

Canonical Kubernetes Explorer

Training Curriculum

Version: 3.1
Prepared by: Andrés Mujica – Cloudbase Solutions
Prepared on: 27 April 2026

Table of Contents

Introduction & Scope.....	3
Agenda.....	3
Day 1.....	5
1. Snaps and Snapcraft.....	5
2. Juju – Application Modelling.....	5
3. Container Fundamentals.....	5
4. Kubernetes Overview and Architecture.....	5
5. Canonical Kubernetes Deployment.....	5
Day 2.....	6
6. Pods and Workloads.....	6
7. Networking, Services and Ingress.....	6
8. Keeping Applications Healthy.....	6
Day 3.....	7
9. Storage and Configuration Data for Applications.....	7
10. Resources and Auto Scaling (HPA).....	7
11. Authentication, Authorization and RBAC.....	7
12. Pod and Network Security.....	7
13. Private Registry.....	8
14. Helm.....	8
15. Upgrading Canonical Kubernetes.....	8
16. Best Practices with Kubernetes.....	8
Additional Topics.....	9
17. Monitoring and Observability.....	9

Introduction & Scope

This document provides a high level overview of the topics covered during the three (3) days of the Canonical Kubernetes Explorer training.

Agenda

The following items will be covered throughout the three-day training:

1. Snaps and Snapcraft

2. Juju – Application Modelling

LAB I: Snaps & Juju basics — bootstrap, models, charms (integrated within the cluster deployment lab).

3. Container Fundamentals

4. Kubernetes Overview and Architecture

5. Canonical Kubernetes Deployment

LAB II: Deploy a Canonical Kubernetes cluster (k8s snap + Juju, MAAS / LXD / cloud providers, Cluster API preview).

6. Pods and Workloads

LAB III: Pods, multi-container patterns, lifecycle hooks, probes, init containers.

7. Networking, Services and Ingress

LAB IV: Cilium CNI, Services (ClusterIP / NodePort / LoadBalancer with MetalLB), CoreDNS, Ingress and Gateway API preview.

8. Keeping Applications Healthy

LAB V: ReplicaSets, Deployments, rolling updates, rollback and rollout history.

9. Storage and Configuration Data for Applications

LAB VI: Volumes, Persistent Volumes, Persistent Volume Claims, Storage Classes (dynamic provisioning), ConfigMaps, Secrets, Downward API.

10. Resources and Auto Scaling (HPA)

LAB VII: Resource requests/limits, LimitRange, ResourceQuota and Horizontal Pod Autoscaler.

11. Authentication, Authorization and RBAC

LAB VIII: ServiceAccounts, Roles / ClusterRoles, RoleBindings / ClusterRoleBindings, Admission Controllers.

12. Pod and Network Security

13. Private Registry

14. Helm

LAB IX: Install Helm, deploy and customize charts (Headlamp), template walkthrough.

15. Upgrading Canonical Kubernetes

LAB X: Backup, snap and Juju upgrade workflow, in-place cluster upgrade (1.34 → 1.35).

16. Best Practices with Kubernetes

Day 1

1. Snaps and Snapcraft

- Introduction to Snaps as the packaging and delivery format for Canonical Kubernetes
- Snap features, channels and tracks (stable / candidate / beta / edge)
- Snap confinement model and security boundaries
- Installing, searching, updating and removing snaps

2. Juju – Application Modelling

- Why application modelling: Juju as the operator framework for Day-1 and Day-2 operations
- Juju components: clouds, controllers, models, machines, units and agents
- Charms: anatomy, files, endpoints and relations
- Core Juju operations: bootstrap, deploy, scale, integrate (relate), expose, status
- Juju bundles, configuration, troubleshooting and helpful operator commands
- Juju HA and lifecycle (removing applications, models and controllers)

Lab I

Snaps & Juju basics — exercised inside the Cluster Deployment lab. Covers snap installation and channel management, Juju controller bootstrap on MAAS / LXD, model creation, charm deployment and integration, status inspection and SSH into units.

3. Container Fundamentals

- Historical overview: VM virtualization vs. container virtualization
- Why containers: advantages, density and developer workflow
- Definition of a container: kernel namespaces and Linux Control Groups (cgroups)
- Container management, container engines and runtimes (containerd, CRI)

4. Kubernetes Overview and Architecture

- What Kubernetes is and what it solves: managing infrastructure and applications at scale
- Kubernetes benefits and value proposition
- Architecture and components: API server, etcd, scheduler, controller manager, cloud controller manager
- Worker node components: kubelet and kube-proxy
- Cluster architecture and chain of events
- Kubernetes data model and resource categories: Pods, Services, ReplicaSets, Deployments, DaemonSets, StatefulSets, ConfigMaps
- Container Network Interface (CNI) and Container Storage Interface (CSI)

5. Canonical Kubernetes Deployment

- Deployment with Snaps (k8s snap)
- Deployment with Juju (managed and external etcd topologies)
- Deployment with Cluster API (CAPI)
- Configuring infrastructure providers: AWS and MAAS
- Supported infrastructure providers and reference architectures

Lab II

Deploy a Canonical Kubernetes cluster. Provision the lab environment (LXD / MAAS), install required tooling, bootstrap a Juju controller, deploy the k8s charm, validate the control plane via juju ssh and snap services, and configure cluster add-ons (metrics-server, DNS, networking).

