

Openshift

Laurent Valeyre

Containors

Containers

Understanding Pod

Patterns

. . .

Dockerfile

Secret Access
The application

Deployment and Service

Prometheu

grafana

Deploy Jenkins a Our Pipeline

Definition
Deploy Our Sample

Openshift

Introduction to the Side Car

Laurent Valeyre

Orange

August 2018



Table of contents

Openshift

Laurent Valeyre

Containers

Understanding Po

Patterns

Apache Status

Dockerfile Secret Access

The application
Deployment and

Prometheus

Prometheus grafana Grafana

Deploy Jenkins and Our Pipeline 1 The base

Containers

Understanding Pods

Main Ways

Patterns

2 Use case

Apache Status

■ Dockerfile

■ Secret Access

■ The application

■ Deployment and Service

3 The Monitoring

■ Prometheus

■ grafana

■ Grafana

4 Pipeline and CI

■ Deploy Jenkins and Our Pipeline Definition





Containers

Openshift

Laurent Valeyre

Containers

Understanding Pod Main Ways

Apache Statu

Secret Access

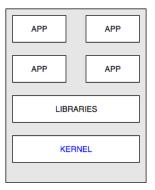
The application
Deployment an

Promethe

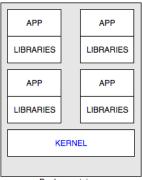
Deploy Jenkins at Our Pipeline

Definition Deploy Our Sample

Figure: Why Containers



Applications on host heavyweight, non-portable Relies on OS package manager



Deploy containers Small and fast, portable Uses OS-Level virtualization



Openshift

Laurent Valevre

Understanding Pods

Secret Access

A Pod is the smallest and simplest unit in the Kubernetes object model that we create or deploy.



Openshift

Laurent Valeyre

Understanding Pods

Understanding Po

Main Ways

Apache Statu Dockerfile

Secret Access

Deployment and

Promether grafana

- A Pod is the smallest and simplest unit in the Kubernetes object model that we create or deploy.
- A Pod represents a running process on our cluster.



Openshift

Laurent Valeyre

Containers
Understanding Pods

onderstanding roo

Dana---

Apache Statu Dockerfile

Secret Access

Deployment and Service

Prometheus

grafana Grafana

Deploy Jenkins an Our Pipeline Definition

- A Pod is the smallest and simplest unit in the Kubernetes object model that we create or deploy.
- A Pod represents a running process on our cluster.
 - A Pod encapsulates an application container (or, in some cases, multiple containers)



Openshift

Laurent Valeyre

Understanding Pods

Onderstanding 1 of

Dattorne

Apache Statu Dockerfile

Secret Access

Deployment and Service

Promethe

Deploy Jenkins and Our Pipeline Definition

- A Pod is the smallest and simplest unit in the Kubernetes object model that we create or deploy.
- A Pod represents a running process on our cluster.
 - A Pod encapsulates an application container (or, in some cases, multiple containers)
 - A Pod storages resources



Openshift

Laurent Valeyre

Containers
Understanding Pods

Main Mann

Iviain vvays

Apache Stati Dockerfile

Secret Access

Deployment and Service

grafana Grafana

Deploy Jenkins and Our Pipeline Definition A Pod is the smallest and simplest unit in the Kubernetes object model that we create or deploy.

- A Pod represents a running process on our cluster.
 - A Pod encapsulates an application container (or, in some cases, multiple containers)
 - A Pod storages resources
 - A Pod has a unique network IP



Openshift

Laurent Valeyre

Understanding Pods

Main Wave

Patterns

Apache Statu Dockerfile

The application Deployment and

Prometheu: grafana

Deploy Jenkins and Our Pipeline Definition A Pod is the smallest and simplest unit in the Kubernetes object model that we create or deploy.

