

Practices

Alauda Container Platform (ACP) Practices

Issue	1.0
Date	2024-06-19



Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Solution Overview..... 1

2 Resource and Cost Planning.....4

3 Implementation Procedure..... 13

3.1 Performing Environment Checks..... 14

3.2 Installing the Media..... 21

3.3 Deploying the Platform..... 23

3.4 Configuring the Platform..... 34

4 Change History..... 36

1 Solution Overview

Scenarios

As emerging business models continue to disrupt traditional sectors such as finance, energy, manufacturing, automobile, and government, enterprises in these sectors are realizing that they cannot rely on traditional competition methods to stay ahead. To remain competitive in today's rapidly evolving landscape, they are turning to digital transformation. Digital transformation allows enterprises to quickly understand user needs, make necessary adjustments, and accelerate product updates. This leads to continuous improvement of user experience and customer satisfaction, ultimately resulting in a competitive advantage in the market. Enterprises often encounter challenges during digital transformation, including:

- Insufficient management support, heavy reliance on technology outsourcing, high costs due to siloed infrastructure management, and difficulty managing complex infrastructure scenarios like multi-cloud environments
- A need for quick resource supply to meet production-level requirements for service development, incomplete proprietary systems, and the need for elastic architectures to support services
- Difficult, error-prone Kubernetes O&M changes, slow service troubleshooting, and low resource utilization

By using advanced technologies such as containers, Kubernetes, and microservices, cloud native solutions can significantly speed up software development and iteration, enhance application architecture agility, improve IT resource elasticity and availability, and ultimately help enterprise customers accelerate their digital transformation efforts.

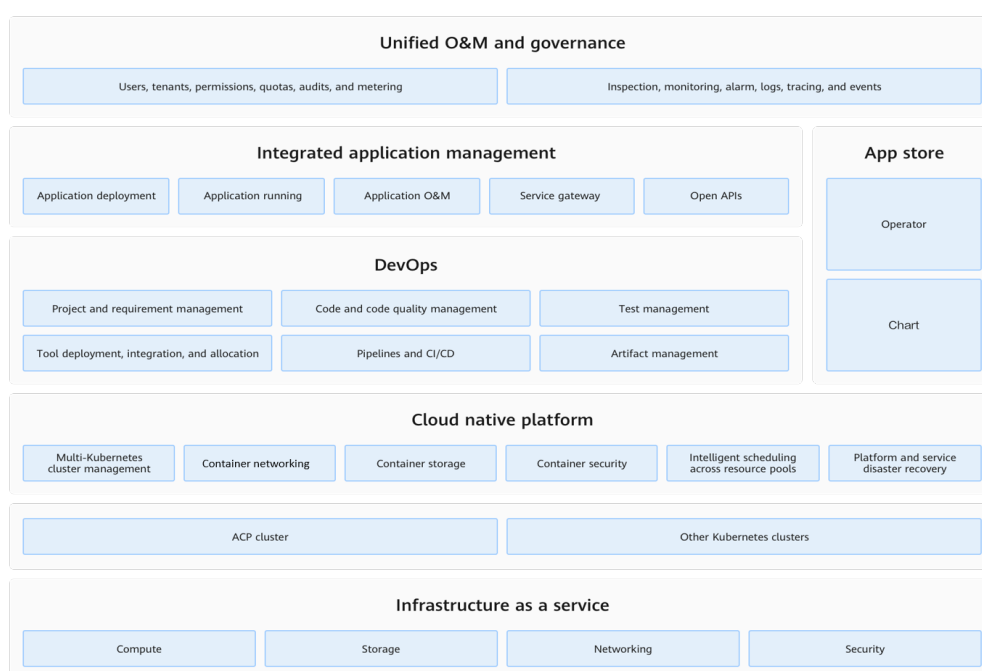
Solution Architecture

Alauda Container Platform (ACP) is an enterprise-grade full-stack cloud native open platform that goes beyond the traditional cloud native framework of container platform, DevOps, and microservices. ACP is built on Kubernetes-centered cloud native technologies and helps enterprises rapidly develop a next-generation full-stack cloud platform. This platform includes modules for cloud native infrastructure, cloud native application management, cloud native DevOps, unified O&M, and governance. ACP uses a container native architecture with Kubernetes as its foundation and control plane. It offers one-click deployment,

automatic O&M, and continuous upgrades, while also providing openness, flexibility, and scalability.

ACP helps to bridge the gaps between different infrastructures and serves as a control plane for hybrid clouds. It supports orchestration of various cloud environments and infrastructures, accelerating the development of application-centric hybrid clouds. ACP is an application-focused platform that can manage the entire application lifecycle using modules such as infrastructure, application management, and DevOps tools. It offers a full-stack, out-of-the-box solution that you can set up and get running fast.

Figure 1-1 Service architecture



Advantages

- Hybrid and multi-cloud management**
 ACP can manage any Kubernetes distribution in any environment, automate Kubernetes cluster lifecycle management at scale, ensure consistent configuration, security, and compliance across all environments, and provide a unified operations and consistent experience.
- Developer self-service**
 With ACP's fully customizable application and infrastructure abstraction, you can easily adopt infrastructure as code (IaC) and GitOps best practices. This simplifies your infrastructure and reduces cognitive load. Additionally, ACP provides built-in and customizable internal developer portals to enhance the developer experience.
- Quick setup of a PaaS container platform**
 ACP assists enterprise customers in various fields to rapidly construct a PaaS container platform, moving their existing infrastructure to a next-generation

cloud container platform with just a few clicks. ACP helps you manage the complete lifecycle of your containers.

- **Continuous expansion and integration of Kubernetes ecosystem tools**

ACP is closely linked with Kubernetes. With the Kubernetes native architecture as its framework, an increasing number of Kubernetes ecosystem tools, systems, add-ons, and solutions and compatible with and have been integrated into ACP. This ensures that you have access to competitive cutting-edge technologies, thereby maintaining your core competitiveness.

- **Automatic container scheduling**

ACP automatically creates Kubernetes resource objects. It establishes application models and creation processes that meet service requirements, and integrates the Kubernetes resource creation processes with the service processes.

- **Secure DevOps**

ACP uses DevSecOps to ensure security throughout the application lifecycle. It automates security protection to safeguard the environment and data, while also implementing CI/CD and ensuring the security of containerized applications. On ACP, centralized management of user identity and access control functions is possible. ACP also integrates with security scanning programs for containers.

- **Continuously improved IT operations**

Quality and security serve as the foundation for the full lifecycle management of software, from requirements to deployment and rollout. The offline IT production process is transformed into a highly automatic and visualized online IT production pipeline, resulting in improved product R&D efficiency, quick response to service requirements, and continuous improvement of IT operations.

