



{KODE}{KLOUD



Red Hat

OPENS SHIFT

for the absolute beginners



I am an IT Solutions Architect focusing on Cloud automation and DevOps. I am passionate about learning new technology and teaching. I believe the best way to learn is to learn by doing and in a fun way. I have authored multiple courses on DevOps and cloud automation technologies. My courses focus on providing students with an interactive and hands-on experience in learning new technology that makes learning really interesting.

Total Students	Courses	Reviews
23,700	7	3,057

Courses you are teaching

HIGHEST RATED

Kubernetes for the Absolute Beginners - Hands-On

Mumshad Mannambeth

★★★★★ 4.5 (161)

HIGHEST RATED

Docker Advanced - SWARM - Hands-on

Mumshad Mannambeth

★★★★★ 4.6 (139)

BEST SELLER

Docker for the Absolute Beginner - Hands On

Mumshad Mannambeth

★★★★★ 4.6 (1,013)

Ansible for the Absolute Beginner - Hands-On

Mumshad Mannambeth

★★★★★ 4.5 (1,423)

About me!

Course Structure

Lecture

Demos

Quiz

Coding Exercises

Assignment

Q&A

Who is this for?

Developers

System Admins

Managers

How to attend this course?

			
 Lectures	✓	✓	✓
 Demos	✓	✓	
 Quizzes	✓		
 Coding Exercises	✓		
 Lab Environment	Local	AWS GCP	Play-with-k8s

Objectives

- Introduction
- Architectural Overview
- Setup - Minishift
- Management – Web, CLI, API
- Projects and Users
- Builds and Image Streams
- Build Triggers
- Deployments
- Networking
- Services and Routes
- Scaling
- Storage
- Catalog and Templates

- Microservices Application
- Example Voting Application Deployment in Openshift

- Pre-Requisite – Containers
- Pre-Requisite – Kubernetes
- Pre-Requisite – Source Code Management
- Pre-Requisite – Builds and CI/CD
- Pre-Requisite – YAML



{KODE{KLOUD

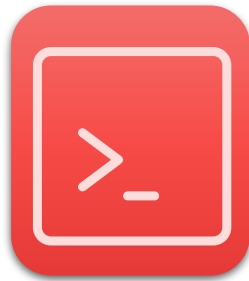


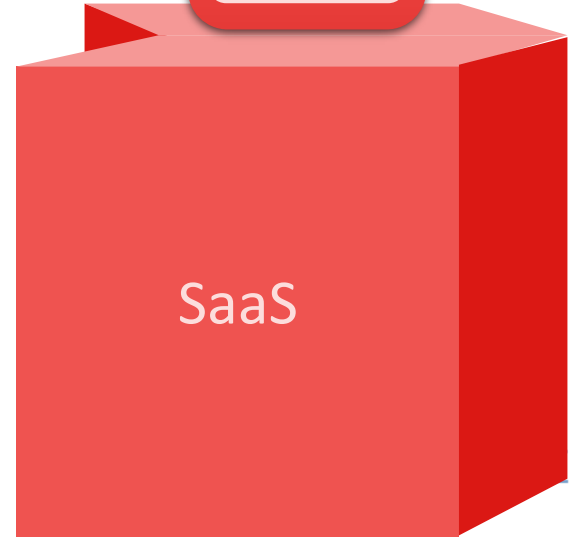
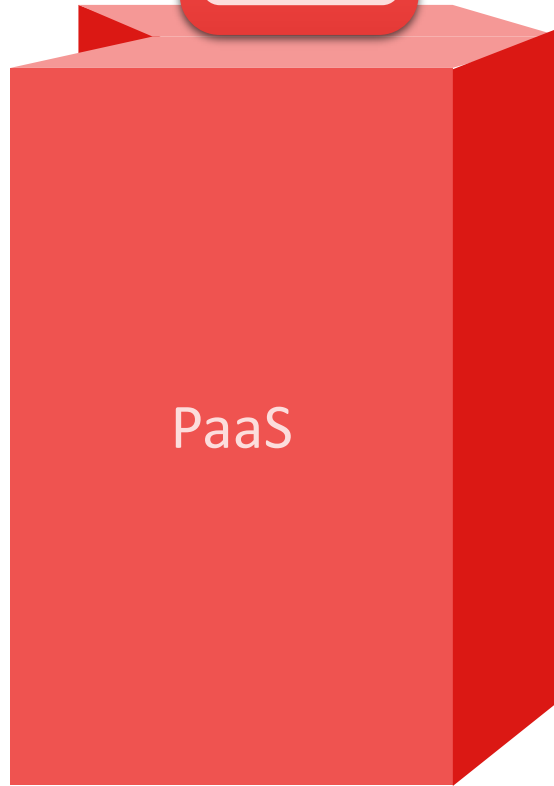
Red Hat

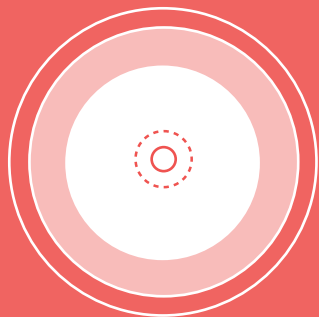
OPENS SHIFT

Introduction









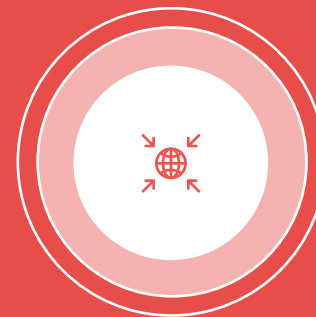
OPENSHIFT
Origin

Open source application
container platform



OPENSHIFT
Online

Public Application
Development hosting service



OPENSHIFT
Dedicated

Managed private cluster
on AWS/Google Clouds



OPENSHIFT
Enterprise

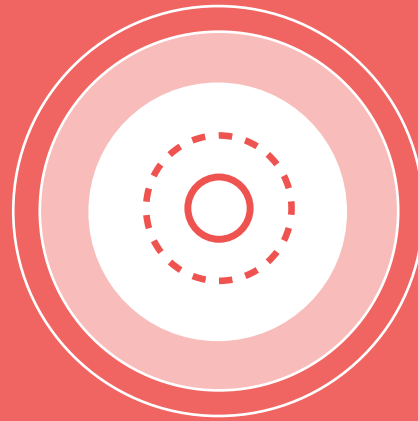
On-Premise private
PaaS



OPENSIFT
Origin



Open source application
container platform



OPENSIFT
Origin

Open source application
container platform

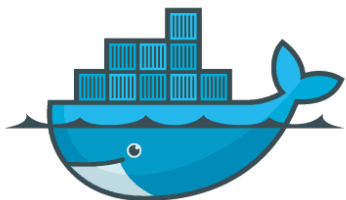
OPENSIFT Origin is based on top of **Docker** containers and the **Kubernetes** cluster manager, with added developer and operational centric **Tools** that enable rapid application development, deployment and lifecycle management.



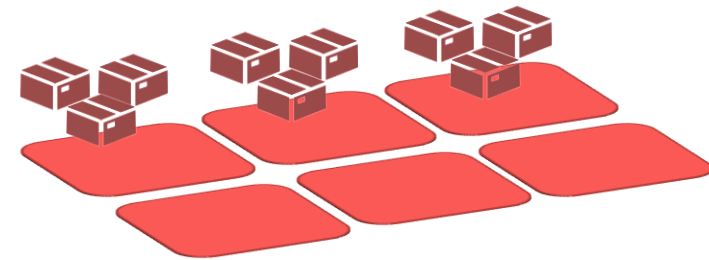
Tools



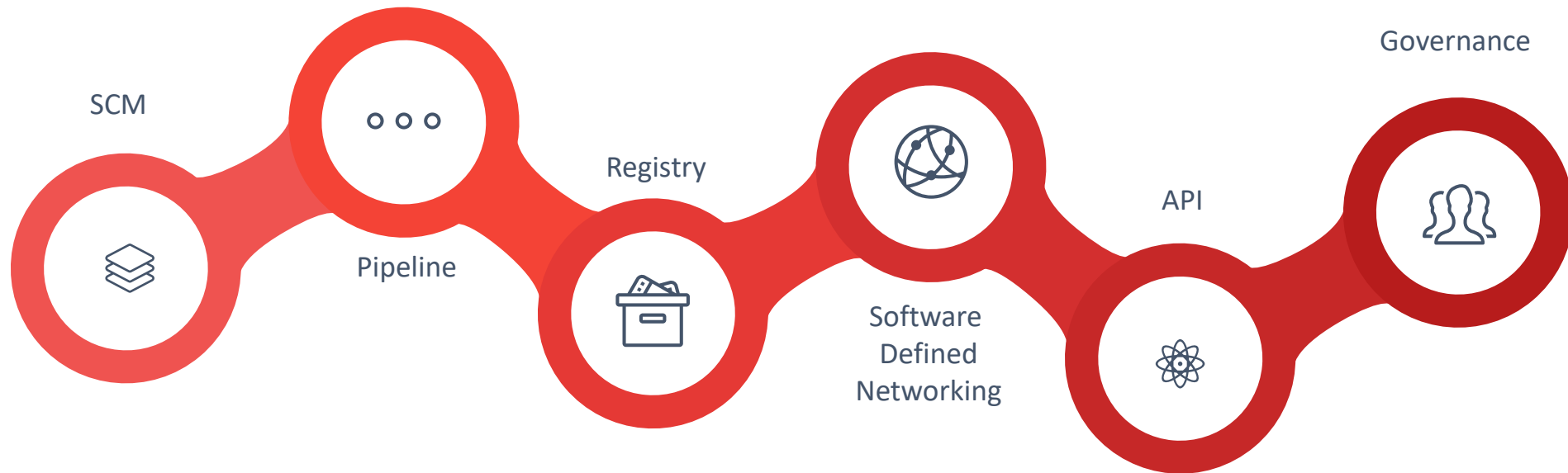
Kubernetes



Docker



Tools





HIGHEST RATED

Kubernetes for the Absolute Beginners -...

