

Install

This document will provide all the information regarding the installation of ACP .

Overview

[Overview](#)

Prepare for Installation

[Prerequisites](#)

[Download](#)

[Node Preprocessing](#)

Installing

Installing

Overview

This document is intended for system administrators with basic Linux knowledge and details the full installation process of the ACP `global` cluster.

In this document, the following terms will be used frequently, please pay attention to the distinction:

- **`global` cluster**: Refers to the cluster responsible for core platform management tasks, such as cluster management and tenant management. Completing the installation of the `global` cluster completes the ACP installation. After installation, workload clusters can be accessed or created as needed.
- **Platform**: Usually refers to the container platform itself, a complete system that provides an application runtime environment and integrates container scheduling, management, networking, storage, and other functions. The **platform** is an overall concept that covers all components, including the `global` cluster and workload clusters.

INFO

Before installation, please ensure that you have completed capacity planning, environment preprocessing, and prerequisite checks to ensure that the hardware, network, and OS of each node meet the requirements. The following content covers platform architecture design, installation methods, and key term explanations to help you grasp the core points during the actual installation process.

TOC

Installation Method

Terminology Explanation

Installation Method

The installation process of the `global` cluster is mainly divided into three stages:

1.

Preparation Stage

- **Prerequisite Check:** Ensure that all node hardware, network, and OS meet the requirements, such as kernel version, CPU architecture, and network configuration.
- **Installation Package Download and Verification:** Log in to the Customer Portal to obtain the latest installation package and signature file, and use the GPG tool to verify the integrity and correctness of the package.
- **Node Preprocessing:** Complete the preprocessing work of all nodes.

2.

Execution Stage

- **Installation Package Upload and Extraction:** Upload the installation package to the target control plane node (recommended directory: `/root/cpaas-install`) and extract the installation resources.
- **Start the Installer:** Execute the installation script (such as `bash setup.sh`) on the control plane node, and select the network plugin (Kube-OVN or Calico), IP protocol mode (IPv4/IPv6/dual stack), and VIP configuration according to the actual environment.
- **Parameter Configuration:** Access the Web UI provided by the installer, and set the Kubernetes version, cluster network, node name, access address, and other key parameters in sequence to complete the installation of the `global` cluster.

3.

Verification Stage

- **System Status Check:** After the installation is complete, log in to the platform Web UI to check the cluster status and the operating status of each component.
- **CLI Verification:** Use command-line tools to check the cluster resource status to ensure that all services are running normally and there are no exceptions or failures.

Terminology Explanation

- **VIP (Virtual IP)** Virtual IP address, used for load balancing the `global` cluster's API Server, Web UI, and other access points to ensure high availability.
-

The following chapters will further explain the detailed operations, configuration parameters, and verification methods of each installation stage. Please read carefully and complete the corresponding preparatory work before the formal installation.

Prepare for Installation

Prerequisites

Download

Node Preprocessing

Prerequisites

Before installing the `global` cluster, you need to prepare hardware, network, and OS that meet the requirements.

INFO

The platform currently does not support direct installation of the `global` cluster in an existing Kubernetes environment. If your environment already has a Kubernetes cluster, please back up your data and clean the environment before installation.

TOC

Capacity Planning

Machines

Basic Requirements

ARM Architecture Requirements

Supported OS and Kernels

Network

Network Resources

Network Configuration

LoadBalancer Forwarding Rules

Capacity Planning

Before installation, you must select an appropriate installation scenario based on your goals and actual needs. Different scenarios have significant differences in infrastructure resource configuration and architecture design requirements. The following are planning recommendations for three typical scenarios:

Single Node

Scope of Application Suitable for platform function verification, demo , or technical feasibility testing. This scenario is only used to verify the core functions of the platform and does not carry production-level application traffic. The resource configuration is at the minimum level.