2 Resource and Cost Planning

Table 2-1 Resource and cost planning

Cloud Service	Usage	Region	Specifications	Quantity	Billing Mode	Billing Period	Billing Unit	Reference Price
ECS	For a global cluster, which carries all ACP platform functions (such as platform components, UI, Kubernetes master node resources, logs, and monitoring)	AP-Bangkok	Specifications: x86 computing General computing-plus c6s.2xlarge.2 8 vCPUs 16 GiB; Image: CentOS CentOS 7.9 64bit; System disk: General-purpose SSD 150 GiB; Data disk 1: General-purpose SSD 500 GiB; Data disk 2: General-purpose SSD 100 GiB	3	Yearly/Monthly	1	Year	USD7,412.25

Cloud Service	Usage	Region	Specifications	Quantity	Billing Mode	Billing Period	Billing Period Unit	Reference Price
ECS	Master node in a service cluster (for Kubernetes master node resources, logs, and monitoring)	AP-Bangkok	Specifications: x86 computing General computing-plus c6s.xlarge.2 4 vCPUs 8 GiB; Image: CentOS CentOS 7.9 64bit; System disk: General-purpose SSD 100 GiB; Data disk 1: General-purpose SSD 100 GiB	3	Yearly/Monthly	1	Year	USD3,132.60
ECS	Prometheus monitoring node in a service cluster	AP-Bangkok	Specifications: x86 computing General computing-plus c6s.xlarge.2 4 vCPUs 8 GiB; Image: CentOS CentOS 7.9 64bit; System disk: General-purpose SSD 100 GiB; Data disk 1: General-purpose SSD 100 GiB	1	Yearly/Monthly	1	Year	USD1,044.20
ECS	Slave node in a service cluster (The price listed is for a single server. The actual server quantity required will depend on the specific requirements.)	AP-Bangkok	Specifications: x86 computing General computing-plus c6s.2xlarge.2 8 vCPUs 16 GiB; Image: CentOS CentOS 7.9 64bit; System disk: General-purpose SSD 100 GiB; Data disk 1: General-purpose SSD 100 GiB	1	Yearly/Monthly	1	Year	USD1,871.25

Cloud Service	Usage	Region	Specifications	Quantity	Billing Mode	Billing Period	Billing Period Unit	Reference Price
Virtual IP address (VIP)	High availability (HA) in a Kubernetes cluster	AP-Bangkok	/	2	Yearly/Monthly	1	Year	USD0.00
Elastic IP (EIP)	Platform access address	AP-Bangkok	Bandwidth price: Dedicated Dynamic BGP Billed by bandwidth 5 Mbit/s	1	Yearly/Monthly	1	Year	USD243.00

 NOTE

- The prices listed in the table are for reference purposes only. They may differ from the actual price, which is displayed on the Huawei Cloud console.
- The table above shows the minimum recommended configurations for deploying ACP. Make sure to allocate resources according to specific service needs. If more information about the pricing of ACP is needed, contact your account manager.

Table 2-2 Hardware configuration (adjusted based on the service pressure)

Server Role	Master, slave, global, log, and monitoring	Master node in a service cluster	Prometheus in a service cluster	Slave node in a service cluster
-------------	--------------------------------------------	----------------------------------	---------------------------------	---------------------------------

Number of Server	3	3 x 2 (two service clusters)	1 x 2	10 x 2
Server Function	Used to carry all functions of the platform	Master node in a service cluster	Monitoring node in service cluster	Slave node in a service cluster
Mandatory or Optional	Mandatory	Mandatory	Mandatory	Mandatory
Number of CPU	8	4	4	8
Memory Capacity	16 GiB	8 GiB	8 GiB	16 GiB
Available Space of the / Partition	150 GiB	50 GiB	50 GiB	50 GiB
/cpaas/data/	500 GiB for storing logs, separate block device	/	/	/
/cpaas/monitoring/	100 GiB for storing monitoring data	/	50 GiB	/
Available Space of /var/lib	50 GiB	50 GiB	50 GiB	50 GiB
Available Space of /opt	30 GiB	30 GiB	30 GiB	30 GiB
/var/lib/docker or /var/lib/containerd	100 GiB	100 GiB	100 GiB	100 GiB
/var/lib/docker or /var/lib/containerd	xfs	xfs	xfs	xfs
(Optional) Separate Block Device	/	100 GiB for topolvm	/	50 GiB x 2

Table 2-3 Hardware requirements

Hardware	Requirement	Model or Minimum Configuration
CPU	The dominant frequency must be at least 2.5 GHz and cannot be overcommitted at the IaaS layer. If requirements still cannot be met, you need to increase the number of CPUs. If Arm CPUs are used, you can increase the number of CPUs by 1.5x. It is advisable to increase the number of CPUs by 2x.	Intel 8255c
Memory	Overcommitment at the IaaS layer is not supported.	6-channel DDR4
Hard disk	The IOPS of a single block device is greater than 2,000, and the throughput is greater than 200 MB/s.	ssd
GPU	GPUs with driver 418.87.00 CUDA 10.1 have been fully tested.	Nvidia

Table 2-4 Network resource requirements

Resource	Mandatory or Optional	Quantity	Description
Certificate	Optional	1	If a certificate is not provided, the deployment script will generate one automatically. However, when a user accesses the platform UI through a browser, a security warning will appear because the certificate is not issued by a trusted certification authority.
Platform access address (external IP address)	Mandatory	1	Domain name or IP address. For details, see Platform Access Address in "Glossary" in <i>Alauda Cloud Native Success Platform Installation Guide</i> .

Resource	Mandatory or Optional	Quantity	Description
Global VIP	Mandatory	1	For details, see global VIP in "Glossary" in <i>Alauda Cloud Native Success Platform Installation Guide</i> .
Kubernetes API server VIP	Mandatory	Multiple	This resource is mandatory in the production environment and is used by kube-api of an HA Kubernetes cluster. Each HA Kubernetes cluster requires a VIP.
ALB VIP	Mandatory	Multiple	If customers require HA with ALB, this resource is necessary. Each service cluster's load balancer needs a VIP. (Note that it is the load balancer, not each ALB instance, that requires a VIP.)
ASM Istio gateway VIP	Optional	Multiple	For each service cluster where ASM Istio is deployed, if the HA Istio gateways are required, a VIP must be provided.
Private load balancer	Mandatory	1	This resource is essential in the production environment. Without it, HA requirements cannot be fulfilled. HA is achieved through load balancers like F5 load balancers. The load balancer is configured with both the Kubernetes API server VIP and the global VIP.
Public load balancer	Mandatory	1	This resource is essential in the production environment. Without it, HA requirements cannot be fulfilled. If the external networks are not distinguished from the internal networks, the private load balancer can be reused. The external address is configured on this load balancer.
More access addresses	Optional	Multiple	To use more IP addresses or domain names other than external address to access the global platform, prepare the domain names and IP addresses and add them in the advanced settings on the installation page during platform deployment.

