Overview

Introduction

Architecture

Platform Mo

Cluster Management Models

Core and Extensions

Availability

tes

Release Notes

Learn More

Introduction

Alauda Container Platform (ACP) is a Kubernetes-based platform for building, deploying, operating, and governing cloud-native applications across private cloud, data center, hybrid cloud, multi-cloud, and edge environments.

ACP keeps Kubernetes APIs and workload primitives at the center of the platform, then adds enterprise capabilities for cluster management, project governance, application delivery, extension management, observability, security, upgrade planning, and recovery.

TOC

What the Platform Provides

Relationship To Kubernetes

Reader Path

What the Platform Provides

Area	What it helps you do	Continue with
Platform management	Install ACP Core, access the web console and APIs, and manage platform-wide services from the `global` cluster.	Architecture

Area	What it helps you do	Continue with
Cluster operations	Create ACP lifecycle-managed workload clusters or onboard third-party Kubernetes environments for centralized governance.	Cluster Management Models
Projects and namespaces	Understand how projects, namespaces, users, quotas, and associated clusters form the platform governance model.	Platform Model
Application delivery	Create and operate applications from images, YAML, charts, source code, and Kubernetes workload resources.	Developer Overview
Extensions	Add capabilities through Operators and Cluster Plugins, and check version compatibility before installation or upgrade.	Core and Extensions
Availability and recovery	Plan deployment topology, high availability, Global Cluster Disaster Recovery, and backup or restore paths.	Availability and Recovery
Version planning	Understand Kubernetes compatibility, third-party cluster onboarding range, runtime baseline, and lifecycle boundaries.	Version and Lifecycle
Security and access	Configure LDAP or OIDC identity providers, Kubernetes RBAC-based authorization, audit, network policy, API filtering, compliance features, and security product integrations.	Security

Relationship To Kubernetes

ACP is built on Kubernetes. Users still work with Kubernetes resources such as clusters, namespaces, workloads, Services, Ingresses, NetworkPolicies, custom resources, and RBAC objects. ACP adds a management plane that helps platform teams apply governance, lifecycle management, observability, and operational workflows across multiple Kubernetes environments.

Some capabilities are included with ACP Core. Other capabilities are provided through separately installed Extensions, such as Operators and Cluster Plugins, or through separate products. Before planning or using those capabilities, check the corresponding documentation for installation, upgrade, compatibility, and limitations.

Reader Path

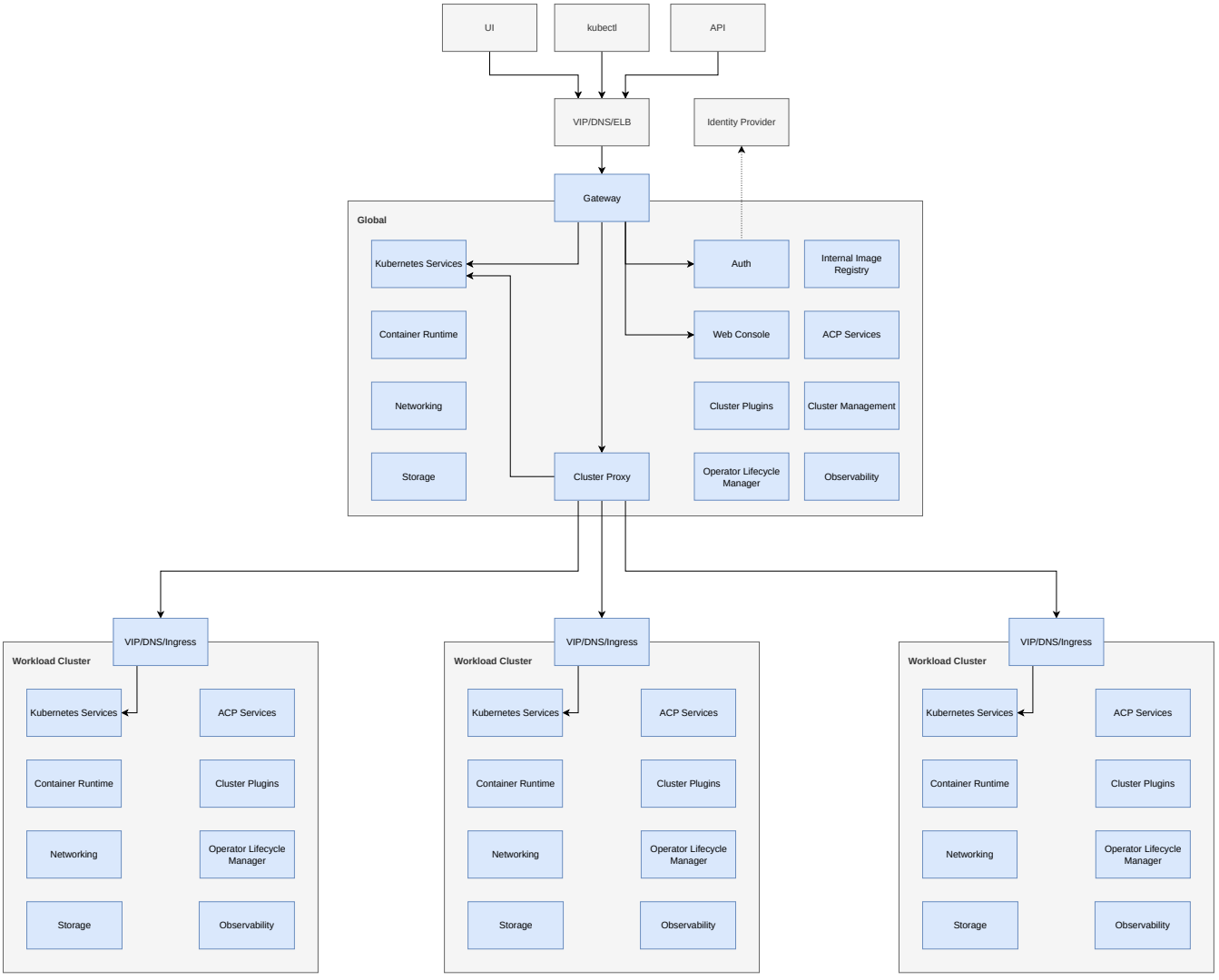
If you are new to ACP, read these pages first:

1. Architecture to understand the hub-and-spoke model.
2. Platform Model to understand the `global` cluster, workload clusters, third-party clusters, projects, and namespaces.
3. Cluster Management Models to understand infrastructure responsibility, control-plane topology, and third-party onboarding.
4. Core and Extensions to understand which capabilities belong to Core and which require an Extension.
5. Learn More to choose the next task path for installation, cluster operations, application delivery, upgrade planning, recovery, security, API, or CLI work.

Architecture

ACP uses a hub-and-spoke architecture. The `global` cluster provides the central management plane, and workload clusters or third-party clusters provide Kubernetes environments where applications and cluster-local components run.

Use the diagram as a high-level view of this model. The architecture details below clarify boundaries that are not visible in the diagram.



For platform-wide terms, see Glossary. For the conceptual platform model, see Platform Model.

TOC

Hub-And-Spoke Model

Access And Request Path

Management Path

State And Data Domains

Core And Extension Boundary

Availability And Recovery Model

Continue Reading

Hub-And-Spoke Model

Part	Role
`global` cluster	Hosts the platform management plane, including web console access, platform APIs, authentication integration, project governance, cluster management, Extension management, and platform-wide coordination.
Workload clusters	Run application workloads and cluster-local components under governance from the `global` cluster.
Third-party clusters	Existing Kubernetes environments onboarded to the platform for centralized governance and operations within documented prerequisites and caveats.

The hub-and-spoke model describes the relationship between the `global` cluster and the Kubernetes environments that it manages. For control-plane topology, including Hosted Control Plane, see Cluster Management Models.

The `global` cluster is the platform hub. Workload clusters and third-party clusters are spokes. A project can span associated clusters, and namespaces provide Kubernetes resource isolation inside those clusters.

Access And Request Path

Users and automation usually access the platform through the Platform Access Address, web console, platform APIs, Kubernetes APIs, or CLI tools.

At a high level:

1. A user, CLI, automation tool, or browser connects to the Platform Access Address.
2. A load balancer, VIP, DNS record, or ingress path routes traffic to platform entry components.
3. Platform API services evaluate authentication and authorization.
4. Requests are handled by the `global` cluster or forwarded to target clusters through the documented cluster access path.
5. Kubernetes API requests to target clusters are evaluated against the user's effective RBAC permissions.

For API usage, see API Introduction. For CLI tools, see CLI Tools. For cluster KubeConfig access, see Access a Cluster with KubeConfig.

