Menu

Overview

Architecture

Release Notes

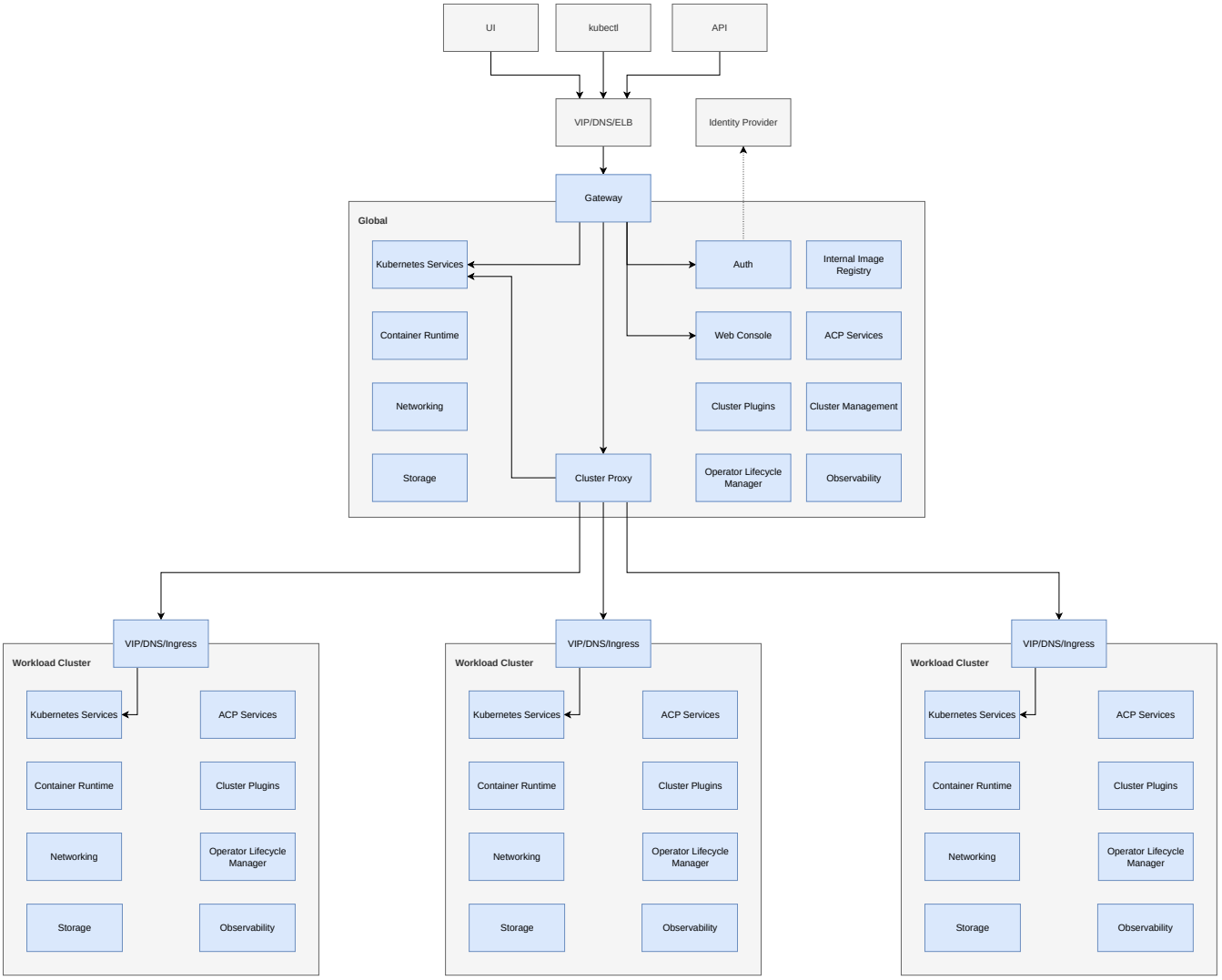
Architecture

TOC

Introduction to Alauda Container Platform

The Alauda Container Platform (ACP) provides an enterprise-grade Kubernetes-based platform that enables organizations to build, deploy, and manage applications consistently across hybrid and multi-cloud environments. ACP integrates core Kubernetes capabilities with enhanced management, observability, and security services, offering a unified control plane and flexible workload clusters.