Table 2-5 Network configuration requirements

Item	Requirement
Network rate	The bandwidth must be at least 1 Gbps. 10 Gbps is recommended. If the global platform and service cluster are deployed in different data centers or hybrid clouds, the network rate between them must be at least 100 Mbps. (1 Gbps will be better.) If there is no need to collect data such as service logs and audits from the service cluster, the required bandwidth can be reduced somewhat.
Network latency	The network latency should be 2 ms or lower. If the global platform and the service cluster are deployed in different data centers or in a hybrid cloud, ensure that the network latency between them is within 30 ms and does not exceed 100 ms.
Security and firewall	<p>There is no firewall between servers on the global platform.</p> <p>There is no firewall between servers in the service cluster.</p> <p>It is recommended that no firewall be deployed between the service cluster and the platform. If a firewall is deployed between them, allow necessary ports to pass through the firewall by referring to Global platform four-layer forwarding rules and Compute cluster forwarding rules in "Network Requirements" in <i>Alauda Cloud Native Success Platform Installation Guide</i>.</p> <p>Calico uses the IP in IP protocol. If the service cluster uses the Calico plug-in, the IP in IP protocol cannot be restricted.</p>
IP address range	IP addresses in the 172.16 to 172.32 CIDR block cannot be used by the server hosting the platform. If any of these IP addresses are already being used, it cannot be altered. To work around this issue, you must modify the Docker configuration on each server and add the bip parameter to avoid using the CIDR block.
Agreement	If dual-stack networking is used, IPv6 must be supported.
Route	The server has a default route or a route pointing to 0.0.0.0 .
Forwarding	All forwarding ports should be allowed.

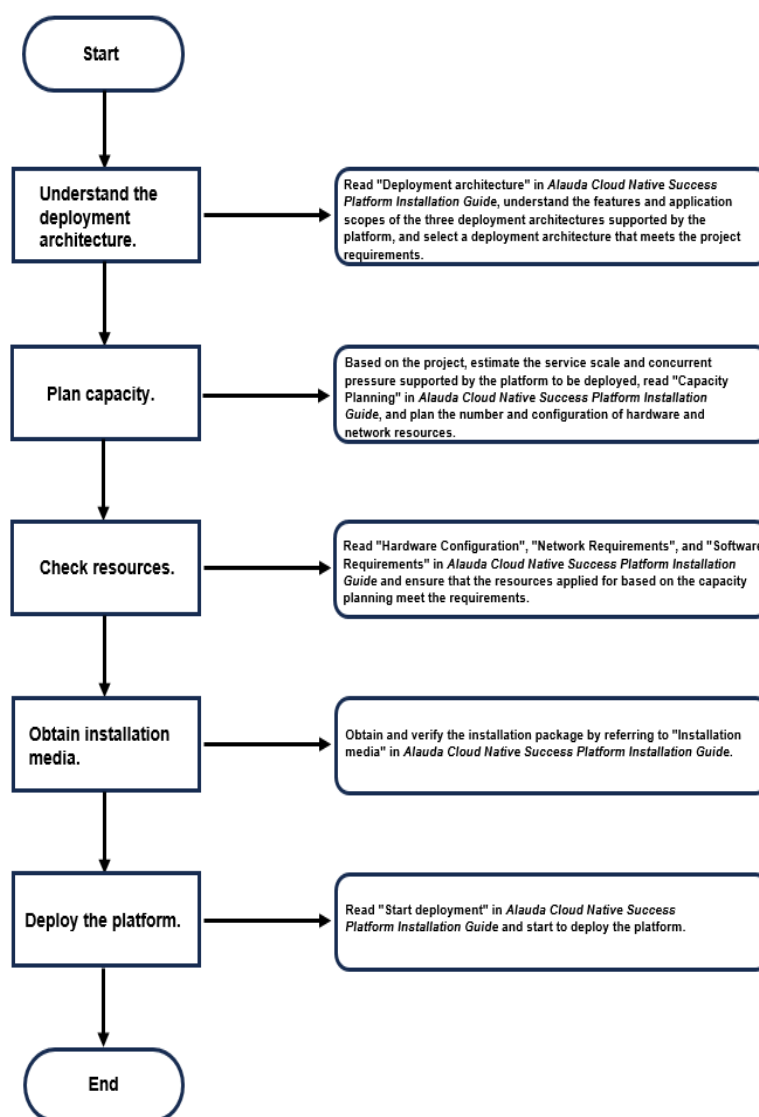
Table 2-6 OSs and kernel versions

Arc hit ect ure	CPU Mod el	Supported OS	Version Information	Re ma rks
Ar m	Kunp eng 920	Kylin	<ul style="list-style-type: none"> Kylin Linux Advanced Server Release (Kylin V10) Kernel version: 4.19.90-11.ky10.aarch64 Kylin Linux Advanced Server Release (Kylin V10 SP1) Kernel version: 4.19.90-17.ky10.aarch64 Kylin Linux Advanced Server Release (Kylin V10 SP2) Kernel version: 4.19.90-24.4.v2101.ky10.aarch64 	Virt uali zati on is not sup por ted.
		openEuler	openEluer 20.03 SP3 Kernel version: 4.19.90-2112.8.0.0131.oe1.aarch64 openEluer 22.03 SP3 Kernel version: 5.10.0-60.18.0.50.oe2203.aarch64	No ne
x86	None	Kylin	<ul style="list-style-type: none"> Kylin Linux Advanced Server Release (Kylin V10) Kernel version: 4.19.90-11.ky10.x86_64 Kylin Linux Advanced Server Release (Kylin V10 SP1) Kernel version: 4.19.90-23.8.v2101.ky10.x86_64 Kylin Linux Advanced Server Release (Kylin V10 SP2) Kernel version: 4.19.90-24.4.v2101.ky10.x86_64 	No ne
		openEuler	<ul style="list-style-type: none"> 20.03 SP3 Kernel version: 4.19.90-2112.8.0.0131.oe1.x86_64 22.03 SP3 Kernel version: 5.10.0-60.18.0.50.oe2203.x86_64 	No ne
		Ubuntu	<ul style="list-style-type: none"> Ubuntu 20.04 Kernel version: 5.4.0-124-generic Ubuntu 22.04 Kernel version: 5.15.0-56-generic 	No ne
		Red Hat	<ul style="list-style-type: none"> Red Hat 7.8 Kernel version: 3.10.0-1127.el7.x86_64 Red Hat 8.0 Kernel version: 4.18.0-80.el8.x86_64 Red Hat 8.6 Kernel version: 4.18.0-372.9.1.el8.x86_64 	No ne

Architecture	CPU Model	Supported OS	Version Information	Remarks
		CentOS	CentOS 7.6, 7.7, 7.8, and 7.9 Kernel version: 3.10.0-1160 and 3.10.0-1127	None

3 Implementation Procedure

Figure 3-1 ACP deployment process



3.1 Performing Environment Checks

[3.2 Installing the Media](#)

[3.3 Deploying the Platform](#)

[3.4 Configuring the Platform](#)