Day 2

6. Pods and Workloads

- Pods as the smallest deployable unit; Pods vs. individual containers
- Pods on Nodes: scheduling and durability
- Pod restart policy and termination process
- Pod lifecycle and hooks (postStart, preStop)
- Probes: startup, liveness and readiness
- Init containers and multi-container patterns
- kubectl create vs. kubectl apply — declarative vs. imperative usage

Lab III

Pods, multi-container patterns and Pod lifecycle. Exercise create vs. apply, image pull policies, lifecycle hooks, startup / liveness / readiness probes and init containers.

7. Networking, Services and Ingress

- Kubernetes networking model: flat network, no NAT and overlay networks
- Canonical Kubernetes networking components and CNI options (Cilium as default, alternative CNIs)
- DNS within the cluster (CoreDNS)
- Services and Endpoints; service types: ClusterIP, NodePort, LoadBalancer (MetalLB)
- kube-proxy modes and worker-node networking
- Labels, selectors and service discovery
- Ingress controllers and Ingress objects (simple fanout and name-based routing)
- Gateway API preview

Lab IV

Networking and Services. Configure Cilium CNI on Canonical Kubernetes, create ClusterIP / NodePort / LoadBalancer services, configure MetalLB, expose applications via Cilium Ingress and validate name-based routing.

8. Keeping Applications Healthy

- ReplicaSets and Deployments
- Rolling updates and rollout strategy
- Rollback and rollout history
- Operational patterns for high availability of stateless workloads

Lab V

Keeping applications healthy — ReplicaSets, Deployments, rolling updates, rollback and rollout history.

Day 3

9. Storage and Configuration Data for Applications

- Storage and ephemeral volumes (emptyDir, hostPath, multi-volume Pods)
- Persistent storage: Persistent Volumes and Persistent Volume Claims
- Reclaim policies and the PV lifecycle
- Storage Classes and dynamic provisioning
- Access modes and volume modes
- Configuration data: ConfigMaps, Secrets and Downward API

Lab VI

Storage and configuration data — emptyDir / hostPath volumes, Persistent Volumes and Claims, dynamic provisioning via Storage Classes, ConfigMaps and Secrets.

10. Resources and Auto Scaling (HPA)

- Resource requests and limits
- LimitRange and ResourceQuota
- Resource utilization and the role of metrics-server
- HorizontalPodAutoscaler — algorithm and resource calculation
- Overview of VerticalPodAutoscaler and Cluster Autoscaler

Lab VII

Resource governance and HPA — define requests/limits, configure LimitRange and ResourceQuota, and validate Horizontal Pod Autoscaler behaviour under load.

11. Authentication, Authorization and RBAC

- Request flow through the API server
- Authentication: users and ServiceAccounts
- Authorization modes; Role-Based Access Control (RBAC)
- Roles, ClusterRoles, RoleBindings and ClusterRoleBindings
- RBAC API verbs and checking effective access
- Admission Controllers — purpose and common examples

Lab VIII

Users, ServiceAccounts and RBAC. Create Roles / ClusterRoles, bind them to users and ServiceAccounts, validate effective permissions and review Admission Controller behaviour.

12. Pod and Network Security

- Pod security: from old-school infra security to Pod-level controls

- Pod Security Standards (PSS) and the PodSecurity admission plugin
- Security Contexts at Pod and container level
- Pod network security with Cilium NetworkPolicy
- Building network policies — sample Cilium policies and best practices

Live demos: SecurityContext + PSS, Cilium NetworkPolicy enforcement.

13. Private Registry

- Why a private registry: supply-chain control and offline / sovereign deployments
- Private registry options and integration with Canonical Kubernetes
- Pulling from authenticated registries and using imagePullSecrets

14. Helm

- Introduction to Helm as the package manager for Kubernetes
- Installing Helm and core Helm operations
- Helm charts and the templating model
- Customising charts and managing releases

Lab IX

Helm — install Helm, deploy a chart (Headlamp), customise values, inspect templates and manage release lifecycle.

15. Upgrading Canonical Kubernetes

- Pre-upgrade checks and backup strategy
- Snap upgrades and channel management
- Juju upgrades — controller and model
- In-place cluster upgrade workflow
- Restore from backup and rollback considerations

Lab X

Upgrading Canonical Kubernetes — perform an in-place upgrade between two consecutive Canonical Kubernetes channels (e.g. 1.34 → 1.35), including backup and validation.

16. Best Practices with Kubernetes

- General cluster operation best practices
- Container build best practices
- Application design best practices on Kubernetes
- Wrap-up and course survey

Additional Topics

If there is any remaining time during the training, the trainer will make best efforts to cover additional topics or provide clarifications based on questions raised by the customer during the sessions.

17. Monitoring and Observability

- Discuss monitoring options for Canonical Kubernetes — what tools can be deployed and supported
- Prometheus and the kube-prometheus-stack
- Logging with Loki / Elasticsearch
- Cilium Hubble for L3-L7 flow visibility and service maps
- Third-party observability products and integrations