Resource Configuration Requirements

Dimension	Specification Requirements
Number of Nodes	1 (physical machine or virtual machine)
CPU	≥16 cores
Memory	≥32GB

Architecture Description

- **All-in-one:** The cluster has only one node, and all control plane components and applications run on that node.
- **Lightweight Load:** Can only load Demo applications with no more than 10 Pods.
- **Non-Production Use:** Does not support horizontal scaling and does not meet application continuity and high availability requirements.

Single Cluster

Scope of Application For the standardized delivery needs of ISVs (Independent Software Vendors), the `global` cluster handles both platform management and direct running of ISV applications, without the need to create additional workload clusters.

Resource Configuration Requirements

Deployment Plan	Node Type	Quantity	Minimum Specifications	Recorder
-----------------	-----------	----------	------------------------	----------

Minimum Configuration	Control Plane Node	3	8 cores 16GB + ISV Requirements	12 cor Requii
	Worker Node	0	—	—
Recommended Configuration	Control Plane Node	3	8 cores 16GB	12 cor
	Worker Node	≥2	According to ISV application load requirements	—

Architecture Description

- **Production-Ready:** The `global` cluster runs both control plane components and ISV applications simultaneously.
- **Platform and Application Isolation:** ISV applications are recommended to run on dedicated worker nodes to avoid resource contention with the control plane.
- **Scaling:** For every 50 new application Pods, add 1 worker node according to ISV application resource requirements.

Multi-Cluster

Scope of Application Suitable for large enterprise data center environments that require unified management of multiple workload clusters across clouds and regions. As the number of workload clusters increases, the `global` cluster must dynamically scale worker nodes and resource configurations to ensure high availability and high performance.

Core Features

- **Hybrid Control:** Supports unified management of cross-cloud and cross-region workload clusters.
- **Elastic Scaling:** `global` cluster resources scale linearly with the number of accessed workload clusters.
- **Physical Isolation:** It is recommended that the control plane and compute plane be physically isolated to ensure system stability.

Baseline Resource Configuration

Node Type	Quantity	Minimum Specifications	Recommended Specification
Control Plane Node	3 or 5	8 cores 16GB	16 cores 32GB
Worker Node	≥2	12 cores 24GB	24 cores 48GB

Dynamic Scaling Rules

1. **Vertical Scaling:** Single nodes can be upgraded in gradients (e.g., 12 cores 24GB → 24 cores 48GB → 48 cores 96GB).
2. **Horizontal Scaling:** For every 10 accessed workload clusters, it is recommended to expand compute resources by 20% (increase node count or upgrade node specifications).
3. **Monitoring Triggered:** When CPU/memory utilization continuously exceeds 70%, initiate scaling measures.

Architecture Description

- The control plane consists of 3 nodes forming an ETCD cluster. It is recommended to enable TLS encryption and periodic snapshots.
- Platform critical components are recommended to be deployed on independent worker nodes to avoid resource contention.
- Disaster recovery recommendation: Deploy across multiple availability zones to ensure worker nodes are distributed in at least 2 physical fault domains.

TIP

1. **Resource Redundancy:** Production environments are recommended to reserve at least 30% resource margin to cope with sudden loads.
2. **Network Planning:** The `global` cluster should be deployed in an independent VPC or VLAN to ensure bandwidth $\geq 1\text{Gbps}$.
3. **Storage Isolation:** ETCD storage is recommended to use NVMe SSD and be physically isolated from application storage.

Machines

INFO

This section describes the minimum hardware requirements for building a highly available `global` cluster. If you have completed capacity planning, please prepare the corresponding resources according to [Capacity Planning](#), or scale up as needed after installation.

Basic Requirements

At least **3** physical machines or virtual machines must be provided as control plane nodes for the cluster. The minimum configuration for each node is as follows:

Category	Minimum Requirements
CPU	≥ 8 cores, clock speed $\geq 2.5\text{GHz}$ No over-provisioning; disable power saving mode
Memory	$\geq 16\text{GB}$ No over-provisioning; recommended to use at least six-channel DDR4
Hard Drive	Single device IOPS ≥ 2000 Throughput $\geq 200\text{MB/s}$ Must use SSD

ARM Architecture Requirements

For ARM architectures (such as Kunpeng 920), it is recommended to increase the configuration to **2 times** that of the x86 minimum configuration, but not less than **1.5 times**.

