qos.reload4j:reload4j	Apache-2.0
chainsaw	MIT
chalk	MIT

Component Name	Licenses
chalk	MIT
cheerio	MIT
cheerio-select	BSD-2-Clause
chownr	0BSD,ISC
chownr	0BSD,ISC
chromium-bidi	Apache-2.0
cidr-regex	BSD-2-Clause
ci-info	MIT
classworlds:classworlds	Plexus
clean-stack	MIT
cli-columns	MIT
cliui	ISC
cliui	ISC
cluster-key-slot	Apache-2.0
cmd-shim	0BSD,ISC
cn.hutool:hutool-all	MulanPSL-2.0
co	MIT
co.elastic.clients:elasticsearch-java	Apache-2.0

Component Name	Licenses
co-body	MIT
code-point-at	MIT
color-convert	MIT
color-convert	MIT
color-name	MIT
color-name	MIT
com.alibaba.fastjson2:fastjson2	Apache-2.0
com.alibaba:dashscope-sdk-java	Apache-2.0
com.alibaba:druid	Apache-2.0
com.alibaba:easyexcel	Apache-2.0
com.alibaba:easyexcel-core	Apache-2.0
com.alibaba:easyexcel-support	Apache-2.0
com.alibaba:fastjson	Apache-2.0
com.azure:azure-ai-openai	MIT
com.azure:azure-core	MIT
com.azure:azure-core-http-netty	MIT
com.azure:azure-json	MIT
com.azure:azure-xml	Classpath-exception-2.0,GPL-2.0,MIT

Component Name	Licenses
com.belerweb:pinyin4j	BSD-2-Clause,BSD-3-Clause
com.cronutils:cron-utils	Apache-2.0
com.esotericsoftware.kryo:kryo	BSD-2-Clause,BSD-3-Clause
com.esotericsoftware.minlog:minlog	BSD-2-Clause,BSD-3-Clause
com.esotericsoftware.reflectasm:reflectasm	BSD-2-Clause,BSD-3-Clause
com.fasterxml.jackson.core:jackson-annotations	Apache-2.0
com.fasterxml.jackson.core:jackson-annotations	Apache-2.0
com.fasterxml.jackson.core:jackson-annotations	Apache-2.0
com.fasterxml.jackson.core:jackson-annotations	Apache-2.0
com.fasterxml.jackson.core:jackson-core	Apache-2.0
com.fasterxml.jackson.core:jackson-core	Apache-2.0
com.fasterxml.jackson.core:jackson-core	Apache-2.0
com.fasterxml.jackson.core:jackson-core	Apache-2.0
com.fasterxml.jackson.core:jackson-databind	Apache-2.0
com.fasterxml.jackson.core:jackson-databind	Apache-2.0
com.fasterxml.jackson.core:jackson-databind	Apache-2.0
com.fasterxml.jackson.core:jackson-databind	Apache-2.0
com.fasterxml.jackson.dataformat:jackson-dataformat-xml	Apache-2.0

Component Name	Licenses
com.fasterxml.jackson.datatype:jackson-datatype-jdk8	Apache-2.0
com.fasterxml.jackson.datatype:jackson-datatype-jdk8	Apache-2.0
com.fasterxml.jackson.datatype:jackson-datatype-jdk8	Apache-2.0
com.fasterxml.jackson.datatype:jackson-datatype-jsr310	Apache-2.0
com.fasterxml.jackson.datatype:jackson-datatype-jsr310	Apache-2.0
com.fasterxml.jackson.datatype:jackson-datatype-jsr310	Apache-2.0
com.fasterxml.jackson.datatype:jackson-datatype-jsr310	Apache-2.0
com.fasterxml.jackson.jaxrs:jackson-jaxrs-base	Apache-2.0
com.fasterxml.jackson.jaxrs:jackson-jaxrs-json-provider	Apache-2.0
com.fasterxml.jackson.module:jackson-module-jaxb-annotations	Apache-2.0
com.fasterxml.jackson.module:jackson-module-parameter-names	Apache-2.0
com.fasterxml.jackson.module:jackson-module-parameter-names	Apache-2.0
com.fasterxml.jackson.module:jackson-module-parameter-names	Apache-2.0
com.fasterxml.woodstox:woodstox-core	Apache-2.0
com.fasterxml:classmate	Apache-2.0
com.github.ben-manes.caffeine:caffeine	Apache-2.0
com.github.binarywang:weixin-java-common	Apache-2.0
com.github.binarywang:weixin-java-miniapp	Apache-2.0

Component Name	Licenses
com.github.binarywang:weixin-java-mp	Apache-2.0
com.github.binarywang:weixin-java-open	Apache-2.0
com.github.jedis-lock:jedis-lock	Apache-2.0
com.github.luben:zstd-jni	BSD-2-Clause
com.github.luben:zstd-jni	BSD-2-Clause
com.github.luben:zstd-jni	BSD-2-Clause
com.github.mwiede:jsch	BSD-3-Clause,ISC
com.github.victools:jsonschema-generator	Apache-2.0
com.github.victools:jsonschema-module-jackson	Apache-2.0
com.github.victools:jsonschema-module-swagger-2	Apache-2.0
com.github.virtuald:curvesapi	BSD-3-Clause
com.github.xiaoymin:swagger-bootstrap-ui	Apache-2.0
com.google.android:annotations	Apache-2.0
com.google.api.grpc:proto-google-common-protos	Apache-2.0
com.google.api.grpc:proto-google-common-protos	Apache-2.0
com.google.code.findbugs:jsr305	Apache-2.0
com.google.code.findbugs:jsr305	Apache-2.0
com.google.code.gson:gson	Apache-2.0

Component Name	Licenses
com.google.code.gson:gson	Apache-2.0
com.google.code.gson:gson	Apache-2.0
com.google.code.gson:gson	Apache-2.0
com.google.errorprone:error_prone_annotations	Apache-2.0
com.google.errorprone:error_prone_annotations	Apache-2.0
com.google.guava:failureaccess	Apache-2.0
com.google.guava:guava	Apache-2.0
com.google.guava:guava	Apache-2.0
com.google.guava:listenablefuture	Apache-2.0
com.google.j2objc:j2objc-annotations	Apache-2.0
com.google.j2objc:j2objc-annotations	Apache-2.0
com.google.protobuf:protobuf-java	Apache-2.0,BSD,BSD-2-Clause,BS
com.google.protobuf:protobuf-java-util	BSD-3-Clause
com.google.protobuf:protobuf-java-util	BSD,BSD-2-Clause,BSD-3-Clause
com.ibm.db2:jcc	IPL-1.0
com.ibm.etcd:etcd-java	Apache-2.0
com.ibm.icu:icu4j	ICU
com.jayway.jsonpath:json-path	Apache-2.0

Component Name	Licenses
com.knuddels:jtokit	MIT
com.lmax:disruptor	Apache-2.0
com.luhuiguo:chinese-utils	Apache-2.0
com.maxmind.db:maxmind-db	Apache-2.0
com.maxmind.geoip2:geoip2	Apache-2.0
com.mchange:c3p0	EPL-1.0,LGPL-2.1
com.mchange:mchange-commons-java	EPL-1.0,LGPL-2.1
com.microsoft.sqlserver:mssql-jdbc	MIT
com.mingdao.whileflow:wf-cache	Unknown
com.mingdao.whileflow:wf-common	Unknown
com.mingdao.whileflow:wf-datasource-metadata	Unknown
com.mingdao.whileflow:wf-external-service	Unknown
com.mingdao.whileflow:wf-generator	Unknown
com.mingdao.whileflow:wf-server	Unknown
com.mingdao:grpc-discussion	Unknown
com.mingdao:grpc-inbox	Unknown
com.mingdao:grpc-report	Unknown
com.mingdao:grpc-workflow	Unknown



Component Name	Licenses
com.mingdao:mdactionlog	Unknown
com.mingdao:md-actionlog	Apache-2.0
com.mingdao:md-actionlog	Apache-2.0
com.mingdao:md-app	Apache-2.0
com.mingdao:md-app	Apache-2.0
com.mingdao:md-app	Apache-2.0
com.mingdao:md-approle	Apache-2.0
com.mingdao:md-approle	Apache-2.0
com.mingdao:md-attachment	Apache-2.0
com.mingdao:md-authorization	Apache-2.0
com.mingdao:md-basic	Apache-2.0
com.mingdao:md-basic	Apache-2.0
com.mingdao:md-basic	Apache-2.0
com.mingdao:md-basic	Apache-2.0
com.mingdao:md-basic	Apache-2.0
com.mingdao:md-cache	Unknown
com.mingdao:md-calendar	Apache-2.0
com.mingdao:md-command	Apache-2.0

Component Name	Licenses
com.mingdao:md-command	Apache-2.0
com.mingdao:mdcommand-puppeteer	Unknown
com.mingdao:md-computingschedule	Apache-2.0
com.mingdao:md-core	Unknown
com.mingdao:md-crm	Apache-2.0
com.mingdao:md-crm	Apache-2.0
com.mingdao:md-crm	Apache-2.0
com.mingdao:md-crm	Apache-2.0
com.mingdao:md-datapipeline	Apache-2.0
com.mingdao:md-dingding	Apache-2.0
com.mingdao:mddiscussion	Unknown
com.mingdao:md-discussion	Apache-2.0
com.mingdao:md-email	Apache-2.0
com.mingdao:md-evalex	Apache-2.0
com.mingdao:md-form	Apache-2.0
com.mingdao:md-group	Apache-2.0
com.mingdao:mdinbox	Unknown
com.mingdao:md-inbox	Apache-2.0