The architecture follows a **hub-and-spoke** model, consisting of a `global` cluster and multiple workload clusters. This design provides centralized governance while allowing independent workload execution and scalability.



Core Architectural Components

Global Cluster

The `global` cluster serves as the centralized management and control hub of ACP. It provides platform-wide services such as authentication, policy management, cluster lifecycle operations, and observability. It's also a central hub for multi-cluster management and provides cross-cluster functionality.

Key components include:

- **Gateway** Acts as the main entry point to the platform. It manages API requests from the UI, CLI (kubectl), and automation tools, routing them to appropriate backend services.
- **Authentication and Authorization (Auth)** Integrates with external Identity Providers (**IdPs**) to provide Single Sign-On (SSO) and RBAC-based access control.

- **Web Console** Provides a web-based interface for ACP. It interfaces with platform APIs through the gateway.
- **Cluster Management** Handles the registration, provisioning, and lifecycle management of workload clusters.
- **ACP Services**
- **Operator Lifecycle Manager (OLM) and Cluster Plugins** Manages the installation, updates, and lifecycle of operators and cluster extensions.
- **Internal Image Registry** Offers an out-of-box integrated container image repository with role-based access.
- **Observability** Provides centralized logging, metrics, and tracing for both the `global` and workload clusters.
- **Cluster Proxy** Enables secure communication between the `global` cluster and workload clusters.

Workload Cluster

Workload clusters are Kubernetes-based environments managed by the `global` cluster. Each workload cluster runs isolated application workloads and inherits governance and configuration from the central control plane.

External Integrations

- **Identity Provider (IdP)** Supports federated authentication via standard protocols (OIDC, SAML) for unified user management.
- **API and CLI Access** Users can interact with ACP through RESTful APIs, the web console, or command-line tools like `kubectl` and `ac`.
- **Load Balancer (VIP/DNS/SLB)** Provides high availability and traffic distribution to the Gateway and ingress endpoints of the `global` and workload Clusters.

Scalability and High Availability

ACP is designed for horizontal scalability and high availability:

- Each component can be deployed redundantly to eliminate single points of failure.
- The `global` cluster supports managing dozens to hundreds of workload clusters.
- Workload clusters can scale independently according to workload demand.
- The use of VIP/DNS/Ingress ensures seamless routing and failover.

Functional Perspective

Alauda Container Platform (ACP)'s complete functionality consists of **ACP Core** and extensions based on two technical stacks: **Operator** and **Cluster Plugin**.

- **ACP Core**

  The minimal deliverable unit of ACP, providing core capabilities such as cluster management, container orchestration, projects, and user administration.

  - Meets the highest security standards
  - Delivers maximum stability
  - Offers the longest support lifecycle

- **Extensions**

  Extensions in both the Operator and Cluster Plugin stacks can be classified into:

  - **Aligned** – Life cycle strategy consisting of multiple maintenance streams, with alignment to ACP.
  - **Agnostic** – Life cycle strategy consisting of multiple maintenance streams, released independently from ACP.

  For more details about extensions, see Extend.

Technical Perspective

Platform Component Runtime All platform components run as containers within a Kubernetes management cluster (the `global` cluster).

High Availability Architecture

- The `global` cluster typically consists of at least three control plane nodes and multiple worker nodes
- High availability of etcd is central to cluster HA; see *Key Component High Availability Mechanisms* for details
- Load balancing can be provided by an external load balancer or a self-built VIP inside the cluster

Request Routing

- Client requests first pass through the load balancer or self-built VIP
- Requests are forwarded to **ALB** (the platform's default Kubernetes Ingress Gateway) running on designated ingress nodes (or control-plane nodes if configured)
- ALB routes traffic to the target component pods according to configured rules

Replica Strategy

- Core components run with at least two replicas
- Key components (such as registry, MinIO, ALB) run with three replicas

Fault Tolerance & Self-healing

- Achieved through cooperation between kubelet, kube-controller-manager, kube-scheduler, kube-proxy, ALB, and other components
- Includes health checks, failover, and traffic redirection

Data Storage & Recovery

- Control-plane configuration and platform state are stored in etcd as Kubernetes resources
- In catastrophic failures, recovery can be performed from etcd snapshots

Primary / Standby Disaster Recovery

- Two separate `global` clusters: **Primary Cluster** and **Standby Cluster**
- The disaster recovery mechanism is based on real-time synchronization of etcd data from the Primary Cluster to the Standby Cluster.

