SHI | TRAINING & ADOPTION

Course Outline

Certified Kubernetes Application Developers
Course CKAD: 5 days Instructor Led

About this course
This class prepares students for the Certified Kubernetes Application Developer (CKAD) exam. Kubernetes is a Cloud Orchestration Platform providing reliability, replication, and stability while maximizing resource utilization for applications and services. By the conclusion of this hands-on training, you will go back to work with all necessary commands and practical skills to empower your team to succeed, as well as gain knowledge of important concepts like Kubernetes architecture and container orchestration.

We prioritize covering all objectives and concepts necessary for passing the Certified Kubernetes Application Developer (CKAD) exam. You will command and configure a high availability Kubernetes environment (and later, build your own!) capable of demonstrating all "K8s'' features discussed and demonstrated in this course. Your week of intensive, hands-on training will conclude with a mock CKAD exam that matches the real thing.

Audience profile
- Anyone who plans to work with Kubernetes at any level or tier of involvement
- Any company or individual who wants to advance their knowledge of the cloud environment
- Application Developers
- Operations Developers
- IT Directors/Managers

At course completion
After completing this course, students will be able to:
- Deploy applications to a Kubernetes cluster
- Use Kubernetes primitives to implement common deployment strategies (e.g., blue/green or canary)
- Define, build and modify container images
- Implement probes and health checks
- Understand multi-container Pod design patterns (e.g., sidecar, init and others)
- Understand ConfigMaps
- Create & consume Secrets
- Troubleshooting and debugging tools
- Provide and troubleshoot access to applications via services
- Use Ingress rules to expose applications

Course Content

From Containers to Kubernetes
- Kubernetes Architecture
- Define, build and modify container images
- Pods and the Control Plane
- Deploy Kubernetes using Ansible

Cluster Basics
- Namespaces and Fundamental Kubectl Commands

SHI | TRAINING & ADOPTION

Course Outline

- Isolating Resources with Kubernetes Namespaces
- Contexts
- Cluster Access with Kubernetes Context
- Kubectl get and sorting
- Listing Resources with kubectl get
- Examining Resources with kubectl describe

Pod Basics

- YAML and how to read it
- Manifests for Pods
- Create and Configure Basic Pods
- API Versioning and Deprecations

Container Health, Security, and Observability

- Kubectl port-forward
- Debugging via kubectl port-forward
- Imperative vs. Declarative Resource Creation
- Kubectl exec and cp
- Performing Commands inside a Pod
- Readiness and Liveness Probes
- Implement Probes and Health Checks
- Security Contexts for Pods
- Understanding Security Contexts

Resource Management

- Limits, Requests, and Namespace ResourceQuotas
- Understanding and Defining Resource Requirements, Limits and Quotas Kubectl Top and Application Monitoring
- Admission Controller
- Create a LimitRange AdmissionController

RBAC

- LEC: Role Based Access Control
- LAB-S1: Role Based Access Control
- LAB-S2: RBAC Distributing Access

Logging

- Logging with kubectl log
- Advanced Logging Techniques
- Advanced Logging Techniques

Ephemeral Storage

- Creating Ephemeral Storage For Fluentd Logging Sidecar
- Best Practices for Container Customization
- ConfigMaps and Volume Mounting
- Persistent Configuration with ConfigMaps
- Secrets
- Create and Consume Secrets

SHI | TRAINING & ADOPTION

Course Outline

Persistent Storage

- Persistent Volumes, Claims, and StorageClasses
- Using PersistentVolumeClaims for Storage
- Stateful Containers for Testing
- Persistent Storage with NFS

Multi-Container Pod Design

- Multi-Container Pods
- Understand the Sidecar Multi-Container Pod Design Pattern
- Init Containers
- Understand the Init Container Multi-Container Pod Design Pattern

Deployments

- Labels
- Understanding Labels and Selectors
- Annotations
- Insert an Annotation
- ReplicaSets
- Create and Configure a ReplicaSet
- DaemonSets
- Deployments- Purpose and Advantages
- Writing a Deployment Manifest
- Deployments- Version Control
- Performing Rolling Updates and Rollbacks with Deployments
- Blue/Green and Canary Deployment Strategies
- Advanced Deployment Strategies
- Deployments- Horizontal Scaling
- Horizontal Scaling with kubectl scale
- Horizontal Pod Autoscaling
- Kubectl patch
- Patching Deployments and Nodes

Jobs and CronJobs

- Jobs and CronJobs
- Understand Jobs and CronJobs

Affinity and Anti-Affinity

- Taints, Tolerations, and Pod Affinity
- Tainted Nodes and Tolerations

NetworkPolicy

- NetworkPolicy
- Deploy a NetworkPolicy

Services and Ingress

- Services- LoadBalancer, NodePort, and ClusterIP

SHI | TRAINING & ADOPTION

Course Outline

- Provide and troubleshoot access to applications via services
- Networking Plugins
- Ingress Controllers
- Use Ingress Rules to Expose Applications

DNS

- LEC: Hostnames and FQDNs
- Hostnames and FQDNs

The Helm Package Manager

- Helm
- Using the Helm Package Manager to Deploy Existing Packages CHALLENGE LAB: Setting up a single tier service mesh A Completed Project

Extending Kubernetes

- Custom Resource Definitions
- Custom Resource Definitions (CRDs)

Troubleshooting

- Troubleshooting

CKAD

- Tips to Pass your CKAD Exam!
- CKAD Practice Drill
- CKAD Exam Bookmarks

Bonus Labs

- Calicoctl
- Deploy a Kubernetes Cluster using Kubeadm
- Create ServiceAccounts for use with the Kubernetes Dashboard
- Sourcing Secrets from HashiCorp Vault
- VNC Desktop
- Saving Your Progress with GitHub
- Create a Cluster Docker Registry
- Advanced Kubernetes Challenge - Build an Application

Index

- Useful Lab Links
- Best Practice Blogs