Component Name	Licenses
com.mingdao:md-inbox	Apache-2.0
com.mingdao:md-inbox	Apache-2.0
com.mingdao:md-logger	Apache-2.0
com.mingdao:md-logger	Apache-2.0
com.mingdao:md-logger	Apache-2.0
com.mingdao:md-logger	Apache-2.0
com.mingdao:md-logger	Apache-2.0
com.mingdao:md-logger	Apache-2.0
com.mingdao:md-logger	Apache-2.0
com.mingdao:md-logger	Apache-2.0
com.mingdao:md-notification	Apache-2.0
com.mingdao:md-notification	Apache-2.0
com.mingdao:mdocr	Unknown
com.mingdao:md-report	Apache-2.0
com.mingdao:mdreport-consumer	Unknown
com.mingdao:mdreport-core	Unknown
com.mingdao:mdreport-kotlin-service	Unknown
com.mingdao:mdreport-web	Unknown
com.mingdao:md-role	Apache-2.0

Component Name	Licenses
com.mingdao:md-role	Apache-2.0
com.mingdao:md-sdm	Apache-2.0
com.mingdao:md-sms	Apache-2.0
com.mingdao:md-structure	Apache-2.0
com.mingdao:md-templatemessage	Apache-2.0
com.mingdao:md-templatemessage	Apache-2.0
com.mingdao:md-tpuser	Apache-2.0
com.mingdao:md-tpuser	Apache-2.0
com.mingdao:md-workflow	Apache-2.0
com.mingdao:md-workflow-integration	Apache-2.0
com.mingdao:md-workflow-plugin	Apache-2.0
com.mingdao:md-worksheet	Apache-2.0
com.mingdao:md-worksheet	Apache-2.0
com.mingdao:md-worksheet	Apache-2.0
com.mingdao:mdworksheetsearch	Unknown
com.mingdao:md-worksheetsearch	Apache-2.0
com.mingdao:md-workweixin	Apache-2.0
com.mingdao:reminder	Unknown

Component Name	Licenses
com.mingdao:tpuser	Unknown
com.mingdao:workflow-consumer	Unknown
com.mingdao:workflow-core	Unknown
com.mingdao:workflow-integration	Unknown
com.mingdao:workflow-plugin	Unknown
com.mingdao:workflow-web	Unknown
com.mingdao:workflow-web-core	Unknown
com.mingdao:wps	Apache-2.0
com.mysql:mysql-connector-j	GPL-2.0,Universal-FOSS-exception
com.novemberain.quartz:quartz-mongodb	Unknown
com.oracle.database.jdbc:ojdbc11	LicenseRef-jfrog-oracle-futc
com.oracle.database.nls:orai18n	LicenseRef-jfrog-oracle-futc
com.qiniu:qiniu-java-sdk	MIT
com.sap.cloud.db.jdbc:ngdbc	LicenseRef-jfrog-sap-developer-li
com.squareup.okhttp3:logging-interceptor	Apache-2.0
com.squareup.okhttp3:logging-interceptor	Apache-2.0
com.squareup.okhttp3:okhttp	Apache-2.0
com.squareup.okhttp3:okhttp	Apache-2.0

Component Name	Licenses
com.squareup.okhttp3:okhttp-sse	Apache-2.0
com.squareup.okio:okio	Apache-2.0
com.squareup.okio:okio	Apache-2.0
com.squareup.okio:okio-jvm	Apache-2.0
com.squareup.okio:okio-jvm	Apache-2.0
com.sun.activation:jakarta.activation	BSD-3-Clause
com.sun.istack:istack-commons-runtime	BSD-3-Clause
com.sun.xml.bind.jaxb:isorelax	CDDL-1.1,GPL-2.0
com.tencentcloudapi:tencentcloud-sdk-java	Apache-2.0
com.thoughtworks.qdox:qdox	Apache-2.0
com.thoughtworks.xstream:xstream	BSD-3-Clause
com.twitter:chill_2.11	Apache-2.0
com.twitter:chill-java	Apache-2.0
com.typesafe.netty:netty-reactive-streams	Apache-2.0
com.typesafe:config	Apache-2.0
com.udojava:EvalEx	MIT
com.ververica:flink-cdc-base	Apache-2.0
com.ververica:flink-connector-debezium	Apache-2.0

Component Name	Licenses
com.zaxxer:HikariCP	Apache-2.0
com.zaxxer:HikariCP-java7	Apache-2.0
com.zaxxer:SparseBitSet	Apache-2.0
combined-stream	MIT
commander	MIT
common-ancestor-path	0BSD,ISC
commons-beanutils:commons-beanutils	Apache-2.0
commons-cli:commons-cli	Apache-2.0
commons-codec:commons-codec	Apache-2.0,Classpath-exception-2.0
commons-codec:commons-codec	Apache-2.0
commons-collections:commons-collections	Apache-2.0
commons-digester:commons-digester	Apache-2.0
commons-io:commons-io	Apache-2.0
commons-io:commons-io	Apache-2.0
commons-lang:commons-lang	Apache-2.0
commons-logging:commons-logging	Apache-2.0
commons-validator:commons-validator	Apache-2.0
concat-map	MIT

Component Name	Licenses
condense-newlines	MIT
config-chain	MIT
confluent.kafka	Apache-2.0
console-control-strings	ISC
consoletables	MIT
consolidate	MIT
content-disposition	MIT
content-type	MIT
cookie	MIT
cookie	MIT
cookie-parser	MIT
cookies	MIT
cookie-signature	MIT
copy-to	MIT
corepack	MIT
core-util-is	MIT
cosmiconfig	MIT
crc	MIT



Component Name	Licenses
crc32	MIT
cross-spawn	MIT
cross-spawn	MIT
csrediscore	MIT
cssesc	MIT
css-select	BSD-2-Clause
css-what	BSD-2-Clause
dapper	Apache-2.0
data-uri-to-buffer	MIT
debug	MIT
debug	MIT
debug	MIT
debug	MIT
debug	MIT
debug	MIT
decamelize	MIT
decompress-response	MIT
decompress-response	MIT

Component Name	Licenses
deep-equal	MIT
deep-extend	MIT
default-user-agent	MIT
define-data-property	MIT
degenerator	MIT
delayed-stream	MIT
delegates	MIT
denque	Apache-2.0
depd	MIT
depd	MIT
destroy	MIT
detect-libc	Apache-2.0
devtools-protocol	BSD-3-Clause
diff	BSD-3-Clause
digest-header	MIT
dnsclient	Apache-2.0
dnsclient	Apache-2.0
domelementtype	BSD-2-Clause

Component Name	Licenses
domhandler	BSD-2-Clause
dom-serializer	MIT
domutils	BSD-2-Clause
dom-walk	MIT
dotnet-etcd	MIT
eastasianwidth	MIT
editorconfig	MIT
ee-first	MIT
elasticsearch	Apache-2.0
emoji-regex	MIT
emoji-regex	MIT
emoji-regex	MIT
encodeurl	MIT
encodeurl	MIT
encoding	MIT,MIT-0
end-of-stream	MIT
entities	BSD-2-Clause
enums.net	MIT

Component Name	Licenses
enums.net	MIT
env-paths	MIT
epplus	Unknown
err-code	MIT
error-ex	MIT
escalade	MIT
escalade	MIT
escape-html	MIT
escape-string-regexp	MIT
escodegen	BSD-2-Clause
es-define-property	MIT
es-errors	MIT
esprima	BSD,BSD-2-Clause
estaverse	BSD-2-Clause
esutils	BSD-2-Clause
etag	MIT
etcd.configuration	Apache-2.0
exceldatareader	MIT

Component Name	Licenses
exceldatareader.dataset	MIT
exif-parser	MIT
expand-template	MIT
exponential-backoff	Apache-2.0
express	MIT
extend-shallow	MIT
extract-zip	BSD-2-Clause
fastest-levenshtein	MIT
fast-fifo	MIT
fd-slicer	MIT
file-type	MIT
file-uri-to-path	MIT
finalhandler	MIT
find-up	MIT
flink-json-1.17.1.jar	Apache-2.0
flink-table-planner.jar	Apache-2.0,MIT
flink-table-planner.jar	Apache-2.0
follow-redirects	MIT

Component Name	Licenses
foreground-child	ISC
foreground-child	ISC
form-data	MIT
formstream	MIT
forwarded	MIT
fresh	MIT
from2	MIT
fs-constants	MIT
fs-extra	MIT
fs-minipass	0BSD,ISC
fs-minipass	0BSD,ISC
function-bind	MIT
gauge	ISC
generic-pool	MIT
get-caller-file	ISC
get-intrinsic	MIT
get-paths	MIT
get-stream	MIT

Component Name	Licenses
get-uri	MIT
gifwrap	MIT
github.com/acarl005/stripansi	MIT
github.com/ahmetb/go-linq	Apache-2.0
github.com/aliyun/aliyun-oss-go-sdk	MIT
github.com/astaxie/beego	Apache-2.0
github.com/aws/aws-sdk-go	Apache-2.0
github.com/aymanbagabas/go-osc52/v2	MIT
github.com/beorn7/perks	MIT
github.com/cespare/xxhash/v2	MIT
github.com/charmbracelet/bubbles	MIT
github.com/charmbracelet/bubbletea	MIT
github.com/charmbracelet/lipgloss	MIT
github.com/cheggaaa/pb	BSD-3-Clause
github.com/containerd/console	Apache-2.0
github.com/coreos/go-semver	Apache-2.0
github.com/coreos/go-systemd/v22	Apache-2.0
github.com/dustin/go-humanize	MIT