Mumshad Mannambeth

★★★★★ 4.5 (136)



HIGHEST RATED

Docker Advanced - SWARM - Hands-on

Mumshad Mannambeth

★★★★★ 4.6 (136)



BEST SELLER

Docker for the Absolute Beginner - Hands On

Mumshad Mannambeth

★★★★★ 4.6 (968)



{KODE{KLOUD



{KODE {KLOUD

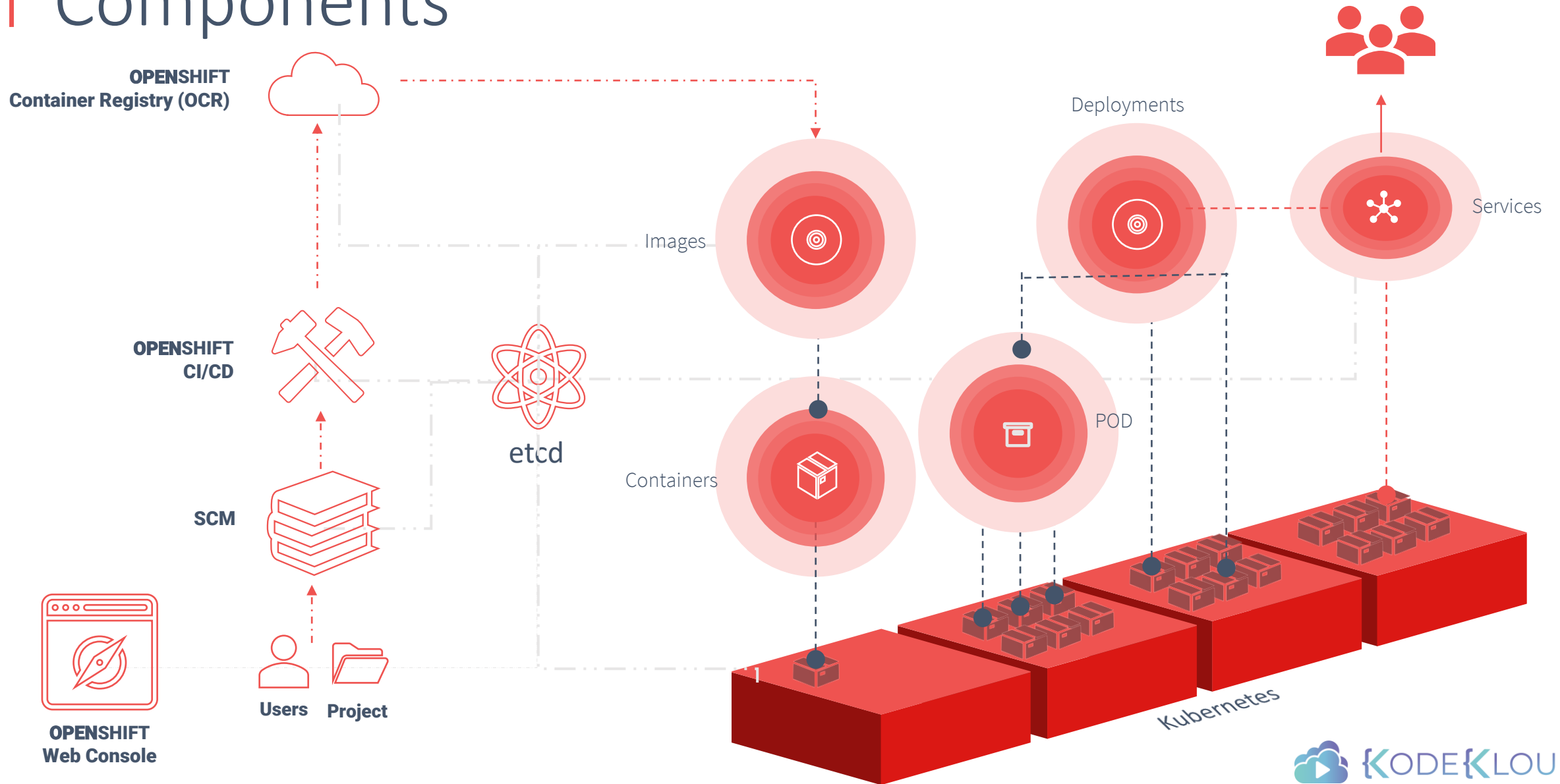


Red Hat

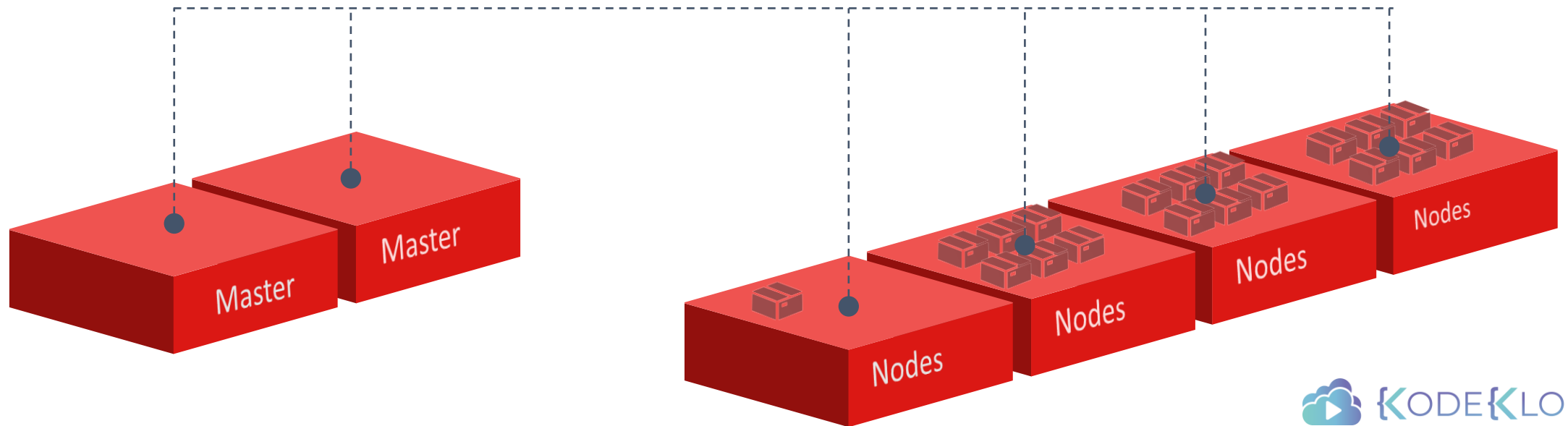
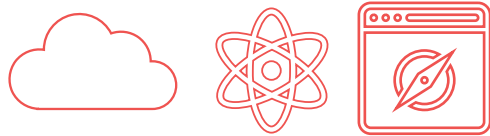
OPENSIFT

Architecture

Components



Master - Node





{KODE}{KLOUD



Red Hat

OPENSIFT

Setup - Minishift



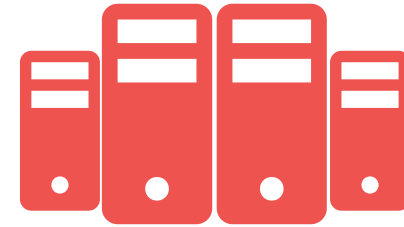
Setup OPENS SHIFT



All-in-One



Single Master
Multiple Nodes



Multiple Master
Multiple Nodes



On-Premise



Cloud



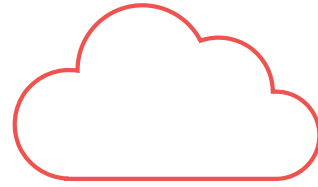
Package Manager (RPM)



Containerized (Docker)



