Ihr Partner für IT-Schulungen.
Seit über 30 Jahren.

GFU Cyrus AG

Docker und Kubernetes: Container-Administration und Orchestrierung - Kompakt (S1873)

Container, und die in ihnen bereitgestellten Microservices, sind mittlerweile zu einem festen Bestandteil der neuen IT-Landschaft geworden. Container basieren aktuell noch zu einem großen Teil auf Docker. Als alternative Container-Engine für erwachsene Orchestrierungslösungen wie Kubernetes oder OpenShift steht jedoch das von Red Hat und Google ins Leben gerufene CRI-O (Container Runtime Interface - OCI Standard) Projekt bereits in den Startlöchern. Zudem bietet Docker (Swarm) im Vergleich zu Kubernetes eher überschaubare Orchestrierungs-Qualitäten. Ein Grund mehr, warum sich Docker Ende 2017 entschieden hat, Kubernetes als alternativen Orchestrator für seine EE/DDC- Plattformen anstelle von Swarm anzubieten.
Kubernetes (K8s), das Googles "Borg" Orchestrierungsplatform entstammt, und nun unter der Federführung der Cloud Native Foundation weiter entwickelt wird, stellt eine der leistungsfähigsten und verbreitetesten Orchestrierungsplattformen weltweit dar.
In Verbindung mit Kubernetes (K8s) können Container viele Aufgaben erfüllen: komplette, sich selbst überwachende, hochverfüg- und skalierbare Container Cluster mit intelligenten Key/Value Backends, vollautomatischer Service Discovery, die selbst Rolling Upgrades / Rollbacks im laufenden Betrieb, unbemerkt für die Anwender, durchführen können. Features wie die Bereitstellung multipler Release-Tracks, Multi-Tenancy-Fähigkeiten, Network Policies und RBAC machen Kubernetes zu einer der leistungsfähigsten Container-Orchestrierungslösungen auf dem Markt.
Flexible, aktuelle SDS (Software Defined Storage)-Lösungen wie z. B. Ceph oder Gluster können nahtlos mit Kubernetes- und OpenShift- Umgebungen zusammenarbeiten, um optimale und Cluster-übergreifende, persistente Datenablagen zu bieten. Storage Classes und Claims bieten sogar die Möglichkeiten, Storage Tiering Modelle transparent zu integrieren.
!LiebelBuch!

Offene Termine

Termin	Tage	Freie Plätze	Ort	Preis
01.07.-04.07.2019 ✔ 3=2	4	>3	Köln	€ 2.170,00 ✱
16.09.-19.09.2019 ✔ 3=2	4	>3	Köln	€ 2.170,00 ✱
04.11.-07.11.2019 ✔ 3=2	4	>3	Köln	€ 2.170,00 ✱

✱ Buchen ohne Risiko
› Keine Vorkasse
› Kostenloses Storno bis zum Vortag des Seminars
› Rechnung nach erfolgreichem Seminar

✔ Garantierter Termin und Veranstaltungsort
€ Preise zzgl. Mehrwertsteuer
3=2 Der dritte Mitarbeiter nimmt kostenlos teil

Weitere Buchungsmöglichkeiten

Firmenschulung	Schulung für Ihre Mitarbeiter mit individuellen Inhalten zum Wunschtermin im GFU-Schulungszentrum.
Inhouse-Schulung	Schulung für Ihre Mitarbeiter mit individuellen Inhalten zum Wunschtermin in Ihrem Hause.
Individualschulung	Schulung für eine Einzelperson mit individuellen Inhalten zum Wunschtermin, wahlweise in Ihrem Hause oder im GFU-Schulungszentrum.