Component Name	Licenses
github.com/fatih/color	MIT
github.com/fatih/structs	MIT
github.com/fsnotify/fsnotify	BSD-3-Clause
github.com/fsnotify/fsnotify	BSD-3-Clause
github.com/gdamore/encoding	Apache-2.0
github.com/gdamore/tcell/v2	Apache-2.0
github.com/globesign/mgo	BSD-2-Clause
github.com/gogo/protobuf	BSD-3-Clause
github.com/go-jose/go-jose/v4	Apache-2.0
github.com/golang/go	BSD-3-Clause
github.com/golang/go	BSD-3-Clause
github.com/golang/protobuf	BSD-3-Clause
github.com/golang/protobuf	BSD-3-Clause
github.com/golang/protobuf	BSD-3-Clause
github.com/golang/snappy	BSD-3-Clause
github.com/golang/snappy	BSD-3-Clause
github.com/golang/snappy	BSD-3-Clause
github.com/golang-jwt/jwt/v4	MIT



Component Name	Licenses
github.com/google/shlex	Unknown
github.com/google/uuid	BSD-3-Clause
github.com/google/uuid	BSD-3-Clause
github.com/go-redis/redis	BSD-2-Clause,Mup
github.com/go-stack/stack	MIT
github.com/grpc-ecosystem/grpc-health-probe	Apache-2.0
github.com/hashicorp/errwrap	MPL-2.0
github.com/hashicorp/go-multierror	MPL-2.0
github.com/hashicorp/hcl	MPL-2.0
github.com/huandu/xstrings	MIT
github.com/imdario/mergo	BSD-3-Clause
github.com/inconshreveable/mousetrap	Apache-2.0
github.com/jedib0t/go-pretty/v6	MIT
github.com/jessevdk/go-flags	BSD-3-Clause
github.com/jmespath/go-jmespath	Apache-2.0
github.com/json-iterator/go	MIT
github.com/juju/ratelimit	LGPL-3.0-linking-exception
github.com/klauspost/compress	Apache-2.0,BSD-3-Clause,MIT

Component Name	Licenses
github.com/klauspost/compress	Apache-2.0,BSD-3-Clause,MIT
github.com/klauspost/cpuid/v2	MIT
github.com/lestrrat-go/backoff/v2	MIT
github.com/lestrrat-go/blackmagic	MIT
github.com/lestrrat-go/httpcc	MIT
github.com/lestrrat-go/iter	MIT
github.com/lestrrat-go/jwx	MIT
github.com/lestrrat-go/option	MIT
github.com/lucasb-eyer/go-colorful	MIT
github.com/magiconair/properties	BSD-2-Clause,Mup
github.com/magiconair/properties	BSD-2-Clause,Mup
github.com/Masterminds/goutils	Apache-2.0
github.com/Masterminds/semver/v3	MIT
github.com/Masterminds/sprig/v3	MIT
github.com/mattn/go-colorable	MIT
github.com/mattn/go-ieproxy	MIT
github.com/mattn/go-isatty	MIT
github.com/mattn/go-localereader	Unknown

Component Name	Licenses
github.com/mattn/go-runewidth	MIT
github.com/mattn/go-runewidth	MIT
github.com/mattn/go-runewidth	MIT
github.com/matttproud/golang_protobuf_extensions	Apache-2.0
github.com/matttproud/golang_protobuf_extensions/v2	Apache-2.0
github.com/minio/cli	Unknown
github.com/minio/colorjson	Unknown
github.com/minio/filepath	Unknown
github.com/minio/md5-simd	Apache-2.0,BSD-3-Clause
github.com/minio/minio-go/v7	Apache-2.0
github.com/minio/selfupdate	Apache-2.0
github.com/minio/sha256-simd	Apache-2.0
github.com/mitchellh/colorstring	MIT
github.com/mitchellh/copystructure	MIT
github.com/mitchellh/go-homedir	MIT
github.com/mitchellh/mapstructure	MIT
github.com/mitchellh/mapstructure	MIT
github.com/mitchellh/reflectwalk	MIT

Component Name	Licenses
github.com/modern-go/concurrent	Apache-2.0
github.com/modern-go/reflect2	Apache-2.0
github.com/mongodb/mongo-tools	Apache-2.0
github.com/montanaflynn/stats	MIT
github.com/montanaflynn/stats	MIT
github.com/muesli/ansi	MIT
github.com/muesli/cancelreader	MIT
github.com/muesli/reflow	MIT
github.com/muesli/termenv	MIT
github.com/navidys/tvxwidgets	MIT
github.com/nsf/termbox-go	MIT
github.com/olekukonko/tablewriter	MIT
github.com/pelletier/go-toml	MIT
github.com/pelletier/go-toml/v2	MIT
github.com/philhofer/fwd	MIT
github.com/pkg/errors	BSD-2-Clause
github.com/pkg/xattr	BSD-2-Clause,Mup
github.com/posener/complete	MIT

Component Name	Licenses
github.com/prometheus/client_golang	Apache-2.0
github.com/prometheus/client_model	Apache-2.0
github.com/prometheus/common	Apache-2.0
github.com/prometheus/procfs	Apache-2.0
github.com/prometheus/prom2json	Apache-2.0
github.com/qiniu/go-sdk/v7	MIT
github.com/qiniu/qshell/v2	MIT
github.com/rivo/tview	MIT
github.com/rivo/uniseg	MIT
github.com/rivo/uniseg	MIT
github.com/rjeczalik/notify	MIT
github.com/robfig/cron	MIT
github.com/rs/xid	MIT
github.com/safchain/ethtool	Apache-2.0
github.com/schollz/progressbar/v3	MIT
github.com/secure-io/sio-go	MIT
github.com/shiena/ansicolor	MIT
github.com/shirou/gopsutil/v3	BSD-3-Clause

Component Name	Licenses
github.com/shopspring/decimal	MIT
github.com/spf13/afero	Apache-2.0
github.com/spf13/afero	Apache-2.0
github.com/spf13/cast	MIT
github.com/spf13/cast	MIT
github.com/spf13/cobra	Apache-2.0
github.com/spf13/jwalterweatherman	MIT
github.com/spf13/pflag	BSD-3-Clause
github.com/spf13/viper	MIT
github.com/spf13/viper	MIT
github.com/spiffe/go-spiffe/v2	Apache-2.0
github.com/subosito/gotenv	MIT
github.com/subosito/gotenv	MIT
github.com/syndtr/goleveldb	BSD-2-Clause
github.com/tidwall/gjson	MIT
github.com/tidwall/match	MIT
github.com/tidwall/pretty	MIT
github.com/tinylib/msgp	MIT

Component Name	Licenses
github.com/tjfoc/gmsm	Apache-2.0
github.com/tklauser/go-sysconf	BSD-3-Clause
github.com/tklauser/numcpus	Apache-2.0
github.com/vbauerster/mpb/v8	Unlicense
github.com/VividCortex/ewma	MIT
github.com/xdg-go/pbkdf2	Apache-2.0
github.com/xdg-go/scram	Apache-2.0
github.com/xdg-go/scram	Apache-2.0
github.com/xdg-go/stringprep	Apache-2.0
github.com/xdg-go/stringprep	Apache-2.0
github.com/youmark/pkcs8	Unknown
github.com/youmark/pkcs8	MIT
github.com/zeebo/errs	MIT
github-from-package	MIT
glob	0BSD,ISC
glob	0BSD,ISC
global	MIT
go.etcd.io/etcd/api/v3	Apache-2.0

Component Name	Licenses
go.etcd.io/etcd/client/pkg/v3	Apache-2.0
go.etcd.io/etcd/client/v3	Apache-2.0
go.mongodb.org/mongo-driver	Apache-2.0
go.mongodb.org/mongo-driver	Apache-2.0
go.mongodb.org/mongo-driver	Apache-2.0
go.uber.org/multierr	MIT
go.uber.org/zap	MIT
golang.org/x/crypto	BSD-3-Clause
golang.org/x/crypto	BSD-3-Clause
golang.org/x/exp	BSD-3-Clause
golang.org/x/net	BSD-3-Clause
golang.org/x/net	BSD-3-Clause
golang.org/x/sync	BSD-3-Clause
golang.org/x/sync	BSD-3-Clause
golang.org/x/sync	BSD-3-Clause
golang.org/x/sys	BSD-3-Clause
golang.org/x/sys	BSD-3-Clause
golang.org/x/sys	BSD-3-Clause



Component Name	Licenses
golang.org/x/term	BSD-3-Clause
golang.org/x/term	BSD-3-Clause
golang.org/x/text	BSD-3-Clause
golang.org/x/text	BSD-3-Clause
golang.org/x/text	BSD-3-Clause
golang.org/x/time	BSD-3-Clause
google.golang.org/genproto	Apache-2.0
google.golang.org/genproto/googleapis/api	Apache-2.0
google.golang.org/genproto/googleapis/rpc	Apache-2.0
google.golang.org/genproto/googleapis/rpc	Apache-2.0
google.golang.org/genproto/googleapis/rpc	Apache-2.0
google.golang.org/genproto/googleapis/rpc	Apache-2.0
google.golang.org/grpc	Apache-2.0
google.golang.org/grpc	Apache-2.0
google.golang.org/grpc	Apache-2.0
google.golang.org/grpc	Apache-2.0
google.golang.org/protobuf	BSD-3-Clause
google.golang.org/protobuf	BSD-3-Clause
google.golang.org/protobuf	BSD-3-Clause