3.1 Performing Environment Checks

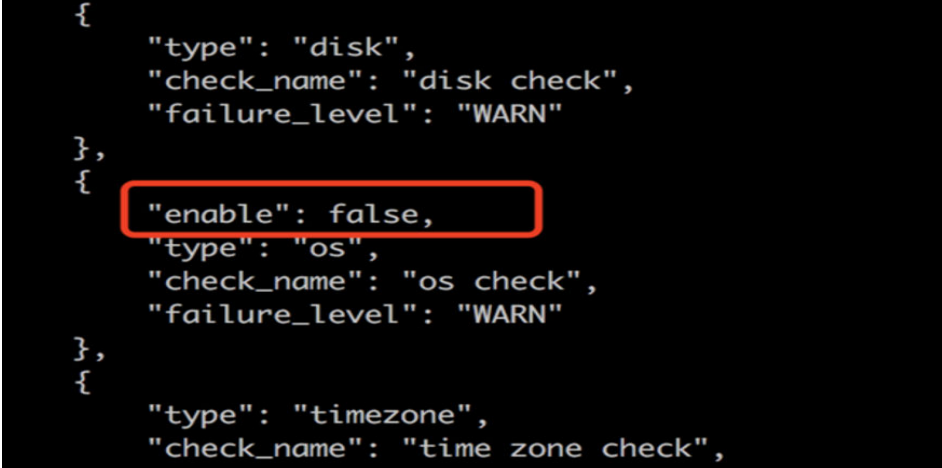
OSs and Kernel Versions

For details, see [3.2 Installing the Media](#).

 NOTE

1. Check for the basic software package.

You need to manually modify the configuration file of the UOS. After the deployment starts, modify the `/cpaas/conf/check_list.json` file, find the `"type": "os"` line, and add `"enable": false`.



```
{
  "type": "disk",
  "check_name": "disk check",
  "failure_level": "WARN"
},
{
  "enable": false,
  "type": "os",
  "check_name": "os check",
  "failure_level": "WARN"
},
{
  "type": "timezone",
  "check_name": "time zone check",
  "failure_level": "WARN"
}
```

2. XFS fragmentation
3. If you encounter a kmem issue, refer to the following:
<https://access.redhat.com/solutions/532663>
<https://github.com/opencontainers/runc/issues/1725>
<https://github.com/kubernetes/kubernetes/issues/61937>
<https://github.com/kubernetes/kubernetes/issues/61937#issuecomment-567042968>
<https://kubeovn.github.io/docs/v1.12.x/en/start/prepare/>

GRUB Startup Parameters

1. Solve the kmem issue.
 - a. Edit `/etc/default/grub` (in CentOS, Red Hat, or tlinux) or `/boot/efi/EFI/kylin/grub.cfg` (in Kylin OS).
In the line containing `GRUB_CMDLINE_LINUX=`, add `cgroup.memory=nokmem` after `crashkernel` and run `grub2-mkconfig -o /boot/grub2/grub.cfg`.
 - b. Restart the system.

If the added parameter can be found in **/proc/cmdline**, the modification was successful.

NOTE

<https://github.com/opencontainers/runc/issues/1725>

<https://github.com/kubernetes/kubernetes/issues/61937>

<https://github.com/kubernetes/kubernetes/issues/61937#issuecomment-567042968>

2. Disable huge pages.

- a. Edit **/etc/default/grub** (in CentOS, Red Hat, or tlinux) or **/boot/efi/EFI/kylin/grub.cfg** (Kylin OS) and add **transparent_hugepage=never** to **GRUB_CMDLINE_LINUX**.
- b. Run **grub2-mkconfig -o /boot/grub2/grub.cfg**.
- c. Restart the server and check the results by against the below image.

NOTE

In an Arm architecture, if the Redis service is not disabled, the performance will be severely affected.

```
1 [root@ ~]# cat /sys/kernel/mm/transparent_hugepage/enabled
2 always madvise [never]
3 [root@ ~]# cat /sys/kernel/mm/transparent_hugepage/defrag
4 always madvise [never]
```

Kernel Modules

Network kernel module requirements:

- If Red Hat is used and the version is earlier than 4.18.0, or Red Hat is not used and the version is earlier than 4.19.0, check **nf_conntrack_ipv4**. If IPv6 is enabled, check **nf_conntrack_ipv6**.
- If kube-ovn is used, check **geneve** and **openvswitch**.
- Check **ip_vs**, **ip_vs_rr**, **ip_vs_wrr**, and **ip_vs_sh**.

NOTE

Take CentOS 7 as an example. Run the following command as user **root**:

```
cat <<EOF > /etc/modules-load.d/cpaas.conf
iptables_nat
EOF
```

Restart the server and run **lsmod | grep iptable_nat**. If the **iptables_nat** module is present, the to-do task is successfully configured.

User Permissions

root

NOTE

You can log in to the system over SSH as a non-root user and run **su -** to gain root access.

sshd Configuration

- Each node in the global cluster can be remotely logged in through SSH.
- The values of **UseDNS** and **UsePAM** in `/etc/ssh/sshd_config` must be **no**.

NOTE

If the user is not **root**, you need to configure the `/etc/sudoers` file so that the user can run **sudo** without entering the password.

If reverse resolution is not set up for the DNS, it may time out.

Swap

Disabled

NOTE

Failing to meet this requirement may cause a sharp increase in system I/O, leading to Docker becoming unresponsive.

Firewall

Disabled

NOTE

This is a requirement from the official Kubernetes documentation.

SELinux

Disabled

NOTE

This is a requirement from the official Kubernetes documentation.

Time Synchronization

The time of all servers must be synchronized, and the time difference cannot exceed 10 seconds.

NOTE

This is a requirement from the official Docker and Kubernetes documentation.

Time Zone

The time zones of all servers must be the same.

NOTE

The time zone should be **Asia/Shanghai**.

/etc/sysctl.conf Kernel Parameters

- **vm.max_map_count=262144**
- **net.ipv4.ip_forward=1**
- **vm.drop_caches=3**
- **net.ipv4.tcp_tw_recycle=0**
- **net.ipv4.tcp_mtu_probing=1**
- **ipv4.conf.all.rp_filter=0**
- **ipv4.conf.eth0.rp_filter=0**
- **net.ipv4.conf.default.rp_filter=0**
- **ipv6.disable=0**

NOTE

vm.max_map_count must be specified for the server where Elasticsearch runs.
The configuration of **net.ipv4.ip_forward** is required in the official Kubernetes document.
You need to disable the file cache.
<https://serverfault.com/questions/646604/what-causes-syn-to-listen-sockets-dropped>
<https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=4396e46187ca5070219b81773c4e65088dac50cc>
<https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt>
rp_filter configurations are required for communications between components in two Calico subnets in different modes.

Hostname Format

Obtain the hostname of the node. If the hostname is used as the node name, it must have a unique value and cannot exceed 36 characters. The following requirements must be met:

- Only lowercase letters, digits, hyphens (-), and periods (.) are allowed.
- The value cannot contain consecutive periods and hyphens (.-), consecutive periods (..), or consecutive hyphens and periods (-.).
- The value must start with a letter or digit.
- The value must end with a letter or digit.