Management Path

The `global` cluster coordinates platform management operations:

- Cluster creation and lifecycle operations for supported ACP lifecycle-managed cluster models.
- Third-party cluster onboarding through import or register workflows.
- Project, namespace, user, role, and quota governance.
- Operator and Cluster Plugin discovery, package upload, installation, and upgrade entry points.
- Central views for observability, audit, inventory, and operations when the required components are installed.

For third-party clusters, the management path depends on version range, connectivity, credentials, provider prerequisites, installed components, and Extension compatibility.

Onboarding a third-party cluster does not make ACP the owner of that cluster's Kubernetes lifecycle, node lifecycle, or provider infrastructure lifecycle.

For model boundaries, see Cluster Management Models.

State And Data Domains

Different data domains have different protection and recovery paths:

Domain	Primary location or owner	Recovery path
`global` cluster Kubernetes resource state	`global` cluster etcd	Global Cluster Disaster Recovery or etcd backup/restore, depending on scenario.
Workload cluster Kubernetes resource state	Each workload cluster etcd	Cluster-specific backup/restore or upgrade procedures.
Third-party cluster lifecycle data	External cluster owner, distribution, or provider	External provider or owner procedures, plus ACP onboarding and operations records.
Registry data	Registry component or backend	Registry backup and recovery procedures.
Monitoring and logging data	Installed monitoring and logging components	Component-specific backup or restore procedures.
Application data	Application namespaces, persistent volumes, databases, and storage backends	Application backup/restore or the data-service-specific recovery procedure.

Global Cluster Disaster Recovery protects only the `global` control-plane DR scenario documented in Availability and Recovery. It is not full platform data DR and is not application data DR.

Core And Extension Boundary

ACP Core provides the platform management plane and the frameworks used to manage clusters, projects, users, APIs, and Extensions.

Additional capabilities can require separately installed Extensions, such as Operators and Cluster Plugins, or separate products. Examples include specific observability backends, registry plugins, API filtering, compliance features, container security, data services, GitOps, logging, Immutable Infrastructure, and Hosted Control Plane.

Before planning those capabilities as part of an architecture, check the corresponding documentation for installation, upgrade, compatibility, and limitations.

For details, see Core and Extensions.

Availability And Recovery Model

Availability is planned by topology:

- Single Node is for testing or proof-of-concept only.
- Single Cluster can support production use when it fits the business scenario and resources are planned for both platform components and workloads.
- Multi-Cluster separates the `global` cluster management plane from workload clusters and should avoid non-platform business workloads on the `global` cluster.
- Three control plane nodes are the HA baseline. Five control plane nodes are scale and reliability guidance, not a universal production gate.

For detailed planning, sizing, component-level high availability, Global DR, and backup/restore entry points, see Availability and Recovery.

Continue Reading

- Introduction
- Platform Model

- Cluster Management Models
- Core and Extensions
- Learn More

Platform Model

ACP uses a hub-and-spoke model. The `global` cluster provides the central platform management plane, while workload clusters and third-party clusters run or expose Kubernetes workloads under platform governance.

Use the platform model to understand the main platform objects and how they relate to each other.

TOC

The `global` Cluster
Workload Clusters
Third-Party Clusters
Hosted Control Plane
Projects And Namespaces
What Happens Where

The `global` Cluster

The `global` cluster is deployed during ACP Core installation. It provides the central management plane for web console access, platform APIs, users and roles, project governance, cluster management, Extension management, and platform-wide coordination.

The `global` cluster is not just another workload cluster in the management model. In Multi-Cluster deployments, avoid running non-platform business workloads on the `global` cluster. In Single Cluster deployments, the `global` cluster also carries business workloads, so resource planning must include both platform components and application workloads.

For installation planning, see Install Overview and Plan.

Workload Clusters

A workload cluster is a Kubernetes cluster that runs application workloads and is managed from the `global` cluster. Workload clusters can be created and lifecycle-managed by ACP when the selected infrastructure model supports that lifecycle.

Workload clusters inherit platform governance and can be associated with projects and namespaces. They can also receive Extensions, observability components, networking components, storage integrations, and application workloads according to the selected cluster model and installed capabilities.

For cluster creation tasks, see Clusters Overview and Creating an On-Premise Cluster.

Third-Party Clusters

A third-party cluster is a Kubernetes environment whose Kubernetes distribution and lifecycle are provided or managed outside Alauda. This includes existing Kubernetes environments, external Kubernetes distributions, and public cloud Kubernetes services.

ACP can onboard third-party clusters for centralized governance, resource visibility, application operations, Extension installation, and observability integration when the required version range, connectivity, credentials, installed components, provider prerequisites, and Extension compatibility are satisfied.

For third-party clusters, ACP does not own the complete Kubernetes lifecycle, node lifecycle, or provider infrastructure lifecycle unless a specific provider or plugin document states that capability. Kubernetes installation, Kubernetes upgrades, node scaling, and infrastructure operations usually remain the responsibility of the cluster owner, external distribution, or cloud provider.

`Third-party cluster` is the product-model term for these onboarded environments. `Managed Cluster` is the current UI and navigation label for the interface that manages them.

For onboarding workflows, see Third-Party Cluster Onboarding, Import Third-Party Clusters, and Register Cluster.

Hosted Control Plane

Hosted Control Plane (HCP) is a control-plane topology, not a cluster model parallel to Installer-Provisioned Infrastructure (IPI) or User-Provisioned Infrastructure (UPI). In HCP, each hosted cluster has its own hosted control plane, and multiple hosted control planes run as workloads on a management cluster. In ACP, HCP is implemented through Kamaji (`TenantControlPlane`).

For current HCP maturity, operating system, connectivity, production-use boundaries, and tasks, see About Hosted Control Plane.

Projects And Namespaces

A project is a governance unit for tenants, teams, or business systems. It can span multiple associated clusters and provides a boundary for users, quotas, policies, and resource visibility.

A namespace is a Kubernetes resource isolation unit. In ACP, a namespace can be created in or imported into a project so that project-level governance applies to the namespace and its workloads.

The relationship is:

Object	Relationship in ACP
`global` cluster	Hosts the central management plane and manages platform governance state.
Workload cluster or third-party cluster	Provides Kubernetes capacity and cluster-local APIs where workloads run.

Object	Relationship in ACP
Project	Defines a tenant, team, or business-system governance boundary and can be associated with one or more clusters.
Namespace	Exists in an associated cluster and can be created in or imported into a project so that project governance applies.
Workloads and resources	Run inside namespaces and inherit the applicable project, namespace, and cluster policies.

For project and namespace concepts, see Project Introduction and Namespace Management.

What Happens Where

Area	Usually happens in the `global` cluster	Usually happens in workload or third-party clusters
Platform access	Web console, platform APIs, central authentication flow, and platform-level controllers	Cluster API endpoints and cluster-local Kubernetes APIs
Governance	Users, roles, projects, quotas, policies, and cluster association	Namespace-level Kubernetes resources and workload permissions
Cluster lifecycle	Cluster creation, onboarding records, upgrade orchestration for supported cluster models, and Extension management	Kubernetes control plane, nodes, workloads, cluster-local components, and provider-owned lifecycle
Observability and audit	Central views, queries, dashboards, and audit interfaces when required services are installed	Metric, log, event, or audit collection components, subject to provider and connectivity boundaries
Recovery	Global Cluster Disaster Recovery and platform-state recovery planning	Cluster-level backup/restore, component data recovery, and application backup/restore

For the high-level architecture, see Architecture. For model comparisons and responsibility boundaries, see Cluster Management Models.

Cluster Management Models

Use the cluster management model to separate three planning decisions:

1. Infrastructure responsibility.
2. Control-plane topology.
3. Kubernetes ownership and onboarding.

Each cluster path combines choices from these axes. Installer-Provisioned Infrastructure (IPI), User-Provisioned Infrastructure (UPI), Hosted Control Plane (HCP), and the current UI label `Managed Cluster` are not one flat list of mutually exclusive cluster types.

TOC

Infrastructure Responsibility

Control-Plane Topology

Kubernetes Ownership And Onboarding

Third-Party Cluster Capability Boundaries

Provider And Workflow Caveats

Infrastructure Responsibility

Infrastructure responsibility explains who provides and manages machines, node operating systems, and Kubernetes lifecycle.