Component Name	Licenses
gopd	MIT
gopkg.in/h2non/filetype.v1	MIT
gopkg.in/ini.v1	Apache-2.0
gopkg.in/ini.v1	Apache-2.0
gopkg.in/mgo.v2	BSD-2-Clause,Mup
gopkg.in/tomb.v2	BSD-3-Clause
gopkg.in/yaml.v2	Apache-2.0,MIT
gopkg.in/yaml.v2	Apache-2.0,MIT
gopkg.in/yaml.v3	Apache-2.0,MIT
gopkg.in/yaml.v3	Apache-2.0,MIT
graceful-fs	ISC
group.springframework.ai:spring-ai-azure-openai	Apache-2.0
group.springframework.ai:spring-ai-core	Apache-2.0
grpc.aspnetcore.healthchecks	Apache-2.0
grpc.aspnetcore.server	Apache-2.0
grpc.aspnetcore.server	Apache-2.0
grpc.aspnetcore.server	Apache-2.0
grpc.aspnetcore.server.clientfactory	Apache-2.0

Component Name	Licenses
grpc.aspnetcore.server.clientfactory	Apache-2.0
grpc.core	Apache-2.0
grpc.core.api	Apache-2.0
grpc.core.api	Apache-2.0
grpc.healthcheck	Apache-2.0
grpc.net.client	Apache-2.0
grpc.net.client	Apache-2.0
grpc.net.clientfactory	Apache-2.0
grpc.net.clientfactory	Apache-2.0
grpc.net.common	Apache-2.0
grpc.net.common	Apache-2.0
grpc.net.common	Apache-2.0
grpc-protobuf-lite-1.53.0.jar	Unknown
has-ansi	MIT
hashring	MIT
hasown	MIT
has-property-descriptors	MIT
has-proto	MIT

Component Name	Licenses
has-symbols	MIT
has-tostringtag	MIT
has-unicode	0BSD,ISC
hosted-git-info	0BSD,ISC
Housekeeper	Unknown
hpagent	MIT
htmlparser2	MIT
http-assert	MIT
http-cache-semantics	BSD-2-Clause
http-errors	MIT
http-errors	MIT
http-errors	MIT
http-proxy-agent	MIT
https	ISC
https-proxy-agent	MIT
https-proxy-agent	MIT
humanize-ms	MIT
iconv-lite	MIT

Component Name	Licenses
iconv-lite	MIT
ieee754	BSD-3-Clause
ignore-walk	0BSD,ISC
image-q	MIT
image-size	MIT
immediate	MIT
import-fresh	MIT
imurmurhash	MIT
indent-string	MIT
inflation	MIT
inherits	0BSD,ISC
inherits	0BSD,ISC
ini	0BSD,ISC
ini	0BSD,ISC
init-package-json	ISC
into-stream	MIT
io.debezium:debezium-embedded	Apache-2.0
io.github.x-stream:mxparser	IndianaUniversityExtreme!LabSoftw

Component Name	Licenses
io.grpc:grpc-api	Apache-2.0
io.grpc:grpc-api	Apache-2.0
io.grpc:grpc-context	Apache-2.0
io.grpc:grpc-context	Apache-2.0
io.grpc:grpc-core	Apache-2.0
io.grpc:grpc-core	Apache-2.0
io.grpc:grpc-netty-shaded	Apache-2.0
io.grpc:grpc-netty-shaded	Apache-2.0
io.grpc:grpc-protobuf	Apache-2.0
io.grpc:grpc-protobuf	Apache-2.0
io.grpc:grpc-protobuf-lite	Apache-2.0
io.grpc:grpc-services	Apache-2.0
io.grpc:grpc-services	Apache-2.0
io.grpc:grpc-stub	Apache-2.0
io.grpc:grpc-stub	Apache-2.0
io.gsonfire:gson-fire	Apache-2.0
io.lettuce:lettuce-core	Apache-2.0
io.micrometer:micrometer-commons	Apache-2.0

Component Name	Licenses
io.micrometer:micrometer-observation	Apache-2.0
io.netty:netty-buffer	Apache-2.0
io.netty:netty-buffer	Apache-2.0
io.netty:netty-codec	Apache-2.0
io.netty:netty-codec	Apache-2.0
io.netty:netty-codec-dns	Apache-2.0
io.netty:netty-codec-http	Apache-2.0
io.netty:netty-codec-http	Apache-2.0
io.netty:netty-codec-http2	Apache-2.0
io.netty:netty-codec-socks	Apache-2.0
io.netty:netty-common	Apache-2.0
io.netty:netty-common	Apache-2.0
io.netty:netty-handler	Apache-2.0
io.netty:netty-handler	Apache-2.0
io.netty:netty-handler-proxy	Apache-2.0
io.netty:netty-resolver	Apache-2.0
io.netty:netty-resolver	Apache-2.0
io.netty:netty-resolver-dns	Apache-2.0

Component Name	Licenses
io.netty:netty-resolver-dns-classes-macos	Apache-2.0
io.netty:netty-resolver-dns-native-macos	Apache-2.0
io.netty:netty-tcnative-boringssl-static	Apache-2.0
io.netty:netty-tcnative-classes	Apache-2.0
io.netty:netty-transport	Apache-2.0
io.netty:netty-transport	Apache-2.0
io.netty:netty-transport-classes-epoll	Apache-2.0
io.netty:netty-transport-classes-kqueue	Apache-2.0
io.netty:netty-transport-native-epoll	Apache-2.0
io.netty:netty-transport-native-kqueue	Apache-2.0
io.netty:netty-transport-native-unix-common	Apache-2.0
io.netty:netty-transport-native-unix-common	Apache-2.0
io.perfmark:perfmark-api	Apache-2.0
io.perfmark:perfmark-api	Apache-2.0
io.projectreactor.netty:reactor-netty-core	Apache-2.0
io.projectreactor.netty:reactor-netty-http	Apache-2.0
io.projectreactor:reactor-core	Apache-2.0
io.prometheus:simpleclient	Apache-2.0



Component Name	Licenses
io.prometheus:simpleclient_common	Apache-2.0
io.prometheus:simpleclient_pushgateway	Apache-2.0
io.prometheus:simpleclient_tracer_common	Apache-2.0
io.prometheus:simpleclient_tracer_otel	Apache-2.0
io.prometheus:simpleclient_tracer_otel_agent	Apache-2.0
io.reactivex.rxjava2:rxjava	Apache-2.0
io.springfox:springfox-core	Apache-2.0
io.springfox:springfox-schema	Apache-2.0
io.springfox:springfox-spi	Apache-2.0
io.springfox:springfox-spring-web	Apache-2.0
io.springfox:springfox-swagger2	Apache-2.0
io.springfox:springfox-swagger-common	Apache-2.0
io.swagger.core.v3:swagger-annotations	Apache-2.0
io.swagger.core.v3:swagger-annotations	Apache-2.0
io.swagger:swagger-annotations	Apache-2.0
io.swagger:swagger-models	Apache-2.0
io.undertow:undertow-core	Apache-2.0
io.undertow:undertow-core	Apache-2.0

Component Name	Licenses
io.undertow:undertow-servlet	Apache-2.0
io.undertow:undertow-servlet	Apache-2.0
io.undertow:undertow-websockets-jsr	Apache-2.0
ioredis	MIT
ipaddr.js	MIT
ip-address	MIT
ip-regex	MIT
isarray	MIT
isarray	MIT
is-arrayish	MIT
is-buffer	MIT
is-cidr	BSD-2-Clause
is-class-hotfix	MIT
is-core-module	MIT
isexe	0BSD,ISC
isexe	0BSD,ISC
is-extendable	MIT
is-fullwidth-code-point	MIT

Component Name	Licenses
is-fullwidth-code-point	MIT
is-fullwidth-code-point	MIT
is-function	MIT
is-generator-function	MIT
is-lambda	MIT
isomorphic-fetch	MIT
isstream	MIT
is-type-of	MIT
is-whitespace	MIT
jackpot	Unknown
jackspeak	BlueOak-1.0.0
jackspeak	BlueOak-1.0.0
jakarta.activation:jakarta.activation-api	BSD-3-Clause
jakarta.annotation:jakarta.annotation-api	Classpath-exception-2.0,EPL-2.0,CC-BY-NC-SA-4.0
jakarta.annotation:jakarta.annotation-api	Classpath-exception-2.0,EPL-2.0,CC-BY-NC-SA-4.0
jakarta.json.bind:jakarta.json.bind-api	Classpath-exception-2.0,EPL-2.0,CC-BY-NC-SA-4.0
jakarta.json:jakarta.json-api	Classpath-exception-2.0,EPL-2.0,CC-BY-NC-SA-4.0
jakarta.servlet:jakarta.servlet-api	Classpath-exception-2.0,EPL-2.0,CC-BY-NC-SA-4.0