NOTE

For details, see <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/>.

/etc/hosts

Hostnames of all servers can be resolved to IP addresses. **localhost** can be resolved to **127.0.0.1**.

CAUTION

The **hosts** file cannot contain duplicate hostnames.

core Files

Run **ulimit -c 0** to disable the generation of **core** files and add '**ulimit -S -c 0**' to the **/etc/profile** file.

NOTE

Sometimes, when a process restarts within a pod, **core** files can be generated. These **core** files take up a significant amount of disk space. This can cause the pod to exit unexpectedly and even impact the entire node.

Requirements for **/etc/resolv.conf**

If a search domain is present, parsing **svc** may result in an error. To resolve this issue, simply remove the **search** field.

There has to be an **/etc/resolv.conf** file and it has to contain the **nameserver** configuration item. IP addresses starting with **127** are not allowed.

DefaultTasks

Run **systemctl show --property=DefaultTasksMax**. If the returned value is not **infinity** or a large number such as **18446744073709551615**, you need to change the value of **DefaultTasksMax**.

NOTE

Change **DefaultTasksMax** to **DefaultTasksMax=infinity** in the **/etc/systemd/system.conf** file.

Impact: When using the global platform as a service cluster in all-in-one or standard deployment architecture, the number of customer services can be affected. This can result in abnormal pods when a large number of services are started.

AppArmor

1. Disable AppArmor:
systemctl stop apparmor.service && systemctl disable apparmor.service;
2. Open **/etc/default/grub** and add **apparmor=0** to **GRUB_CMDLINE_LINUX**.

NOTE

When the UOS is used, if the runtime version is containerd 1.6.4 or later, AppArmor must be disabled to avoid deployment errors.

Software Package and System Tools

- The following system tools must exist on the host:
ip, ss, tar, swapoff, modprobe, sysctl, md5sum, and either SCP or SFTP
- bc must be installed for EulerOS.
- To deploy topolvm and rook, lvm2 must be installed.

Software Package Removal

Kylin OS comes with runC pre-installed, which conflicts with the runC deployed on the platform, so, runC must be removed prior to deployment.

/tmp Access

The account must have the permissions to run **ls** and **cat** in the **/tmp** directory.

GPU Devices

Check whether the device exists when a GPU device is used.

CPU Cores

The number of CPU cores must be at least 2.

Memory Size

The memory size is at least 2 GiB.

Kubelet Service Checks

The **/etc/systemd/system/kubelet.service** file cannot exist.

Default Route

The server has a default route or a route pointing to **0.0.0.0**.

Whether Ports Are Occupied

Check whether the following listening ports are occupied:

- All ports to be checked: 10249, 10250, and 10256
- Master node ports: 2379, 2380, 6443, 10249, 10250, 10251, 10252, and 10256
- Kube-OVN ports: 6641 and 6642
- Calico port: 179

Network Interfaces

The network interfaces configured for the cluster and node are present.

Hardware Architecture

The hardware architecture (x86 or Arm) of the node must be the same as that of the cluster.

IP Address

The IP address must be valid and exist. If IPv6 is enabled, the IPv6 address must also be valid and exist.

The node IP address cannot be a loopback IP address.

- **127.0.0.1**
- **0:0:0:0:0:0:0:0** or **::** (The node IP address cannot be a multicast address.)
- **224.0.0.0** to **239.255.255.255**
- An IPv6 address starting with **FF**. The node IP address cannot be a link-local address.
- **169.254.0.0/16*** address block
- **fe80::/10*** address block: The node IP address cannot be all-0 IP address or a broadcast address.
- **255.255.255.255**

IP Address Segment

The IP addresses in the **172.16.x.x** to **172.32.x.x** CIDR block required by Docker are not occupied.

If the IP addresses within the CIDR block are already in use and cannot be changed, modify the Docker configuration files on all nodes by adding the **bip** parameter to prevent Docker from using those occupied IP addresses.

Node Access

The node and its SSH port are accessible.

Whether the Node Can Access the Platform

The node can access the platform through the platform address.

Whether the Node Can Access the Platform Image Repository

The node can access the image repository of the platform.

Whether the Node IP Address Is in Any Configured CIDR Blocks

The node IP address is not within any of the configured CIDR blocks, including the default subnet CIDR block, container CIDR block, service CIDR block, or join CIDR block.

CIDR Block Checks

- When the default subnet is underlay, the system skips the verification process of whether the node IP address is within the default subnet CIDR block, service CIDR block, or join CIDR block.

- When the default subnet is a non-underlay network, the node IP address (including IPv6) must not be within any of the configured CIDR blocks, which include the default subnet CIDR block, container CIDR block, service CIDR block, or join CIDR block.

Master Port Access

Check the connectivity between the host and each port of all master nodes (6443, 2379, and 2380).

Checks for the pki Directory

Check whether the `/var/lib/kubelet/pki` directory is either empty or non-existent on the target host.

Checks for the cri Directory Space

Check the available size of the specified directory (`/var/lib/containerd` or `/var/lib`).

Check Timeout

By default, it takes about 110s to add a node on the UI.

Checks for Directories That Cannot Exist

The `/var/log/pods` directory cannot exist.

/usr/bin Service

If the docker, containerd, or runc exists, it must be in `/usr/bin`.

3.2 Installing the Media

How to Download

Use a tenant account to access the platform to download the installation package and related documents, or contact the service manager. For details, see <https://www.alauda.io/>.

Verifying the Media

NOTE

Check whether the installation package is secure and reliable and has not been tampered with. This operation is optional.

Background

Before delving into GNU Privacy Guard (GnuPG or GPG), it is important to have a grasp of Pretty Good Privacy (PGP). In 1991, programmer Phil Zimmermann created PGP, an encryption software designed to evade government surveillance. PGP quickly gained popularity due to its user-friendly interface and has since become an essential tool for many programmers. However, PGP is commercial software and cannot be used freely. To address this, the Free Software Foundation (FSF) developed an open-source alternative called GnuPG or GPG. Nowadays, there is an open-source product available to everyone. Unlike PGP, GPG does not include patented algorithms and can be used without restrictions for commercial applications. For details, see HOWTO <https://www.gnupg.org/howtos/en/GPGMiniHowto.html>.

Step 1 Install the GPG.

You can install it in either of the following ways:

Download the source code and compile and install it.

```
./configure  
make  
make install
```

Install the software from a standard software repository.

```
Ubuntu:  
sudo apt-get install gnupg  
Centos:  
yum install gnupg -y  
Mac:  
brew install gpg
```

Step 2 Import a public key.