Model	Machines and nodes	Node operating system	Kubernetes lifecycle	Main docs
Installer-Provisioned Infrastructure (IPI)	Provisioned through the platform and its infrastructure provider integration.	Managed by the platform with Immutable OS.	Managed by the platform, including supported provisioning, scaling, and upgrade flows.	About Immutable Infrastructure
User-Provisioned Infrastructure (UPI)	Prepared by the user as physical or virtual machines.	Managed by the user.	Installed and managed by the platform after the user prepares nodes.	Creating an On-Premise Cluster

IPI and UPI describe responsibility boundaries. They do not by themselves describe whether the control plane is hosted or dedicated, and they do not describe whether a third-party cluster has been imported or registered.

Existing third-party Kubernetes environments are covered under Kubernetes Ownership And Onboarding, because their Kubernetes distribution and lifecycle usually exist outside Alauda.

Control-Plane Topology

Control-plane topology explains where Kubernetes control plane components run.

Topology	Meaning	Support boundary
Dedicated control plane	Control plane components run on control plane nodes in the target cluster.	This is the normal topology for current production guidance.
Hosted Control Plane	Each hosted cluster has its own control plane, but the control plane runs as workloads on a management cluster. In ACP,	See About Hosted Control Plane for current maturity, operating system,

Topology	Meaning	Support boundary
	HCP is implemented through Kamaji (`TenantControlPlane`).	connectivity, and production-use boundaries.

HCP is an architecture for the control plane. It is not a ACP Core default capability and is not a peer concept to IPI or UPI.

For more information, see About Hosted Control Plane.

Kubernetes Ownership And Onboarding

Kubernetes ownership explains whether ACP owns the Kubernetes lifecycle or whether the Kubernetes environment already exists outside Alauda.

Model	How it enters the platform	Lifecycle boundary
ACP lifecycle-managed workload cluster	Created through supported ACP cluster workflows.	ACP manages the supported Kubernetes lifecycle for the selected model.
Third-party cluster	Onboarded through import or register workflows.	ACP provides central governance and operations within documented prerequisites and caveats. The external cluster owner, distribution, or provider usually owns Kubernetes lifecycle, node lifecycle, and provider infrastructure lifecycle.

`Managed Cluster` is the current UI and navigation label for third-party cluster onboarding and management areas. For product-model planning, use `Third-party cluster`.

`Import cluster` and `Register cluster` are onboarding methods for third-party clusters:

Onboarding method	Connection model	Use when
Import cluster	The `global` cluster connects to the target cluster API server with	The platform can reach the target cluster API server and the operator

Onboarding method	Connection model	Use when
	supplied address, CA, and credentials.	can provide the required cluster information.
Register cluster	A reverse proxy service in the target cluster initiates registration and establishes a tunnel to the platform.	The target cluster should initiate the connection, or direct platform access to the target API server is restricted.

After onboarding, expected day-2 management is treated as the same at the Overview level. Provider and workflow-specific caveats still apply.

Third-Party Cluster Capability Boundaries

ACP can provide these entry capabilities for third-party clusters when prerequisites are met:

- Resource visibility and centralized governance.
- Project and namespace association.
- Application operations.
- Operator and Cluster Plugin installation, subject to Extension compatibility.
- Observability integration, subject to installed components, network paths, credentials, and provider caveats.

Third-party cluster onboarding does not mean that ACP manages every Kubernetes version, provider operation, node operation, certificate, control-plane metric, audit source, ingress path, storage class, or Extension image on every third-party cluster.

Use the product-supported onboarding range in the Kubernetes Support Matrix to confirm Kubernetes-version eligibility. Check provider prerequisites and exact Extension compatibility separately.

For the exact matrix and upgrade relationship, see Kubernetes Support Matrix and Version and Lifecycle.

Provider And Workflow Caveats

Use the following caveat categories to decide which provider or workflow documentation to check for provider-specific details.

Caveat category	What to check	Continue with
Kubernetes lifecycle	Whether Kubernetes installation or upgrade is managed by ACP or by the external owner.	Clusters Overview
Node lifecycle	Whether adding, deleting, or scaling nodes is supported from the platform UI for the selected model.	Node Management
Connectivity	Whether the `global` cluster and target cluster can reach the required endpoints and whether a platform URL annotation is needed.	Network Configuration for Imported Clusters
Certificates	Which certificates are visible or rotated by the platform for the selected workflow.	Provider or onboarding workflow docs under Import Third-Party Clusters
Audit and metrics	Whether audit data and control-plane metrics are available from the target environment.	Audit and provider workflow docs
Ingress and storage	Whether post-import initialization is required for ingress, load balancing, or storage classes.	Public Cloud Cluster Initialization
Extension compatibility	Whether the exact Operator or Cluster Plugin version supports the target ACP version and cluster model.	Core and Extensions

If you need to choose what to read next, see Learn More.

Core and Extensions

ACP capabilities are delivered through ACP Core and Extensions that are distributed as separate packages. Understanding this boundary helps you plan installation, upgrade, compatibility checks, and troubleshooting.

TOC

Core
Package Boundary
Extension Types
Lifecycle Types
Compatibility Source
Capability Boundaries

Core

ACP Core deploys the `global` cluster and provides the platform management plane, platform API and CLI access, cluster management, project and namespace management, users and RBAC, Extension management frameworks, and core communication paths such as Cluster Proxy. Customer-facing capabilities such as the Web Console are delivered by Aligned Extensions.

Core capabilities are installed with ACP Core or are part of the platform management plane deployed with the `global` cluster. Capabilities that require a separate installation flow, Operator, Cluster Plugin, or separate product are not default Core functionality.

For installation scope, see Install Overview.

Package Boundary

The ACP Core Package contains Core artifacts only. It does not include Aligned or Agnostic Extension packages.

Some installation and upgrade procedures require you to download specific Aligned Extension packages separately and copy them into the `plugins/` directory of the extracted Core Package. This is a manual artifact preparation step. Placing an Extension package in that directory does not make the Extension part of Core or change its lifecycle type.

Extension Types

Extension is the umbrella term for the two main extension mechanisms in ACP:

Extension type	Technology	Where installation is executed	Start here
Operator	Operator Lifecycle Manager (OLM), OperatorHub, `CatalogSource`, `Subscription`, `InstallPlan`, and `ClusterServiceVersion`.	In the target cluster where the Operator runs.	Operator
Cluster Plugin	`ModulePlugin`, `ModuleConfig`, and `ModuleInfo` custom resources.	YAML-based installation creates `ModuleInfo` in the `global` cluster; the plugin target depends on affinity and configuration.	Cluster Plugin

OperatorHub is the interface for discovering, installing, upgrading, and managing Operators. Cluster Plugins are published to the `global` cluster and installed through the platform according to each plugin's configuration and affinity.

For the Extension section, see Extend.

Lifecycle Types

Operators and Cluster Plugins use the same lifecycle model:

Lifecycle	Meaning	Upgrade behavior
`Core`	Follows the ACP Core release and cluster Distribution Version.	Updated by upgrading the cluster. Standalone upgrade is not supported.
`Aligned`	Follows the ACP release stream but can be updated independently when a compatible version is published.	Can be upgraded independently when the exact version is compatible.
`Agnostic`	Released independently from ACP.	Can be upgraded independently when the exact version is compatible.

Cluster Plugins expose this model through the `cpaas.io/lifecycle-type` label. For Operators, check compatibility through the Alauda Customer Portal metadata, Extension documentation, or release guidance instead of expecting every lifecycle label to be visible in the product interface.

For Cluster Plugin details, see Cluster Plugin.

Compatibility Source

For a specific Operator or Cluster Plugin version, use the Alauda Customer Portal `ACP compatible versions` field as the compatibility authority. The exact Extension version must list the target ACP release as compatible.

Use product documentation for each Extension to understand capabilities, installation steps, upgrade steps, limitations, known issues, supported Kubernetes versions, architectures, providers, and cluster models. Release notes are useful for version changes, but they are not the primary compatibility source for every specific Extension version. The Kubernetes support matrix describes platform and Kubernetes version relationships and the default product-validation scope for third-party clusters; it is not an Extension compatibility matrix.

For package handling, see Upload Packages and Download Packages.

Capability Boundaries

Use these boundaries when planning installation, upgrade, and operations:

Capability area	How to read it
Registry	The platform can provide registry entry points and registry-related workflows. Registry plugins, backends, backup, and recovery have their own installation and operational boundaries.
Observability	Centralized observability depends on installed monitoring, logging, event, tracing, or inspection components. Provider and connectivity caveats apply to third-party clusters.
Audit	Audit uses platform audit views and depends on logging service components. Third-party provider audit data is not universally available.
Security and compliance	Authentication, RBAC, NetworkPolicy, API filtering, compliance features, and security products have separate component boundaries and installation requirements.
Separate products and independently installed capabilities	Service Mesh, AI, Hyperflux, Data Services, Cost Management, DevOps, Container Security, Compliance Service, GitOps, Logging Service, Immutable Infrastructure, Hosted Control Plane, and Cluster Authentication are not default Core functionality. Check the corresponding documentation before planning installation, upgrade, compatibility, or limitations.