- A Pod represents a running process on our cluster.
 - A Pod encapsulates an application container (or, in some cases, multiple containers)
 - A Pod storages resources
 - A Pod has a unique network IP
- A *Pod* represents a unit of deployment:



Main ways

Openshift

Laurent Valeyre

Main Ways

Secret Access



Deploy Our Sample

Pods in a Kubernetes cluster can be used in two main ways:



Main ways

Openshift

Laurent Valevre

Main Ways

Secret Access

Pods in a Kubernetes cluster can be used in two main ways:

Pods that run a single container (most common Kubernetes use case)



Main ways

Openshift

Laurent Valeyre

Understanding F

Main Ways Patterns

Apache Statu
Dockerfile

The application
Deployment and

Promethe grafana Grafana Pods in a Kubernetes cluster can be used in two main ways:

- Pods that run a single container (most common Kubernetes use case)
- Pods that run multiple containers that need to work together(encapsulate an application composed of multiple co-located containers that tightly coupled)



Patterns for Composite Containers: Sidecar

Openshift

Laurent Valeyre

Containers

Understanding Po

Patterns

Apache Stat

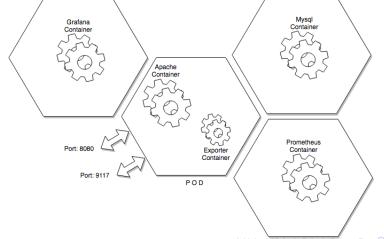
Dockerfile Secret Access

The sections

Deployment and

Prometheus grafana

Deploy Jenkins an Our Pipeline Definition Figure: schema of our Sidecar





Apache status

Openshift

Laurent Valeyre

Containers

Main Ways

Apache Status

Dockerfile Secret Acces

The application Deployment and

Prometheus

grafana Grafana

Deploy Jenkins ar Our Pipeline Definition Module to enable the output statistic of *Apache*.

```
<Location /server-status>
   SetHandler server-status
   Order deny,allow
   Allow from all
</Location> ExtendedStatus On>
```

Figure: status.conf

This module has to be copied in the /etc/apache2/mods-enabled/ directory.



Dockerfile

Openshift

Laurent Valeyre

Containers

Main Ways

A---b- Casa

Dockerfile

The application

The Monitorii

Prometheu grafana Grafana

Deploy Jenkins and Our Pipeline Definition The *Dockerfile* includes the copy of the *Apache* module Important to add the switching between *root* and *1001* user

```
FROM ubuntu:latest
USER root
...
RUN a2enmod status
COPY status.conf /etc/apache2/mods-enabled/
EXPOSE 8080
USER 1001
CMD ["/usr/sbin/apache2ctl", "-DFOREGROUND"]
```

Figure: Dockerfile



Secret Access

Openshift

Laurent Valeyre

Containers

Main Ways

Patterns

Apache Statu

Secret Access

The application
Deployment and

Prometheus grafana

Deploy Jenkins and Our Pipeline Definition And because the credential of *GITLAB*, we'll use the login/password based on the token initialized in our profile

apiVersion: v1 kind: Secret

metadata:

metadata:

name: gitlab-secret
namespace: sidecar

type: kubernetes.io/basic-auth

data:

username: c3Bpa2U=

password: dmFsZW50aW51

Figure: gitlab-secret.yaml



Secret Access

Openshift

Laurent Valeyre

Containers
Understanding Pod

Patterns

Apache Status

Dockerfile

Secret Access

The application

Deployment and Service

Promethe

The *username* and *password* are encoded to Base64 format. Finally we load the new *secret*

```
$ echo -n 'spike' | base64
c3Bpa2U=
$ echo -n 'valentine' | base64
dmFsZW50aW51
$ oc create -f gitlab-secret.yaml
```



New Project

Openshift

Laurent Valeyre

Containers

Main Ways

Apache Statu

Dockerfile Secret Access

The application Deployment and

Promethe grafana

Deploy Jenkins ar Our Pipeline Definition It's time to create our new project *sidecar*, similar to a namespace

```
$ oc new-project sidecar \
--display-name='Side Car Project' \
--description='Side Car Project'
```



New Application

Openshift

Laurent Valevre

The application

It's time to create our application

```
$ oc new-app https://gitlab.forge.orange-labs.fr/
laov6410/cdnselect.git --name sidecar
$ oc set build-secret --source bc/sidecar gitlab-se
```

- \$ oc expose service sidecar
- \$ oc get all name --selector app=sidecar



Item To Modify

Openshift

Laurent Valevre

Secret Access

Deployment and Service

2 parts will be modified to adapted to our application

- DeploymentConfig
- Service



DeploymentConfig

Openshift

Laurent Valeyre

Containers
Understanding Pod
Main Ways

Use case

Dockerfile Secret Access

The application

Deployment and

Service

Prometheus grafana Grafana

Deploy Jenkins and Our Pipeline Definition

We edit *DeploymentConfig*

```
$ oc edit dc/sidecar
```

and we add

```
spec:
  containers:
  - name: apache-exporter
   image: previousnext/apache-exporter
   command: [ "apache_exporter", "-scrape_uri", \
    "http://127.0.0.1:8080/server-status/?auto" ]
   ports:
   - containerPort: 9117
```



Service

Openshift

Laurent Valeyre

Understanding Po Main Ways

Apache Status

Secret Access

The application

Deployment and

Service

Prometheus grafana

Deploy Jenkins and Our Pipeline Definition

We edit service

\$ oc edit svc/sidecar

spec:

. . .