For example: If x86 requires 8 cores 16GB, then ARM should reach at least 12 cores 24GB, and the recommended configuration is 16 cores 32GB.

Supported OS and Kernels

INFO

- Kernel Version Requirements:** These kernel versions have been officially released and validated by our platform tests. In your actual deployment, adherence to the **A.B.C major version numbers** is crucial, while subsequent minor versions can vary.
- Unsupported Environments:** If the OS, kernel version, or CPU architecture does not meet the requirements, please contact technical support.

Red Hat Enterprise Linux (RHEL)

- RHEL 7.8: `3.10.0-1127.el7.x86_64`
- RHEL 8.0 to 8.6: `4.18.0-80.el8.x86_64` to `4.18.0-372.9.1.el8.x86_64`

WARNING

RHEL 7.8 does not support **Calico Vxlan IPv6**.

CentOS

- CentOS 7.6 to 7.9: `3.10.0-1127` to `3.11`

WARNING

CentOS does not support **Calico Vxlan IPv6**.

Ubuntu

- Ubuntu 20.04 LTS: `5.4.0-124-generic`
- Ubuntu 22.04 LTS: `5.15.0-56-generic`

WARNING

Ubuntu HWE (Hardware Enablement) versions are not supported.

Kylin Linux Advanced Server

- Kylin V10 SP3: `4.19.90-52.22.v2207.ky10.x86_64`

WARNING

- Kylin V10, V10-SP1, and V10-SP2 have known kernel issues that may cause **NodePort network access failures**; it is recommended to upgrade to **Kylin V10-SP3**.
- ARM architecture only supports `Kunpeng 920`. For other models, please contact technical support.

Network

Before installation, the following network resources must be pre-configured. If a hardware LoadBalancer cannot be provided, the installer supports configuring **haproxy + keepalived** as a software load balancer, but you need to understand:

- **Poorer Performance:** Software load balancing performance is lower than hardware LoadBalancer.
- **Higher Complexity:** If you are not familiar with keepalived, it may cause the `global` cluster to be unavailable, problem troubleshooting will take a long time, and seriously affect platform reliability.

Network Resources

Resource	Mandatory	Quantity	Description
<code>global</code> VIP	Mandatory	1	<p>Used for nodes in the cluster to access kube-apiserver, configured in the load balancing device to ensure high availability.</p> <p>This IP can also be used as the access address for the platform Web UI.</p> <p>Workload clusters in the same network as the <code>global</code> cluster can also access the <code>global</code> cluster through this IP.</p>
External IP	Optional	On Demand	<p>When there are workload clusters that are not in the same network as the <code>global</code> cluster, such as a hybrid cloud scenario, it must be provided. Workload clusters in other networks access the <code>global</code> cluster through this IP.</p> <p>This IP needs to be configured in the load balancing device to ensure high availability.</p>

Resource	Mandatory	Quantity	Description
			This IP can also be used as the access address for the platform Web UI.
Domain Name	Optional	On Demand	If you need to access the <code>global</code> cluster or platform Web UI through a domain name, please provide it in advance and ensure that the domain name resolution is correct.
Certificate	Optional	On Demand	It is recommended to use a trusted certificate to avoid browser security warnings; if not provided, the installer will generate a self-signed certificate, but there may be security risks when using HTTPS.

INFO

A domain name must be provided in the following cases:

1. The `global` cluster needs to support IPv6 access.
2. A disaster recovery plan for the `global` cluster is planned.

NOTE

If the platform needs to configure multiple access addresses (for example, addresses for internal and external networks), please prepare the corresponding IP addresses or domain names in advance according to the table above. You can configure them in the installation parameters later, or add them according to the product documentation after installation.