- If the Primary Cluster becomes unavailable due to a failure, services can quickly switch to the Standby Cluster.

Key Component High Availability Mechanisms

etcd

- Deployed on three (or five) control plane nodes
- Uses the RAFT protocol for leader election and data replication
- Three-node deployments tolerate up to one node failure; five-node deployments tolerate up to two
- Supports local and remote S3 snapshot backups

Monitoring Components

- **Prometheus**: Multiple instances, deduplication with Thanos Query, and cross-region redundancy
- **VictoriaMetrics**: Cluster mode with distributed VMStorage, VMInsert, and VMSelect components

Logging Components

- **Nevermore** collects logs and audit data
- **Kafka / Elasticsearch / Razor / Lanaya** are deployed in distributed and multi-replica modes

Networking Components (CNI)

- **Kube-OVN / Calico / Flannel**: Achieve HA via stateless DaemonSets or triple-replica control plane components

ALB

- Operator deployed with three replicas, leader election enabled
- Instance-level health checks and load balancing

Self-built VIP

- High-availability virtual IP based on Keepalived
- Supports heartbeat detection and active-standby failover

Harbor

- ALB-based load balancing
- PostgreSQL with Patroni HA
- Redis Sentinel mode
- Stateless services deployed in multiple replicas

Registry and MinIO

- Registry deployed with three replicas
- MinIO in distributed mode with erasure coding, data redundancy, and automatic recovery

Release Notes

TOC

4.1.2

Fixed Issues

- After a global cluster upgrade, monitoring dashboards for all Applications and all kinds of Workloads in non-upgraded worker clusters will not display any data.

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.

4.1.1

Fixed Issues

- Fixed an issue where running `violet push` before upgrading caused functional components to become abnormal, blocking the upgrade. The command has been enhanced to separate image push and CR creation, allowing users to push only images without creating CRs.

Known Issues

- After a global cluster upgrade, monitoring dashboards for all Applications and all kinds of Workloads in non-upgraded worker clusters will not display any data.
- 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.

4.1.0

Features and Enhancements

Immutable Infrastructure

Released:

- **Alauda Container Platform DCS Infrastructure Provider**
- **Alauda Container Platform Kubeadm Provider**

Both plugins have an *Agnostic* lifecycle and release asynchronously with Alauda Container Platform (ACP).

- **DCS Infrastructure Provider** implements the Cluster API Infrastructure Provider interface, integrating with Huawei Datacenter Virtualization Solution (DCS).
- **Kubeadm Provider** installs and configures the Kubernetes control plane and nodes on VMs provisioned by the infrastructure provider.

Together, these plugins enable fully automated cluster management on DCS.

Documentation is under preparation and will be published in the online documentation upon release.

Machine Configuration

Released: **Alauda Container Platform Machine Configuration** Lifecycle: *Agnostic*, releases asynchronously with ACP.

Machine Configuration manages file updates, systemd units, and SSH public keys across cluster nodes, providing:

- `MachineConfig` CRD for writing configurations to hosts.
- `MachineConfigPool` CRD for grouping and managing node configurations based on role labels.
- Upon cluster installation, two default MachineConfigPools are automatically created — one for control plane nodes and one for worker nodes. Additionally, users can create custom MachineConfigPools as needed.

The system continuously monitors configuration drift, marking affected nodes as *Degraded* until resolved.

For detailed feature information, see Machine Configuration.

etcd Encryption

Released: **Alauda Container Platform etcd Encryption Manager** Lifecycle: *Agnostic*, releases asynchronously with ACP.

Provides periodic rotation of etcd data encryption keys on workload clusters using AES-GCM for secrets and configmaps. Supports seamless re-encryption and key reload without workload disruption, maintaining backward compatibility with the last 8 keys.