All-in-One



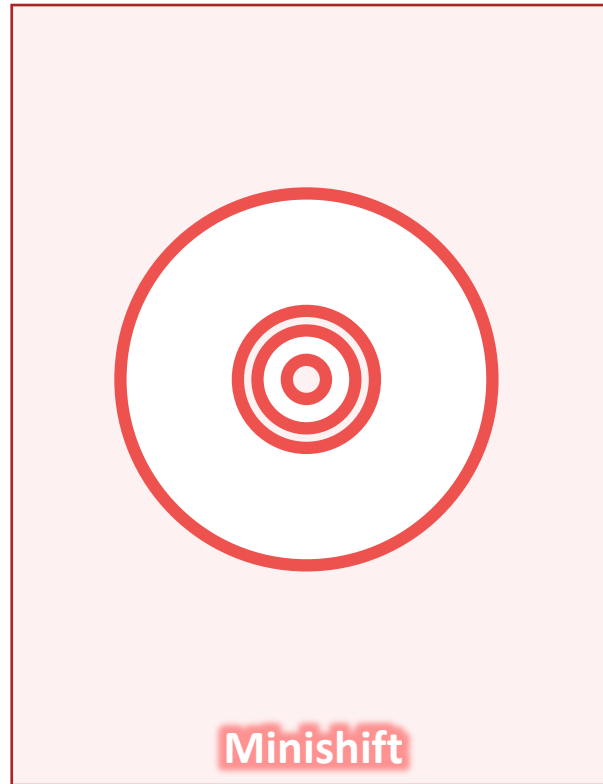
Kubernetes

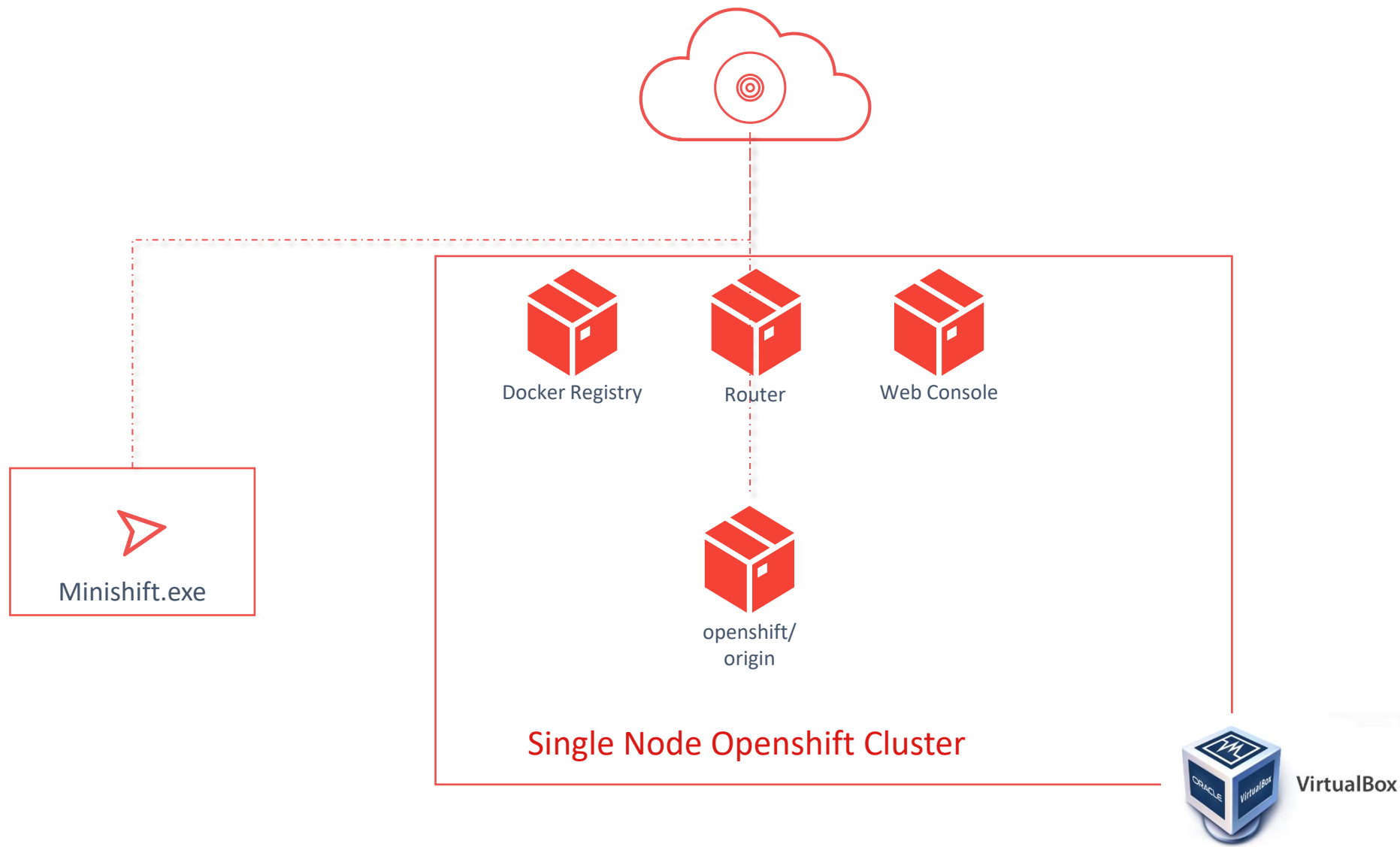


etcd



OpenShift







{KODE {KLOUD



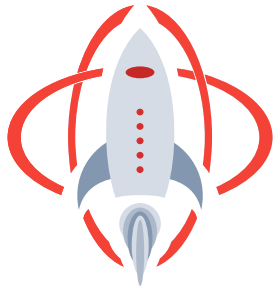
Red Hat

OPENSSHIFT

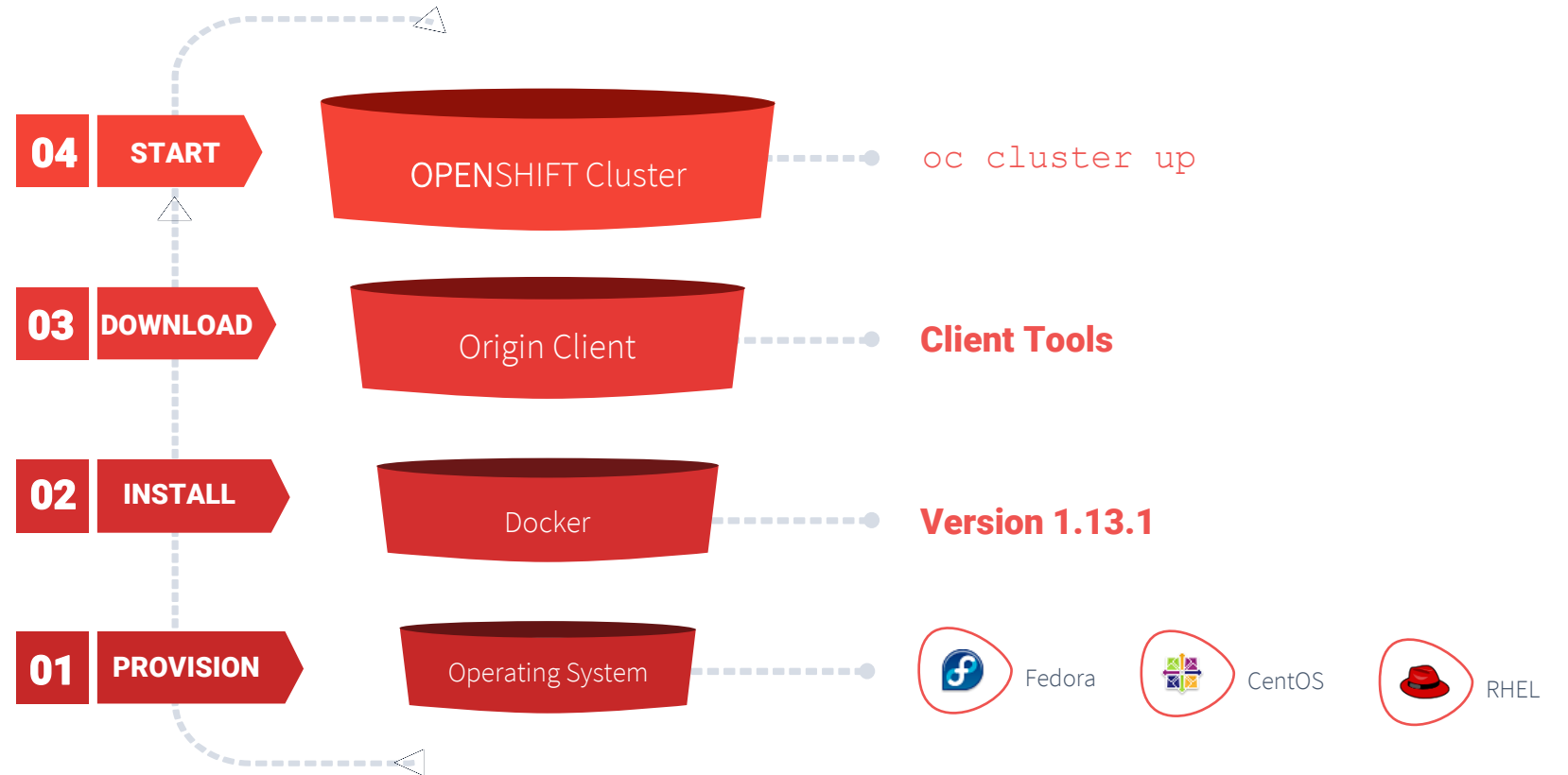
Setup – Using Docker



All-in-One



without
Minishift



Containerized (Docker)



KODEKLOUD



{KODE{KLOUD



Red Hat

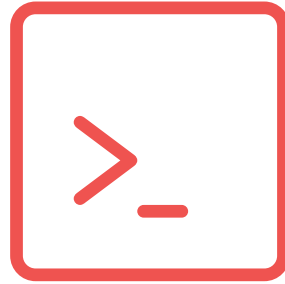
OPENSHIFT

Web Console and CLI

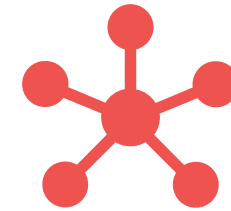
Management Tools



Web Console



CLI



REST API



Web Console - Homepage

OPENSIFT ORIGIN ? mmumshad

Search Catalog

Browse Catalog

Deploy Image | Import YAML / JSON | Select from Project

All | Languages | Databases | Middleware | CI/CD | Other

Filter ▾ 20 Items

.NET .NET Core Builder Images	Apache HTTP Server (httpd)	php CakePHP + MySQL	Dancer + MySQL	Django + PostgreSQL
Jenkins	MariaDB	MongoDB	MySQL	Nginx HTTP server and a reverse proxy (nginx)
Node.js	Node.js + MongoDB	Perl	PHP	Pipeline Build Example

My Projects

[+ Create Project](#)

5 of 8 Projects [View All](#)

- [kube-system](#)
created a day ago
- [openshift-web-console](#)
created a day ago
- [openshift](#)
created a day ago
- [default](#)
created a day ago
- [kube-public](#)
created a day ago



Web Console - Project View

OPENSIFT ORIGIN mmumshad

openshift-web-console

Name List by

APPLICATION
openshift-web-console

DEPLOYMENT
webconsole, #1

CONTAINERS

webconsole
Image: openshift/origin-web-console:v3.9.0
Ports: 8443/TCP

NETWORKING

Service - Internal Traffic
webconsole
443/TCP (https) → 8443

Routes - External Traffic

1 pod

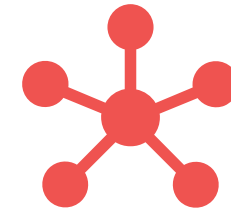
Management Tools



Web Console



CLI



REST API



CLI

```
> oc
```

OpenShift Client

This client helps you develop, build, deploy, and run your applications on any OpenShift or Kubernetes compatible platform. It also includes the administrative commands for managing a cluster under the 'adm' subcommand.