For a task-oriented route map, see Learn More.

Availability and Recovery

Plan availability and recovery by matching your environment to the appropriate deployment topology, high availability baseline, Global Cluster Disaster Recovery path, and backup or restore procedure.

For detailed sizing, backup, restore, and disaster recovery procedures, follow the linked topic-specific procedures. The guidance below focuses on planning choices and support boundaries.

TOC

Deployment Topology
High Availability Baseline
Component-Level High Availability
Global Cluster Disaster Recovery
Backup And Restore Paths
Recovery Checklist

Deployment Topology

Choose topology by scenario, not by a single production minimum table.

Topology	Use when	Availability notes
Multi-Cluster	One `global` cluster manages multiple workload clusters.	Avoid non-platform business workloads on the `global` cluster. Size the `global` cluster for the number of managed clusters, user concurrency, API traffic, and installed plugins.
Single Cluster	One cluster intentionally runs both platform components and business workloads.	Production-capable when it fits a single-business-system scenario. Plan resources for both platform components and workloads. The `global` cluster control plane uses 3 nodes in this mode.
Single Node	Test or proof-of-concept environments.	Do not use for production.

For installation planning, see Plan and Prerequisites.

High Availability Baseline

For production-like environments, use 3 control plane nodes as the HA baseline. A 5 control plane node topology can improve scale and reliability for larger environments, but it is not a universal hard requirement for every production deployment.

Infra nodes or custom role nodes are useful for isolating platform components or high-load components, but they are not a universal requirement unless a sizing tier or component document says so. For Extra Large `global` cluster sizing, follow the sizing guidance for dedicated infra nodes.

Planning topic	Source
`global` cluster sizing by managed cluster count	Evaluating Resources for Global Cluster
Workload cluster control plane sizing and scale factors	Evaluating Resources for Workload Cluster
Node roles and infra/custom role nodes	Cluster Node Planning

Planning topic	Source
Node requirements and OS/kernel support	Node Preprocessing

Component-Level High Availability

Standard HA topology deployments rely on the following mechanisms across platform components. This section is reference context for architects; you do not configure these per component yourself when the platform is deployed in an HA topology.

etcd

- Deployed on the `global` cluster control plane nodes.
- Uses the RAFT consensus protocol for leader election and replicated state.
- A three control plane node deployment tolerates one node failure; a five control plane node deployment tolerates two.
- Supports local and remote object storage snapshot backups for recovery.

Platform access and ingress

- The cluster ingress gateway runs with multiple replicas and leader election.
- An external load balancer, DNS record, or a self-built VIP with heartbeat detection and active-standby failover provides the platform access entry point.

Image registry and object storage

- The image registry component runs in multiple replicas behind the cluster ingress, with a highly available database backend and a highly available cache backend.
- The object storage backend used by the platform runs in distributed mode with erasure coding, data redundancy, and automatic recovery.

Monitoring and logging

- Monitoring runs as multiple instances with query-time deduplication, distributed storage components, and cross-region redundancy options.

- Logging components are deployed in distributed, multi-replica modes across ingestion, transport, storage, and query.

Container network

- The container network interface (CNI) achieves high availability through stateless per-node agents and triple-replica control plane components.

Self-healing

- Health checks, failover, and traffic redirection are coordinated by the Kubernetes control plane and platform components.
- Transient failures are retried and workloads are rescheduled onto healthy nodes when a node becomes unavailable.

For topology-level planning, see High Availability Baseline. For cluster-level disaster recovery, see Global Cluster Disaster Recovery.

Global Cluster Disaster Recovery

Global Cluster Disaster Recovery protects the platform management entry point and `global` control-plane services when the Primary `global` cluster becomes unavailable.

Global DR has the following scope:

- It uses Primary and Standby `global` clusters.
- It relies on real-time synchronization of resource state stored in the Primary `global` cluster etcd, except excluded namespaces.
- It restores the platform entry point and `global` control-plane services by switching DNS or VIP access to the Standby cluster.
- Primary and Standby should follow the validated path of aligned versions, patches, component versions, and key configuration.

Do not treat Global DR as full platform data DR, application data DR, automatic failover, or an SLA-backed RPO/RTO commitment. Global DR does not cover registry data, chartmuseum data, other component data, application data, or resources excluded from etcd synchronization.

For the procedure and supported scenarios, see Global Cluster Disaster Recovery.

Backup And Restore Paths

Recovery is composed by scenario. Each mechanism protects a specific data domain and has its own procedure, prerequisite, and limitation.

Scenario	Use	Source
Primary `global` cluster failure	Global Cluster Disaster Recovery.	Global Cluster Disaster Recovery
Accidental cluster-state deletion or rollback	etcd backup and restore.	etcd Backup and Restore
Registry image repository data	Registry backup and recovery.	Registry Data Backup and Recovery
Monitoring data	Monitoring component backup or restore.	VictoriaMetrics Backup and Recovery
Logging data	Logging component backup or restore, according to the installed logging backend.	Logging Service
Application resources and persistent volumes	Application backup and restore with Data Backup Essentials and Data Backup for Velero.	Backup Overview

Application backup can protect namespaces, Kubernetes resources, and persistent volume data according to the backup configuration. It does not support every storage or application data pattern. For example, `hostPath` PersistentVolumes are not supported by the documented application backup path, and database workloads should follow data-service-specific backup guidance.

Recovery Checklist

Use this checklist to choose follow-up work. For detailed steps, follow the linked procedures:

1. Choose Single Cluster, Multi-Cluster, or Single Node during installation planning.
2. Size the `global` cluster and workload clusters from the scalability guidance.
3. Decide whether Global Cluster Disaster Recovery is required before installing ACP Core.
4. Configure backups for etcd, registry data, monitoring data, logging data, and applications according to the data domains you need to recover.
5. Run regular recovery checks and failover drills where your operational process requires them.
6. Verify platform access, `global` services, connected cluster access, and component-level recovery after failover or restore.

For related planning and operations paths, see Learn More.

Version and Lifecycle

Use this guidance to plan ACP version compatibility, Kubernetes support, third-party cluster onboarding, runtime and OS boundaries, and Extension lifecycle checks.

Use Kubernetes Support Matrix for the exact Kubernetes version table.

TOC

Product And Kubernetes Versions
Third-Party Cluster Onboarding Range
Runtime, OS, And CPU Architecture
Extension Lifecycle
Feature Maturity
Upgrade Reading Path

Product And Kubernetes Versions

The Kubernetes support matrix separates three decisions: which Kubernetes targets can be used for platform-managed cluster creation, which workload-cluster versions satisfy the `global` cluster upgrade prerequisite, and which Kubernetes minors are product-supported for third-party cluster onboarding. Do not infer one range from another.

For ACP 4.3 and later, workload clusters only need to remain within the documented Compatible Versions before a `global` cluster upgrade. In ACP 4.2 and earlier, workload clusters must be upgraded to the latest Kubernetes version in the compatible list before upgrading the `global` cluster.

Exact Kubernetes builds and creation targets come from versions offered by the installed platform patch and release-specific artifacts, not from this lifecycle overview.

For the exact table, see Kubernetes Support Matrix.

Third-Party Cluster Onboarding Range

Use Product-Tested Scope for Third-Party Clusters for the current product-supported Kubernetes range, the ongoing validation policy, and the default validation scope.

The product-tested onboarding range is shared by ACP 4.3 and later releases that provide third-party cluster onboarding. It is maintained independently from the release-specific creation and Compatible Versions rows, and the shared range is updated when product validation expands.

This range is separate from:

- The Kubernetes version supported for platform-managed cluster creation.
- The workload-cluster compatible versions used before upgrading the `global` cluster.
- Support for every provider, capability, operation, or Extension.

The published Kubernetes range establishes version eligibility. Follow Third-Party Cluster Onboarding and the selected Import, Register, or provider-specific procedure for connectivity, credentials, and environment-specific preparation.

For cluster model boundaries, see Cluster Management Models.

Runtime, OS, And CPU Architecture

Read runtime, OS, CRI, and CPU architecture information by responsibility boundary.