See etcd Encryption for details.

Kubernetes Certificates Rotator

Released: **Alauda Container Platform Kubernetes Certificates Rotator** Lifecycle: *Agnostic*, releases asynchronously with ACP.

Enables automated rotation of certificates used by Kubernetes components.

See Automated Kubernetes Certificate Rotation for details.

Cluster Enhancement

Released: **Alauda Container Platform Cluster Enhancer** Lifecycle: *Aligned*.

New features and changes:

- **etcd Backup**: Migrated etcd backup functionality from Backup & Recovery to Cluster Enhancer due to differences in usage and implementation. Optimized the deployment method to avoid conflicts during configuration changes and upgrades.
- **Event Cleanup**: Implements active cleanup of expired Kubernetes events externally to prevent accumulation in etcd, reducing etcd load and instability risks during restarts.
- **Certificate Monitoring**: Converts certificate management into certificate monitoring with alert rules and dashboards, replacing the previous Certificates management functionality. Implements a more efficient monitoring approach while monitoring loopback certificates used by kube-apiserver.
- **Cluster Monitoring Dashboard Migration**: Migrates cluster monitoring resources from `chart-cpaas-monitor` to Cluster Enhancer.
- **Cluster Details Chart Migration**: Switches monitoring charts in cluster details to custom monitoring dashboards.

Chinese Language Pack

Chinese language support has been decoupled from the platform and released as the **Chinese Language Pack** plugin. The platform defaults to English upon installation; users can install this plugin if Chinese language support is needed.

Create On-Premise Cluster

From ACP 4.1 onward, creating on-premise clusters supports only the latest Kubernetes version provided by the platform, replacing the previous option to choose among four Kubernetes versions.

Logging

- Upgraded ClickHouse to version v25.3.
- Added `POD IP` tags to application logs, allowing filtering by `POD IP`.
- Improved standard output log collection: The timestamp field now reflects the actual print time of the log, instead of the collection component's time, ensuring logs are displayed in the correct order.

Monitoring

- Upgraded Prometheus to version v3.4.2.
- Custom variables now support three types: Constant, Custom, and Textbox.
  - **Constant:** A fixed value that does not change.
  - **Custom:** A value selected from a predefined list.
  - **Textbox:** A value entered manually by the user.
- Stat Chart now supports Graph mode, which displays a trend curve for the selected period below the statistic.
- Value Mapping now supports regular expressions and special values.
- Panels can now be copied, allowing you to duplicate a panel within the current dashboard.

Tenant Management

- Project quotas now support custom resource quotas and storage class quotas.
- The plugin provides new metrics: `cpaas_project_resourcequota` and `cpaas_project_resourcequota_aggregated`, which can be used to display project quotas in

dashboards.

- `cpaas_project_resourcequota` : Available in every cluster.
- `cpaas_project_resourcequota_aggregated` : Available in the `global` cluster and aggregates data from all clusters.

- Custom Roles now have additional restrictions, allowing assignment of permissions only within the corresponding role type:
  - **Platform Role:** Can assign all permissions.
  - **Project Role:** Can assign only permissions within the scope of the platform's preset `project-admin-system` role.
  - **Namespace Role:** Can assign only permissions within the scope of the platform's preset `namespace-admin-system` role.
  - Permissions that the current user does not possess cannot be assigned.

Automated UID/GID Allocation Solution for Secure Pod Execution

In Kubernetes, you can configure a dedicated User ID (UID) and Group ID (GID) range for each namespace. When users deploy Pods within such a namespace, we automatically set the RunAsUser and fsGroup for all containers within the Pod, based on the namespace's predefined security policies. These users and groups will be dynamically allocated from the UID/GID range authorized for that specific namespace.

Key Capabilities and Value:

- **Enhanced Security**: By enforcing containers to run as non-privileged users and restricting their UID/GID ranges, this solution effectively mitigates security risks like container escapes and privilege escalation, adhering to the principle of least privilege.

- **Simplified Management**: Developers no longer need to manually specify UID/GID in each container or Pod configuration. Once a namespace is configured, all Pods deployed within it automatically inherit and apply the correct security settings.