To create a new application, login to your server and then run new-app:

```
oc login https://mycluster.mycompany.com
oc new-app centos/ruby-22-centos7~https://github.com/openshift/ruby-ex.git
oc logs -f bc/ruby-ex
```



CLI - Login

```
> oc login
```

```
OpenShift server [https://localhost:8443]: https://openshift.example.com
```

```
Username: developer
```

```
Authentication required for https://openshift.example.com (openshift)
```

```
Password: *****
```

```
Login successful.
```

```
> oc login -u developer -p developer
```

```
Login successful.
```

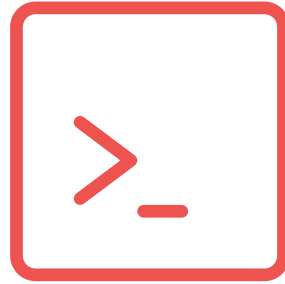
```
> oc logout
```

```
User, developer, logged out of https://openshift.example.com
```

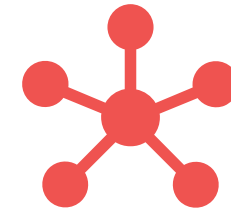

Management Tools



Web Console



CLI



REST API



REST API

```
> curl https://localhost:8443/oapi/v1/users \  
-H "Authorization: Bearer <Token>"
```

```
"kind": "UserList",  
"apiVersion": "v1",  
"metadata": {  
  "selfLink": "/oapi/v1/users",  
},  
"items": [  
  {  
    "metadata": {  
      "name": "developer",  
      "selfLink": "/oapi/v1/users/developer",  
      "uid": "271e2b49-47f0-11e8-afb8-4a1a95a6dbc1",  
      "resourceVersion": "1289",  
      "creationTimestamp": "2018-04-24T18:49:00Z"  
    },  
    "identities": [  
      "anypassword:developer"  
    ],  
    "groups": null  
  },  
  {  
    "metadata": {  
      "name": "mmumshad",  
      "selfLink": "/oapi/v1/users/mmumshad",  
      "uid": "c905343d-48a6-11e8-afb8-4a1a95a6dbc1",  
      "resourceVersion": "108365",  
      "creationTimestamp": "2018-04-25T16:36:20Z"  
    },  
    "identities": [  
      "anypassword:mmumshad"  
    ],  
    "groups": null  
  },  
]
```

```
> oc whoami -t
```

```
27pZjHw9GU5B5De4H3IC26ZBd0JtzQJMyz79
```



{KODE}{KLOUD

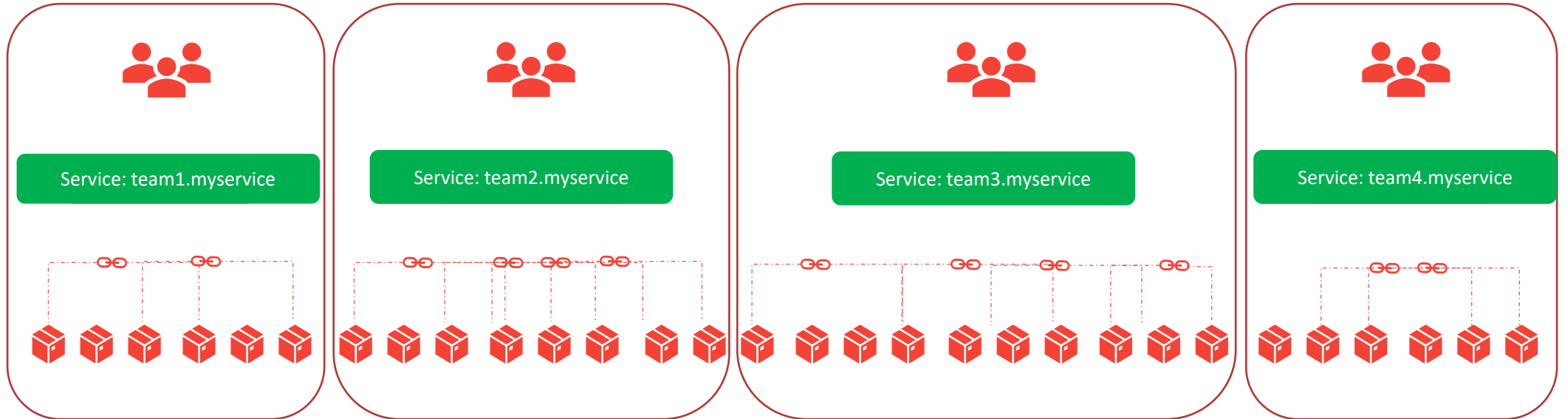


Red Hat

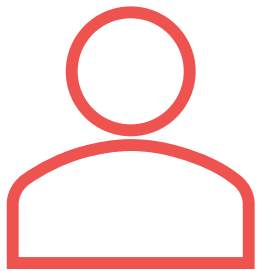
OPENS SHIFT

Projects and Users

Projects



Users



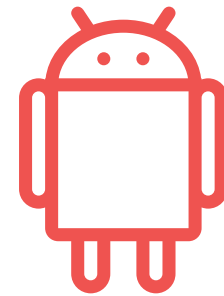
Regular

developer



System

system:admin
system:master



Service

system:serviceaccount:proj1:db_user

OAuth Server



Allow All



Deny All

`/etc/openshift/master/master-config.yaml`



{KODE}{KLOUD

Red Hat **OPENS**HIFT

Pre-Requirement

Source Code Management

Source Code Management



Application



Web Console



API Backend



Source Code Management System



Developers



```
|-- README.md
|-- Dockerfile
|-- bower.json
|-- package.json
|-- app
  |-- app.js ← [User]
  |-- web
    |-- index.html ← [User]
    |-- mmdir.js
    |-- routing.js
    |-- server.js
  |-- api ← [User]
    |-- api.groups.js
    |-- api.posts.js
    |-- api.users.js
    |-- api.widgets.js
    |-- authentication ← [User]
      |-- oauth.js
      |-- vendor
  |-- database ← [User]
    |-- db.js
  |-- integration ← [User]
    |-- servicenow.js
```

Source Code Management

Revision History

File Merging



Anal

```
1 client/app/main/main.html
@@ -18,6 +18,7 @@
18 <div class="container"
19
20 <div class="row" st
21
22 <div class="col-r
repeat="video in $ctr
22 <a href="" ng-c
style="color: grey">
23 <div style="c

2 server/config/envi
@@ -37,7 +37,7 @@ exp
37 {
38   title: 'Overview
39   type: 'overview'.
40 -   video_id: '6sE0G
41 },
42 {
43   title: 'Getting Started',
```

Code Pull requests 161 Projects 0 Insights

Releases Tags

- 3 days ago rel/release-3.0.2 ...
5c141f7 zip tar.gz
- 9 days ago release-2.9.1-RC0 ...
e30710a zip tar.gz
- 8 days ago rel/release-2.7.6 ...
085099c zip tar.gz
- 9 days ago release-3.0.2-RC1 ...
5c141f7 zip tar.gz
- 16 days ago release-2.7.6-RC0 ...
d4edbac zip tar.gz
- 20 days ago release-3.0.2-RC0 ...
5c141f7 zip tar.gz
- 21 days ago rel/release-3.1.0 ...
16b7061 zip tar.gz