Network Configuration

Type	Requirement Description
Network Speed	Speed of <code>global</code> cluster and workload cluster in the same network $\geq 1\text{Gbps}$ (recommended 10Gbps); cross-network speed $\geq 100\text{Mbps}$ (recommended 1Gbps). Insufficient speed will significantly reduce data query performance.
Network Latency	Latency $\leq 2\text{ms}$ in the same network; latency $\leq 100\text{ms}$ (recommended $\leq 30\text{ms}$) across networks.
Network Policy	Please refer to LoadBalancer Forwarding Rules to ensure that the necessary ports are open; when using Calico CNI, ensure that the IP-in-IP protocol is enabled.
IP Address Range	The <code>global</code> cluster nodes should avoid using the 172.16-32 network segment. If it has been used, please adjust the Docker configuration (add the <code>bip</code> parameter) to avoid conflicts.

LoadBalancer Forwarding Rules

This rule is designed to ensure that the `global` cluster can receive traffic from the LoadBalancer normally. Please check the network policy according to the following table to ensure that the relevant ports are open.

Source IP	Protocol	Destination IP	Destination Port	Description
<code>global</code> VIP, External IP	TCP	All control plane node IPs	443	Provides access services for the platform Web UI, image repository, and Kubernetes API Server through the HTTPS protocol. The default port is <code>443</code> . If you need to use a custom HTTPS port,

Source IP	Protocol	Destination IP	Destination Port	Description
				<p>please do the following:</p> <ul style="list-style-type: none">• Replace the destination port in the port forwarding rule with your custom port number.• Later, in the Web UI installation parameters, fill in your custom port number.
<code>global</code> VIP, External IP	TCP	All control plane node IPs	6443	This port provides access to the Kubernetes API Server for nodes within the cluster.
<code>global</code> VIP, External IP	TCP	All control plane node IPs	11443	<p>This port provides access to the image repository for nodes within the cluster.</p> <p>Note: If you plan to use an external image repository instead of the default image repository</p>

Source IP	Protocol	Destination IP	Destination Port	Description
				provided by the <code>global</code> cluster, you do not need to configure this port.

TIP

- It is recommended to configure health checks on the LoadBalancer to monitor the port status.
- If you plan to implement a disaster recovery plan for the `global` cluster, you need to open port `2379` for all control plane nodes for ETCD data synchronization between the primary and disaster recovery clusters.
- The platform only supports HTTPS by default. If HTTP support is required, you need to open the HTTP port for all control plane nodes.

Download Installation Package

Before installation, you need to download and extract the installation packages. If you require the Extensions Package, please extract the Core Package first, followed by the Extensions Package, into the same directory before starting the installation.

1 Download the Installation Packages

Log in to the **Customer Portal** to download the installation packages. If you have not registered an account, please contact technical support.

Two types of installation packages are available:

- **Core Package:** Required for all installations
- **Extensions Package:** Optional, download only if needed for additional functionality

2 Core Package Documentation

Download the core package installer-core-v4.x.y-zzzz.tar. Here, "zzzz" may be x86, arm, or hybrid, depending on the architecture chosen by the user (the same applies below).

Extract the core package using the tar command:

```
tar xvf installer-core-v4.x.y-zzzz.tar
```

3 Extensions Package Documentation

Download the extensions package installer-extensions-v4.x.y-zzzz.tar. Here, "zzzz" must match the core package. The extensions package contains plugins that are not included in the core package. If you downloaded the Extensions Package, you need to extract the extensions package to the same directory before installation:

```
tar xvf installer-extensions-v4.x.y-zzzz.tar -C <the core package extrac
```

Example: If the core package was extracted to /root, there would be an installer directory under /root after extraction. You should execute the following command in the directory where installer-extensions-v4.x.y-zzzz.tar is located:

```
tar xvf installer-extensions-v4.x.y-zzzz.tar -C /root/
```

The installation guide will detail the Core Package installation process. The Extensions Package documentation will provide specific instructions for using and configuring the extensions.

Node Preprocessing

Before installing the `global` cluster, all nodes (control plane nodes and worker nodes) must complete preprocessing.