CAUTION

- The public key is a trusted method to verify that the installation media has not been tampered with. It can be downloaded from <https://www.alauda.io/>.
- After downloading, verify that the public key MD5 is 2eaddfab97d2951a8915f327acb53562 and ensure that it has not been tampered with.
- Once you import the public key, run **gpg --list-keys** and verify that the public key ID is **BB097AE6**. Make sure that the public key ID has not been tampered with.

```
curl https://www.alauda.cn/download/verify-key.pub | gpg --import  
# Run the preceding command. The information similar to the following is displayed:  
gpg: key BB097AE6: public key "cpaas (Special for packing) <wht@126.com>" imported  
gpg: Total number processed: 1  
gpg: imported: 1 (RSA: 1)
```

Step 3 Check the public key.

```
gpg --list-keys  
# Run the preceding command. The public key information is displayed.  
/root/.gnupg/pubring.gpg  
-----  
pub 4096R/BB097AE6 2020-08-11  
uid cpaas (Special for packing) <wht@126.com>  
sub 4096R/3750351A 2020-08-11
```

Step 4 Verify the public key signature.

```
gpg --fingerprint BB097AE6
# Run the preceding command. The public key information is displayed.
pub 4096R/BB097AE6 2020-08-11
Key fingerprint = 09EE E7B9 A30C F4B3 5E31 A91B 2704 1C16 BB09 7AE6
uid cpaas (Special for packing) <wht@126.com>
sub 4096R/3750351A 2020-08-11
```

Step 5 Verify the signature.

Download the signature file and get the signature file by referring to the installation media.

```
gpg --verify <Signature file> <Installation package>
# If the verification is successful, the information similar to the following is displayed:
gpg: Signature made Thu 03 Sep 2020 03:51:35 PM CST using RSA key ID BB097AE6
gpg: Good signature from "cpaas (Special for packing) <wht@126.com>"
gpg --verify finger/cpaas-devops-2.14.0-20200901.tgz.sig 42.50s user 3.08s system 68% cpu 1:06.76 total
# If the warning message similar to the following is displayed, check the public key. If the public key is the
same as that provided in the download link, ignore the warning.
gpg: WARNING: This key is not certified with a trusted signature!
```

Step 6 Perform quick configurations.

On each node, run the **init.sh** script in the **res** directory as the user **root** after decompressing the installation package to quickly configure the OS.

 **CAUTION**

- The quick configuration commands do not include modifying **/etc/hosts**, upgrading the kernel version, or configuring the NTP service.
 - These commands may not fulfill all software requirements, and running quick configuration scripts does not guarantee successful deployment. To avoid deployment failures due to unqualified software configurations, it is essential to verify that all required items are correctly configured according to "Software Requirements" in *Alauda Cloud Native Success Platform Installation Guide*.
-

----End

3.3 Deploying the Platform

 **NOTE**

In this section, the default image repository is used to deploy the platform. To obtain the necessary platform component images for the cluster, you must obtain them from external image repositories and set up an image repository beforehand. For more information, contact technical support.

Procedure

Step 1 Decompress the main installation package to the **/root/cpaas-install** directory on the first master node of the global platform. (You can also decompress it to another directory. The directory for storing decompressed files must have at least 100 GiB space. After the deployment is complete, you can delete the directory.)

Step 2 Run the following commands on the first master node to decompress the package and go to the installation directory:

```
tar -xvf < Installation package file path, for example, installer-v3.6.0.tar> -C /root/cpaas-install #  
Decompress the installation package.  
cd /root/cpaas-install/installer
```

Step 3 Select the global cluster network and run the corresponding command.

- To use a Kube-OVN overlay network to deploy the global cluster, run the following command:

```
bash setup.sh
```

- To use a Calico network to deploy the global cluster, run the following command:

```
bash setup.sh --network-mode calico
```

Step 4 Enter the URL in the address box of the browser to access the platform deployment page based on the command output.

 **NOTE**

It takes about 5 minutes to wait for the minialauda to be ready.

Figure 3-2 Basic settings

Product Deployment v3.16.1

1 Basic Settings

2 Advanced

Accounts Setting

* Username: admin

The system's default administrator account.

* Password:

* Confirm Password:

System Settings

Kubernetes Version: 1.25.16-1 1.26.15 1.27.12 1.28.8

The service mesh version depends on the Kubernetes version. When choosing a version, you need to fully consider functional dependencies

Container Runtime: Containerd v1.6.28-4

Cluster Network Protocol: IPv4 Single Stack IPv6 Single Stack IPv4/IPv6 Dual Stack

* Cluster Endpoint:

Self-built VIP:

IP/Domain: 192.168.1.105 :6443

Global cluster API Server address, fill in the load balancing address if there is a high availability requirement.

GPU Type: Disable Virtualized GPU Physical GPU

* Platform Access Address

IP/Domain: 192.168.1.105

The access address of the platform, which is also the address used by the business cluster to interact with the global cluster.

Advanced

Certificate: Self-signed Certificate An Existing Certificate

Image Repository

Image Repository: Platform Deployment External

Image Repository Address

* IP/Domain: 192.168.1.105 :11443

The address of the mirror repository used by the platform is the same as the platform's access address, and the business cluster will dock to this address to obtain the mirrors used by the components.

Username: Start with a letter or number, use lower case letters, numbers, or L...

User information needed when the business cluster pulls images. After setting, the platform will automatically create a Registry user for you. The username cannot be admin.

Password:

Container Network

* Default Subnet: 10 3 0 0 / 16

After the cluster is created, new subnets are supported.

* Service CIDR: 10 4 0 0 / 16

* Join CIDR: Custom 100.64.0.0/16

Node Settings

Network Interface Card

The host network card used by the cluster network plugin. If not filled, can be configured in the node.

* Node Name: Use IP Use Hostname

Using hostname as node name requires ensuring that all node host names within the cluster are unique.

Global Cluster Platform Node Isolation

Once opened, you need to set the "Platform Exclusive" node. When you add "Control node", "Platform Exclusive" is on by default, and when you add "Compute node", "Platform Exclusive" is off by default. When you add "Compute Node", "Platform Exclusive" is turned off by default.

* Node:

SSH Port: 22 Authentication Method: Password Username: root

Controller Node: 192.168.1.105 GPU Node: SSH Connection IP Network Interface Card:

Recovery From Draft Add Node

Next Step

Issue 1.0 (2024-06-19)

Copyright © Huawei Technologies Co., Ltd.