Environment	How to read support information
ACP lifecycle-managed environments	The platform installs and manages containerd. The exact runtime build follows the installed ACP patch and is not a stable minor-release documentation value. For immutable nodes, use the patch-specific OS Support Matrix.
HCP environments	Use About Hosted Control Plane for the current maturity, runtime, operating system, connectivity, and production-use boundaries.
Third-party clusters	Node OS, container runtime, CRI, and CPU architecture usually remain the responsibility of the external cluster owner, external distribution, or cloud provider.
Immutable Infrastructure	Use the Immutable Infrastructure documentation for Alauda OS images provided by ACP and provider-specific requirements.
Extensions	Use the exact Extension documentation and Customer Portal compatibility metadata. Package availability for x86, ARM, or hybrid architectures is not a universal guarantee for every Extension image or provider scenario.

Do not infer complete Docker, CRI-O, Alauda OS, mixed-architecture workload cluster, provider architecture, or universal Extension image support unless an explicit support matrix, Extension document, provider document, or release guidance states that support.

For node prerequisites, see Node Preprocessing. For package architecture, see Download. For Immutable Infrastructure, see About Immutable Infrastructure.

Extension Lifecycle

Operators and Cluster Plugins use the `Core`, `Aligned`, and `Agnostic` lifecycle model.

Lifecycle	Compatibility and upgrade meaning
`Core`	Follows the ACP Core release and cluster Distribution Version. Updated through cluster upgrade.

Lifecycle	Compatibility and upgrade meaning
`Aligned`	Follows the ACP release stream but can be upgraded independently when a compatible version is published.
`Agnostic`	Released independently from ACP and upgraded independently when a compatible version is published.

For a specific Operator or Cluster Plugin version, use the Alauda Customer Portal `ACP compatible versions` field as the compatibility authority. Release notes are useful for version changes. The Kubernetes support matrix is not the authority for one exact Extension version.

For details, see Core and Extensions.

Feature Maturity

When reading Overview, release notes, or component documentation, treat maturity labels as support boundaries:

Label	Meaning for readers
GA	The feature is generally available within the documented scope and prerequisites.
Technology Preview	The feature is available for evaluation within the documented limits and should not be treated as production-supported unless product documentation explicitly says so.
Deprecated	The feature remains available but is planned for removal or replacement. Plan migration according to release guidance.
Removed	The feature is no longer available in the documented release.

For current HCP maturity and evaluation boundaries, see About Hosted Control Plane. Global Cluster Disaster Recovery is generally available only within the documented `global` control-plane DR scope in Availability and Recovery.

Upgrade Reading Path

Before upgrading:

1. Read Kubernetes Support Matrix to confirm compatible workload-cluster versions.
2. Read Upgrade Overview to understand the CVO-based upgrade model.
3. Read Pre-Upgrade before upgrading the `global` cluster.
4. Check Extension compatibility in the Alauda Customer Portal and the Extension documentation.
5. Review Release Notes for version changes, behavior changes, technical preview status, deprecated features, and known issues.

Kubernetes Support Matrix

Use the Kubernetes support matrix to confirm Kubernetes version relationships for ACP releases and to distinguish cluster creation, upgrade compatibility, and third-party cluster onboarding boundaries.

For the conceptual lifecycle guidance, see Version and Lifecycle.

TOC

- How To Read This Page
- Version Support Matrix
- Product-Tested Scope for Third-Party Clusters
  - Product-Supported Kubernetes Range
  - Default Validation Scope
- Upgrade Requirements

How To Read This Page

Term	Meaning
Supported for cluster creation	The Kubernetes version that can be used when ACP creates a new platform-managed cluster for that release.

Term	Meaning
Compatible Versions	The Kubernetes versions that workload clusters can run before the `global` cluster is upgraded to that ACP release.
Third-party cluster product-tested scope	The product-supported Kubernetes onboarding range and the default capabilities covered by product validation on third-party clusters.

Version Support Matrix

The following table shows the Kubernetes version support for each ACP release.

INFO

This documentation is maintained for a ACP minor release line and does not publish a separate documentation set for each patch release. The table reflects the Kubernetes minors supported by the current documented state of that release line. If a later patch release adds another Kubernetes minor, the same row and its compatible-version range are updated. Use the target versions offered by the platform and release-specific artifacts for exact Kubernetes builds.

At any point in the release line, create clusters only with a Kubernetes target offered and validated by the installed ACP patch. The minor-level row can expand when a later patch adds another validated Kubernetes minor.

ACP Version	Supported for cluster creation	Compatible Versions
ACP 4.4	1.35	1.35, 1.34, 1.33, 1.32
ACP 4.3	1.34	1.34, 1.33, 1.32, 1.31
ACP 4.2	1.33	1.33, 1.32, 1.31, 1.30
ACP 4.1	1.32	1.32, 1.31, 1.30, 1.29
ACP 4.0	1.31, 1.30, 1.29, 1.28	1.31, 1.30, 1.29, 1.28

Product-Tested Scope for Third-Party Clusters

Product-Supported Kubernetes Range

Third-party Kubernetes clusters have completed product onboarding validation for Kubernetes 1.28 through 1.35, inclusive. Kubernetes minors outside this range are not declared product-supported for third-party cluster onboarding.

The current product-tested range applies to ACP 4.3 and later releases that provide third-party cluster onboarding. It is maintained as one shared range instead of being copied into each release row, and is updated when product validation covers additional Kubernetes minors.

The product validation program continues to test additional Kubernetes minor versions. A version that is planned for testing or still being evaluated is not product-supported. After a version completes product validation, the range on this page is updated. Use the currently published range as the authority for product support.

This range is separate from the Compatible Versions column. Compatible Versions determine whether workload clusters satisfy the prerequisite for upgrading the `global` cluster. The third-party onboarding range determines whether a third-party Kubernetes cluster can be onboarded.

Default Validation Scope

For third-party clusters within the published Kubernetes range, default product validation covers:

- Core Kubernetes workload and resource operations, such as creating and managing Deployments.
- Extend workflows used to install and upgrade Operators and Cluster Plugins.
- ClickHouse-based logging installed and upgraded through Extend.
- VictoriaMetrics-based monitoring installed and upgraded through Extend.

Extend is included in the default validation scope because the validated logging and monitoring capabilities depend on Extend for installation and upgrade. Customers can also

use Extend to install other capabilities when the exact Operator or Cluster Plugin declares compatibility with the target environment.

Default product validation does not mean that every specific Operator or Cluster Plugin is validated. For any other Operator or Cluster Plugin, check its product documentation and Alauda Customer Portal compatibility metadata, or contact technical support.

The published Kubernetes range and default validation scope do not replace the prerequisites of the selected onboarding method. Follow Third-Party Cluster Onboarding and the applicable Import, Register, or provider-specific procedure to verify connectivity, credentials, and environment-specific preparation.

For model boundaries, see Cluster Management Models. For upgrade planning, see Upgrade Overview and Pre-Upgrade.

Upgrade Requirements

For ACP 4.3 and later, workload clusters only need to remain within the documented compatible version range before upgrading the `global` cluster.

In ACP 4.2 and earlier, all workload clusters must be upgraded to the latest Kubernetes version in the compatible versions list before upgrading the `global` cluster.

For upgrade procedures, see Upgrade.

Release Notes

Review release notes with the Kubernetes Support Matrix, Version and Lifecycle, and feature-specific documentation to understand release changes and exact support boundaries.

TOC

- 4.4.0
  - Features and Enhancements
    - Kubernetes 1.35 and Upgrade Readiness
    - More Consistent Platform Installation and Upgrades
    - Log Collection Plugin Updates
    - Platform Image Sources
    - Application Image and Workload Operations
    - Monitoring, Dashboards, and Cost Management
    - Networking
    - Storage
    - Virtualization
    - Global Cluster Disaster Recovery Enhancements
    - Alauda OS Backup and Restore
    - Access and Authorization
    - Immutable Infrastructure
    - Upcoming Immutable Infrastructure Provider Releases
  - Deprecations and Removals
  - Fixed Issues

Known Issues

Related Product Sites

4.4.0

Issued: 2026-08-18

Features and Enhancements

Kubernetes 1.35 and Upgrade Readiness

ACP 4.4 upgrades the platform baseline to **Kubernetes 1.35**. Before upgrading, verify every production node meets the new Kubernetes requirements: the kernel must be version 5.8 or later, and the node must use cgroup v2. The upgrade preflight check blocks the upgrade until an administrator confirms that these checks are complete.

For the required checks and acknowledgement procedure, see Prepare for an Upgrade and Upgrade a Global Cluster.

More Consistent Platform Installation and Upgrades

ACP 4.4 improves installation and upgrade reliability. On a new `global` cluster, the upgrade-management component is installed automatically. Platform components are also installed in a fixed dependency order, reducing installation and upgrade failures when optional platform services are installed independently.

For upgrade procedures, see Upgrade.