TOC

Execute the Quick Configuration Script

Node Checks

Remove Conflicting Packages

Configure Search Domain

Execute the Quick Configuration Script

The ACP installation package provides a script for quickly configuring nodes.

Unzip the installation package to obtain the `init.sh` script file in the `res` directory. Copy the script file to the nodes and ensure that you have `root` privileges.

Execute the script:

```
bash init.sh
```



WARNING

`init.sh` cannot guarantee that all of the following checks are properly handled. You still need to continue with the steps below.

Node Checks








The following lists all the checks that must be completed on the nodes. Depending on the node's role, the required checks will vary. For example, some checks apply only to control plane nodes.

Checks are divided into two categories:


-  Indicates a check that must pass.
-  Indicates a check that must be met in specific scenarios. Please determine whether the corresponding conditions are met according to the instructions. If they are, you must resolve them.



The following is the list of checks:

- **OS and Kernel**









-  The machine's grub boot configuration must have the `transparent_hugepage=never` parameter.
-  CentOS 7.x system machine's grub boot configuration must have the `cgroup.memory=nokmem` parameter.
-  Check whether the kernel modules `ip_vs` , `ip_vs_rr` , `ip_vs_wrr` , and `ip_vs_sh` are enabled.
-  When the kernel version is lower than 4.19.0 (or RHEL is lower than 4.18.0), check whether the kernel modules `nf_conntrack_ipv4` and (for IPv6) `nf_conntrack_ipv6` are enabled.
-  If the `global` cluster plans to use `Kube-OVN` CNI, the kernel modules `geneve` and `openvswitch` must be enabled.
-  Disable apparmor/selinux and firewall.
-  Disable `swap` .

- **Users and Permissions**

-  The node's SSH user has `root` privileges and can use `sudo` without the password.

-  The `UseDNS` and `UsePAM` parameters in `/etc/ssh/sshd_config` must be set to `no`.
-  Executing `systemctl show --property=DefaultTasksMax` returns `infinity` or a very large value; otherwise, adjust `/etc/systemd/system.conf`.

- **Node Network**

-  `hostname` must comply with the following rules:
 - No more than 36 characters.
 - Starts and ends with a letter or number.
 - Contains only lowercase letters, numbers, `-`, and `.`, but cannot contain `.-`, `..`, or `-.` .
-  `localhost` in `/etc/hosts` must resolve to `127.0.0.1`.
-  The `/etc/resolv.conf` file must exist and contain `nameserver` configurations, but must not contain addresses starting with 172 (disable systemd-resolved).
-  The `/etc/resolv.conf` file should not configure search domains (if you must configure them, see [Configure Search Domain](#)).
-  The machine's IP address cannot be a loopback, multicast, link-local, all-0, or broadcast address.
-  Executing `ip route` must return a default route or a route pointing to `0.0.0.0`.
-  The nodes must not occupy the following ports:
 - **Control plane nodes:** `2379`, `2380`, `6443`, `10249` ~ `10256`
 - **Node where the installer is located:** `8080`, `12080`, `12443`, `16443`, `2379`, `2380`, `6443`, `10249` ~ `10256`
 - **Worker nodes:** `10249` ~ `10256`
-  If the `global` cluster uses **Kube-OVN** or **Calico**, ensure that the following ports are not occupied:
 - **Kube-OVN:** `6641`, `6642`
 - **Calico:** `179`

- ⚠ Ensure that the IP addresses in the network segment `172.16.x.x ~ 172.32.x.x` required by Docker are not occupied. If the IPs in this network segment are occupied and cannot be changed, please contact technical support.

