



OpenShift Installation & Administration

Tobias Derksen



Über mich ...



Tobias Derksen

- IT Consultant @codecentric
- DevOps Specialist
- OpenShift Trainer
- RedHat Certified Engineer



Vorstellung

Agenda

- Einführung in OpenShift
- Cluster Konzeption
- Installation
- Web Interface & CLI Basics
- Hochverfügbarkeit
- Networking / SDN
- Persistent Storage
- Best Practices
- OpenShift 4
- Hochverfügbarkeit für Applikationen

Einführung in OpenShift

Was ein Chaos ...

OPENSIFT

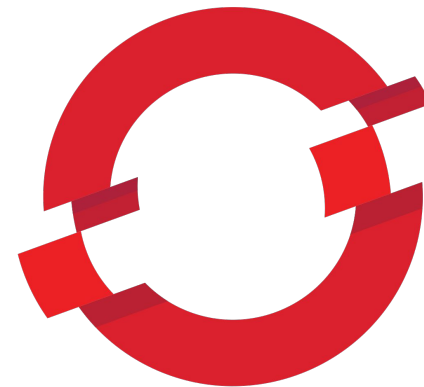
origin



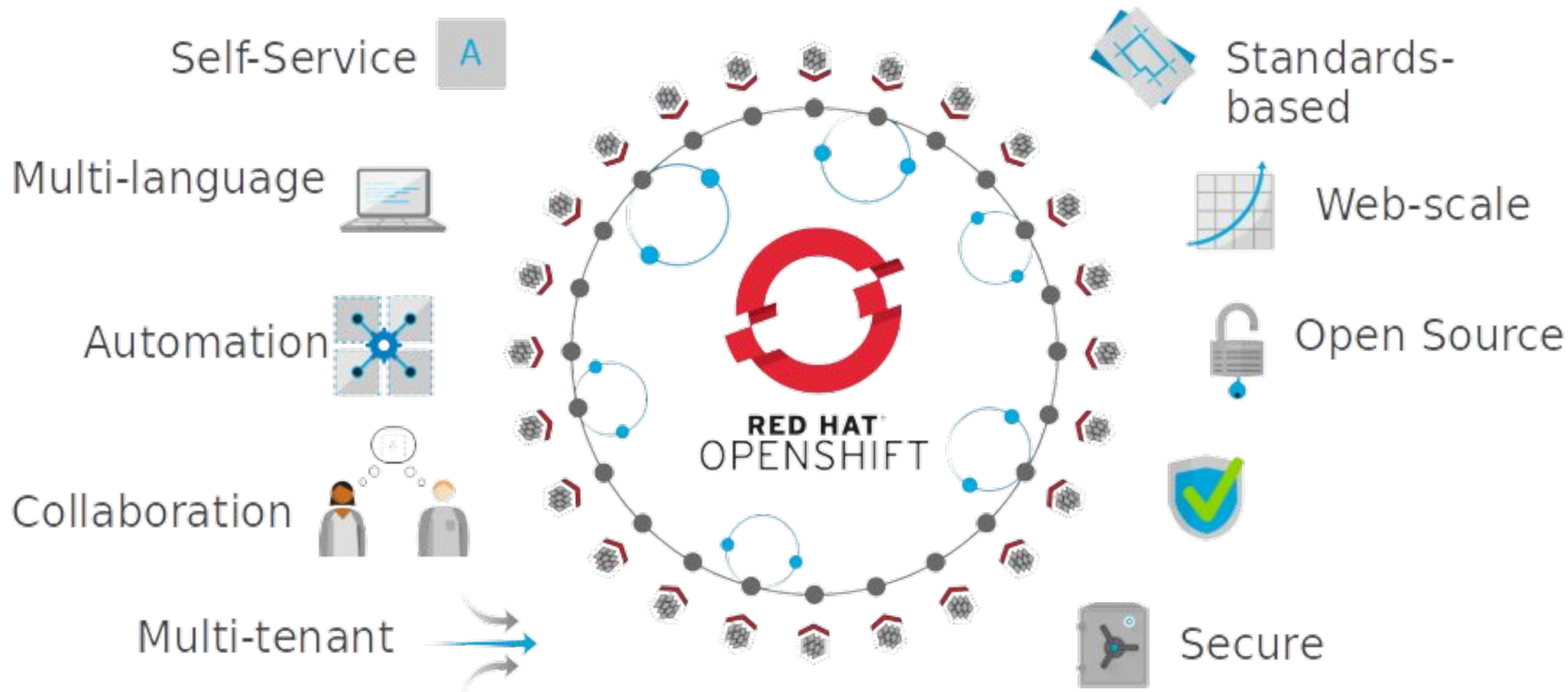
redhat.