Component Name	Licenses
jakarta.servlet:jakarta.servlet-api	Classpath-exception-2.0,EPL-2.0,C
jakarta.validation:jakarta.validation-api	Apache-2.0
jakarta.websocket:jakarta.websocket-api	Classpath-exception-2.0,EPL-2.0,C
jakarta.ws.rs:jakarta.ws.rs-api	Classpath-exception-2.0,EPL-2.0,C
jakarta.xml.bind:jakarta.xml.bind-api	BSD-3-Clause
javax.activation:activation	CDDL-1.0
javax.activation:javax.activation-api	CDDL-1.0,Classpath-exception-2.0
javax.annotation:jsr250-api	CDDL-1.0
javax.cache:cache-api	Apache-2.0
javax.inject:javax.inject	Apache-2.0
javax.servlet:javax.servlet-api	CDDL-1.0,CDDL-1.1,Classpath-exception-2.0
javax.ws.rs:javax.ws.rs-api	Classpath-exception-2.0,EPL-2.0,C
javax.xml.bind:jaxb-api	CDDL-1.1,Classpath-exception-2.0
jimp	MIT
jint	BSD-2-Clause
jpeg-js	BSD-3-Clause
jrt-fs.jar	Unknown

Component Name	Licenses
jrt-fs.jar	Unknown
js-beautify	MIT
jsbn	MIT,MIT-Wu
js-cookie	MIT
jsonfile	MIT
json-parse-even-better-errors	MIT
json-parse-even-better-errors	MIT
json-stringify-nice	OBSD,ISC
js-tokens	MIT
js-yaml	MIT
jszip	GPL-3.0,MIT
just-diff	MIT
just-diff-apply	MIT
kafka-node	MIT
kareem	Apache-2.0
keygrip	MIT
kind-of	MIT
koa	MIT

Component Name	Licenses
koa-bodyparser	MIT
koa-compose	MIT
koa-compose	MIT
koa-convert	MIT
koa-favicon	MIT
koa-router	MIT
koa-send	MIT
koa-send	MIT
koa-session	MIT
koa-static	MIT
koa-views	MIT
libnpmaccess	0BSD,ISC
libnpmdiff	0BSD,ISC
libnpmexec	0BSD,ISC
libnpmfund	0BSD,ISC
libnpmhook	ISC
libnpmorg	0BSD,ISC
libnpm-pack	0BSD,ISC

Component Name	Licenses
libnmpublish	0BSD,ISC
libnmpsearch	0BSD,ISC
libnmpteam	0BSD,ISC
libnmpversion	0BSD,ISC
LICENSE.gz	Apache-2.0,BSD-2-Clause,License iccjpeg
lie	MIT
line-reader	MIT
lines-and-columns	MIT
load-bmfont	MIT
locate-path	MIT
long	Apache-2.0
long	Apache-2.0
longbow.cronos	MIT
longbow.logging	Unknown
longbow.tasks	Unknown
lru-cache	0BSD,ISC
lru-cache	0BSD,ISC

Component Name	Licenses
lru-cache	0BSD,ISC
lru-cache	0BSD,ISC
make-fetch-happen	ISC
mapster	MIT
mapster.core	MIT
marked	BSD-3-Clause,MIT
mavenEcjBootstrapAgent.jar	Unknown
mavenEcjBootstrapAgent.jar	Unknown
MDDataManager	Unknown
media-typer	MIT
memcached	MIT
memory-pager	MIT
merge-descriptors	MIT
methods	MIT
microsoft.aspnetcore.jsonpatch	MIT
microsoft.aspnetcore.mvc.newtonsoftjson	MIT
microsoft.dotnet.platformabstractions	MIT
microsoft.dotnet.platformabstractions	MIT



Component Name	Licenses
microsoft.extensions.configuration.binder	MIT
microsoft.extensions.dependencymodel	MIT
microsoft.openapi	MIT
microsoft.win32.systemevents	MIT
mime	MIT
mime	MIT
mime-db	MIT
mime-types	MIT
mimic-response	MIT
mimic-response	MIT
min-document	MIT
mingdaoyun-community	Unknown
minimatch	0BSD,ISC
minimatch	0BSD,ISC
minimatch	0BSD,ISC
minimatch	0BSD,ISC
minimist	MIT
minipass	0BSD,ISC

Component Name	Licenses
minipass	0BSD,ISC
minipass	0BSD,ISC
minipass	0BSD,ISC
minipass-collect	0BSD,ISC
minipass-fetch	MIT
minipass-flush	0BSD,ISC
minipass-json-stream	MIT
minipass-pipeline	0BSD,ISC
minipass-sized	0BSD,ISC
minizlib	MIT
mitt	MIT
mkdirp	MIT
mkdirp	MIT
mockdate	MIT
mongo:mongo	Unknown
mongodb	Apache-2.0
mongodb.driver.core	Apache-2.0
mongodb.libmongocrypt	Apache-2.0

Component Name	Licenses
mongodb-connection-string-url	Apache-2.0
mongoose	MIT
mongoose	MIT
mongoose-legacy-pluralize	Apache-2.0
morelinq	Apache-2.0,MIT
morelinq	Apache-2.0
morelinq	Apache-2.0
morgan	MIT
mpath	MIT
mquery	MIT
ms	MIT
ms	MIT
ms	MIT
mssqlhelper	Unknown
mute-stream	0BSD,ISC
mz	MIT
nan	MIT
napi-build-utils	MIT

Component Name	Licenses
negotiator	MIT
nested-error-stacks	MIT
net.bytebuddy:byte-buddy	Apache-2.0
net.devh:grpc-client-spring-boot-autoconfigure	MIT
net.devh:grpc-client-spring-boot-autoconfigure	MIT
net.devh:grpc-client-spring-boot-starter	MIT
net.devh:grpc-client-spring-boot-starter	MIT
net.devh:grpc-common-spring-boot	MIT
net.devh:grpc-common-spring-boot	MIT
net.devh:grpc-server-spring-boot-autoconfigure	MIT
net.devh:grpc-server-spring-boot-autoconfigure	MIT
net.devh:grpc-server-spring-boot-starter	MIT
net.devh:grpc-server-spring-boot-starter	MIT
net.devh:grpc-spring-boot-starter	MIT
net.java.dev.jna:jna	Apache-2.0,LGPL-2.1
net.java.dev.msv:xsdlib	BSD-3-Clause
net.javacrumbs.shedlock:shedlock-core	Apache-2.0
net.javacrumbs.shedlock:shedlock-core	Apache-2.0

Component Name	Licenses
net.javacrumbs.shedlock:shedlock-core	Apache-2.0
net.javacrumbs.shedlock:shedlock-core	Apache-2.0
net.javacrumbs.shedlock:shedlock-provider-mongo	Apache-2.0
net.javacrumbs.shedlock:shedlock-provider-redis-spring	Apache-2.0
net.javacrumbs.shedlock:shedlock-provider-redis-spring	Apache-2.0
net.javacrumbs.shedlock:shedlock-provider-redis-spring	Apache-2.0
net.javacrumbs.shedlock:shedlock-spring	Apache-2.0
net.javacrumbs.shedlock:shedlock-spring	Apache-2.0
net.javacrumbs.shedlock:shedlock-spring	Apache-2.0
net.javacrumbs.shedlock:shedlock-spring	Apache-2.0
net.jodah:typetools	Apache-2.0
net.minidev:accessors-smart	Apache-2.0
net.minidev:json-smart	Apache-2.0
net.sourceforge.args4j:args4j	MIT
netmask	MIT
newtonsoft.json.bson	MIT
node:node	Unknown
node-abi	MIT

Component Name	Licenses
node-fetch	MIT
node-gyp	MIT
node-hex	MIT
node-xlsx	Apache-2.0
noop-logger	MIT
nopt	0BSD,ISC
nopt	0BSD,ISC
normalize-package-data	BSD-2-Clause
npgsql	PostgreSQL
npm	Artistic-2.0
npm-audit-report	ISC
npm-bundled	0BSD,ISC
npm-install-checks	BSD-2-Clause
npmlog	0BSD,ISC
npm-normalize-package-bin	0BSD,ISC
npm-package-arg	0BSD,ISC
npm-packlist	0BSD,ISC
npm-pick-manifest	ISC

Component Name	Licenses
npm-profile	ISC
npm-registry-fetch	ISC
npm-user-validate	BSD-2-Clause
npoi	Apache-2.0
nth-check	BSD-2-Clause
number-is-nan	MIT
object-assign	MIT
object-inspect	MIT
omggif	MIT
once	0BSD,ISC
on-finished	MIT
on-finished	MIT
on-headers	MIT
only	Unknown
optional	MIT
optional-require	Apache-2.0
optional-require	Apache-2.0
org.antlr:antlr4-runtime	ANTLR-PD,BSD-3-Clause

Component Name	Licenses
organtlr:antlr-runtime	ANTLR-PD,BSD
organtlr:stringtemplate	ANTLR-PD,BSD-3-Clause
org.apache.commons:commons-collections4	Apache-2.0
org.apache.commons:commons-compress	Apache-2.0,Classpath-exception-2.0
org.apache.commons:commons-compress	Apache-2.0
org.apache.commons:commons-csv	Apache-2.0
org.apache.commons:commons-lang3	Apache-2.0
org.apache.commons:commons-lang3	Apache-2.0
org.apache.commons:commons-math3	Apache-2.0
org.apache.commons:commons-pool2	Apache-2.0
org.apache.commons:commons-text	Apache-2.0
org.apache.flink:flink-annotations	Apache-2.0
org.apache.flink:flink-annotations	Apache-2.0
org.apache.flink:flink-cep	Apache-2.0
org.apache.flink:flink-clients	Apache-2.0
org.apache.flink:flink-clients	Apache-2.0
org.apache.flink:flink-connector-base	Apache-2.0
org.apache.flink:flink-connector-base	Apache-2.0