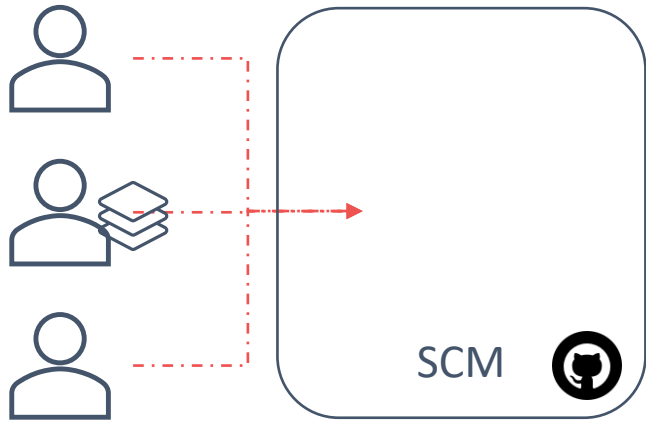
{KODE}{KLOUD

Red Hat **OPENS**HIFT


Pre-Requirement

CI/CD - Builds and Pipeline


Build



Java




Maven



app.jar

```
> java -jar app.jar
```

Python




DistUtils




app.tar.gz

```
> pip install app.tar.gz
```

Ruby



Bundler



app.gem

```
> gem install app.gem
```

Build

Java



Maven



app.jar

```
> java -jar app.jar
```

Python



DistUtils



app.tar.gz

```
> pip install app.tar.gz
```

Ruby



Bundler



app.gem

```
> gem install app.gem
```



Dockerfile

01

Install Platform

```
> docker build Dockerfile
```

03

Install and Configure Pre-requisites



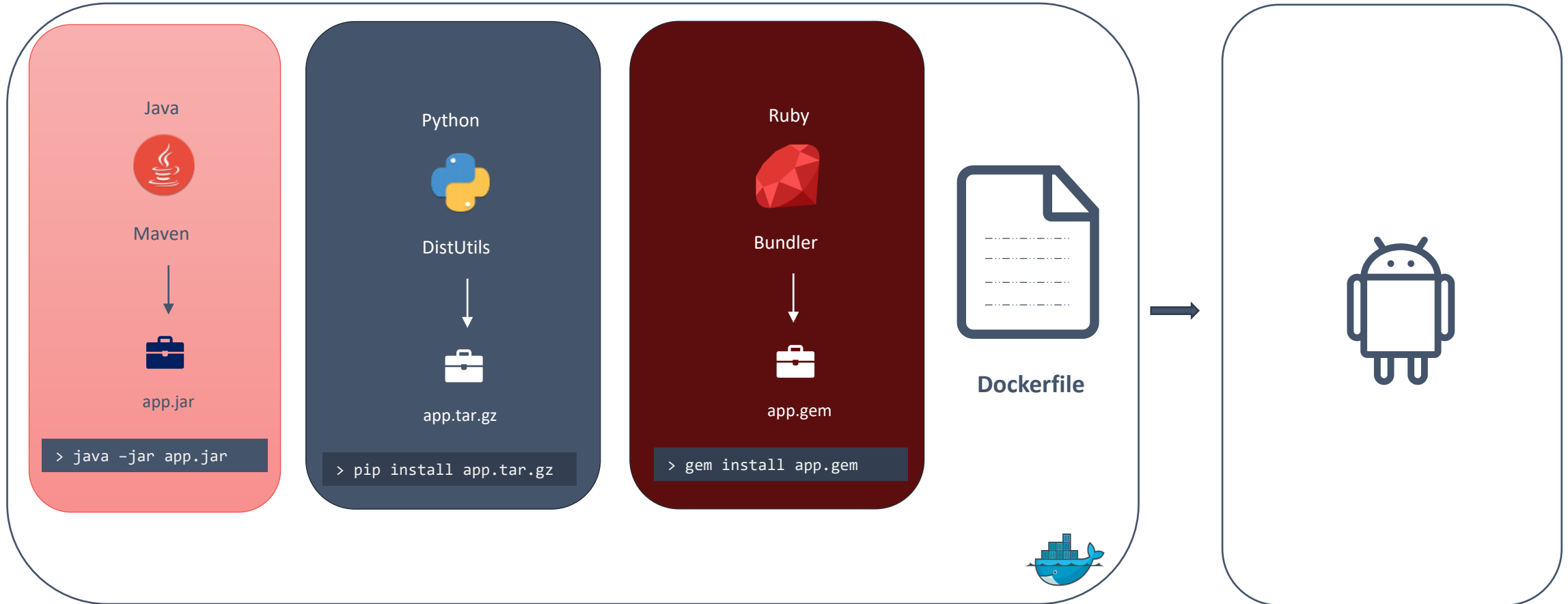
Configure Services

04

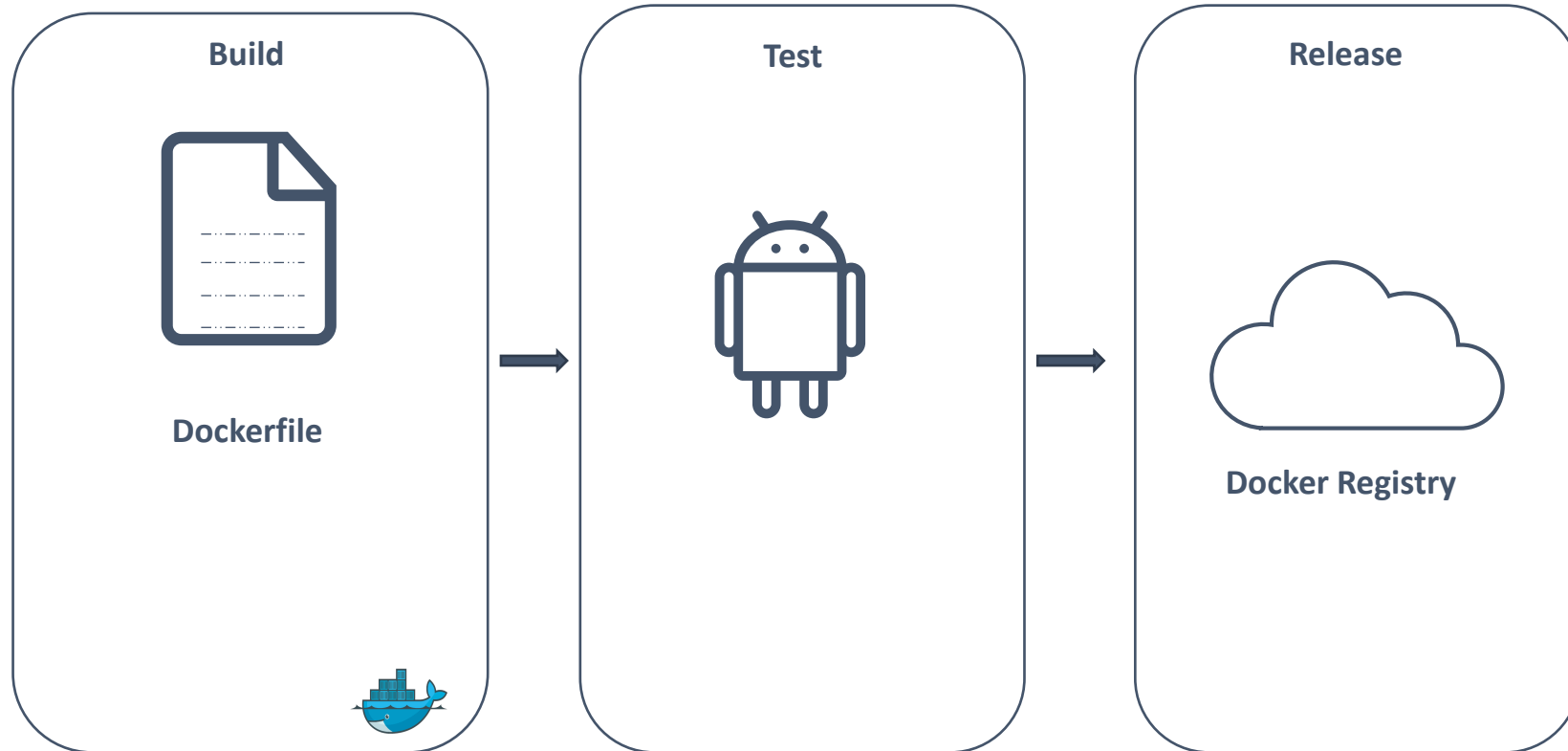
Start Services

```
> docker run app
```