kubernetes



OPENSIFT



OpenShift ist ... kubernetes plus

- Routing
- Metriken
- Logging
- Web Oberfläche
- Builds
- Image Registry
- Sicherheitsmaßnahmen
- SDN
- Templates

Mit Redhat Subscription:

- Trusted Registry
- Security Newsletter
- **Support**

Begriffe

- Container
 - Pod
 - Node
 - Projekt
 - Namespace
 - etcd
 - Gluster
- Ansible
 - Inventory
 - Playbook

Cluster Konzeption

Verschiedene Node Typen

Master Nodes

API - Server

ETCD

Web Console

Infrastructure Nodes

Router

Image Registry

Logging Stack

Metriken

Storage Controller

Compute Nodes

Applikationen

Services

Datenbanken

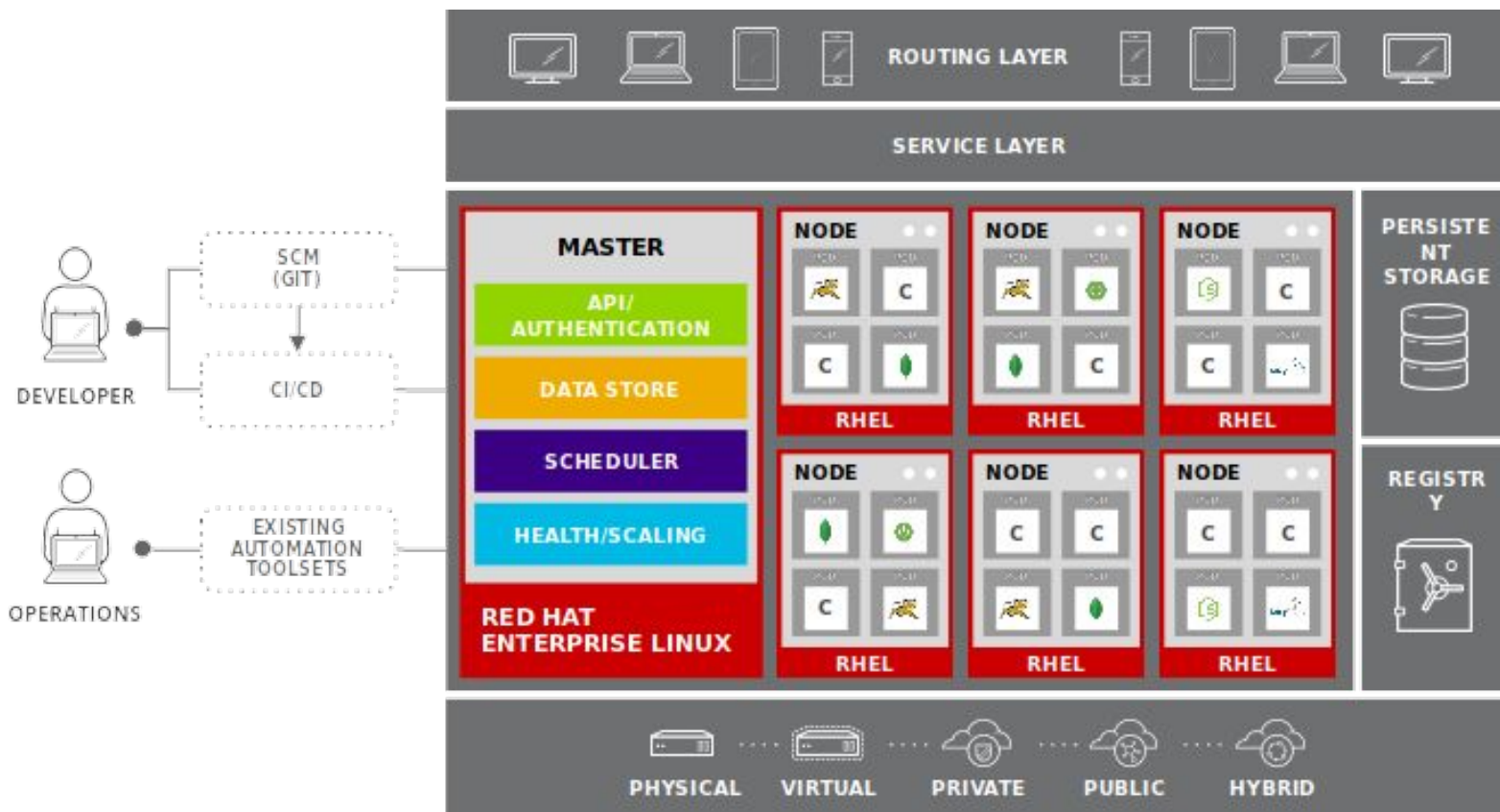
Builds

Andere Workloads

Storage Nodes

Nur beim Einsatz von
Gluster

Nodes mit physischem
Speicher



Minimum Cluster Sizing

Master Nodes	Infrastructure Nodes	Compute Nodes
<ul style="list-style-type: none">• Fedora, CentOS oder RHEL• 4 (v)CPU• 16GB RAM• 50GB disk	<ul style="list-style-type: none">• Fedora, CentOS oder RHEL• 2 (v)CPU• 8 GB RAM• 50GB disk	<ul style="list-style-type: none">• Fedora, CentOS oder RHEL• 1 (v)CPU• 8 GB RAM• 35GB disk

Recommended Cluster Sizing

Master Nodes	Infrastructure Nodes	Compute Nodes
<ul style="list-style-type: none">• Fedora, CentOS oder RHEL• 4 (v)CPU• 16GB RAM• 100GB disk	<ul style="list-style-type: none">• Fedora, CentOS oder RHEL• 4 (v)CPU• 16GB RAM• 100GB root disk• ≥ 250GB registry storage	<ul style="list-style-type: none">• Fedora, CentOS oder RHEL• ≥ 2 (v)CPU• ≥ 8GB RAM• ≥ 50GB disk