- name: 9117-tcp

port: 9117

protocol: TCP

targetPort: 9117

9117 is The port related to exporter apache



Finally

Openshift

Laurent Valeyre

Containers

Understanding Pool

Apache Status

Dockerfile

The application
Deployment and
Service

Promether grafana

Deploy Jenkins an Our Pipeline Definition

Finally we create our new application from this yaml file

```
$ oc get svc --selector "app=selector"

NAME TYPE CLUSTER-IP PORT(S)

selector ClusterIP 172.30.127.98 8080/TCP,911

$ oc describe dc/sidecar
```

Et voila...



Pull Image

Openshift

Laurent Valevre

We pull the image for *Prometheus* from the public Docker Hub registry.

- \$ oc new-app prom/prometheus
- \$ oc new-app grafana/grafana
- oc expose service prometheus
- \$ oc expose service grafana



Image and configuration decoupled

Openshift

Laurent Valevre

Prometheus

```
qlobal:
  scrape interval:
                        5.5
  evaluation interval: 5s
scrape configs:
  - job name: 'apache-exporter'
    scheme: http
    static configs:
    - targets: ['faye:9117']
      labels: {'host': 'cOmerade'}
```



Prometheus Configmap

Openshift

Laurent Valevre

Prometheus

We create the *configmap* including the prometheus configuration and finally edit the deployment config prometheus

```
$ oc create configmap prom-config \
--from-file=prometheus.yml
```

\$ oc edit dc/prometheus



DeploymentConfig and ConfigMap

Openshift

Laurent Valevre

Prometheus

Add these 2 blocks of code

name: prom-config-volume

confiqMap:

name: prom-config defaultMode: 420

name: prom-config-volume mountPath: /etc/prometheus/



Grafana Configmap

Openshift

Laurent Valeyre

Container

Understanding Pod Main Ways

Apache Statu

Dockerfile Secret Access

The application Deployment and

Prometheus

grafana

Deploy Jenkins at Our Pipeline Definition We create the *configmap* including the grafana configuration

```
$ oc create configmap grafana-config \
--from-file=grafana.ini
```

\$ oc edit dc/grafana



DeploymentConfig and ConfigMap

Openshift

Laurent Valeyre

Containers

Main Ways

Apache Statu

Secret Access

The application
Deployment and

Prometheus

grafana Grafana

Deploy Jenkins and Our Pipeline Definition

Modification in spec-containers

```
- image:
    ...
    volumeMounts:
    - name: grafana-config
        mountPath: /etc/grafana/
    ...
volumes:
- name: grafana-config
    configMap:
```

name: grafana-config

defaultMode: 420



Openshift

Laurent Valeyre

Containers

Understanding

Main Ways

Patterns

Apache Stati

Dockerfile Secret Access

The application

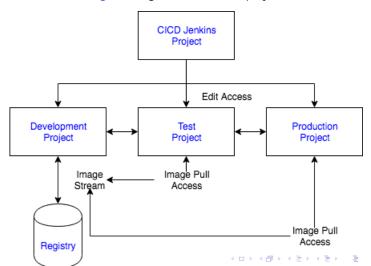
Promethe grafana

Pipeline and

Deploy Jenkins at Our Pipeline

Deploy Our Sample

Figure: diagram of different projects





Openshift

Laurent Valevre

- Project CICD Containing our Jenkins instance
- Development For building and developing our application images
- Testing For testing our application
- Production Hosting our production application



Create Projects

Openshift

Laurent Valeyre

Containers
Understanding Pod

Main Ways Patterns

Apache Status
Dockerfile
Secret Access

The application Deployment and Service

Prometho

Pipeline and

Deploy Jenkins and Our Pipeline Definition

```
$ oc login -u developer -p developer

$ oc new-project cicd --display-name='CICD Jenkins --description='CICD Jenkins'

$ oc new-project development \ --display-name='Development' --description='Development' soc new-project testing --display-name='Testing' --description='Testing'

$ oc new-project production --display-name='Production-description='Production'
```