Log Collection Plugin Updates

The following updates are delivered by independently versioned Log Collection plugin releases. The matching plugin release can be published after the ACP 4.4 platform release; install it when the package is available. Its product documentation and release notes will provide the detailed configuration and upgrade procedure.

- **Vector log collection** — The Log Collection plugin moves to **Vector** as the log collector. Vector provides a higher-performance collection path for large log volumes and a migration path from the previous collector.
- **OpenSearch log storage** — The matching plugin release replaces its embedded Elasticsearch service with a connection to a customer-provided OpenSearch service. New log data is written to OpenSearch. Existing Elasticsearch data is not automatically migrated and can be retained as read-only historical data. The plugin release documentation will describe supported historical-data handling and the upgrade procedure.

Platform Image Sources

For new production deployments, ACP 4.4 changes the recommended platform image-source strategy: use an external image registry, either your organization's existing registry or a separately deployed registry. Use it as the central image source for the platform, managed clusters, and plugins.

The platform built-in registry remains supported, but it is no longer the recommended default for new production environments.

For registry requirements, payload upload, installation, validation, upgrade preparation, and troubleshooting, see Prepare an External Platform Image Registry.

The next separately versioned component release in the ACP 4.4 delivery cycle will allow Alauda OS-based Workload Clusters to use a third-party registry independently of the Global Cluster registry. This lets you place the image source nearer to a Workload Cluster in geographically separated environments. The related product documentation and release notes will identify supported scenarios and configuration after the component release is available.

Application Image and Workload Operations

- **Operator-managed Registry v2 for application images** — ACP provides an Operator-managed integrated registry for application developers and workloads to push, pull, and manage application images. Administrators install it from OperatorHub and configure storage, namespace-scoped access, and managed ServiceAccount pull secrets. Workloads use the in-cluster Registry service; administrators can expose it to developer machines and CI when required. It supports scheduled image pruning with a configured retention policy; run registry garbage collection separately when you need to reclaim

unreferenced storage. The legacy Registry Cluster Plugin remains available for existing legacy deployments. For installation, configuration, access, and cleanup procedures, see Registry v2 administration.

- **Policy-based workload rebalancing** — Install the **Alauda Build of Descheduler** Cluster Plugin when workloads need to be rebalanced after changes in utilization, node configuration, affinity, taints, or topology. The plugin evicts eligible Pods according to the configured policies; for Pods managed by a controller, the default scheduler places the replacements after an eviction. It is not installed by default, does not schedule replacement Pods itself, and respects PodDisruptionBudgets. For installation, policy, and verification guidance, see Workload Rebalancing (Descheduler).
- **In-place Pod resource resizing** — ACP 4.4 provides guidance for using the Kubernetes Pod `resize` subresource to adjust CPU and memory requests and limits on a running Pod when the target cluster supports it. Use `kubectl` or an API client that supports the subresource. This is not a general web-console editing feature, and `resizePolicy` determines whether a container must restart; VPA `InPlaceOrRecreate` can still fall back to Pod recreation. For prerequisites, examples, and limitations, see Adjust Pod Resource Levels Without Pod Disruption.
- **PodDisruptionBudget operational guidance** — New operational and API guidance explains how to use `minAvailable` or `maxUnavailable` to protect replicated workloads during voluntary disruptions such as node drain, maintenance, and upgrades. PodDisruptionBudgets do not protect workloads from involuntary failures such as node hardware faults. For examples and API details, see Using PodDisruptionBudgets.

Monitoring, Dashboards, and Cost Management

- **Perses Monitoring Dashboards** — Create and manage metric dashboards with the Perses dashboard experience, import supported Perses or Grafana dashboard JSON, and migrate existing `MonitorDashboard` resources when needed. Existing Monitoring Dashboards remain available during the transition. For details, see Perses Monitoring Dashboards.
- **Fleet Monitoring** — Platform administrators can see connected clusters, monitoring-data freshness, resource capacity and utilization, and project quota allocation and usage in one multi-cluster view. Fleet Monitoring complements, rather than replaces, detailed monitoring and troubleshooting of an individual cluster. For details, see Fleet Monitoring.
- **VictoriaMetrics for Cost Management** — Cost Management and metering can use VictoriaMetrics as their metrics data source. Queries are scoped to the relevant cluster,

helping ensure that cost and resource-usage data is returned for the correct cluster in multi-cluster environments. For details, see Cost Management.

Networking

- **SR-IOV support** — ACP 4.4 supports passing SR-IOV network interfaces directly through to Pods, so network-intensive workloads such as carrier core-network applications can run on the platform.
- **OVN security group FQDN rules** — SecurityGroup rules support domain-name (FQDN) matching, so administrators can define egress policies with domain names instead of only static IP or CIDR targets.
- **OVN database SSL encryption** — Connections to the OVN database can be encrypted with SSL, including certificate-based access and certificate rotation, improving control-plane communication security.
- **AdminNetworkPolicy GA** — AdminNetworkPolicy adapts to the new API version and is now generally available, with web console management. For details, see Admin Network Policy.
- **IPVLAN CNI binary** — The Kube-OVN delivery now ships the `ipvlan` CNI binary alongside `macvlan`, enabling secondary-network-interface scenarios such as VIP egress over an underlay BGP fabric.
- **Gateway and Ingress observability** — New monitoring dashboards cover the Ingress Controller and Gateway, and MetalLB dashboards add BGP session state and IP address pool utilization. `ingress-nginx` adds OpenTelemetry support and authenticated access to its metrics endpoint.
- **Visual IP pool management** — Create and manage IP pools from the web console for fine-grained IP management in multi-VLAN clusters.

Storage

- **Unified storage component management** — The Alauda storage operator becomes the unified entry point for storage components, managing installation, upgrade, status, and dependencies of built-in Ceph, TopoLVM, local storage, and external Ceph.
- **Ceph encryption in transit and at rest** — Ceph supports in-transit encryption and KMS-backed at-rest encryption. For details, see In-Transit Encryption and Persistent Volume Encryption.

- **Ceph multi-NIC networking** — The Ceph container network supports multiple network interfaces, separating client traffic from replication traffic.
- **Object storage usability** — CephObjectStoreUser quota usage can be viewed, monitored, and alerted on. Bucket claim Secrets expose standard `accessKey` / `secretKey` fields that AWS SDKs and common tools can consume directly, and the console shows the Secret associated with each bucket claim.

Virtualization

- **CPU and memory hotplug** — Add CPU and memory to a running virtual machine without shutting it down. For procedures and guest OS support details, see CPU and Memory Hotplug.
- **Physical GPU passthrough** — Attach a physical GPU directly to a virtual machine for GPU-accelerated workloads.
- **Virtual machine migration (V2V)** — Integrates the forklift plugin to migrate virtual machines from traditional virtualization platforms to ACP.

Global Cluster Disaster Recovery Enhancements

Global Cluster Disaster Recovery remains supported for both traditional operating system deployments and Alauda OS-based Immutable Infrastructure deployments. In 4.4, the synchronization path is more resilient and covers more of the `global` cluster lifecycle. This capability protects the `global` control plane; it does not protect application data.

The standby cluster now preserves the active cluster's update order, including when recovery takes long enough for the active etcd to compact its history. The DR configuration can adapt when platform components change without replacing the DR service. The standby cluster also receives global-cluster upgrade state and workload-cluster lifecycle information, so it has the information required to take over more safely.

Before a DR switchover, or before a DR-aware `global` cluster upgrade removes the synchronizer, continue to validate that the active and standby clusters are consistent. This prevents stale standby data from causing incorrect recovery actions.

For details, see Global Cluster Disaster Recovery and Global Cluster Disaster Recovery on Immutable Infrastructure↗.

Alauda OS Backup and Restore

Clusters that use Alauda OS can now use **Cluster Enhancer** for etcd backup and restore. You can run scheduled or on-demand backups, retain them on the control plane nodes, and optionally keep an additional copy in S3-compatible object storage. This protects cluster configuration data; an etcd restore overwrites the existing etcd data and must be performed according to the documented recovery procedure.

For configuration and recovery steps, see etcd Backup and Restore.

Access and Authorization

- **Violet API-token publishing** — `violet`, the CLI used to package and publish plugins, now accepts a platform API token when publishing a package to the platform. Administrators and automation can publish without an interactive login. Download `violet` from the Alauda Customer Portal. For details, see Upload Packages.
- **In-platform CLI downloads** — Users can download ACP CLI packages from the web console for Linux (`amd64` and `arm64`), macOS (`amd64` and `arm64`), and Windows (`amd64`).
- **Safer role delegation** — The platform prevents users from granting permissions that exceed their own authorization boundary. The console also includes the related ClusterRole capabilities.