25

Figure 3-3 Advanced settings

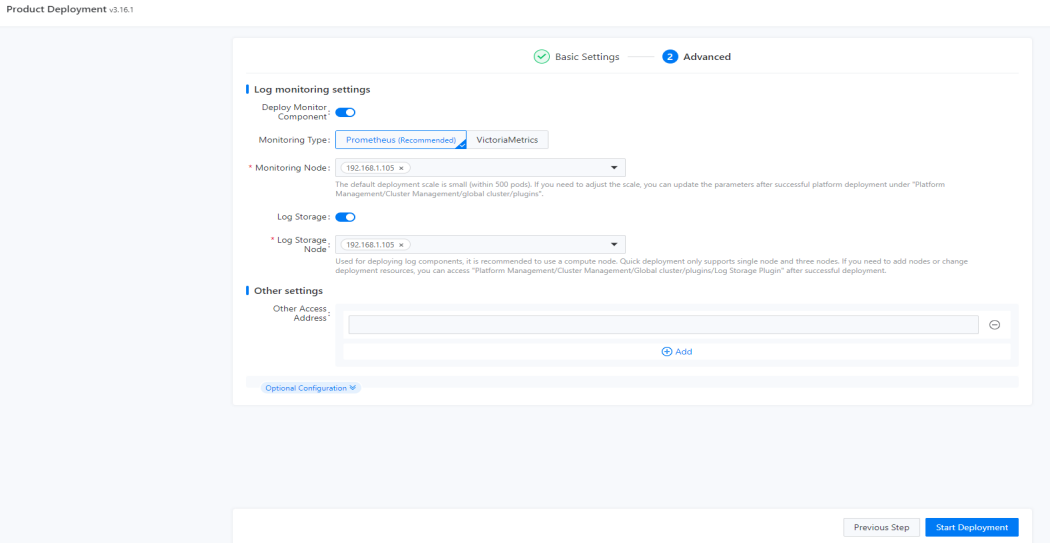


Table 3-1 Parameters

Conf igur atio n	Parame ter	Description	Remarks
Syst em Setti ngs	Kuberne tes Version	Select the Kubernetes component version.	To ensure proper function dependency, it is important to carefully consider the version of Kubernetes selected for the service mesh. The product baseline document can provide information on function dependency that may be helpful in making this decision.
	Contain er Runtim e	Select the container runtime version. containerd is used by default, helping you quickly set up a container. <ul style="list-style-type: none">containerd containerd has a shorter tracing and fewer components. It is more stable and occupies fewer node resources.	None

Configuration	Parameter	Description	Remarks
	Cluster Endpoint	<p>API server address exposed by the Kubernetes cluster where global is in. Determine the deployment architecture by referring to "Capacity Planning" in <i>Alauda Cloud Native Success Platform Installation Guide</i>.</p> <ul style="list-style-type: none"> There are two ways to configure cluster HA. You can use Keepalived or a load balancer: <ol style="list-style-type: none"> Keepalived: Enable on-premises VIP (virtual router ID) and set IP/Domain to the applied VIP. Once the cluster is successfully created and using the on-premises VIP, the access address will be <VIP>:6443. Load balancer: Disable the on-premises VIP and set IP/Domain to the IP address or domain name of the load balancer (such as the F5, IaaS load balancer, or HAProxy) outside the cluster. To quickly experience the platform's functions, you can deploy a single node and use it as both the master and compute node. You can disable the on-premises VIP and set IP address/domain name to the IP address of a prepared node. If you plan on adding master nodes to the cluster after it is created, it is recommended that you enter the IP address or domain name of the load balancer outside the cluster in the designated text box. This will make future scale-out easier. 	<p>Note:</p> <ul style="list-style-type: none"> To enable the on-premises VIP, Keepalived needs the host network to support Virtual Router Redundancy Protocol (VRRP) and all nodes to be in the same subnet as the VIP. To ensure HA of the cluster IP address, you can use an on-premises VIP. However, you will need to contact the platform network administrator or O&M personnel beforehand to apply for the VIP and virtual router ID. It is important to note that the virtual router ID must be unique within the same subnet.
	GPU Type	Select the corresponding GPU as needed.	Make sure that the node has the GPU driver installed.

Configuration	Parameter	Description	Remarks
	Hardware architecture	Select x86 or Arm.	It specifies the hardware architecture for the cluster. Once selected, only nodes with the corresponding architecture can be added to the cluster.
	Platform Access Address	<p>Access address of the platform, which is used for communications between the global cluster and the service cluster. By default, it is the same as the cluster IP address. You can configure it by referring to the following:</p> <ul style="list-style-type: none">• If you need both internal and external network access, enter an internal network address. Other access addresses of the platform can be added in the advanced settings on this installation page.• If you need DR, the address must be a domain name.• If you need a valid certificate, enter the domain name and configure a self-signed certificate in the Certificate parameter or upload a certificate issued by the certification authority.	The platform is deployed using the HTTPS protocol by default. If you want to use the insecure HTTP protocol, you can enable the option in the advanced settings.

Configuration	Parameter	Description	Remarks
Image Repository	Image Repository	<p>Repository for storing platform component images.</p> <ul style="list-style-type: none"> • Platform Deployment: A built-in image repository is created during deployment of global. All component images on the platform are pulled from this repository. • External: You need to enter necessary details of an external image repository. In DR environments, you also need to ensure that the repository IP address is accessible to all DR nodes. It is recommended to set up an image repository beforehand to store platform component images. For more information, contact technical support. 	None
Container Network	Cluster Network Protocol	<p>Implementing IPv4/IPv6 dual-stack can effectively address the issue of limited IPv4 network address resources.</p> <p>Before enabling dual-stack, make sure that all nodes have correctly configured IPv6 network addresses. Once a cluster has been created with dual-stack enabled, it cannot be reverted back to IPv4 single stack.</p> <p>In the following scenarios, it is recommended that you enable dual-stack:</p> <ul style="list-style-type: none"> • Your applications need to provide access services to users with IPv6 terminals. • You need to analyze and process the access sources that use IPv6 terminals to access services provided by your applications. • Your application system needs to use IPv6 to access other systems (such as the database system) or application systems. 	The global platform uses the Kube-OVN overlay network by default. Make sure that the container network and host network belong to different network CIDR blocks to avoid any exceptions during system deployment.

Configuration	Parameter	Description	Remarks
	Default Subnet	Default subnet CIDR block, which is the cluster CIDR block	
	Service CIDR	IP address range used by the Kubernetes ClusterIP Services. It must not overlap with the default subnet's network CIDR block.	
	Join CIDR	IP address range used for communications between nodes and container groups in Kube-OVN overlay transmission mode. It cannot overlap with the default subnet or service CIDR block.	
Node Settings	Network Interface Card	Node network interface used by the cluster network add-on. If this parameter is not specified, the system automatically obtains the network interface corresponding to the default route of the node.	None
	Node Name	Select Use IP or Use Hostname .	When selecting a hostname as the node name, you need to ensure that the hostname is unique in the cluster.
	Global Cluster Platform Node Isolation	<p>If you need both the global cluster to run platform components and the global cluster to run service applications, you can enable this function to prevent platform components and service applications from preempting resources.</p> <p>Once you enable this function, you must configure the Application Deployable for any node you add. This configuration ensures that only platform components can be deployed on the node, and that platform components (excluding certain DaemonSets) will not be scheduled to nodes without Application Deployable configured.</p>	None

Configuration	Parameter	Description	Remarks
	Add Node	<ul style="list-style-type: none"> Only one or three master nodes are supported. If there are three master nodes in the cluster, the global cluster is an HA cluster. Application Deployable: <ol style="list-style-type: none"> For a controller node: When you enabled Application Deployable, related functions are forcibly disabled. In this case, service applications cannot be deployed on this node. When you disabled Application Deployable, related functions are disabled by default, but you can still determine whether to deploy service applications on this node. For a compute node: When you enabled Application Deployable, related functions are forcibly disabled. In this case, service applications cannot be deployed on this node. When you disabled Application Deployable, related functions are forcibly enabled, so that service applications can be deployed on this node. When adding a master node or compute node, if the GPU node is enabled, you need to manually install the GPU driver and container runtime. When the Kube-OVN overlay network is used to deploy the platform, if the network interface name of a node is specified, the node uses the configured network interface. 	<ul style="list-style-type: none"> For details about how to select a deployment architecture, see "Capacity Planning" in <i>Alauda Cloud Native Success Platform Installation Guide</i>. After you click Add Node, the platform checks the availability of the node. If the verification fails, adjust the configuration following instructions and add the node again.