- **Software and Directory Requirements:**

- ✅ Must have the following installed: `ip` , `ss` , `tar` , `swapoff` , `modprobe` , `sysctl` , `md5sum` , and `scp` or `sftp` .
- ⚠ If you plan to use local storage **TopoLVM** or **Rook**, you need to install `lvm2` .
- ✅ The `/etc/systemd/system/kubelet.service` file is not allowed to exist.
- ✅ `/tmp` mount parameters must not contain `noexec` .
- ✅ Remove packages that conflict with `global` cluster components (see [Remove Conflicting Packages](#)).
- ✅ The following files must be deleted if they exist:
 - `/var/lib/docker`
 - `/var/lib/containerd`
 - `/var/log/pods`
 - `/var/lib/kubelet/pki`

- **Cross-Node Checks**

- ✅ There must be no network firewall restrictions between nodes in the `global` cluster.
- ✅ The `hostname` of each node in the cluster must be unique.
- ✅ The time zones of all nodes must be unified, and the time synchronization error must be ≤ 10 seconds.

Appendix

Remove Conflicting Packages

Before installation, applications may already be running in the docker/containerd environment on the nodes, or software conflicting with the `global` cluster may have been installed.

Therefore, it is necessary to check and uninstall conflicting packages.

DANGER

- To avoid application interruption or data loss, be sure to confirm whether there are conflicting software packages. When a conflict is found, please develop an application switching plan and back up your data before uninstalling.
- After uninstalling conflicting packages, you still need to check whether there are other potentially conflicting binary files in directories such as `/usr/local/bin/` (such as software related to docker, containerd, runc, podman, container network, container runtime, or Kubernetes).

The following commands can be used for reference.

CentOS / RedHat**Check:**

```
for x in \
  docker docker-client docker-common docker-latest \
  podman-docker podman \
  runc \
  containernetworking-plugins \
  apptainer \
  kubernetes kubernetes-master kubernetes-node kubernetes-client \
; do
  rpm -qa | grep -F "$x"
done
```

Uninstall:

```
for x in \
  docker docker-client docker-common docker-latest \
  podman-docker podman \
  runc \
  containernetworking-plugins \
  apptainer \
  kubernetes kubernetes-master kubernetes-node kubernetes-client \
; do
  yum remove "$x"
done
```

Ubuntu

Check:

```
for x in \
    docker.io \
    podman-docker \
    containerd \
    rootlesskit \
    rkt \
    containernetworking-plugins \
    kubernetes \
; do
    dpkg-query -l | grep -F "$x"
done

for x in \
    kubernetes-worker \
    kubectl kube-proxy kube-scheduler kube-controller-manager kube-apiserver
    k8s microk8s \
    kubeadm kubelet \
; do
    snap list | grep -F "$x"
done
```

Uninstall:

```

for x in \
    docker.io \
    podman-docker \
    containerd \
    rootlesskit \
    rkt \
    containernetworking-plugins \
    kubernetes \
; do
    apt-get purge "$x"
done

for x in \
    kubernetes-worker \
    kubectl kube-proxy kube-scheduler kube-controller-manager kube-apiserver
    k8s microk8s \
    kubeadm kubelet \
; do
    snap remove --purge "$x"
done

```

Kylin

Check:

```

for x in \
    docker docker-client docker-common \
    docker-engine docker-proxy docker-runc \
    podman-docker podman \
    containernetworking-plugins \
    apptainer \
    containerd \
    kubernetes kubernetes-master kubernetes-node kubernetes-client kubernetes
; do
    rpm -qa | grep -F "$x"
done

```

Uninstall:

```

for x in \
    docker docker-client docker-common \
    docker-engine docker-proxy docker-runc \
    podman-docker podman \
    containernetworking-plugins \
    apptainer \
    containerd \
    kubernetes kubernetes-master kubernetes-node kubernetes-client kubernetes
; do
    yum remove "$x"
done

```

Configure Search Domain

In Linux OS, the `/etc/resolv.conf` file is used to configure DNS client domain name resolution settings. The `search` line specifies the domain search path for DNS queries.

Configuration Requirements

- **Number of Domains:** The number of domains in the `search` line should be less than `domainCountLimit - 3` (default `domainCountLimit` is 32).
- **Length of Single Domain:** Each domain name must not exceed 253 characters.
- **Total Character Length:** The total character count of all domain names and spaces must not exceed `MaxDNSSearchListChar` (default is 2048).