Immutable Infrastructure

ACP 4.4 continues to expand the documented Immutable Infrastructure path for Alauda OS, the supported operating system for immutable nodes in the documented provider scenarios. The following capabilities are available in their supported provider scenarios:

- **Bare Metal cluster lifecycle** — Create and manage both `global` and workload clusters on supported Bare Metal environments. The `global` cluster path includes Global Cluster Disaster Recovery.
- **Control plane and storage resilience** — Improve control-plane availability through provider placement rules or a self-built control-plane VIP where no load balancer is available, and preserve declared local disks when nodes are replaced during a rolling upgrade.

- **Alauda OS node storage on Huawei DCS** — Huawei DCS Provider `v1.0.16` or later supports dedicated `/var/lib/kubelet` and `/var/lib/containerd` disks. When disks are declared as persistent, the provider detaches and reattaches them during rolling node replacement, reducing Btrfs I/O pressure on the system disk and protecting declared node-local data.
- **Network and host configuration** — Configure multiple network interfaces for node traffic isolation where supported. On Huawei Cloud Stack, a node can retain both its short hostname and its FQDN.
- **Huawei DCS Provider compatibility for Alauda OS** — For Huawei DCS clusters that use the Alauda OS image for ACP `v4.3.2` or later, including ACP `v4.4.0` and later, install Huawei DCS Provider `v1.0.21` or later. Select the Alauda OS image from the OS Support Matrix row for the same ACP release. Do not pair DCS Provider `v1.0.21` or later with an Alauda OS image for an ACP release earlier than `v4.3.2`. See Alauda OS and Provider Compatibility for Huawei DCS↗.
- **Safer Huawei DCS virtual-machine operations** — Stop a virtual machine before deleting its node, and remove the boot CD-ROM attachment after startup so it does not prevent later VM migration.

For your provider and exact supported version, see About Immutable Infrastructure and its provider release notes.

Upcoming Immutable Infrastructure Provider Releases

The following updates are planned for separately versioned provider releases in the ACP 4.4 delivery cycle. Upgrading ACP to 4.4.0 alone does not install them; use the matching provider release when it becomes available. Provider product documentation and release notes will publish the detailed support boundaries and procedures.

- **VMware vSphere fixed-address node lifecycle** — The VMware vSphere Provider update improves MachineConfigPool health and bootstrap diagnostics, persistent and ephemeral disk handling, and rolling-update safeguards for fixed-address nodes. A virtual machine that an administrator intentionally powers off for maintenance can remain powered off.

Deprecations and Removals

- **System ALB removed** — The platform-level system ALB is removed in 4.4.

- **MinIO Operator removed** — The MinIO Operator is no longer shipped with the platform starting in 4.4. Use Ceph object storage for S3-compatible object storage.

Fixed Issues

- Fix metis component storage limit configuration is too small and causes metis container to restart after exceeding the limit

Known Issues

- Application creation failure triggered by the defaultMode field in YAML.
  Affected Path: Alauda Container Platform → Application Management → Application List → Create from YAML. Submitting YAML containing the defaultMode field (typically used for ConfigMap/Secret volume mount permissions) triggers validation errors and causes deployment failure.
  Workaround: Manually remove all defaultMode declarations before application creation.
- When pre-delete post-delete hook is set in helm chart.
  When the delete template application is executed and the chart is uninstalled, the hook execution fails for some reasons, thus the application cannot be deleted. It is necessary to investigate the cause and give priority to solving the problem of hook execution failure.

Related Product Sites

The following product sites provide component-specific documentation, compatibility information, upgrade guidance, and, where published, release notes. Their versions and release schedules are independent of ACP 4.4, so product documentation and release notes can be published after the ACP 4.4 platform release.

- Alauda Service Mesh v2 ↗
- Alauda Build of OpenTelemetry v2 ↗
- Alauda Distributed Tracing ↗
- Alauda AI ↗
- Alauda Hyperflux ↗

- Alauda Build of HAMi ↗
- Alauda NVIDIA GPU ↗
- Alauda Ascend NPU ↗
- Alauda Database Service for MySQL-PXC ↗
- Alauda Database Service for MySQL-MGR ↗
- Alauda Cache Service for Redis OSS ↗
- Alauda Streaming Service for Kafka ↗
- Alauda Build of Keycloak ↗
- Alauda Streaming Service for RabbitMQ ↗
- Alauda support for PostgreSQL ↗
- Alauda support for CloudNativePG ↗
- Alauda Cost Management ↗
- Alauda DevOps ↗
- Alauda Container Security ↗
- Immutable Infrastructure ↗
- Compliance Service ↗
- Alauda Container Platform GitOps ↗
- Hosted Control Plane ↗
- Logging Service ↗
- Cluster Authentication ↗

Glossary

Use this glossary to interpret platform-wide terms in Alauda Container Platform. For workflow-specific terms, use the corresponding workflow or component documentation.

TOC

Platform And Cluster Terms

Extension And Packaging Terms

Identity, Access, And Security Terms

Networking And Access Terms

Disaster Recovery, Backup, And Upgrade Terms

Terminology Conventions

Platform And Cluster Terms

Term	Definition	Related doc
ACP Core	The platform management foundation installed first. It deploys the `global` cluster and provides core management-plane capabilities such as web console access, platform APIs, users and RBAC, cluster management, project governance, and Extension management frameworks.	Core and Extensions

Term	Definition	Related doc
`global` cluster	The central management cluster deployed by ACP Core. It hosts platform management services and coordinates cluster, project, user, Extension, and platform operations.	Platform Model
Workload cluster	A Kubernetes cluster that runs application workloads under governance from the `global` cluster. Depending on the selected model, ACP can create and lifecycle-manage workload clusters.	Platform Model
Third-party cluster	A Kubernetes environment whose Kubernetes distribution and lifecycle are provided or managed outside Alauda. ACP can onboard third-party clusters for centralized governance and operations within documented prerequisites and caveats.	Cluster Management Models
Managed Cluster	The current UI/navigation label for the interface that manages onboarded third-party clusters. It is not a conceptual model parallel to `third-party cluster`.	Managed Clusters Overview
Installer-Provisioned Infrastructure (IPI)	An infrastructure responsibility model where the platform provisions machines, manages node operating systems through Immutable OS, and manages the supported Kubernetes lifecycle.	Cluster Management Models
User-Provisioned Infrastructure (UPI)	An infrastructure responsibility model where users prepare physical or virtual machines and retain node OS responsibility, while the platform installs and manages Kubernetes on those nodes.	Cluster Management Models
Hosted Control Plane (HCP)	A control-plane topology where each hosted cluster has its own control plane, and multiple hosted control planes run as workloads on a management cluster. In ACP, HCP is implemented through Kamaji (`TenantControlPlane`).	About Hosted Control Plane
Immutable Infrastructure	A provisioning and operating model where node configurations are baked into images and changes are applied by replacing nodes with new images.	About Immutable Infrastructure

Term	Definition	Related doc
Immutable OS	An immutable operating system used by installer-provisioned nodes. Node state is kept consistent by treating the operating system layer as centrally managed and replaced through image-based updates.	About Immutable Infrastructure
Project	A platform governance unit for a tenant, team, or business system. A project can span multiple associated clusters and acts as a boundary for quotas, policies, users, and namespace ownership.	Project Introduction
Namespace	A Kubernetes namespace managed directly or indirectly by the platform. In ACP, a namespace can belong to a project and inherit project-level governance.	Namespace Management
Control plane	The Kubernetes management layer that runs components such as the API server, scheduler, controller manager, and etcd.	Architecture
Control plane node	A node that runs Kubernetes control plane components for a cluster using a dedicated control-plane topology.	Node Management
Worker node	A node that runs application workloads and cluster-local supporting components.	Node Management

Extension And Packaging Terms

Term	Definition	Related doc
Extension	The umbrella term for Operator and Cluster Plugin based mechanisms that add capabilities to ACP environments.	Core and Extensions