Schulungs-Ziel

Nach dem Workshop, welcher auf den Einsatz von Self-Hosted Container Clustern mit Docker und Kubernetes unter Linux fokussiert ist, haben die Teilnehmer ein fundiertes Know-How über die eingesetzten Container-Technologien, sind in der Lage, eigene Applikationscontainer zu erstellen und zu verwalten, und kennen die notwendigen Tools, um Container im täglichen Betrieb zu administrieren. Der stark praxisorientierte Workshop vermittelt ebenfalls das Know How über die Stärken und Schwächen der Container Technologien, gibt wichtige Ansätze zum Debugging und Troubleshooting, sowie Vorgehensweisen und designtechnische Konzepte an die Hand, um Container nahtlos in bestehende IT-Landschaften zu integrieren.
Zudem wird betrachtet, wie mit Kubernetes massiv skalierbare und hochverfügbare Container-Infrastrukturen zur Verfügung gestellt werden können.
Alle Übungen werden auf der Basis ausführlicher Workshop-Unterlagen und zugehöriger Beispieldateien/VMs ausgeführt, sodass der Teilnehmer auf diese Informationen auch nach dem Workshop jederzeit reproduzierbar zugreifen kann.
Die Teilnehmer erhalten zudem ausführliche und umfangreiche Workshop-Unterlagen (~ 1.500 Seiten) in PDF-/Buch-Form.

Wer sollte teilnehmen

Dieser Kurs richtet sich inhaltlich an Admins/DevOps Teams aus dem Linux-Umfeld, die sich tieferes Verständnis über Docker und Kubernetes aneignen wollen. Die Teilnehmer sollten folgende Vorkenntnisse besitzen:
- Solide Vorkenntnisse in der Administration von

Inhalt

Administration von Docker Containern

**** Für dieses Seminar sind solide Vorkenntnisse in den Bereichen "Linux-System und Netzwerk-Administration" erforderlich! ****

Erstellen / Administrieren von Images und Containern

- **Docker / Container Basics und Administration**
  - Container vs. VM: Einsatzgebiete, Vor- und Nachteile, Portierbarkeit
  - Continuous Delivery / Continuous Integration, Microservices
  - Übersicht: dedizierte Container-Plattformen wie Red Hats CoreOS/Atomic und SUSEs CaaSP
  - Docker und alternative Container-Engines: CRI-O, cri-containerd
  - Funktionale Übersicht: Container und Kernel-Namespaces, Kernel-Capabilities verstehen, auslesen und setzen
  - Docker Registry, Daemon und Client-Frontend (Docker CLI)
  - Docker Installation/Pakete, Versionsfragen und Plattformen
  - Images pullen, inspizieren und verwalten
  - Container-Instanzen starten und verwalten, limitieren
  - Applikationen im Container starten
  - Änderungen an Containern durchführen und commiten, Tagging-Regelwerke
  - Eigene 'trusted' Images erzeugen: Dockerfile, Build-Direktiven und Build-Prozess, Buildah als Docker-"lose" Alternative
  - (Zentralisiertes) Logging
  - Troubleshooting/Debugging und Best Practices
- **Storage**
  - Daten- / Layer- Strukturen unter /var/lib/docker

Linux-Systemen und ein sicherer Umgang mit der Kommandozeile sind zwingend erforderlich. Die Teilnehmer sollten zuvor mindestens das Seminar **Linux System-Administration** oder vergleichbare Seminare besucht haben, oder einen vergleichbaren Kenntnisstand aufweisen.
- Grundlegende Vorkenntnisse im Bereich Docker/Container, Virtualisierung und HA- sowie Storage- Clustering sind hilfreich

Organisation

Teilnehmerzahl
min. 1, max. 6 Personen

Seminarzeiten
4 Tage, 1. Tag 10:00 - 17:00 Uhr, Folgetage 09:00 - 16:00 Uhr

Ort der Schulung
GFU-Schulungszentrum Köln oder bei Ihnen als Inhouse-Schulung

Enthaltene Leistungen

Im Preis enthalten:
- Voll ausgestatteter Arbeitsplatz pro Teilnehmer
- Fachbuch zum Seminar
- Teilnahmezertifikat
- Kostenloser persönlicher Parkplatz
- Kostenloser Shuttle-Service
- Frühstück, Snacks und Getränke ganztägig
- Mittagessen im eigenen Restaurant, täglich 6 Menüs, auch vegetarisch

Haben Sie Fragen?

