Automation for Setup and Configuration of OpenShift

Andy Wirtz

15th October 2019
Andy Wirtz

Andy:
- IT Consultant at ATIX AG, Germany
- Automation of data centers
- Deployment of cloud native services
- Expertise in Docker, Kubernetes, Istio

Contact:
- Phone: +49 (0)89 452 35 38-248
- Mail: wirtz@atix.de
- www.xing.com/profile/Andy_Wirtz2
- www.linkedin.com/in/andy–wirtz

#atix #osad #container #automation
Agenda

1. Motivation and Requirements
2. Automation of the Setup
3. Automation in the Cluster
4. Automation of the Configuration
5. Summary

#atix #osad #container #automation
Agenda

1. Motivation and Requirements
2. Automation of the Setup
3. Automation in the Cluster
4. Automation of the Configuration
5. Summary
Container Platforms

Container platforms:
- Hardware abstraction
- Cloud for containers
- Efficient resource management

Automation:
- Scaling dependent on the requests
- Some ready to use pods in stock
- Self-healing
- Updates without downtime
On-Premises:

- Cloud-native technologies for the own data centers
- Private cloud for containers
- Defining own security and protection of data privacy

Offline:

- Disconnected installation and operation
- No direct connection to the internet
Automation

Automation:
- Standardization
- Reproducibility
- Necessary in a highly dynamic IT world

Container platforms:
- Automation of the setup
- Automation of the cluster
- Automation of the configuration
Agenda

1. Motivation and Requirements
2. Automation of the Setup
3. Automation in the Cluster
4. Automation of the Configuration
5. Summary
Setup overview

- Deployment
- Repos/Images
- Preparation
- Installation

Container Platform

- Pod 1
- Pod 2
- Pod 3
- Pod 4
- Pod 5
- Pod 6
Offline installation:

- Hosts are disconnected from internet
- They need repositories/images for installation

Orcharhino:

- Lifecycle management tool
- Based on TheForeman & Katello
- Provides repositories/images
- Smart proxy synchronizes
- Smart proxy is local mirror for offline hosts
Agenda

1. Motivation and Requirements
2. Automation of the Setup
3. Automation in the Cluster
4. Automation of the Configuration
5. Summary

#atix #osad #container #automation
Cluster overview

- **Ingress**
- **Service**
- **Deployment**
- **ReplicaSet**
- **Pod**
- **Resource Quota**
- **HorizontalPod Autoscaler**
- **StorageClass**
- **Persistent VolumeClaim**
Deployment Object

Deployment object:

- Update apps declaratively
- Trigger rollout by applying a change
- Tool reacts

Update strategies:

- Recreate strategy
- Rolling update
- Blue green deployment
- Canary release
Recreate Strategy

replicas = 3

Pod c
Pod b
Pod a

Pod f
Pod e
Pod d

Pod f
Pod e
Pod d
Rolling Update 2

maxSurge = 1
replicas = 3
maxUnav. = 1

Pod a  v1
Pod b  v1
Pod c  v1
Pod d  v1
Pod e  v2
Pod f  v2
Blue Green Deployment
Canary Release

replicas = 3

Pod a
v1
Pod b
v1
Pod c
v1
Pod d
v2

pause ...

resume ...

Pod a
v1
Pod b
v1
Pod c
v1
Pod d
v2

Pod e
v2
Pod f
v2
Pod d
Automatic Stoppage

Automatic stoppage:
- Block rollouts of bad versions
- Define readiness probe
- Use parameter “minReadySeconds”

Readiness probes:
- HTTP GET probe
- Exec probe
- TCP socket probe
Agenda

1. Motivation and Requirements
2. Automation of the Setup
3. Automation in the Cluster
4. Automation of the Configuration
5. Summary
Configuration overview

- Storage
- Service Mesh
- Central Logging
- Security
- Monitoring
- CI/CD
- Backup
- Infrastructure as Code

Container Platform:
- Pod 1
- Pod 2
- Pod 3
- Pod 4
- Pod 5
- Pod 6
Infrastructure as Code:

- Declarative model
- Define your desired state
- Tool compares desired and actual state
- Tool acts if need be

Cluster administration via code:

- Use version control repositories
- Separate code and data
- Make use of idempotence
Namespaces:
- Separate teams and applications

Projects:
- OpenShift object
- For multitenancy
Role-Based-Access-Control:

- Service accounts: “who”
- (Cluster) role bindings: “is allowed to”
- (Cluster) roles: “perform what to whom”

Users and groups:

- OpenShift objects
- For humans instead of processes in pods
- Sync with corporate database
Pod Security Policies

Pod security policies:
- Restrict security-related features
- Ensure isolation to hosts
- Prevent priviledged escalation

Security context constraints:
- OpenShift object
- Same objective
Network Policies

Network policies:
- Isolates the pod network
- Limit inbound and/or outbound traffic
- Cluster admin can define default deny

Isolation via:
- Namespaces
- Labels
- Pod IPs
Resource Management

Resource management:
- Limit range: defines min, max, default limits and requests for pods
- Resource quota: defines amount of resources available for pods in namespace

Resources:
- CPU and memory
- Storage
- Number of objects
Agenda

1. Motivation and Requirements
2. Automation of the Setup
3. Automation in the Cluster
4. Automation of the Configuration
5. Summary
Automation of the Setup

Orcharhino:

- Deployment of hosts (VMs)
- Provisioning of repositories/images
- Preparation of the hosts
- Installation of the container platform
Automation in the Cluster

Kubernetes/OpenShift:
- Reconciliation loop
- Rolling updates
- Service discovery
- Load balancing
- Storage provisioning
- Storage binding
- Quota enforcement
- Horizontal scaling
Automation of the Configuration

Git:
- Distributed version control system
- Infrastructure as Code
- Separate code and data
- Use CI/CD

Kubectl apply:
- Apply changes declaratively
- Rollout new configurations
- Make use of idempotence