Term	Definition	Related doc
Operator	An Extension mechanism built on Kubernetes custom resources and controllers. In ACP, Operators are managed through Operator Lifecycle Manager and OperatorHub.	Operator
Operator Lifecycle Manager (OLM)	The operator management framework that handles Operator installation, upgrades, dependency resolution, and related resources such as `CatalogSource`, `Subscription`, `InstallPlan`, and `ClusterServiceVersion`.	Operator
OperatorHub	The platform interface for discovering, installing, upgrading, and managing Operators through OLM.	Operator
Cluster Plugin	The platform Extension mechanism for chart-based plugins managed through `ModulePlugin`, `ModuleConfig`, and `ModuleInfo` custom resources.	Cluster Plugin
`Core` lifecycle	Extension lifecycle type that follows the ACP Core release and cluster Distribution Version. Standalone upgrade is not supported.	Core and Extensions
`Aligned` lifecycle	Extension lifecycle type that follows the ACP release stream but can be upgraded independently when a compatible version is published.	Core and Extensions
`Agnostic` lifecycle	Extension lifecycle type released independently from ACP and upgraded independently when a compatible version is published.	Core and Extensions
Customer Portal compatible versions	The compatibility field in the Alauda Customer Portal that indicates which ACP versions a specific Operator or Cluster Plugin version supports.	Core and Extensions

Identity, Access, And Security Terms

Term	Definition	Related doc
Identity Provider (IdP)	An external identity system used for platform account authentication, such as LDAP or OIDC.	Identity Provider Introduction
LDAP	A directory access protocol supported by the platform for enterprise user authentication. Active Directory can be integrated through LDAP.	LDAP Management
OpenID Connect (OIDC)	An identity layer based on OAuth 2.0 that the platform supports for third-party user authentication.	OIDC Management
Platform Roles	System-provided role templates used by the platform after the RBAC refactor. They are bound to users or groups and convert to Kubernetes permissions.	Roles Introduction
Kubernetes Roles	Native Kubernetes `Role` and `ClusterRole` objects used for fine-grained permissions. They are bound with `RoleBinding` or `ClusterRoleBinding`.	Roles Introduction
Audit	Platform operation records for users and system actions. Audit views depend on logging service components.	Audit Introduction
NetworkPolicy	A Kubernetes resource for namespace-scoped network traffic policy. Enforcement depends on the selected CNI.	NetworkPolicy
API Refiner	API filtering capability that can filter or mask Kubernetes API responses based on user permissions, project, cluster, and namespace.	API Security
Compliance features	Kyverno-based features for policy enforcement, violation monitoring, and reporting. Use the Compliance Service documentation for scope and prerequisites.	Compliance Service

Networking And Access Terms

Term	Definition	Related doc
Platform Access Address	The external address used to access platform services such as the web console and platform APIs. It can be the same as the Cluster Endpoint or a separate address for external access scenarios.	Installing
Cluster Endpoint	The address used by cluster components and administrators to reach the target Kubernetes control plane endpoint.	Installing
LoadBalancer	A Kubernetes Service type that exposes a Service through an external load balancer. In the platform, load-balancing can also involve external devices, VIPs, or provider-specific services.	Configure Services
Self-built VIP	The built-in virtual IP option used when an external load balancer is not provided for the Cluster Endpoint.	Prerequisites
Cluster Proxy	A platform communication capability used to connect the `global` cluster and managed clusters for supported management operations.	Architecture
Import cluster	An onboarding method where the `global` cluster connects to the target third-party cluster API server with supplied address, CA, and credentials.	Import Clusters
Register cluster	An onboarding method where a reverse proxy service in the target third-party cluster initiates registration and establishes a tunnel to the platform.	Register Cluster

Disaster Recovery, Backup, And Upgrade Terms

Term	Definition	Related doc
Global Cluster Disaster Recovery	A Primary and Standby `global` cluster DR mechanism based on `global` cluster etcd synchronization and DNS/VIP failover. It	Availability and Recovery

Term	Definition	Related doc
	protects the `global` control-plane scenario only within documented boundaries.	
etcd backup and restore	Backup and restore path for Kubernetes control-plane resource state stored in etcd.	etcd Backup and Restore
Application backup and restore	Velero-based backup and restore path for application resources and persistent volumes, delivered through Data Backup components.	Backup Overview
Cluster Version Operator (CVO)	The upgrade workflow and controller used to coordinate target versions, preflight checks, status, and execution progress for `global` and workload cluster upgrades.	Upgrade Overview
ClusterVersionShadow	A custom resource used by the CVO-based upgrade workflow to expose desired version, status, preflight results, stages, and upgrade history.	Upgrade Overview
Distribution Version	The cluster distribution version used to coordinate upgrade gating and version alignment for `global` and workload cluster upgrades.	Upgrade Overview
Preflight	The set of checks run before upgrade execution to reduce upgrade risk.	Pre-Upgrade

Terminology Conventions

- The `global` cluster is the platform management cluster.
- `Third-party cluster` refers to externally provided Kubernetes environments. `Managed Cluster` is the current UI or navigation label.
- UI labels, API resource names, protocol names, and feature names keep their official capitalization.
- Acronyms are expanded on first mention when the surrounding page needs the expansion.

Learn More

Choose the next task path for installation, cluster operations, application delivery, upgrade planning, availability and recovery, security, API, or CLI work.

TOC

Understand The Platform
Install And Plan
Manage Clusters
Build And Operate Applications
Install And Use Extensions
Plan Versions And Upgrades
Plan Availability, Backup, And Recovery
Secure And Integrate

Understand The Platform

Goal	Start here
Understand what ACP is and how it relates to Kubernetes.	Introduction
Understand the hub-and-spoke architecture.	Architecture

Goal	Start here
Understand the `global` cluster, workload clusters, third-party clusters, HCP, projects, and namespaces.	Platform Model
Understand the three-axis cluster model and responsibility boundaries.	Cluster Management Models
Find platform-wide terms.	Glossary

Install And Plan

Goal	Start here
Understand what ACP Core installation deploys.	Install Overview
Choose Multi-Cluster, Single Cluster, or Single Node.	Plan
Prepare resource, network, OS, and load-balancing prerequisites.	Prerequisites
Download the Core Package and choose package architecture.	Download
Install ACP Core.	Installing
Validate a completed installation.	Validation
Decide whether to plan Global Cluster Disaster Recovery.	Global Cluster Disaster Recovery

Manage Clusters

Goal	Start here
Create ACP lifecycle-managed workload clusters.	Clusters Overview

Goal	Start here
Create on-premises workload clusters.	Creating an On-Premise Cluster
Use Immutable Infrastructure.	About Immutable Infrastructure
Evaluate Hosted Control Plane (HCP).	About Hosted Control Plane
Onboard existing third-party Kubernetes environments by import.	Import Third-Party Clusters
Onboard existing third-party Kubernetes environments by register.	Register Cluster
Access a cluster with KubeConfig.	Access a Cluster with KubeConfig

Build And Operate Applications

Goal	Start here
Understand developer workflows.	Developer Overview
Create applications from images, charts, YAML, source code, or Operator-backed catalogs.	Create Applications
Manage Kubernetes workloads.	Application Workloads
Manage namespaces.	Namespace Management
View application logs, events, and monitoring dashboards.	Application Observability
Work with container images and registries.	Images and Registry

Install And Use Extensions

Goal	Start here
Understand Core and Extension boundaries.	Core and Extensions
Understand the Extension system.	Extend Overview
Install and manage Operators.	Operator
Install and manage Cluster Plugins.	Cluster Plugin
Upload Extension packages.	Upload Packages
Check version compatibility for a specific Extension.	Use the Alauda Customer Portal `ACP compatible versions` field, then read the Extension documentation.

Plan Versions And Upgrades

Goal	Start here
Read Kubernetes version support and the third-party onboarding range.	Kubernetes Support Matrix
Understand version, lifecycle, runtime, OS, and architecture boundaries.	Version and Lifecycle
Read release changes.	Release Notes
Plan cluster upgrades.	Upgrade Overview
Prepare before upgrading the `global` cluster.	Pre-Upgrade
Upgrade workload clusters.	Upgrade Workload Cluster

Plan Availability, Backup, And Recovery

Goal	Start here
Understand topology, HA, Global DR, and recovery boundaries.	Availability and Recovery
Size the `global` cluster.	Evaluating Resources for Global Cluster
Size workload cluster control planes.	Evaluating Resources for Workload Cluster
Plan and operate Global Cluster Disaster Recovery.	Global Cluster Disaster Recovery
Back up and restore etcd, registry, monitoring, logging, or applications.	Backup Overview

Secure And Integrate

Goal	Start here
Configure identity providers with LDAP or OIDC.	Identity Provider Introduction
Manage users, roles, and Kubernetes RBAC.	Roles Introduction
Manage projects and namespaces for tenant isolation.	Project Introduction
View audit records.	Audit Introduction
Configure NetworkPolicy.	NetworkPolicy
Use API Refiner for API response filtering.	API Security
Use compliance features.	Compliance Service
Use container security capabilities.	Alauda Container Security

Goal	Start here
Use platform APIs.	API Introduction
Use CLI tools.	CLI Tools