Configuration	Parameter	Description	Remarks
Log monitoring settings	Monitoring Type	<p>It is recommended to use Prometheus for monitoring. When choosing a node for setting up the monitoring service, it is advisable to select a non-master node.</p> <p>If VictoriaMetrics is selected, you must configure the number of VictoriaMetrics agent instances, that is, the number of VMAgents. You are advised to add one. A maximum of three can be added.</p>	<p>To properly configure the monitoring node, you must meet the requirements listed in "Hardware Configuration" in <i>Alauda Cloud Native Success Platform Installation Guide</i>. You can deploy HA monitoring components in scenarios that demand HA.</p> <p>The monitoring component is typically deployed on a small scale by default. If you need to make changes to the deployment, you can disable the monitoring component in platform management after deployment, redeploy the component, and select nodes to deploy monitoring components of varying scales.</p>
	Log Storage Node	<p>Deploy components like Elasticsearch on the node. It is recommended to choose a non-master node for setting up the log service. The default log service has limited capacity, so if you require a larger log scale, contact the relevant personnel.</p>	<p>Only one or three log nodes can be selected. If more log nodes are required, see <i>Changing the Number of Nodes Where Elasticsearch Runs</i>.</p>

Configuration	Parameter	Description	Remarks
Other settings	Other Access Addresses	You can enter multiple IP addresses or domain names.	<ul style="list-style-type: none">When entering IP addresses, make sure they can be forwarded to the cluster IP address.When entering domain names, make sure they have been resolved to the cluster IP address.
	Pod Number Limit	Maximum number of pods on each node. The default value is 110 . If all-in-one deployment architecture is used, you can increase the maximum number of pods on a node to 255 manually to ensure that there are enough pod IP addresses.	None
	Product	Select the name of the product to be deployed.	None
	extension parameters	It is not recommended to manually configure extension parameters. Doing so may result in the cluster becoming unavailable, and it cannot be modified after creation. If you need to make changes, contact technical support for assistance.	None

After setting these preceding parameters, click **Start Deployment** in the lower right corner.

----End

Verifying the Platform

To check whether the global platform has been successfully deployed, run the following command on the master node of the Kubernetes cluster where the global component is running:

```
#Check whether the Sentry component is successfully deployed. Run the following command to search for the chart that fails to be deployed:
kubectl get apprelease --all-namespaces
#Check whether all the pods are running properly. If there is a problem, run the following command to search for the failed pod:
```

```
kubectl get pod --all-namespaces | awk '{if ($4 != "Running" && $4 != "Completed")print}' | awk -F'/' '{if ($3 != $4)print}'
```

Accessing the Platform

NOTE

Using different browsers to access the platform may cause the platform UI to display incorrectly or certain functions to be unavailable due to browser compatibility issues. To avoid these issues, use the recommended browser versions listed below:

- Google Chrome 93 or later
 - Firefox 92 or later
1. After the deployment and installation are complete, click the access button on the browser page to go to the platform portal. You can download and check the deployment list.
 2. Platform maturity setting (Alpha function switch setting): *<Platform access address >/console-platform/feature-gate*

Deleting the Installer

Run **docker rm -f minialauda-control-plane** on the host where the platform is deployed to delete the installer. After the installer is deleted, the deployment list cannot be downloaded.

NOTE

You do not need to manually delete the container. The system automatically deletes the installer 10 minutes later.

3.4 Configuring the Platform

NOTE

Once the deployment is finished, certain functions or configurations on the platform may not align with the project's requirements. If this occurs, you can refer to this section to properly configure the platform.

Modifying Component Software Configurations

NOTE

To ensure your services can handle the expected scale and pressure, make sure your hardware configuration meets the requirements outlined in "Capacity Planning" of the *Alauda Cloud Native Success Platform Installation Guide*. Additionally, adjust the software configuration of your components based on this section.

Modify the log collection scope and the storage duration of logs, audit data, and events.

- **Log Collection Scope**

It can be modified on the platform UI. For details, see the *Alauda Cloud Native Success Platform User Guides*.

- **Log Retention Period**

It can be modified on the platform UI. For details, see the *Alauda Cloud Native Success Platform User Guides*.

- **Monitoring Data Retention Period**

It can be modified on the platform UI. For details, see the *Alauda Cloud Native Success Platform User Guides*.

- **Limit Modification**

Run the following command on the first master node to search for the component to be modified:
kubectrl get deploy,sts,ds -A | grep apoll # Run this command to find the Apollo resource name and then run the following command to make changes:
kubectrl edit -n cpaas-system deployment.apps/apollo

- **apollo -es-enablealias Modification**


Changing the value of this parameter to **false** will improve the log query speed. The default value is **false**. If it is set to **true**, it allows Elasticsearch index names that do not follow specifications (for example, **log-workload-20230208**) and supports log queries based on aliases in customers' on-premises Elasticsearch instances.
Run the following command on the first master node:
kubectrl edit prdb base
Search for **valuesOverride** in **.spec**. If **valuesOverride** is not present, add the key and then add the following content:
valuesOverride:
ait/chart-alauda-base:
logging:
esEnableAliases: false # A Boolean parameter which has only two values: **true** or **false**

- **Changing the Elasticsearch Fragments (ALAUDA_ES_SHARDING)**

For details, see "Platform Center" > "Platform management" > "Clusters" > "Clusters" > "Plugin management" > "Deploying the Log Storage Component" in the *Alauda Cloud Native Success Platform User Guides*.

- **Changing the Number of Nodes Where Elasticsearch Runs**

You can modify the configuration on the UI. You can log in to the platform as an administrator, choose **Platform Management** > **Clusters** > **Clusters** >

Plugins > **ElasticSearch Log Center**, click , and select **Update** from the drop-down list.

 **NOTE**

Elasticsearch can run on one node or three nodes. It is not possible to change the node to a higher number. To use an Elasticsearch that runs on more than three nodes, you will have to make some manual parameter modifications.

1. Run **moduleinfo(kubectrl get moduleinfo |grep logcenter | grep <cluster-name>)** and get the **spec.config.components.elasticsearch.nodes** part.
2. Add the name of the required node below. (You can run **kubectrl get nodes** to obtain the node name.)

4 Change History

Table 4-1 Change history

Release On	Description
2024-05-29	The issue is the first official release.