Gerne beraten wir Sie persönlich per Mail oder Telefon.

- info@gfu.net
- Infoline 0221 82 80 90

  - Storage Backends im Überblick, Vor- und Nachteile: AuFS, Overlay2, Devicemapper/direct-lvm, ZFS, BTRFS
  - Docker Volumes: Datenaustausch zwischen Container und Host, Externe Docker Volumes für Container
  - Clusterweit konsistentes ID-Mapping für Container-Daten per SSSD und externem, zentralem IDM wie Active Directory oder LDAP sicherstellen
- **Docker Networking**
  - Docker Networking: Aufbau und Funktionsweise
  - Docker Portmapping und iptables-Rulesets, Nachteile des Docker-Netzwerkstacks
  - Übergreifende Netzwerk-Konzepte abseits von Docker in echten Container-Clustern
- **Container-Security**
  - Grundsätzliche Überlegungen zur Security
  - SSL/TLS
  - Self-Hosted, Multi-Purpose Trusted Registry am Beispiel von Nexus Repository Manager oder Artifactory
  - Images: Vulnerability (CVE) Scanning, Image Signing und Verify. Host-Security mit CIS Benchmarks
  - Registry Authentifizierung per LDAP / AD
  - Rollen und Privilegien / RBAC Modelle

Container Cluster / Orchestrierung

- **Konzeptionelle Betrachtungen**
  - Überblick: Designansätze, Konzepte und Orchestrierungs-Tools für Container Cluster
  - Use-Cases für Vanilla Kubernetes, OpenShift, DC/OS
  - Monitoring-/HA-Mechanismen
  - Service Discovery und Key/Value Stores
- **Container Cluster mit Kubernetes (K8s)**
  - Kubernetes-Komponenten und -Architektur: Master + Worker, Nodes, etcd als Key Value Store, Networking mit flannel, weave und Alternativen
  - API-server, Controller Manager, Scheduler, Kubelets
  - Setup eines Multi-Node K8s-Clusters
  - Rollout: Bare Metal / VM, Pod-basiert per kubeadm,
  - Alternativen
  - Management Tools: Management des Clusters per
  - kubectl, grafische UI's (Kubernetes-Dashboard)

- K8s und RBAC: User- und Systemrollen, Authentifizierung
- und Autorisierung, RoleBindings, Contexts,
- Zugriffslimitierungen, NetworkPolicies und mehr
- API-Schemata/Versionen und K8s Ressourcen/Workloads
- Container, Pods, Replica Sets, Deployments, DaemonSets,
- ConfigMaps, Namespaces, Services, Ingress, Service-Meshes und mehr
- Health-Checks: Liveness und Readiness Probes
- Service-Exponierung an die Außenwelt, Headless Services,
- Label und Selektoren
- Auto-Respawning und Skalierbarkeit: Scale-up/Out and Scale Down, Autoscaling (HPA / Horizontal Pod Autoscale)
- Rolling Updates / Rollbacks, Revisions-History
- Node / Pod Metrics: Auswertung und Visualisierung
- K8s und Prometheus, Resource Monitoring und Logging
- Scheduler und Placement-Strategies: Limits Requests und Quotas
- Persistente und dynamische Storage-Volumes, Bereitstellung über Volume Claims, Storage Classes und mehr
- Strategien für Bare-Metal / VM HA Cluster Integrationen der K8s Kernkomponenten
- Konzeptionelle Betrachtungen: Hoch-Skalierbare und -verfügbare Software Defined Storage Backends für Container-Cluster
- Konzeptioneller Ausblick: Operator Ressourcen und Scale-Out abseits "dummer" Stateless Applications

Ausblicke

- Das "Enterprise Kubernetes": OpenShift - ein konzeptioneller Überblick
- Komplett Docker-lose Kubernetes Cluster mit CRI-O, Buildah, Podman
- Komplette Security Lösungen für Container Cluster
- Kubernetes in der Cloud
- The Road Ahead