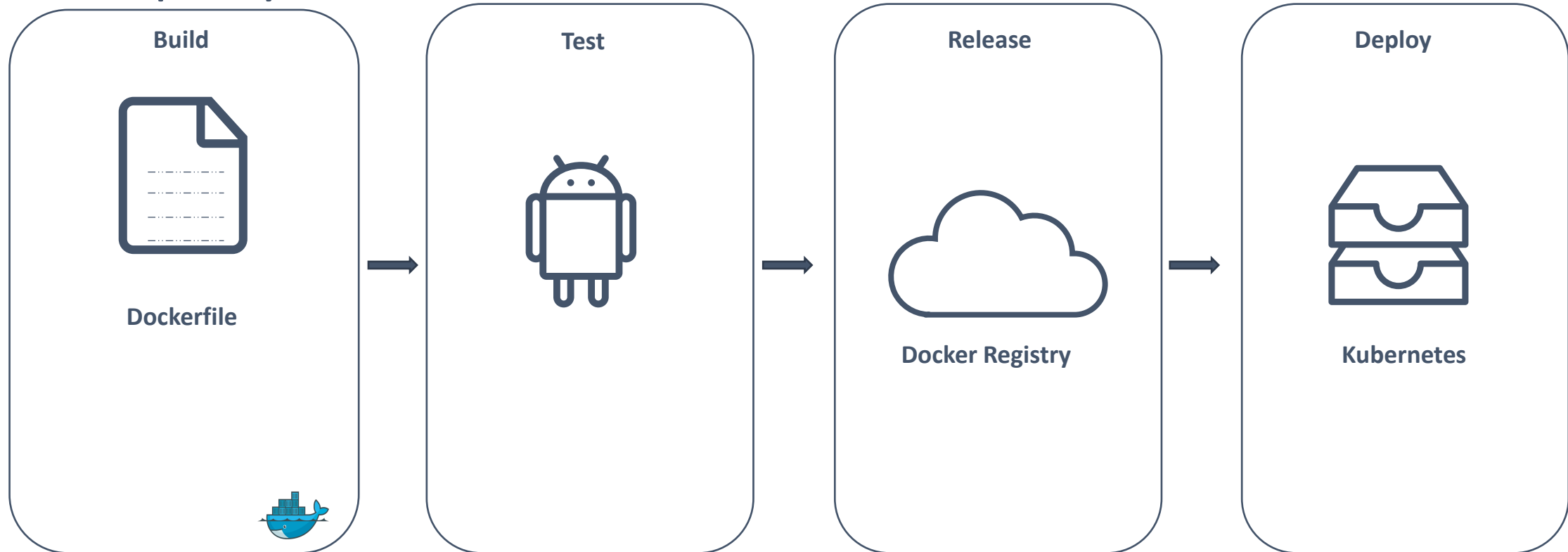
Test



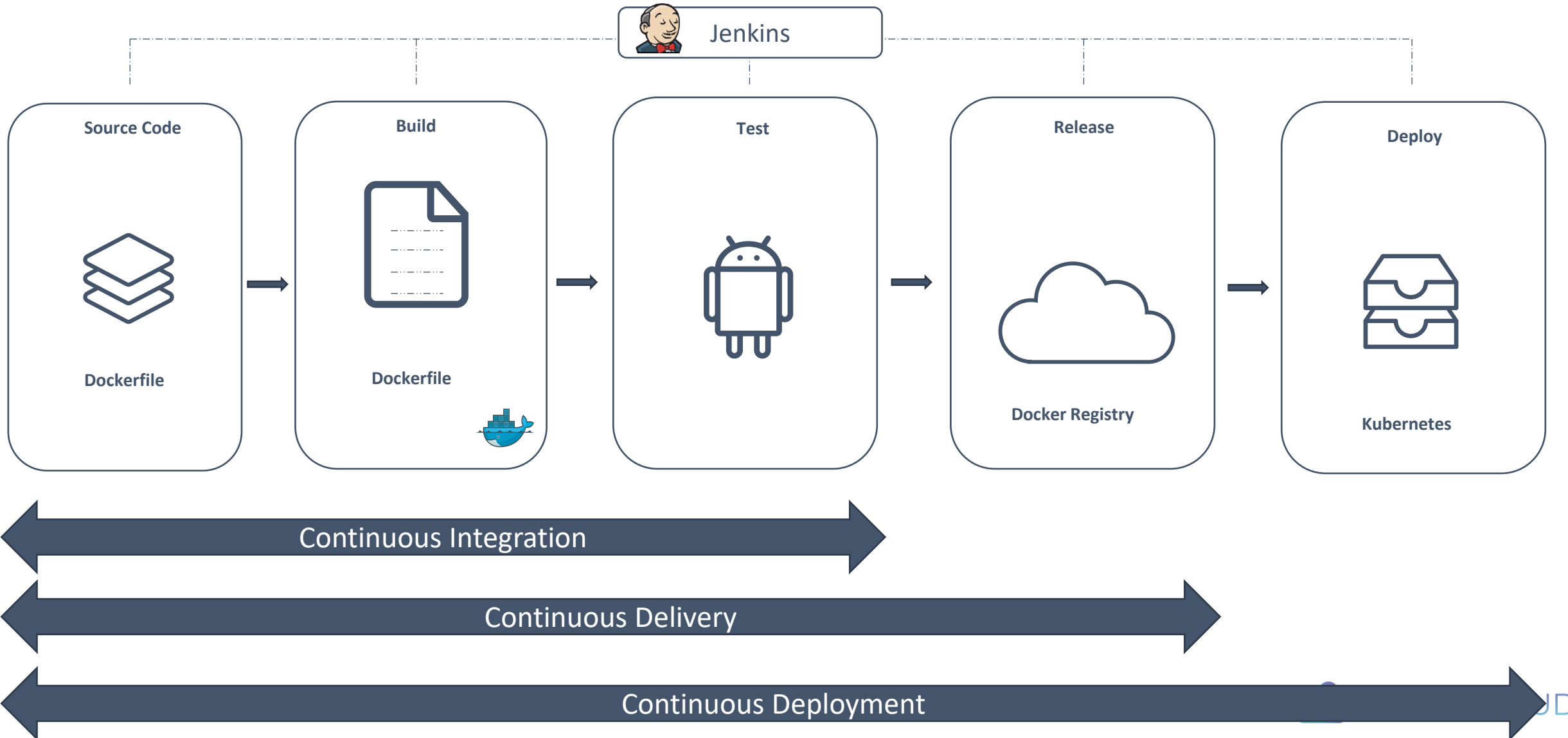
Release



Deploy



Build Pipeline





{KODE{KLOUD



Red Hat

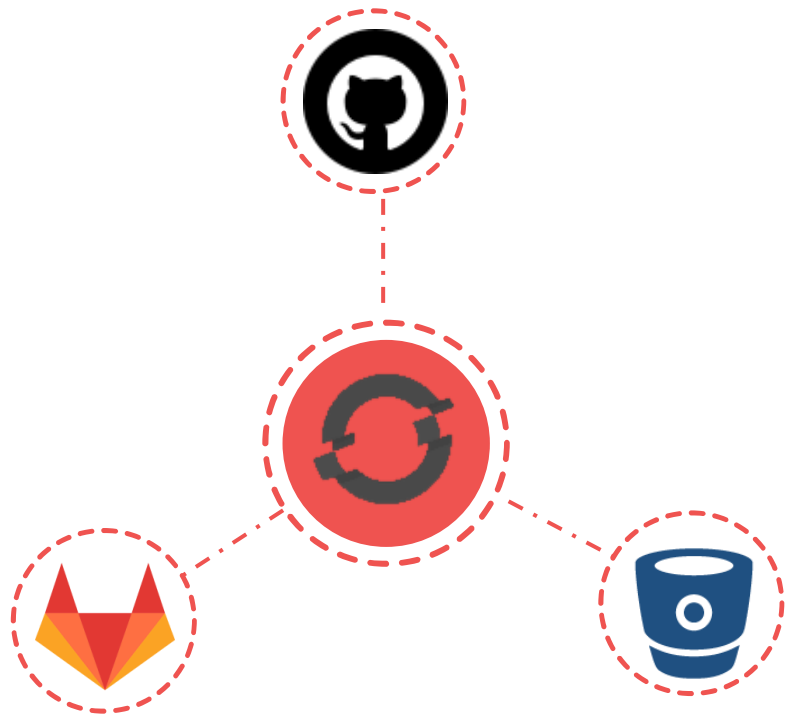
OPENSIFT

SCM, Builds and Deployments in **OPENSIFT**

Pre-Requisites

- Docker
- Docker Image
- Docker Registry

Source Code Management



Python

Information Configuration Results

1 2 3

Version

3.6 — latest

* Application Name

* Git Repository

`https://github.com/openshift/django-ex.git`

[Try Sample Repository ↕](#)

If you have a private Git repository or need to change application defaults, view [advanced options](#).

Cancel < Back Create

Build

<https://github.com/openshift/django-ex.git>

01 Create Build



02 Download Source



03 Build Image



04 Push to Registry



Docker Registry

The screenshot shows the OpenShift Origin interface for a 'test-project'. The 'Builds' section is active, displaying a table with one build named 'django-ex' with status 'Complete'. Below the table, the 'Logs' tab is selected, showing a log of the build process. The log includes cloning the source, installing dependencies, and pushing the image to the registry.

Name	Last Build	Status
django-ex	#2	✓ Complete

```
1 Cloning "https://github.com/openshift/django-ex.git" ...
2 Commit: 37f7fc41432b9c07265c5896a4fb226caa870427 (Merge pull request #115 from hhorak/python-3.6)
3 Author: Honza Horak <hhorak@redhat.com>
4 Date: Tue Apr 24 18:48:25 2018 +0200
5 ---> Installing application source ...
6 ---> Installing dependencies ...

21 Installing collected packages: pytz, django, sqlparse, django-debug-toolbar, gunicorn, psycogp2, whitenoise
22 Successfully installed django-1.11.12 django-debug-toolbar-1.8 gunicorn-19.4.5 psycogp2-2.7.3.1 pytz-2018.4 sqlparse-0.2.4 whitenoise-3.3.1
23 You are using pip version 9.0.1, however version 10.0.1 is available.
24 You should consider upgrading via the 'pip install --upgrade pip' command.
25 ---> Collecting Django static files ...

205 Pushing image 172.30.1.1:5000/test-project/django-ex:latest ...
206 Pushed 0/10 layers, 10% complete
207 Pushed 1/10 layers, 11% complete
208 Pushed 2/10 layers, 21% complete
215 Pushed 9/10 layers, 97% complete
216 Pushed 10/10 layers, 100% complete
217 Push successful
```


Deployment

04 Push to Registry



Docker Registry

05 Deploy

test-project

Deployments [Learn More](#)

Filter by label

Name	Last Version	Status
django-ex	#1	Active, 1 replica

```
apiVersion: apps/v1  
kind: Deployment
```



```
apiVersion: apps.openshift.io/v1  
kind: DeploymentConfig
```



Deployment

<https://github.com/openshift/django-ex.git>

01 Create Build



02 Download Source



03 Build Image



04 Push to Registry



Docker Registry

05 Deploy



Python

Information Configuration Results

1 2 3

Version

3.6 — latest

* Application Name

* Git Repository

<https://github.com/openshift/django-ex.git>

[Try Sample Repository ↕](#)

If you have a private Git repository or need to change application defaults, view [advanced options](#).

Cancel < Back Create



{KODE{KLOUD



Red Hat

OPENSIFT

Builds in **OPENSIFT**

Objectives

- Build Strategies
- Create a new Build Configuration

Pre-Requisites

- YAML Files

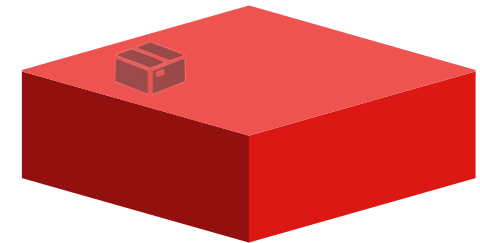
Build Strategy

1 Docker Build

app.py

Dockerfile

```
1 FROM ubuntu:16.04
2
3 RUN apt-get update && apt-get install -y python python-pip
4
5 RUN pip install flask
6
7 COPY app.py /opt/
8
9 ENTRYPOINT FLASK_APP=/opt/app.py flask run --host=0.0.0.0
10 def hello():
11     return 'I am good, how about you?'
12
13 if __name__ == "__main__":
14     app.run(host="0.0.0.0", port=8080)
```



Build Strategy

2 Source-To-Image (S2I)

app.py

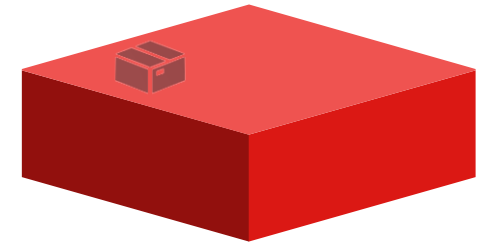
```
1 import os
2 from flask import Flask
3 app = Flask(__name__)
4
5 @app.route("/")
6 def main():
7     return "Welcome!"
8
9 @app.route('/how are you')
10 def hello():
11     return 'I am good, how about you?'
12
13 if __name__ == "__main__":
14     app.run(host="0.0.0.0", port=8080)
```



Python Image



App Image



Build Strategy

3

Custom Build

Java



app.jar

Python



app.tar.gz

Ruby



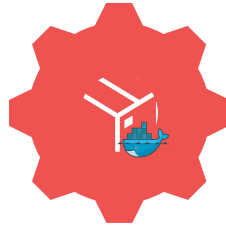
app.gem

Image Stream

Python Image

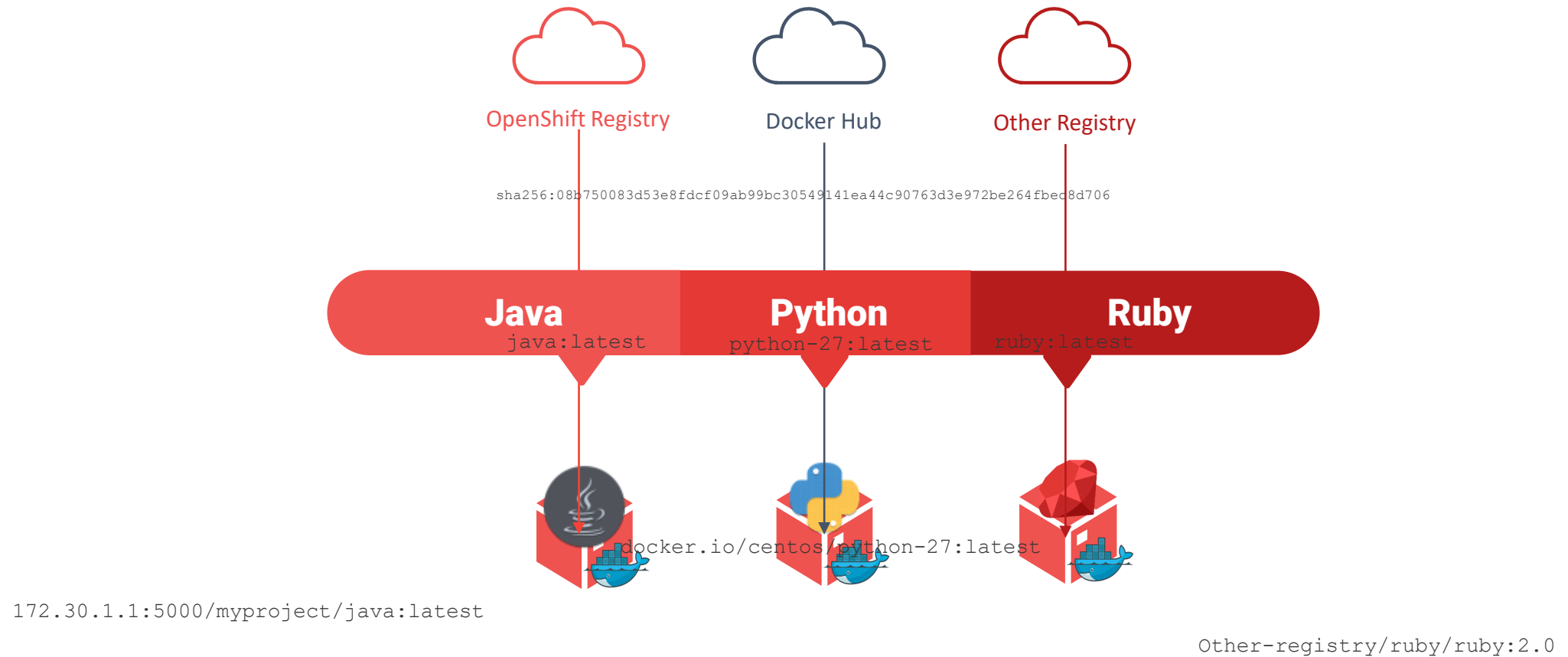


Code



Application Image

Image Streams



View Build Config

The screenshot shows the OpenShift Build Configuration page for an application named 'simple-webapp'. The page is titled 'simple-webapp' and was created 3 hours ago. It has tabs for 'app' and 'simple-webapp'. Below the tabs are sections for 'History', 'Configuration', 'Environment', and 'Events'. The 'Configuration' tab is active. Under 'Details', the following information is displayed:

- Build Strategy:** Source
- Source Repo:** <https://github.com/mmumshad/simple-webapp-flask.git>
- Source Ref:** master
- Builder Image:** openshift/python:3.6
- Output To:** [test-project/simple-webapp:latest](#)
- Run Policy:** Serial

A red circle highlights the 'Source' link in the 'Build Strategy' section, and a red dashed line connects it to the corresponding 'Source' configuration in the YAML code block on the right. Another red circle highlights the 'Actions' dropdown menu, and a red dashed line connects it to the 'Start Build' button in the screenshot.

S2i-build-config.yaml

```
kind: "BuildConfig"
apiVersion: "v1"
metadata:
  name: "simple-webapp"
spec:
  runPolicy: "Serial"
  triggers:
  -
    type: "GitHub"
    github:
      secret: "b5e471d57f79f52e"
  - type: "Generic"
    generic:
      secret: "4be5b473f9985dcf"
  -
    type: "ImageChange"
  source:
    git:
      uri: "https://github.com/mmumshad/simple-webapp-flask.git"
  strategy:
    type: Source
    sourceStrategy:
      from:
        kind: "ImageStreamTag"
        name: "python:3.6"
  output:
    to:
      kind: "ImageStreamTag"
      name: "simple-webapp:latest"
```

Create Build Configuration

S2i-build-config.yaml

```
kind: "BuildConfig"
apiVersion: "v1"
metadata:
  name: "simple-webapp"
spec:
  runPolicy: "Serial"
  triggers:
  -
    type: "GitHub"
    github:
      secret: "b5e471d57f79f52e"
  - type: "Generic"
    generic:
      secret: "4be5b473f9985dcf"
  -
    type: "ImageChange"
  source:
    git:
      uri: "https://github.com/mmumshad/simple-webapp-
flask.git"
  strategy:
    type: Source
    sourceStrategy:
      from:
        kind: "ImageStreamTag"
        name: "python:3.6"
  output:
    to:
      kind: "ImageStreamTag"
      name: "simple-webapp:latest"
```

Docker-build-config.yaml

```
kind: "BuildConfig"
apiVersion: "v1"
metadata:
  name: "simple-webapp-docker"
spec:
  runPolicy: "Serial"
  triggers:
  -
    type: "GitHub"
    github:
      secret: "b5e471d57f79f52e"
  - type: "Generic"
    generic:
      secret: "4be5b473f9985dcf"
  -
    type: "ImageChange"
  source:
    git:
      uri: "https://github.com/mmumshad/simple-webapp-docker.git"
  strategy:
    type: Docker
    dockerStrategy:
      from:
        kind: "DockerImage"
        name: "ubuntu:16.04"
  output:
    to:
      kind: "ImageStreamTag"
      name: "simple-webapp:latest"
```

Create Build Configuration

Docker-build-config.yaml

```
kind: "BuildConfig"
apiVersion: "v1"
metadata:
  name: "simple-webapp-docker"
spec:
  runPolicy: "Serial"
  triggers:
  -
    type: "GitHub"
    github:
      secret:
- type: "Generic"
  generic:
    secret:
- type: "ImageStream"
  source:
    git:
      uri: "https://github.com/mmumshad/simple-webapp-docker.git"
    strategy:
      type: Docker
      sourceStrategy:
        from:
          kind: "DockerImage"
          name: "ubuntu:16.04"
      output:
        to:
          kind: "ImageStreamTag"
          name: "simple-webapp:latest"
```

```
apiVersion: build.openshift.io/v1
kind: BuildConfig
metadata:
  name: simple-webapp-docker-2
spec:
  output:
    to:
      kind: ImageStreamTag
      name: 'simple-webapp-docker:latest'
  runPolicy: Serial
  source:
    git:
      ref: master
      uri: 'https://github.com/mmumshad/simple-webapp-docker.git'
      type: Git
      strategy:
        dockerStrategy:
          from:
```

OPENSIFT ORIGIN

test-project

Search Catalog

Add to Project

Overview

Applications

Builds

APPLICATION

django-ex

<http://django-ex-test-project.192.168.99.100.nip.io>

Import YAML / JSON

Import YAML / JSON

simple-webapp-docker created 3 hours ago

app simple-webapp-docker

History Configuration Environment Events

Build Strategy:	Docker
Source Repo:	https://github.com/mmumshad/simple-webapp-docker.git
Source Ref:	master
Builder Image:	ubuntu:16.04
Output To:	test-project/simple-webapp-docker:latest
Run Policy:	Serial ?

Triggers [Learn More](#)

Cancel < Back Create

Start a build

Builds » simple-webapp-docker

simple-webapp-docker created 3 hours ago

Start Build

Actions ▾

app simple-webapp-docker

History

Configuration

Environment

Events

✓ Build #1 is complete. [View Log](#)
started 3 hours ago

Filter by label

Add

Build	Status	Duration	Created
#4	⌛ New		a few seconds ago
#3	🔄 Running	5 seconds	a few seconds ago
#2	✓ Complete	9 seconds	a minute ago



{KODE{KLOUD

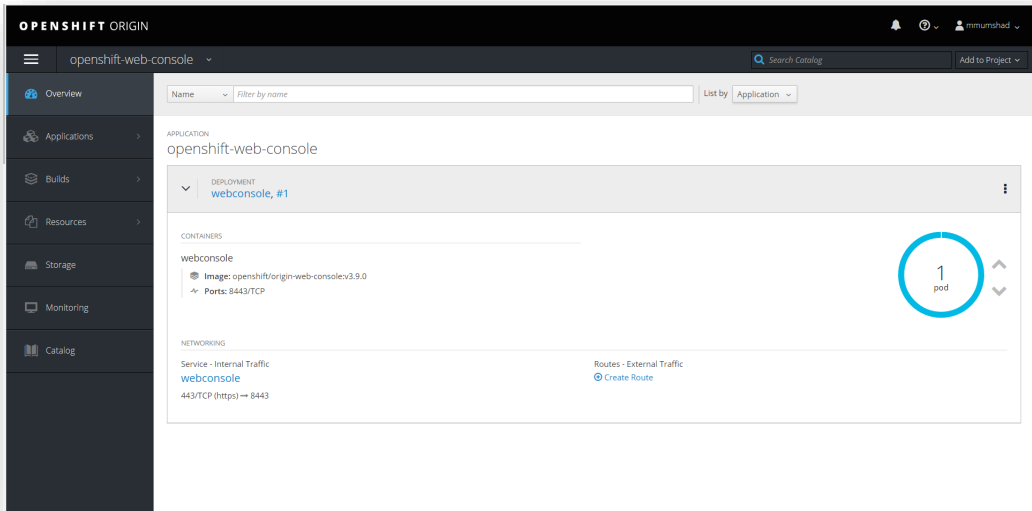
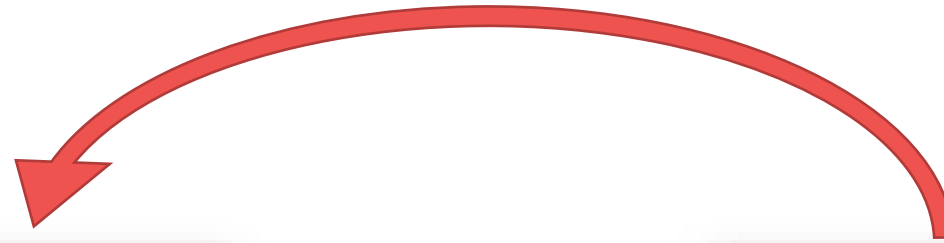