Example

```
search domain1.com domain2.com domain3.com
```

- The total number of domains is 3.
- The length of a single domain, such as `domain1.com`, is 11.
- The total character length is 35, i.e., $11 + 11 + 11 + 2$ (two spaces).

WARNING

- If the `search` line in the `/etc/resolv.conf` file does not meet the above limitations, it may cause DNS query failures or performance degradation.
- Before modifying the `/etc/resolv.conf` file, it is recommended to back up the file.

Installing

This section describes the specific steps for installing the `global` cluster.

Before starting the installation, please ensure that you have completed the prerequisite checks, installation package download and verification, node preprocessing, and other preparatory work.

TOC

Process

- Upload and Extract Installation Package

- Start the Installer

 - Network Mode and IP Family

- Parameter Configuration

- Verify Successful Installation

- Install Product Docs Plugin

- Parameter Description

- Installer Cleanup

Process

1 Upload and Extract Installation Package

Upload the Core Package installation package to any machine of the `global` cluster control plane nodes, and extract it according to the following command:

```
# Assume that the /root/cpaas-install folder already exists on the machine
tar -xvf {Path to Core Package File}/{Core Package File Name} -C /root/c
cd /root/cpaas-install/installer || exit 1
```

INFO

- This machine will become the first control plane node after the `global` cluster installation is complete.
- After the Core Package is extracted, at least **100GB** of disk space is required. Please ensure sufficient storage resources.
- If you have downloaded the Extensions Package, please extract it and refer to the documentation included within the package before proceeding to the next steps.

2

Start the Installer

Execute the following installation script to start the installer. After the installer starts successfully, the command line terminal will output the web console access address.

After waiting for about 5 minutes, you can use a browser on your PC to access the web console provided by the installer.

```
bash setup.sh
```

WARNING

Ensure that the IP address and port 8080 of the node where the installer is located can be accessed normally, so that the web console provided by the installer can be accessed smoothly after the installer starts successfully.

Network Mode and IP Family

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

The `--network-mode` parameter affects the CNI of the `global` cluster created by the installer. If this parameter is not specified, the CNI of the `global` cluster will default to Kube-OVN. If you want Calico as the CNI, you must explicitly specify `--network-mode calico`.

```
bash setup.sh --ip-family ipv6
```

If you plan to create a `global` cluster with Single-stack Network IPv6, you must explicitly specify `--ip-family ipv6` when starting the installer. Without this parameter, the `global` cluster created by the installer will support Single-stack Network IPv4 and Dual-stack Network by default.

3 Parameter Configuration

After completing the installation parameter configuration according to the page guide, confirm the installation.

[Parameter Description](#) provides detailed descriptions of key parameters. Please read carefully and configure according to actual needs.

4 Verify Successful Installation

After the installation is complete, the platform access address will be displayed on the page. Click the **Access** button to jump to the platform Web UI.

In the **Platform Management** view, click **Cluster Management > Clusters** in sequence, and find the cluster named `global`.

Select **CLI Tools** from the drop-down menu on the right, and execute the following command to verify the installation status:

```
# Check if there are failed Charts
kubectl get apprelease --all-namespaces
# Check if there are failed Pods
kubectl get pod --all-namespaces | awk '{if ($4 != "Running" && $4 != "C
```

5 Install Product Docs Plugin

INFO

The **Alauda Container Platform Product Docs** plugin provides access to product documentation within the platform. All help links throughout the platform will direct users to this documentation. If this plugin is not installed, clicking help links in the platform will result in 404 access errors.

1.

Navigate to **Administrator**.

2.

In the left sidebar, click **Marketplace > Cluster Plugins** and select the `global` cluster.

3.

Locate the **Alauda Container Platform Product Docs** plugin and click **Install**.