Mehr RAM => mehr disk (+25GB disk / 8GB RAM)

Anzahl der Nodes

	Minimal	Development	Production	Production (HA)
Master	1	1	1	3
Infrastructure			1+	2+
Compute		2+	3+	6+

Und wie viele Nodes brauche ich jetzt genau?

Einzelfall abhängig!

Kriterien:

- Erwarteter Workload der Applikationen
- Fest allokierte Ressourcen der Applikationen
- Gewünschte Pods per Node
- Hochverfügbarkeit (HA)
- Cluster Reserven
- Automatische Skalierung
- Mehr Ressourcen sind besser als mehr Nodes

Cluster Limits (OKD 3.11)

Anzahl der Nodes	2.000
Anzahl der Pods	150.000
Pods per Node	250
Namespaces / Projekte	10.000
Pods per Namespace	3.000
Pods per CPU	entfallen

Installation vorbereiten

Bastion Host

- Sprung-Host für SSH
- Zentrale Verwaltung der Konfiguration
- Zentrale Verwaltung der OpenShift-Version
- Keine Ansible / Python Versionsprobleme
- Installer benötigt Abhängigkeiten

Schritt für Schritt zur Installation

1. Infrastruktur provisionieren
2. System Updates und Abhängigkeiten installieren
3. DNS Einträge erstellen und prüfen
4. Inventory erstellen
5. Playbook: prerequisites.yml
6. Playbook: deploy_cluster.yml
7. Zusätzliche Aufgaben nach der Installation

Besonderheiten & Abhängigkeiten

- x86_64 Architecture
- Kein Support für IPv6 cluster-intern
- SELinux benötigt (enforcing)
- NetworkManager
- firewalld (recommended)
- rngd (rng-tools)

