

## Openshift Container Deployment

### Doelgroep Cursus OpenShift Container Deployment

De cursus OpenShift Container Deployment is bestemd voor developers, DevOps engineers en systeembeheerders die applicaties willen deployen, beheren en schalen met OpenShift.

### Voorkennis Cursus OpenShift Container Deployment

Ervaring met Linux, basiskennis van containers en Kubernetes wordt aanbevolen. Ervaring met DevOps-principes en CI/CD processen is bevorderlijk voor de begripsvorming.

### Uitvoering Training OpenShift Container Deployment

De training bestaat uit interactieve presentaties afgewisseld met live demo's en praktische hands-on labs onder begeleiding van een ervaren trainer.

### Certificaat OpenShift Container Deployment

De deelnemers krijgen na het goed afronden van de cursus een certificaat van deelname aan OpenShift Container Deployment.

Duur: 2 dagen

Prijs: € 1499

[Open Rooster](#)



### Openshift Container Deployment



## Inhoud Cursus OpenShift Container Deployment

In de cursus OpenShift Container Deployment leren deelnemers hoe zij applicaties kunnen deployen, beheren en schalen binnen een OpenShift omgeving. De training behandelt containertechnologie, OpenShift architectuur, CI/CD pipelines en autoscaling.

### OpenShift Fundamentals

De cursus start met een overzicht van OpenShift, de verschillen met Kubernetes en het gebruik van de webconsole en oc CLI. Onderwerpen als projecten, RBAC, templates en services worden geïntroduceerd.

### Containers en Images

In deze module leren deelnemers over containertechnologie, het bouwen van images met Docker en Podman, gebruik van image registries, Source-to-Image (S2I) en security best practices voor containers.

### OpenShift Architectuur

Er wordt dieper ingegaan op de architectuur van OpenShift, met aandacht voor master- en workernodes, de control plane, etcd datastore, networking, storage oplossingen, multi-tenancy en monitoring.

### Applicaties Deployen

Deze module behandelt het deployen van applicaties via DeploymentConfigs, rolling updates, blue-green deployments, resource management, health checks en het gebruik van ConfigMaps en Secrets.

### CI/CD Pipelines Configureren

Deelnemers leren CI/CD pipelines op te zetten met OpenShift Pipelines en Tekton. Integratie met Git, gebruik van triggers en webhooks, beveiliging en best practices komen uitgebreid aan bod.

### Applicaties Beheren en Schalen

Focus ligt op het beheren en automatisch schalen van applicaties, gebruik van autoscalers, self-healing mechanismen, resource limits, monitoring via Prometheus en logging met Elasticsearch en Kibana.

## Modules Cursus OpenShift Container Deployment

<b>Module 1: OpenShift Fundamentals</b>	<b>Module 2: Containers and Images</b>	<b>Module 3: OpenShift Architecture</b>
What is OpenShift? Kubernetes vs OpenShift OpenShift Web Console oc CLI Tool Projects and Namespaces Role-Based Access Control OpenShift Templates Operators Overview Developer vs Admin Perspective Intro to OpenShift Services	Containers vs Virtual Machines Container Lifecycle Building Container Images Docker vs Podman Image Registries Pushing and Pulling Images Image Streams BuildConfigs in OpenShift Source-to-Image (S2I) Security Best Practices	Master and Worker Nodes Control Plane Components etcd Data Store Networking in OpenShift SDN vs OVN Ingress and Routes Storage in OpenShift Persistent Volumes Monitoring and Logging Multi-Tenancy and Isolation
<b>Module 4: Deploying Applications</b>	<b>Module 5: Configuring CI/CD Pipelines</b>	<b>Module 6: Scaling Applications</b>
Deployment Strategies Creating DeploymentConfigs Rolling Updates Blue-Green Deployment Canary Deployment Managing Resources Environment Variables Secrets and ConfigMaps Health Checks Automatic Rollbacks	OpenShift Pipelines Overview Tekton Pipelines Basics Pipeline Resources and Tasks Building CI/CD Pipelines Integrating with Git Repositories Triggers and Webhooks Managing Pipeline Runs Pipeline as Code (YAML) Security and Permissions Best Practices for CI/CD	Scaling Applications Manually Horizontal Pod Autoscaler Cluster Autoscaling Self-healing Applications Managing Application Updates Resource Quotas and Limits Observability with Prometheus Application Metrics and Alerts Using OpenShift Console Logging with ELK Stack