Component Name	Licenses
org.apache.flink:flink-connector-files	Apache-2.0
org.apache.flink:flink-connector-jdbc	Apache-2.0
org.apache.flink:flink-connector-kafka	Apache-2.0
org.apache.flink:flink-connector-oracle-cdc	Apache-2.0
org.apache.flink:flink-core	Apache-2.0
org.apache.flink:flink-core	Apache-2.0
org.apache.flink:flink-file-sink-common	Apache-2.0
org.apache.flink:flink-file-sink-common	Apache-2.0
org.apache.flink:flink-hadoop-fs	Apache-2.0
org.apache.flink:flink-hadoop-fs	Apache-2.0
org.apache.flink:flink-java	Apache-2.0
org.apache.flink:flink-java	Apache-2.0
org.apache.flink:flink-json	Apache-2.0
org.apache.flink:flink-json	Apache-2.0
org.apache.flink:flink-metrics-core	Apache-2.0
org.apache.flink:flink-metrics-core	Apache-2.0
org.apache.flink:flink-optimizer	Apache-2.0
org.apache.flink:flink-optimizer	Apache-2.0

Component Name	Licenses
org.apache.flink:flink-queryable-state-client-java	Apache-2.0
org.apache.flink:flink-queryable-state-client-java	Apache-2.0
org.apache.flink:flink-rpc-akka	Apache-2.0
org.apache.flink:flink-rpc-akka-loader	Apache-2.0
org.apache.flink:flink-rpc-core	Apache-2.0
org.apache.flink:flink-runtime	Apache-2.0
org.apache.flink:flink-runtime	Apache-2.0
org.apache.flink:flink-scala_2.11	Apache-2.0
org.apache.flink:flink-shaded-asm-9	Apache-2.0
org.apache.flink:flink-shaded-asm-9	Apache-2.0
org.apache.flink:flink-shaded-force-shading	Apache-2.0
org.apache.flink:flink-shaded-force-shading	Apache-2.0
org.apache.flink:flink-shaded-guava	Apache-2.0
org.apache.flink:flink-shaded-jackson	Apache-2.0
org.apache.flink:flink-shaded-jackson	Apache-2.0
org.apache.flink:flink-shaded-netty	Apache-2.0
org.apache.flink:flink-shaded-netty	Apache-2.0
org.apache.flink:flink-shaded-zookeeper-3	Apache-2.0

Component Name	Licenses
org.apache.flink:flink-shaded-zookeeper-3	Apache-2.0
org.apache.flink:flink-sql-client_2.11	Apache-2.0
org.apache.flink:flink-sql-connector-mongodb	Apache-2.0
org.apache.flink:flink-statebackend-rocksdb	Apache-2.0
org.apache.flink:flink-state-processor-api_2.11	Apache-2.0
org.apache.flink:flink-streaming-java	Apache-2.0
org.apache.flink:flink-streaming-java	Apache-2.0
org.apache.flink:flink-table-api-bridge-base	Apache-2.0
org.apache.flink:flink-table-api-bridge-base	Apache-2.0
org.apache.flink:flink-table-api-java	Apache-2.0
org.apache.flink:flink-table-api-java	Apache-2.0
org.apache.flink:flink-table-api-java-bridge	Apache-2.0
org.apache.flink:flink-table-api-java-bridge	Apache-2.0
org.apache.flink:flink-table-api-java-uber	Apache-2.0
org.apache.flink:flink-table-api-java-uber	Apache-2.0
org.apache.flink:flink-table-api-scala_2.11	Apache-2.0
org.apache.flink:flink-table-api-scala-bridge_2.12	Apache-2.0
org.apache.flink:flink-table-common	Apache-2.0

Component Name	Licenses
org.apache.flink:flink-table-common	Apache-2.0
org.apache.flink:flink-table-planner_2.11	Apache-2.0
org.apache.flink:flink-table-planner-blink_2.11	Apache-2.0
org.apache.flink:flink-table-planner-loader	Apache-2.0
org.apache.flink:flink-table-planner-loader	Apache-2.0
org.apache.flink:flink-table-runtime	Apache-2.0
org.apache.flink:flink-table-runtime	Apache-2.0
org.apache.flink:flink-table-runtime-blink_2.11	Apache-2.0
org.apache.flink:force-shading	Apache-2.0
org.apache.httpcomponents:httpasyncclient	Apache-2.0
org.apache.httpcomponents:httpclient	Apache-2.0
org.apache.httpcomponents:httpclient	Apache-2.0
org.apache.httpcomponents:httpcore	Apache-2.0
org.apache.httpcomponents:httpcore-nio	Apache-2.0
org.apache.httpcomponents:httpmime	Apache-2.0
org.apache.kafka:connect-file	Apache-2.0
org.apache.kafka:connect-json	Apache-2.0
org.apache.kafka:connect-transforms	Apache-2.0

Component Name	Licenses
org.apache.kafka:kafka-clients	Apache-2.0
org.apache.kafka:kafka-clients	Apache-2.0
org.apache.kafka:kafka-clients	Apache-2.0
org.apache.kafka:kafka-tools	Apache-2.0
org.apache.logging.log4j:log4j-api	Apache-2.0
org.apache.logging.log4j:log4j-api	Apache-2.0
org.apache.logging.log4j:log4j-to-slf4j	Apache-2.0
org.apache.logging.log4j:log4j-to-slf4j	Apache-2.0
org.apache.maven.wagon:wagon-provider-api	Apache-2.0
org.apache.maven:maven-artifact	Apache-2.0
org.apache.oltu.oauth2:org.apache.oltu.oauth2.client	Apache-2.0
org.apache.oltu.oauth2:org.apache.oltu.oauth2.common	Apache-2.0
org.apache.poi:poi	Apache-2.0
org.apache.poi:poi-ooxml	Apache-2.0
org.apache.poi:poi-ooxml-schemas	Apache-2.0
org.apache.tomcat.embed:tomcat-embed-core	Apache-2.0
org.apache.tomcat.embed:tomcat-embed-el	Apache-2.0
org.apache.tomcat.embed:tomcat-embed-el	Apache-2.0

Component Name	Licenses
org.apache.tomcat.embed:tomcat-embed-websocket	Apache-2.0
org.apache.tomcat:tomcat-annotations-api	Apache-2.0
org.apache.xmlbeans:xmlbeans	Apache-2.0
org.apiguardian:apiguardian-api	Apache-2.0
org.aspectj:aspectjweaver	EPL-2.0
org.aspectj:aspectjweaver	EPL-2.0
org.assertj:assertj-core	Apache-2.0
org.asynchttpclient:async-http-client	Apache-2.0
org.asynchttpclient:async-http-client-netty-utils	Apache-2.0
org.attoparser:attoparser	Apache-2.0
org.awaitility:awaitility	Apache-2.0
org.bouncycastle:bcpkix-jdk15on	MIT
org.bouncycastle:bcutil-jdk15on	MIT
org.checkerframework:checker-compat-qual	Classpath-exception-2.0,GPL-2.0,MIT
org.checkerframework:checker-qual	MIT
org.clapper:grizzled-slf4j_2.11	BSD-3-Clause
org.codehaus.groovy:groovy	Apache-2.0
org.codehaus.janino:janino	BSD-3-Clause

Component Name	Licenses
org.codehaus.mojo:animal-sniffer-annotations	MIT
org.codehaus.mojo:animal-sniffer-annotations	MIT
org.codehaus.plexus:plexus-utils	Apache-2.0
org.codehaus.woodstox:stax2-api	BSD-2-Clause,BSD-3-Clause
org.dom4j:dom4j	Plexus
org.dom4j:dom4j	BSD-3-Clause
org.eclipse.parsson:parsson	Classpath-exception-2.0,EPL-2.0,CC-BY-NC-SA-4.0
org.ehcache.modules:ehcache-107	Apache-2.0
org.ehcache.modules:ehcache-api	Apache-2.0
org.ehcache.modules:ehcache-core	Apache-2.0
org.ehcache.modules:ehcache-impl	Apache-2.0
org.ehcache.modules:ehcache-xml	Apache-2.0
org.ehcache.modules:ehcache-xml-spi	Apache-2.0
org.ehcache:ehcache	Apache-2.0
org.ehcache:sizeof	Apache-2.0
org.elasticsearch.client:elasticsearch-rest-client	Apache-2.0
org.freemarker:freemarker	Apache-2.0
org.glassfish.hk2.external:aopalliance-repackaged	Classpath-exception-2.0,EPL-2.0,CC-BY-NC-SA-4.0