DNS Einträge

Eintrag	Master (extern)	Master (intern)	Routes
Beispiel	master.openshift.com	internal.openshift.com	*.apps.openshift.com
Ziel	Master Nodes (8443)	Master Nodes (8443)	Infra Nodes (80, 443)
Benutzung	Externer Zugriff auf Master für CLI und Web Oberfläche.	Interne Kommunikation der Nodes mit dem Master	Eintrittspunkt für externen Traffic. Konkrete Routen werden von OpenShift generiert.

```

[OSEv3:children]
masters
nodes
etcd

[OSEv3:vars]
ansible_user=centos
ansible_become=true
ansible_ssh_common_args='-o StrictHostKeyChecking=no'

deployment_type=origin
openshift_deployment_type=origin

openshift_disable_check=docker_storage,memory_availability
openshift_clock_enable=true
openshift_use_dnsmasq=true
os_firewall_use_firewalld=true

osm_use_cockpit=true
openshift_release='v3.11'

openshift_master_default_subdomain='apps.training0.cc-openshift.de'
openshift_master_cluster_hostname='master0.training0.cc-openshift.de'
openshift_master_cluster_public_hostname='master0.training0.cc-openshift.de'

openshift_master_identity_providers=[{'name': 'htpasswd_auth', 'login': 'true', 'challenge': 'true', 'kind': 'HTPasswdPasswordIdentityProvider'}]
openshift_master_htpasswd_users={'admin': '$apr1$zGsjCrLt$1KSuj66CggeWSv.D.BX0A1', 'user': '$apr1$.gw8w9i1$ln9bfTRiD60wuNTG5LVW50'}

[masters]
master0.training0.cc-openshift.de openshift_node_group_name='node-config-master-infra' openshift_schedulable=true

[etcd]
master0.training0.cc-openshift.de
|

[nodes]
master0.training0.cc-openshift.de openshift_node_group_name='node-config-master-infra' openshift_schedulable=true
app0.training0.cc-openshift.de openshift_node_group_name='node-config-compute' openshift_schedulable=true
app1.training0.cc-openshift.de openshift_node_group_name='node-config-compute' openshift_schedulable=true
app2.training0.cc-openshift.de openshift_node_group_name='node-config-compute' openshift_schedulable=true

```

Node Group Config

- `node-config-master`
- `node-config-infra`
- `node-config-compute`

- `node-config-master-infra`
- `node-config-all-in-one`

Nach der Installation

- Cluster Administrator ernennen

```
oc adm policy add-cluster-role-to-user cluster-admin <username>
```

Wichtige Cluster Komponenten

- Master API
- etcd
- Web Console
- Router
- Registry

Zertifikate

- OpenShift Root CA wird bei Installation generiert
- Zertifikate werden erstellt für:
 - Nodes
 - etcd
 - Router
 - Services (Metriken, Logging, etc)

Achtet auf das Ablaufdatum!!!!!!

Erneuerung der Zertifikate mit Playbook

Nachinstallation von Komponenten

- Einige Komponenten lassen sich einfach nachinstallieren
- Man kann das “deploy_cluster” Playbook nochmal laufen lassen
- Man kann das entsprechende Komponentenplaybook starten

```
openshift_logging_install_logging=true  
openshift_metrics_install_metrics=true  
openshift_logging_es_nodeselector={"node-role.kubernetes.io/infra":"true"}
```

Objekte & Ressourcen

Alles nur Objekte

- Der Zustand des Clusters wird mit den verschiedenen Objekten abgebildet.
- Cluster Objekte (z.B. Namespaces, Persistent Volumes)
- Projekt Objekte (z.B. Deployments, Builds)
- Die Objekte werden im etcd gespeichert

Wichtige Objekt Typen

- Clusterroles
- Rolebindings
- Persistent Volumes
- Persistent Volume Claims
- Template
- Pod
- ConfigMap
- Secret
- Deployment
- DeploymentConfig
- Build
- Route
- Service

OpenShift CLI Basics

User Management

OpenShift Identity Provider

Möglichkeiten zur User Verwaltung

HTPASSWD

Hard-coded
Passwörter im
htpasswd Format
welche lokal auf den
Mastern liegen.

LDAP

Generischer LDAP
Authenticator. Kann
mit jedem
handelsüblichen
LDAP Server
verbunden werden.

Social Logins

Github

Gitlab

Google

OpenID Connect

Generischer OpenID
Connect
Authenticator. Kann
jeden OAuth2 oder
OIDC Provider
anbinden.