RBAC

Openshift

Laurent Valeyre

Container

Understanding Pod Main Ways

Patterns

Apache Stati

Dockerfile Secret Access

The application
Deployment and

Promet

grafana Grafana

Pipeline and

Deploy Jenkins an Our Pipeline Definition \$ oc policy add-role-to-user edit \
system:serviceaccount:cicd:jenkins -n development
\$ oc policy add-role-to-user edit \
system:serviceaccount:cicd:jenkins -n testing
\$ oc policy add-role-to-user edit \
system:serviceaccount:cicd:jenkins -n production



RBAC

Openshift

Laurent Valeyre

Containers

Understanding Pool

Main Ways Patterns

Apache Stat

Dockerfile Secret Access

The application
Deployment and

The Mor

> Promethe grafana Grafana

Pipeline and

Deploy Jenkins as Our Pipeline Definition

Definition Deploy Our Samp \$ oc policy add-role-to-group system:image-puller system:serviceaccounts:testing -n development \$ oc policy add-role-to-group system:image-puller system:serviceaccounts:production -n development



Openshift

Laurent Valeyre

Containers

Main Ways

Patterns

Apache Status Dockerfile

Secret Access The applicatio

Prometheus grafana

grafana Grafana

Deploy Jenkins and Our Pipeline Definition

Deploy a Jenkins ephemeral instance in cicd project

```
$ oc project cicd
```

\$ oc new-app --template=openshift/jenkins-persister

\$ oc status

Let's create the pipeline itself.

\$ oc create -n cicd -f \

https://raw.githubusercontent.com/devops-with-open./pipeline-configs/master/pipeline.yaml



Openshift

Laurent Valevre

Deploy Our Sample

```
$ oc project development
```

```
$ oc create new-app --name=cdnapi \
https://github.com/gandalf-the-white/faye.git
```

\$ oc expose svc/cdnapi



Openshift

Laurent Valevre

```
Deploy Our Sample
```

```
$ oc project testing
$ oc create dc cdnapi \
```

```
--image=172.30.1.1:5000/development/cdnapi:promoted
```

```
$ oc rollout cancel dc/cdnapi
```



Patch

Openshift

Laurent Valevre

Deploy Our Sample

```
$ oc patch dc/cdnapi -p \
'{"spec":{"template":\
    {"spec":{"containers":\
        [{"name":"default-
         container", "imagePullPolicy": "Always"
        $ oc rollout cancel dc/cdnapi
```



Expose

Openshift

Laurent Valeyre

Secret Access

The application

Deploy Our Sample

\$ oc expose dc cdnapi --port=8080

\$ oc expose svc/cdnapi



Openshift

Laurent Valevre

Deploy Our Sample

```
$ oc project production
$ oc create dc cdnapi \
--image=172.30.1.1:5000/development/cdnapi:promote
$ oc rollout cancel dc/cdnapi
```



Patch

Openshift

Laurent Valevre

```
$ oc patch dc/cdnapi -p \
'{"spec":{"template":\
    {"spec":{"containers":\
        [{"name":"default-
         container", "imagePullPolicy": "Always"
        $ oc rollout cancel dc/cdnapi
```



Expose

Openshift

Laurent Valeyre

Container

Containers

Understanding Po

Main Ways

Pattern:

Use casi

Apache Stat

Secret Access

The application Deployment and

The

Prometh

grafana Grafana

Deploy Jenkins a

Deploy Jenkins and Our Pipeline Definition Deploy Our Sample

```
$ oc expose dc cdnapi --port=8080
```

\$ oc expose svc/cdnapi



links

Openshift

Laurent Valeyre

Understanding F

Main Ways Patterns

Apache Status

Dockerfile Secret Access

The application
Deployment and
Service

Promethe

grafana Grafana

Deploy Jenkins a

Definition

Deploy Our Sample

```
https://kubernetes.io/blog/2015/06/
the-distributed-system-toolkit-patterns/ https:
//www.robustperception.io/openshift-and-prometheus
http://widerin.net/blog/
official-grafana-docker-image-on-openshift/
```