Component Name	Licenses
org.glassfish.hk2.external:jakarta.inject	Classpath-exception-2.0,EPL-2.0,CC-BY
org.glassfish.hk2:hk2-api	Classpath-exception-2.0,EPL-2.0,CC-BY
org.glassfish.hk2:hk2-locator	Classpath-exception-2.0,EPL-2.0,CC-BY
org.glassfish.hk2:hk2-utils	Classpath-exception-2.0,EPL-2.0,CC-BY
org.glassfish.jaxb:jaxb-runtime	BSD-3-Clause
org.glassfish.jaxb:txw2	BSD-3-Clause
org.glassfish.jersey.containers:jersey-container-servlet	Apache-2.0,BSD-2-Clause,BSD-3-1.0,Classpath-exception-2.0,EPL-2.0,LicenseRef-jfrog-public-domain
org.glassfish.jersey.containers:jersey-container-servlet-core	Apache-2.0,BSD-2-Clause,BSD-3-1.0,Classpath-exception-2.0,EPL-2.0,LicenseRef-jfrog-public-domain
org.glassfish.jersey.inject:jersey-hk2	Apache-2.0,BSD-2-Clause,BSD-3-1.0,Classpath-exception-2.0,EPL-2.0,LicenseRef-jfrog-public-domain
org.glassfish.jakarta.el	Classpath-exception-2.0,EPL-2.0,CC-BY
org.graalvm.js:js-language	MIT,UPL-1.0
org.graalvm.polyglot:polyglot	UPL-1.0
org.graalvm.regex:regex	UPL-1.0
org.graalvm.sdk:collections	UPL-1.0
org.graalvm.sdk:jniutils	UPL-1.0



Component Name	Licenses
org.graalvm.sdk:nativebridge	UPL-1.0
org.graalvm.sdk:nativeimage	UPL-1.0
org.graalvm.sdk:word	UPL-1.0
org.graalvm.shadowed:icu4j	ICU
org.graalvm.truffle:truffle-api	UPL-1.0
org.graalvm.truffle:truffle-compiler	UPL-1.0
org.graalvm.truffle:truffle-enterprise	MS-RL
org.graalvm.truffle:truffle-runtime	UPL-1.0
org.hamcrest:hamcrest	BSD-3-Clause
org.hibernate.validator:hibernate-validator	Apache-2.0
org.ini4j:ini4j	Apache-2.0
org.javassist:javassist	Apache-2.0,LGPL-2.1,MPL-1.1
org.javassist:javassist	Apache-2.0,LGPL-2.1,MPL-1.1
org.jboss.logging:jboss-logging	Apache-2.0
org.jboss.logging:jboss-logging	Apache-2.0
org.jboss.threads:jboss-threads	Apache-2.0
org.jboss.threads:jboss-threads	Apache-2.0
org.jboss.xnio:xnio-api	Apache-2.0

Component Name	Licenses
org.jboss.xnio:xnio-nio	Apache-2.0
org.jctools:jctools-core	Apache-2.0
org.jctools:jctools-core	Apache-2.0
org.jetbrains.kotlin:kotlin-reflect	Apache-2.0
org.jetbrains.kotlin:kotlin-stdlib	Apache-2.0
org.jetbrains.kotlin:kotlin-stdlib	Apache-2.0
org.jetbrains.kotlin:kotlin-stdlib-common	Apache-2.0
org.jetbrains.kotlin:kotlin-stdlib-common	Apache-2.0
org.jetbrains.kotlin:kotlin-stdlib-jdk7	Apache-2.0
org.jetbrains.kotlin:kotlin-stdlib-jdk7	Apache-2.0
org.jetbrains.kotlin:kotlin-stdlib-jdk8	Apache-2.0
org.jetbrains.kotlin:kotlin-stdlib-jdk8	Apache-2.0
org.jetbrains:annotations	Apache-2.0
org.jruby.jcodings:jcodings	MIT
org.json:json	LicenseRef-jfrog-public-domain
org.lz4:lz4-java	Apache-2.0
org.mapstruct:mapstruct	Apache-2.0
org.mapstruct:mapstruct	Apache-2.0

Component Name	Licenses
org.mapstruct:mapstruct	Apache-2.0
org.mnode.ical4j:ical4j	BSD-3-Clause
org.mongodb:bson	Apache-2.0
org.mongodb:bson-record-codec	Apache-2.0
org.mongodb:mongodb-driver-core	Apache-2.0
org.mongodb:mongodb-driver-sync	Apache-2.0
org.objenesis:objenesis	Apache-2.0
org.ow2.asm:asm	BSD-3-Clause
org.postgresql:postgresql	BSD-2-Clause,PostgreSQL
org.projectlombok:lombok	MIT
org.projectlombok:lombok	MIT
org.projectlombok:lombok-mapstruct-binding	MIT
org.quartz-scheduler:quartz	Apache-2.0
org.quartz-scheduler:quartz-jobs	Apache-2.0
org.reactivestreams:reactive-streams	MIT-0
org.realityforge.org.jetbrains.annotations:org.jetbrains.annotations	Apache-2.0
org.reflections:reflections	BSD-2-Clause,BSD-3-Clause,WTF
org.slf4j:jcl-over-slf4j	Apache-2.0

Component Name	Licenses
org.slf4j:jul-to-slf4j	MIT
org.slf4j:jul-to-slf4j	LicenseRef-jfrog-oracle-nftc-2021
org.slf4j:slf4j-api	MIT
org.slf4j:slf4j-api	GPL-2.0,MIT
org.slf4j:slf4j-simple	MIT
org.springframework.boot:spring-boot	Apache-2.0
org.springframework.boot:spring-boot	Apache-2.0
org.springframework.boot:spring-boot-autoconfigure	Apache-2.0
org.springframework.boot:spring-boot-autoconfigure	Apache-2.0
org.springframework.boot:spring-boot-jarmode-layertools	Apache-2.0
org.springframework.boot:spring-boot-jarmode-layertools	Apache-2.0
org.springframework.boot:spring-boot-jarmode-layertools	Apache-2.0
org.springframework.cloud:spring-cloud-function-context	Apache-2.0
org.springframework.cloud:spring-cloud-function-core	Apache-2.0
org.springframework.data:spring-data-commons	Apache-2.0
org.springframework.data:spring-data-keyvalue	Apache-2.0
org.springframework.data:spring-data-mongodb	Apache-2.0
org.springframework.data:spring-data-redis	Apache-2.0,GPL-2.0

Component Name	Licenses
org.springframework.integration:spring-integration-core	Apache-2.0
org.springframework.integration:spring-integration-redis	Apache-2.0
org.springframework.kafka:spring-kafka	Apache-2.0
org.springframework.kafka:spring-kafka	Apache-2.0
org.springframework.plugin:spring-plugin-core	Apache-2.0
org.springframework.plugin:spring-plugin-metadata	Apache-2.0
org.springframework.retry:spring-retry	Apache-2.0
org.springframework.security:spring-security-config	Apache-2.0
org.springframework.security:spring-security-core	Apache-2.0
org.springframework.security:spring-security-crypto	Apache-2.0
org.springframework.security:spring-security-web	Apache-2.0
org.springframework.session:spring-session-core	Apache-2.0
org.springframework.session:spring-session-data-mongodb	Apache-2.0
org.springframework:spring-aop	Apache-2.0
org.springframework:spring-aop	Apache-2.0
org.springframework:spring-beans	Apache-2.0
org.springframework:spring-beans	Apache-2.0
org.springframework:spring-context	Apache-2.0

Component Name	Licenses
org.springframework:spring-context	Apache-2.0
org.springframework:spring-context-support	Apache-2.0
org.springframework:spring-core	Apache-2.0
org.springframework:spring-core	Apache-2.0
org.springframework:spring-expression	Apache-2.0
org.springframework:spring-expression	Apache-2.0
org.springframework:spring-expression	Apache-2.0
org.springframework:spring-jcl	Apache-2.0
org.springframework:spring-jcl	Apache-2.0
org.springframework:spring-jdbc	Apache-2.0
org.springframework:spring-messaging	Apache-2.0
org.springframework:spring-messaging	Apache-2.0
org.springframework:spring-oxm	Apache-2.0
org.springframework:spring-tx	Apache-2.0
org.springframework:spring-tx	Apache-2.0
org.springframework:spring-web	Apache-2.0
org.springframework:spring-web	Apache-2.0
org.springframework:spring-webmvc	Apache-2.0

Component Name	Licenses
org.springframework:spring-webmvc	Apache-2.0
org.terracotta:offheap-store	Apache-2.0
org.terracotta:statistics	Apache-2.0
org.terracotta:terracotta-utilities-tools	Apache-2.0
org.threeten:threetenbp	BSD-3-Clause
org.thymeleaf.extras:thymeleaf-extras-java8time	Apache-2.0
org.thymeleaf:thymeleaf	Apache-2.0
org.thymeleaf:thymeleaf-spring5	Apache-2.0
org.unescape:unescape	Apache-2.0
org.wildfly.client:wildfly-client-config	Apache-2.0
org.wildfly.common:wildfly-common	Apache-2.0
org.wildfly.common:wildfly-common	Apache-2.0
org.xerial.snappy:snappy-java	Apache-2.0
org.xerial.snappy:snappy-java	Apache-2.0
org.yaml:snakeyaml	Apache-2.0
org.yaml:snakeyaml	Apache-2.0
os-homedir	MIT
os-name	MIT

Component Name	Licenses
osx-release	MIT
package-json-from-dist	BlueOak-1.0.0
pacote	0BSD,ISC
pac-proxy-agent	MIT
pac-resolver	MIT
pako	MIT,Zlib
parent-module	MIT
parse5	MIT
parse5-htmlparser2-tree-adapter	MIT
parse-bmfont-ascii	MIT
parse-bmfont-binary	MIT
parse-bmfont-xml	MIT
parse-conflict-json	ISC
parse-headers	MIT
parse-json	MIT
parseurl	MIT
path-exists	MIT
path-is-absolute	MIT