- **Ensured Compliance**: This helps customers better meet internal security policies and external compliance requirements, ensuring containerized applications run in a regulated environment.

Usage:

- Add the label `security.cpaas.io/enabled` to your Namespace.

Productized Solution Based on Argo Rollouts

Our productized solution, built on open-source Argo Rollouts, empowers users with fine-grained control over their release processes. By implementing progressive and controlled deployment strategies, it minimizes business interruptions or failures that can arise from launching new features or versions, significantly reducing release risks.

Key Capabilities and Value:

- **Blue-Green Deployment**: Achieve zero-downtime updates by deploying new versions alongside your existing production environment. After thorough testing, traffic can be instantly or rapidly switched from the old version to the new one.

- **Canary Deployment**: Gradually introduce new versions by directing a small percentage (e.g., 5%) of production traffic to them, allowing you to observe their performance and stability. Based on predefined metrics (such as error rates or latency), the system can automatically increase traffic or roll back if issues are detected, limiting the impact of potential problems.

- **Platform-Certified Argo Rollout Chart**: You can download the community's open-source Argo Rollouts directly, or opt for the platform-certified version available through Alauda Cloud.

Alauda Container Platform Registry: Deep Integration with Platform User Permissions

To provide a more secure and convenient image management experience, we've deepened the integration of our lightweight image registry with the platform's existing user permission system.

Key Capabilities and Value:

- **Deep Integration with Platform User System**: The image registry is seamlessly integrated with the platform's user authentication and Role-Based Access Control (RBAC) mechanisms. Developers, testers, and administrators can directly use their existing platform credentials, eliminating the need for additional configuration or separate account management. The platform automatically maps user permissions within a Namespace to

corresponding access rights for images in the registry. For example, users can only push and pull images in "specific Namespaces" they have access to.

- **Smoother Command-Line Operations**: Supports image `pull` and `push` operations via CLI tools, significantly improving operational efficiency and convenience.

Warning:

- Only supports installing Alauda Container Platform Registry via the solution.

KEDA-Based Auto-Scaling Solution

To enable applications to intelligently respond to actual load, our platform offers an auto-scaling solution built on KEDA (Kubernetes Event-driven Autoscaling).

Key Capabilities and Value:

- **Event-Driven Elastic Scaling**: KEDA supports over 70 types of scalers to automatically scale applications (such as Deployments, Jobs, etc.). Beyond traditional CPU and memory utilization, it can monitor metrics like message queue length (e.g., Kafka, RabbitMQ), database connection counts, HTTP request rates, and custom metrics.

- **Platform-Certified KEDA Operator**: Download and install the platform-certified version via Alauda Cloud.

Solutions:

- The product provides two solutions: auto-scaling based on Prometheus metrics and scaling down to zero.

Cross-Cluster Application Disaster Recovery Solution (Alpha)

Our platform now offers a new GitOps-based Cross-Cluster Application Disaster Recovery (DR) solution, designed to significantly enhance application resilience and availability.

Key Capabilities and Value:

- **Diverse DR Models**: Flexibly supports Active-Active (AA-DR) for global, high-concurrency demands; Active-Standby Dual-Active (AS-DR) to optimize resource utilization; and Active-Passive (AP-DR) to strictly ensure data consistency.

- **Automated GitOps Sync**: Leverages the power of GitOps, combined with ApplicationSet and Kustomize, to automate cross-cluster configuration synchronization, ensuring the DR environment is always in a ready state.
- **Flexible Traffic Management**: Utilizes third-party provided DNS and GSLB functionalities to achieve intelligent, health-check-driven traffic redirection and rapid failover, minimizing service disruption.
- **Multi-Dimensional Data Synchronization**: The solution provides guidance on various methods, including database-level, storage-level, and application-level synchronization, to ensure eventual data consistency between clusters, laying the foundation for business continuity.
- **Streamlined Failover Process**: Clearly defines detailed steps for failure detection, traffic redirection, state promotion, and service recovery, ensuring efficient and orderly failover during a disaster.

Note:

- The data synchronization aspect of the disaster recovery solution is closely tied to the customer's business characteristics and data volume, and thus can vary significantly. Therefore, actual implementation requires specific handling tailored to the customer's particular scenario.