LDAP Anbindung im Inventory

```
openshift_master_identity_providers=[
{
    'name': 'ldap_auth',
    'challenge': 'true',
    'login': 'true',
    'kind': 'LDAPPasswordIdentityProvider',
    'attributes': {'id': ['uid'], 'email': ['mail'], 'name': ['cn'], 'preferredUsername': ['cn']},
    'bindDN': 'cn=openshift,dc=cc-openshift,dc=de',
    'bindPassword': 'OpenShiftLdap',
    'insecure': 'true',
    'url': 'ldap://ldap.cc-openshift.de:389/dc=cc-openshift,dc=de?cn'
}
]
```

LDAP Gruppen synchronisieren

- Mapping von LDAP Gruppen auf OpenShift Rollen
- Manuelle Konfiguration
- Manuelles Synchronisieren
- https://docs.okd.io/3.11/install_config/syncing_groups_with_ldap.html

```
groupUIDNameMapping:  
  "cn=group1,ou=groups,dc=example,dc=com": cluster-admin  
  "cn=group2,ou=groups,dc=example,dc=com": cluster-reader  
  "cn=group3,ou=groups,dc=example,dc=com": project-admin
```

```
oc adm groups sync --sync-config=config.yaml --confirm
```

Rollen & Rechte

- Cluster Rollen
- Projekt Rollen
- Rechte bestehen aus Verb + Objekttype (Beispiel: get projects)
- Rechte eines Accounts = Summe aller erlaubten Aktionen
- Systemaccounts

Cluster Rollen:

- cluster-admin
- cluster-reader
- self-provisioner

Projekt Rollen:

- admin
- edit
- view

Skalierung & HA

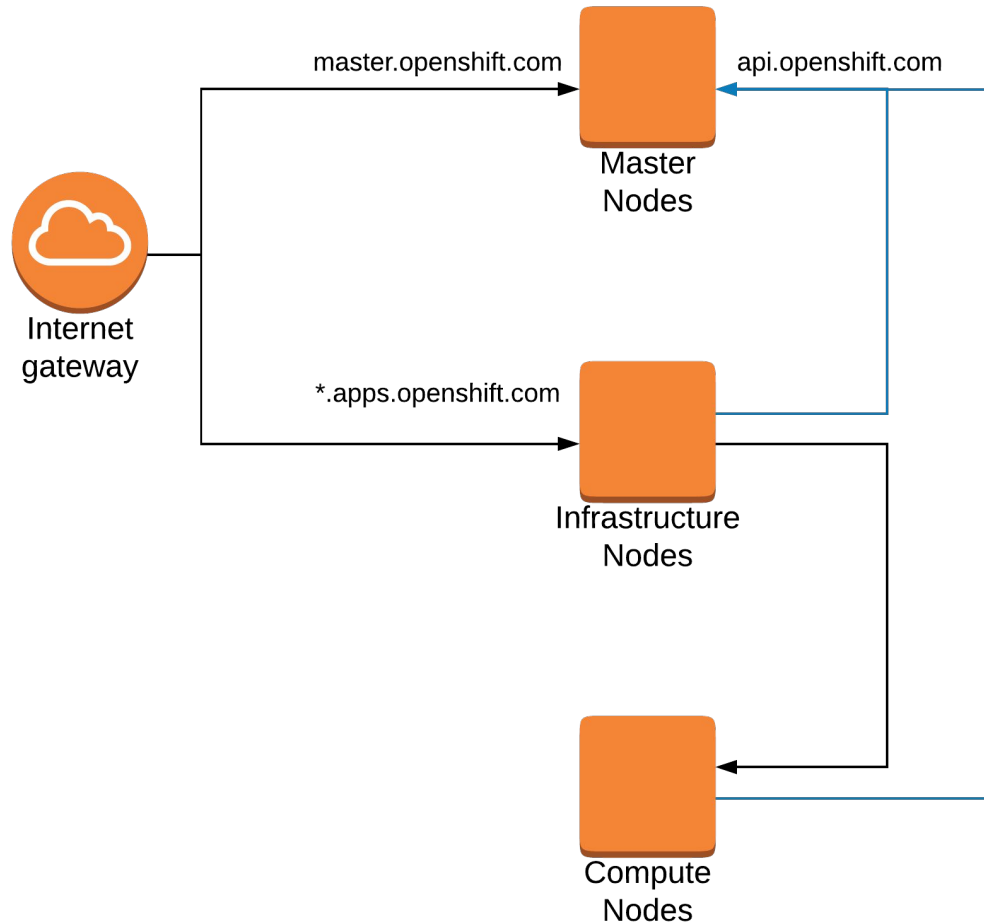
Skalierung

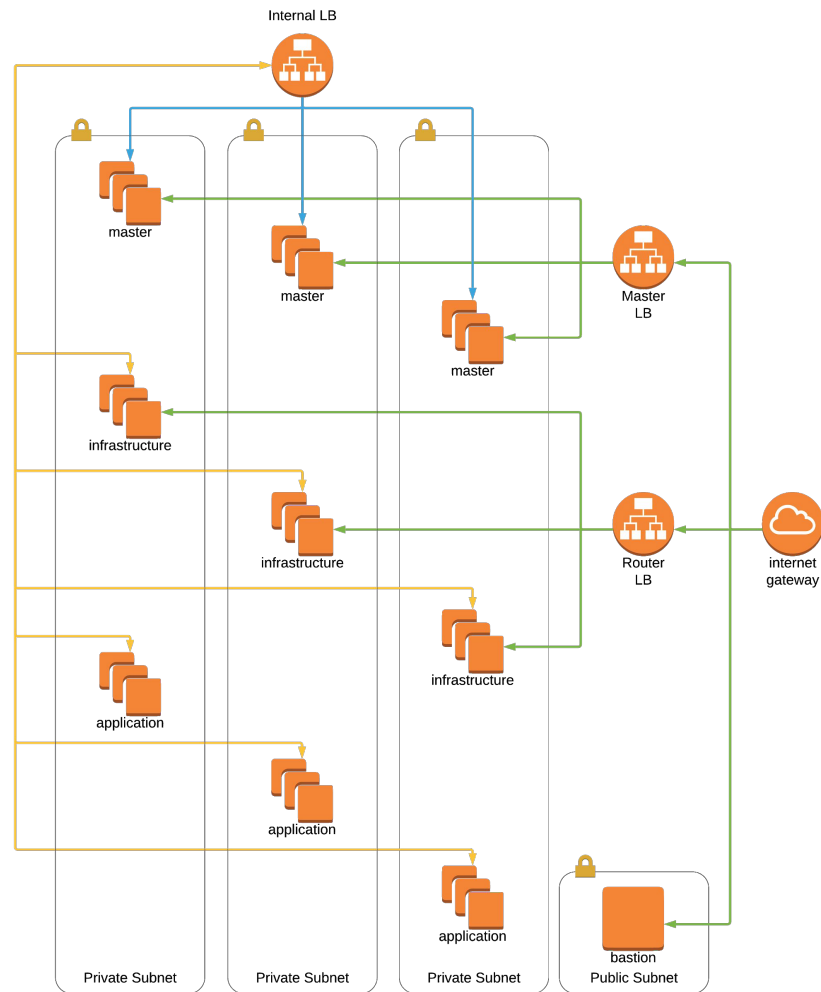
- Master hinzufügen
- Node hinzufügen
- Node entfernen

- Node updaten (System updates)
- Cluster updaten

Hochverfügbarkeit

- min. 3 Master Nodes
- min. 2 Infrastructure Nodes
- Genug Compute Nodes um die Workload zu übernehmen
- Loadbalancer für Infrastructure Nodes
- Loadbalancer für Master API
- Vorsicht vor DNS Problemen
- HA im DNS
- HA im Storage System





Zones & Region

- /etc/origin/master/scheduler.json
- Zone: Anti-Affinität
- Region: Affinität
- Custom Configuration:
 - Racks
 - Build Nodes
 - Enforce Labeling

```
[root@ip-10-1-5-240 master]# oc label node master-1 zone="zone-1" region="frankfurt"
```

```
[
  {
    "argument": {
      "serviceAntiAffinity": {
        "label": "zone"
      },
      "name": "Zone",
      "weight": 2
    }
  },
  {
    "argument": {
      "serviceAffinity": {
        "label": "region"
      },
      "name": "Region",
      "weight": 2
    }
  }
]
```

OpenShift SDN

Network Plugins

- ovs-subnet
- ovs-networkpolicy
- ovs-multitenant

- Unterschiede in Isolationsgrade

```
os_sdn_network_plugin_name='redhat/openshift-ovs-networkpolicy'
```

Ingress Network Policy

- Objekttyp: NetworkPolicy
- Kontrolliert eingehenden Traffic per Pod
- Kann einzelne Pods im **selben** Namespace freischalten
- Kann **ganze** externe Namespaces freischalten

```
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: allow-http-and-https
spec:
  podSelector:
    matchLabels:
      role: frontend
  ingress:
    - ports:
      - protocol: TCP
        port: 80
      - protocol: TCP
        port: 443
```