Component Name	Licenses
path-key	MIT
path-scurry	BlueOak-1.0.0
path-scurry	BlueOak-1.0.0
path-to-regexp	MIT
path-to-regexp	MIT
pause-stream	Apache-2.0,MIT
pdfsharp	MIT
peek-readable	MIT
pend	MIT
phin	MIT
picocolors	0BSD,ISC
pify	MIT
pipelines.sockets.unofficial	MIT
p-is-promise	MIT
pixelmatch	ISC
p-limit	MIT
p-locate	MIT
p-map	MIT

Component Name	Licenses
p-map	MIT
pngjs	MIT
pngjs	MIT
polly	BSD-3-Clause
polly.core	BSD-3-Clause
postcss-selector-parser	MIT
ppt_tools.jar	Unknown
pptxgenjs	MIT
prebuild-install	MIT
pretty	MIT
process	MIT
process-nextick-args	MIT
proc-log	ISC
proc-log	ISC
proggy	ISC
progress	MIT
promise	MIT
promise-all-reject-late	0BSD,ISC

Component Name	Licenses
promise-call-limit	0BSD,ISC
promise-inflight	ISC
promise-retry	MIT
promzard	0BSD,ISC
protobufjs	BSD-3-Clause,LicenseRef-jfrog-pr
proto-list	0BSD,ISC
proxy-addr	MIT
proxy-agent	MIT
proxy-from-env	MIT
p-try	MIT
pump	MIT
pump	MIT
pump	MIT
pump	MIT
pump	MIT
punycode	MIT
puppeteer	Apache-2.0
puppeteer-core	Apache-2.0
qiniu	MIT

Component Name	Licenses
qrcoder	MIT
qrcoder	MIT
qrcode-terminal	Apache-2.0
qs	BSD-3-Clause
queue	MIT
queue-tick	MIT
range-parser	MIT
raw-body	MIT
rc	Apache-2.0,BSD-2-Clause,BSD-2-FreeBSD,MIT
read	0BSD,ISC
readable-stream	MIT
readable-stream	MIT
readable-web-to-node-stream	MIT
read-cmd-shim	0BSD,ISC
README.gz	Zlib
README.gz	FSFAP
read-package-json-fast	0BSD,ISC

Component Name	Licenses
redis	MIT
redis.clients:jedis	MIT
redis.clients:jedis	MIT
redis-commands	MIT
redis-errors	MIT
redis-parser	MIT
regenerator-runtime	MIT
regexp-clone	MIT
require-at	Apache-2.0
require-directory	MIT
require-main-filename	ISC
resolve-from	MIT
resolve-path	MIT
retry	MIT
retry	MIT
retry	Apache-2.0,MIT
rsaextensions	Apache-2.0
safe-buffer	MIT

Component Name	Licenses
safe-buffer	MIT
safer-buffer	MIT
saslprep	MIT
savedmodel:preloaded_data.pb	Unknown
sax	ISC,MIT
scala-java8-compat_2.11-0.7.0.jar	Unknown
secure-json-parse	BSD-3-Clause
semver	0BSD,ISC
semver	0BSD,ISC
semver	0BSD,ISC
send	MIT
serilog	Apache-2.0
serilog.sinks.file	Apache-2.0
serilog.sinks.rollingfile	Apache-2.0
serve-static	MIT
servicestack.redis	Unknown
set-blocking	ISC
set-function-length	MIT

Component Name	Licenses
setimmediate	MIT
setprototypeof	0BSD,ISC
setprototypeof	0BSD,ISC
sharpcompress	MIT
shebang-command	MIT
shebang-regex	MIT
side-channel	MIT
sift	MIT
signal-exit	ISC
signal-exit	ISC
sigstore	Apache-2.0
simple-concat	MIT
simple-get	MIT
simple-lru-cache	MIT
sizeof-agent.jar	Unknown
skyapm.abstractions	Apache-2.0
skyapm.agent.aspnetcore	Apache-2.0
skyapm.agent.generalhost	Apache-2.0

Component Name	Licenses
skyapm.agent.hosting	Apache-2.0
skyapm.core	Apache-2.0
skyapm.diagnostics.aspnetcore	Apache-2.0
skyapm.diagnostics.entityframeworkcore	Apache-2.0
skyapm.diagnostics.entityframeworkcore.npgsql	Apache-2.0
skyapm.diagnostics.entityframeworkcore.pomelo.mysql	Apache-2.0
skyapm.diagnostics.entityframeworkcore.sqlite	Apache-2.0
skyapm.diagnostics.grpc	Apache-2.0
skyapm.diagnostics.grpc.net.client	Apache-2.0
skyapm.diagnostics.httpclient	Apache-2.0
skyapm.diagnostics.sqlclient	Apache-2.0
skyapm.transport.grpc	Apache-2.0
skyapm.transport.grpc.protocol	Apache-2.0
skyapm.utilities.configuration	Apache-2.0
skyapm.utilities.dependencyinjection	Apache-2.0
skyapm.utilities.logging	Apache-2.0
sliced	MIT
smart-buffer	MIT



Component Name	Licenses
snappy	MIT
socks	MIT
socks-proxy-agent	MIT
socks-proxy-agent	MIT
sourcecode.mingdao.net/mingdao/MDServices-Golang	Unknown
source-map	BSD-3-Clause
sparse-bitfield	MIT
spdx-correct	Apache-2.0
spdx-exceptions	CC-BY-3.0
spdx-expression-parse	MIT
spdx-expression-parse	MIT
spdx-license-ids	CC0-1.0
sprintf-js	BSD-3-Clause
ssri	ISC
stackexchange.redis	MIT
standard-as-callback	MIT
statuses	MIT
statuses	MIT

Component Name	Licenses
statuses	MIT
streamsearch	MIT
streamx	MIT
string_decoder	MIT
string_decoder	MIT
string-hash	CC0-1.0
string-width	MIT
string-width	MIT
string-width	MIT
string-width	MIT
strip-ansi	MIT
strip-ansi	MIT
strip-ansi	MIT
strip-ansi	MIT
strip-json-comments	MIT
strtok3	MIT
supports-color	MIT
supports-color	MIT

Component Name	Licenses
swashbuckle.aspnetcore.cli	MIT
swashbuckle.aspnetcore.cli	MIT
swashbuckle.aspnetcore.swaggergen	MIT
swashbuckle.aspnetcore.swaggergen	MIT
swashbuckle.aspnetcore.swaggerui	MIT
swashbuckle.aspnetcore.swaggerui	MIT
swashbuckle.aspnetcore.swaggerui	MIT
system.configuration.configurationmanager	MIT
system.configuration.configurationmanager	MIT
system.configuration.configurationmanager	MIT
system.diagnostics.performancecounter	MIT
system.drawing.common	MIT
system.drawing.common	MIT
system.drawing.common	MIT
system.drawing.common	MIT
system.security.cryptography.protecteddata	MIT
system.security.cryptography.protecteddata	MIT
system.security.cryptography.protecteddata	MIT

Component Name	Licenses
system.security.permissions	MIT
system.security.permissions	MIT
system.windows.extensions	MIT
system.windows.extensions	MIT
system.windows.extensions	MIT
tar	0BSD,ISC
tar-fs	MIT
tar-fs	MIT
tar-stream	MIT
tar-stream	MIT
text-decoder	Apache-2.0
text-table	MIT
thenify	MIT
thenify-all	MIT
through	Apache-2.0,MIT
timm	MIT
tinycolor2	MIT
tiny-relative-date	MIT

Component Name	Licenses
to-buffer	MIT
toidentifier	MIT
token-types	MIT
toolgood.words	Apache-2.0
tr46	MIT
tr46	MIT
traverse	MIT,X11
treeverse	0BSD,ISC
tslib	0BSD
tslib	0BSD
tsscmp	MIT
tuf-js	MIT
tunnel-agent	Apache-2.0
TypeBuilder-1.0.jar	Unknown
type-is	MIT
typescript	Apache-2.0
unbzip2-stream	MIT
undici	MIT

Component Name	Licenses
undici	MIT
undici-types	MIT
undici-types	MIT
unescape	MIT
unique-filename	ISC
unique-slug	0BSD,ISC
universalify	MIT
unpipe	MIT
urijs	MIT
urlib	MIT
urlpattern-polyfill	MIT
utif2	MIT
util-deprecate	MIT
utility	MIT
utils-merge	MIT
uuid	MIT
validate-npm-package-license	Apache-2.0
validate-npm-package-name	ISC

Component Name	Licenses
vary	MIT
walk-up-path	0BSD,ISC
wangkanai.detection.core	Apache-2.0
wangkanai.detection.device	Apache-2.0
webidl-conversions	BSD-2-Clause
webidl-conversions	BSD-2-Clause
whatwg-fetch	MIT
whatwg-url	MIT
whatwg-url	MIT
which	0BSD,ISC
which	0BSD,ISC
which-module	ISC
which-pm-runs	MIT
wide-align	0BSD,ISC
win-release	MIT
wrap-ansi	MIT
wrap-ansi	MIT
wrap-ansi	MIT

Component Name	Licenses
wrappy	0BSD,ISC
write-file-atomic	ISC
ws	MIT
xhr	MIT
xlsx	Apache-2.0
xml2js	MIT
xmlbuilder	MIT
xml-parse-from-string	MIT
xmlpull:xmlpull	LicenseRef-jfrog-public-domain
xtend	MIT
y18n	ISC
y18n	ISC
yallist	0BSD,ISC
yargs	MIT
yargs	MIT
yargs-parser	ISC
yargs-parser	ISC
yauzl	MIT



Component Name	Licenses
ylru	MIT
zkweb.system.drawing	MIT
zod	MIT