Comprehensive Upgrade of Dependency Components for Enhanced Stability and Security

This release includes upgrades to the following core components:

- KubeVirt upgraded to v1.5.2
- Ceph upgraded to 18.2.7
- MinIO upgraded to RELEASE.2025-06-13T11-33-47Z

Other open-source dependencies have also been synchronized to their latest community versions, addressing numerous known issues and security vulnerabilities to ensure improved system stability and reliability.

Enhanced Virtualization Features for Improved Business Continuity and Security

Based on practical application requirements in virtualization environments, this update introduces several key enhancements:

- **High Availability Migration**: Automatically migrates virtual machines to healthy nodes during node failures, ensuring uninterrupted business continuity.
- **Virtual Machine Cloning**: Quickly create new virtual machines from existing ones, significantly improving deployment efficiency.
- **Virtual Machine Templates**: Supports converting existing virtual machines into templates for rapid, batch deployment of similarly configured environments.
- **Trusted Computing (vTPM)**: Virtual machines now support trusted computing features, enhancing overall security.

Detailed instructions and guidelines for these new features have been updated in the user manual.

Object Storage Service Based on COSI v2 Provides More Flexible and Efficient Storage Management

The Container Object Storage Interface (COSI) has been upgraded to version v2 (alpha), bringing enhancements such as:

- **Multi-Cluster Access**: Supports simultaneous access to multiple different Ceph or MinIO storage clusters, enabling more efficient centralized management.
- **Fine-Grained Quota Management**: Allows flexible quota settings for different storage categories, optimizing resource utilization.
- **Enhanced Permission Management**: Supports the creation of various user access permissions, including read-write, read-only, and write-only modes.
- **Anonymous Access Support**: The Ceph COSI Driver now supports anonymous access, enabling quick external HTTP program access through Ingress configuration.

ALB Enters Maintenance Mode

> **WARNING**
>
> ALB will stop new feature development and only receive maintenance and security fixes. Version 4.1 supports ingress-nginx, and version 4.2 supports Envoy Gateway.

Future Plan:

- For ingress users, directly use ingress-nginx
- Future new features will only be supported on GatewayAPI
- Avoid mentioning ALB unless there are strong requirements for ALB-exclusive capabilities (e.g., project port allocation)

Currently unsupported ALB-exclusive features in GatewayAPI:

- Port-based gateway instance allocation
- Traffic forwarding based on IP and IP ranges
- EWMA algorithm for load balancing
- WAF usage
- Rule-level monitoring views

Using ingress-nginx to provide Ingress capabilities

Introduce the community's most mainstream Ingress controller implementation to replace existing ALB-based Ingress scenarios.

Key Capabilities and Value:

- Compatibility with mainstream community practices to avoid communication ambiguities
- Ingress UI supports custom annotations for leveraging ingress-nginx's rich extension capabilities
- Security issue fixes

Kube-OVN Supports New High-Availability Multi-Active Egress Gateway

A new Egress mechanism addresses limitations of previous centralized gateways. The new Egress Gateway features:

- Active-Active high availability via ECMP for horizontal throughput scaling
- Sub-1s failover via BFD
- Reuse of underlay mode, with Egress Gateway IPs decoupled from Nodes
- Fine-grained routing control through Namespace selectors and Pod selectors
- Flexible Egress Gateway scheduling via Node selectors

Support for AdminNetworkPolicy-type cluster network policies

Kube-OVN Supports Community's New Cluster Network Policy API. This API allows cluster administrators to enforce network policies without configuring them in each Namespace.

Advantages over previous cluster network policies:

- Community-standard API (replacing internal APIs)
- No conflicts with NetworkPolicy (higher priority than NetworkPolicy)
- Supports priority settings

For more information: Red Hat Blog on AdminNetworkPolicy ↗

Deprecated and Removed Features

Docker Runtime Removal

- Previously, the platform provided Docker runtime images even though it was not the default runtime for new clusters. Starting with ACP 4.1, Docker runtime images will no longer be provided by default.

Template Application Removal

- The **Application → Template Application** entry point has been officially removed. Please ensure all Template Applications are upgraded to "Helm Chart Application" before upgrading.