Egress Network Policy

- Objekttyp: EgressNetworkPolicy
- Kontrolliert **cluster-externen** Traffic
- Ein Policy Objekt pro Namespace
- Kann mit einigen Techniken umgangen werden

```
kind: EgressNetworkPolicy
apiVersion: v1
metadata:
  name: default
spec:
  egress:
    - type: Allow
      to:
        cidrSelector: 1.2.3.0/24
    - type: Allow
      to:
        dnsName: www.foo.com
    - type: Deny
      to:
        cidrSelector: 0.0.0.0/0
```

Third-Party-Plugins

- https://docs.okd.io/3.11/architecture/networking/network_plugins.html

Backup & Restore

Backup Möglichkeiten

1. Snapshot der Maschinen
2. Backup der Konfigurationen und wichtigen Daten (Velero)
3. etcd Backup
4. Objekt-Export als YAML oder JSON
5. Infrastructure-as-Code

etcd Backup

- Backup der etcd Datenbank
- Bringt den Cluster in den **exakt** selben Zustand wie zur Zeit des Backups

```
etcdctl3 snapshot save /backup/db
```

```
etcdctl3 member list
```

DR Szenarien

1. Node(s) fällt aus
2. Master fällt aus
3. Projekt(e) wird gelöscht / verschwindet
4. Rechenzentrum fällt aus (mit HA)
5. Cluster fällt aus
6. etcd fehlerhaft

Persistent Storage

Persistent Storage Provider

- HostPath
- EmptyDir (Ephemeral Storage)
- GlusterFS / OpenShift Container Storage
- NFS (unsupported)
- iSCSI
- Ceph
- Diverse Cloud Mechanismen (AWS, GCE, Azure, etc)
- Dynamic Provisioning

Best Practices

Externe Image Registry

Vorteile:

- Keine Abhängigkeiten an die interne Registry
- Hochverfügbarkeit wird ausgelagert

Nachteile:

- Wartung
- evt. Lizenzkosten
- Hardware

Best Practices - Cluster betreiben

- Nicht alle Applikationen eignen sich dafür
 - Monolithen -> schlechte Skalierung
 - Datenbanken -> von schneller Storage abhängig
 - Nicht HTTP basierter Traffic
- Infrastructure-as-Code
- “/var/log” läuft schnell voll
- Monitoring der Ressourcen

Best Practices - Security

- SELinux nicht deaktivieren
- Cluster Nodes nur intern (über Bastion) erreichbar
- non-root Container
- Container Scanning nach Sicherheitslücken
- Blocken von offenen Registries (Docker Hub, Quay.io)
- EgressIP für Firewalls / Network Policies
- Traffic Encryption (Service Mesh)
- Regelmäßige Updates im Cluster
- **Regelmäßige Updates der Base Images**

OpenShift 4

What's new ...

- Neuer Installer
- Over-the-air Updates
- Cluster Autoscaling
- Neues User Interface
- Developer CLI Tools (ODO)
- Service Mesh (Istio)
- Quay
- Operators & Operator Hub

Installation

- Installer provisioned Infrastructure (IPI)
- User provisioned Infrastructure (UPI)
- AWS
- Azure
- VMware
- Bare Metal

HA for Applications

Hochverfügbarkeit done right

- TODO

Best Practices

- Ressource Allocation / Quality of Service
- Quotas
- PodDisruptionBudget
- Deployment Strategy

Ende

Upcoming Events

- 30.9. - OpenShift Anwendertreffen Frankfurt
- 19. - 21.11. - kubecon San Diego (USA)

Stay connected



Adresse
codecentric AG
Köpenicker Straße 31
10179 Berlin - Mitte













Contact Info
E-Mail: tobias.derksen@codecentric.de
www.codecentric.de



Telephone
Telefon: +49 (0) 170 2295 733

Hello, World!

cc_primary template colours (included in master template)

	#FFFFFF		#15584C
	#000000		#1FB18A
	#F0F6F4		#2CE6AF
	#004452		
	#007891		
	#00AED2		
	#03BDEC		

Link colour

	#D6B32C
	#9C954E

cc_secondary template colours (you need to build by yourself)

	#EF5E1B
	#D6B32C
	#E61B77

cc_icons