Parameter Description

Parameter	Description
Kubernetes Version	All optional versions are rigorously tested for stability and compatibility. Recommendation: Choose the latest version for optimal features and support.
Cluster Network Protocol	Supports three modes: IPv4 single stack, IPv6 single stack, IPv4/IPv6 dual stack. Note: If you select dual stack mode, ensure all nodes have correctly configured IPv6 addresses; the network protocol cannot be changed after setting.

Cluster Address	<p>Enter the pre-prepared domain name. If no domain name is available, enter the pre-prepared <code>global VIP</code>.</p> <p><code>Self-Built VIP</code> is disabled by default, only enable it if you have not provided a LoadBalancer. After enabling, the installer will automatically deploy <code>keepalived</code> to provide software load balancing support.</p> <p>Note: The following conditions must be met when using <code>Self-Built VIP</code>,</p> <ul style="list-style-type: none">• A usable VRID is available;• The host network supports the VRRP protocol;• All control plane nodes and the VIP must be on the same subnet. <p>Tip: For single-node deployments in feature experience scenarios, you can directly enter the node IP. There is no need to enable <code>Self-Built VIP</code> or prepare network resources such as <code>global VIP</code>.</p>
Platform Access Address	<p>If you do not need to distinguish between Cluster Address and Platform Access Address, enter the same address as the Cluster Address.</p> <p>If you need to distinguish, for example, if the <code>global</code> cluster is only for internal network access and the platform needs to provide external network access, enter the pre-prepared domain name or <code>External IP</code>.</p> <p>The platform uses HTTPS access by default and does not enable HTTP. If you need to enable HTTP access, enable it in Advanced Settings (not recommended).</p> <p>Note: A domain name must be entered in the following cases,</p> <ul style="list-style-type: none">• A disaster recovery plan for the <code>global</code> cluster is planned;• The platform needs to support IPv6 access. <p>Tip: If you need to configure more platform access addresses, you can add them in Other Settings > Other Platform</p>

	<p>Access Addresses in the next step. Or, after installation, add them in platform management according to the user manual.</p>
Certificate	<p>The platform provides self-signed certificates to support HTTPS access by default.</p> <p>If you need to use a custom certificate, you can upload an existing certificate.</p>
Image Repository	<p>The <code>Platform Deployment</code> image repository is used by default, which contains images of all components.</p> <p>If you need to use an <code>External</code> image repository, please contact technical support to obtain the image synchronization plan before configuring.</p>
Container Network	<p>The default subnet and Service network segment of the cluster cannot overlap.</p> <p>When using the Kube-OVN Overlay network, ensure that the container network and the host network are not in the same network segment, otherwise it may cause network exceptions.</p>
Node Name	<p>If you select <code>Host Name as Node Name</code>, ensure that the host names of all nodes are unique.</p>
<code>global</code> Cluster Platform Node Isolation	<p>Enable only when you plan to run application workloads in the <code>global</code> cluster.</p> <p>After enabling:</p> <ul style="list-style-type: none"> Nodes can be set to <code>Platform Exclusive</code>, i.e., only run platform components, ensuring platform and application workloads are isolated; Workloads of the DaemonSet type are excluded.
Add Node	Control Plane Node:

- Supports adding 1 or 3 control plane nodes (3 for high availability configuration);
- If `Platform Exclusive` is enabled, `Deployable Applications` is forced to be disabled, and control plane nodes only run platform components;
- If `Platform Exclusive` is disabled, you can choose whether to enable `Deployable Applications`, allowing control plane nodes to run application workloads.

Worker Node:

- If `Platform Exclusive` is enabled, `Deployable Applications` is forced to be disabled;
- If `Platform Exclusive` is disabled, `Deployable Applications` is forced to be enabled.

When using Kube-OVN, you can specify the node network card by entering the gateway name.

If the node availability check fails, please adjust it according to the page prompt and add it again.

Installer Cleanup

Normally, the installer will be automatically deleted after installation. If the installer is not automatically deleted after 30 minutes of installation, please execute the following command on the node where the installer is located to force delete the installer container:

```
docker rm -f minialauda-control-plane
```