Red Hat

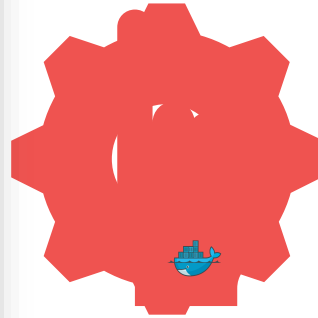
OPENS SHIFT

Build Triggers

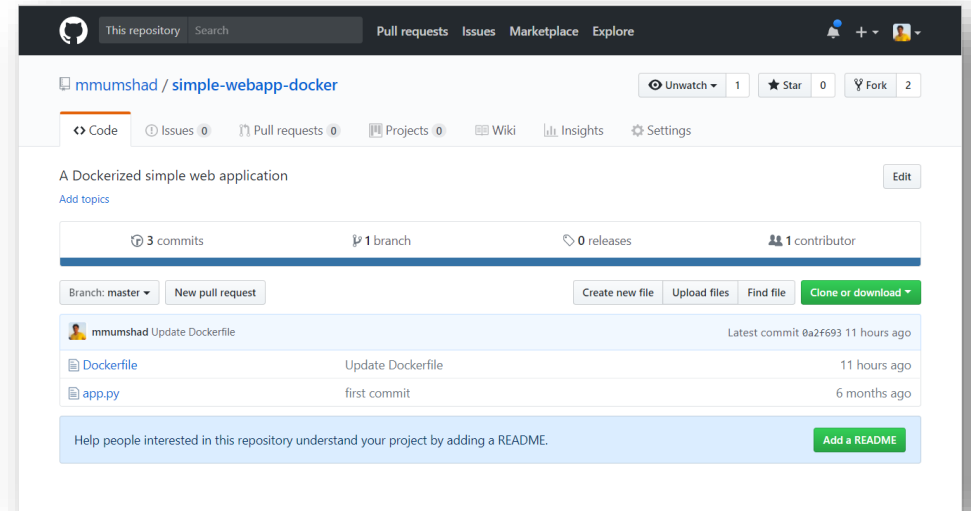
Trigger Build



OpenShift



Automated Build



Code Repository



Webhook

The screenshot shows the OpenShift build configuration page for a build named 'sample-webapp-docker'. The 'Configuration' tab is selected and highlighted with a red dashed box. The 'Details' section on the left lists build parameters: Build Strategy (Source), Source Repo (http://192.168.56.117:8080/root/test-project.git), Source Ref (master), Builder Image (openshift/python:3.6), Output To (myproject/sample-webapp-docker:latest), and Run Policy (Serial). The 'Triggers' section on the right lists various triggers: Config Change For (Build config sample-webapp-docker), New Image For (openshift/python:3.6), Generic Webhook URL (https://192.168.56.115:8443/oapi/v1/nam), GitHub Webhook URL (https://192.168.56.115:8443/oapi/v1/namespaces/.....), GitLab Webhook URL (https://192.168.56.115:8443/oapi/v1/nam), and Manual (CLI) (oc start-build sample-webapp-docker -n r).

Openshift

The screenshot shows the GitHub repository settings page for 'mmumshad / simple-webapp-docker'. The 'Settings' tab is selected and highlighted with a red dashed box. The 'Webhooks' section is also highlighted with a red dashed box. The 'Add webhook' form is visible, showing the 'Payload URL' field with the value 'https://example.com/postreceive' and the 'Content type' dropdown set to 'application/x-www-form-urlencoded'.

Code Repository



{KODE{KLOUD

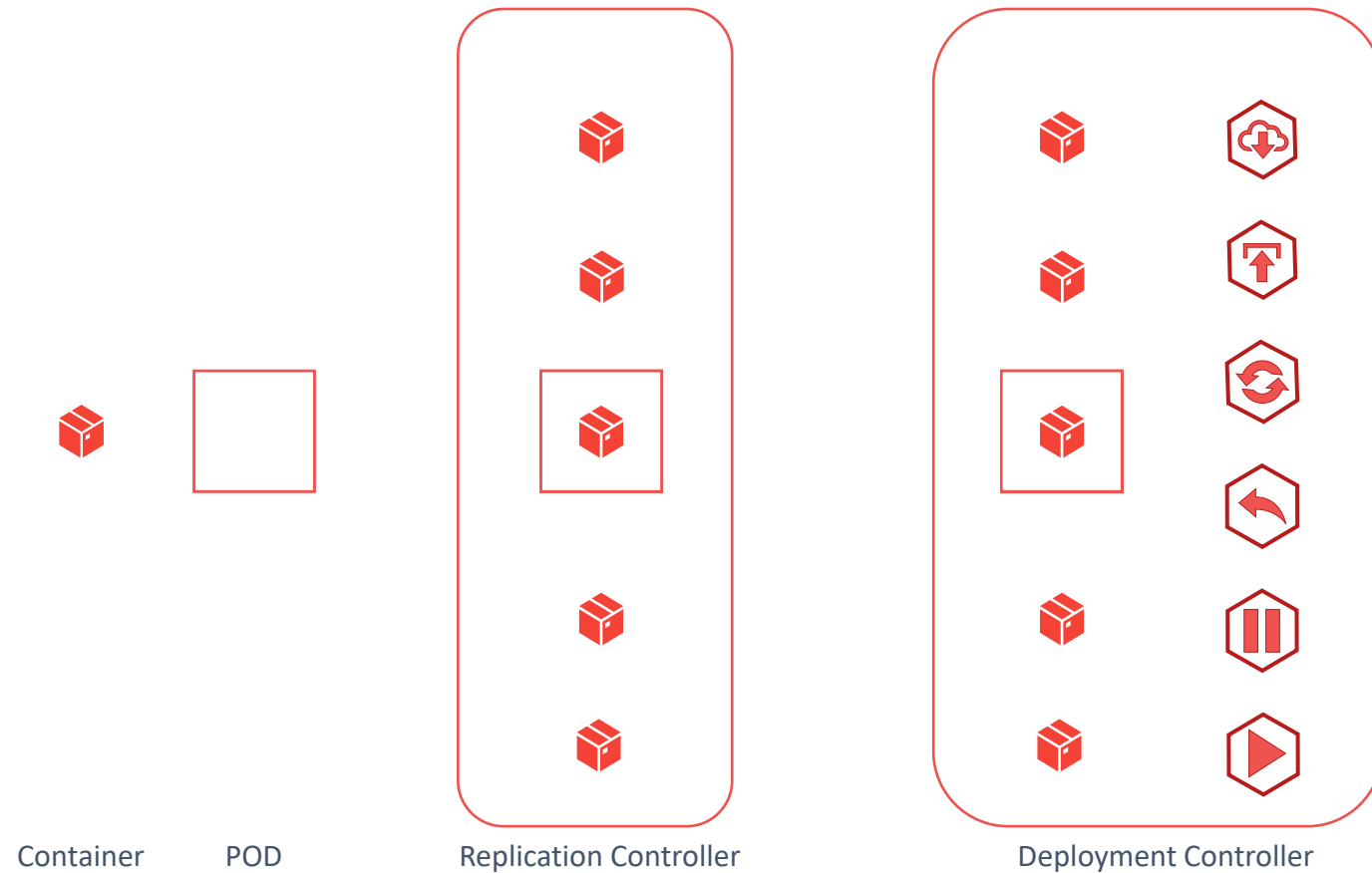


Red Hat

OPENSSHIFT

Deployments

Deployment Controller



Deployment

The screenshot shows the KodeCloud interface for a project named "My Project". The left sidebar contains navigation options: Overview, Applications (selected), and Builds. The main content area is titled "Deployments" and includes a search bar for the catalog and an "Add to Project" button. Below the title is a filter input field labeled "Filter by label" with an "Add" button. A table lists the deployment details:

Name	Last Version	Status	Created	Trigger
sample-webapp-docker	#4	Active, 1 replica	12 hours ago	Image change

Deployment

Deployments » sample-webapp-docker

sample-webapp-docker created 13 hours ago

app sample-webapp-docker

History Configuration Environment Events

Details

Selectors:	deploymentconfig=sample-webapp-docker
Replicas:	1 replica
Strategy:	Rolling
Timeout:	600 sec
Update Period:	1 sec
Interval:	1 sec
Max Unavailable:	25%
Max Surge:	25%

Containers

sample-webapp-docker

Deploy

Actions ▾

Image: myproject/sample-webapp-docker 08b7500 211.1 MiB

Build: sample-webapp-docker, #5

Source: Update app.py ca92a69 authored by Administrator

Ports: 8080/TCP

Volumes

[Add Storage](#) | [Add Config Files](#)

Triggers

Manual (CLI):

[Learn More](#)

New Image For:


Change Of:

oc rollout latest dc/sample-webapp-docker -n r

myproject/sample-webapp-docker:latest





Config

Details

Selectors:	deploymentconfig=sample-webapp-docker
Replicas:	1 replica 
Strategy:	Rolling
Timeout:	600 sec
Update Period:	1 sec
Interval:	1 sec
Max Unavailable:	25%
Max Surge:	25%

Containers



sample-webapp-docker

-  **Image:** myproject/sample-webapp-docker 08b7500 211.1 MiB
-  **Build:** sample-webapp-docker, #5
-  **Source:** Update app.py ca92a69 authored by Administrator
-  **Ports:** 8080/TCP

Volumes

[Add Storage](#) | [Add Config Files](#)

Triggers

Manual (CLI):	<input type="text" value="oc rollout latest dc/sample-webapp-docker -n r"/> 
Learn More 	
New Image For:	myproject/sample-webapp-docker:latest
Change Of:	Config

deployment-config.yaml

```
apiVersion: apps.openshift.io/v1
kind: DeploymentConfig
metadata:
  name: sample-webapp-docker
spec:
  replicas: 1
  selector:
    deploymentconfig: sample-webapp-docker
  strategy:
    type: Rolling
  template:
    metadata:
      labels:
        app: sample-webapp-docker
        deploymentconfig: sample-webapp-docker
    spec:
      containers:
        - image: myproject/sample-webapp-docker
          imagePullPolicy: Always
          name: sample-webapp-docker
          ports:
            - containerPort: 8080
              protocol: TCP
      triggers:
        - imageChangeParams:
            automatic: true
            containerNames:
              - sample-webapp-docker
          from:
            kind: ImageStreamTag
            name: 'sample-webapp-docker:latest'
            namespace: myproject
            lastTriggeredImage: >-
              172.30.1.1:5000/myproject/sample-webapp-
docker@sha256:08b750083d53e8fdc09ab99bc30549141ea44c90763d3e972be264fbec8d706
          type: ImageChange
        - type: ConfigChange
```


Edit Deployment Configuration

Edit Deployment Config sample-webapp-docker

Deployment Strategy

Strategy Type

Rolling

The rolling strategy will wait for pods to pass their readiness check, scale down old components and then scale up. [Learn More](#)

Timeout

600 seconds

How long to wait for a pod to scale up before giving up.

Maximum Number of Unavailable Pods

25%

The maximum number of pods that can be unavailable during the rolling deployment. This can be either a percentage (10%) or a whole number (1).

Maximum Number of Surge Pods

25%

The maximum number of pods that can be scheduled above the original number of pods while the rolling deployment is in progress. This can be either a percentage (10%) or a whole number (1).


To set additional parameters or edit lifecycle hooks, view [advanced strategy options](#).





History

sample-webapp-docker created 15 hours ago

app **sample-webapp-docker**

[History](#) Configuration Environment Events

 Deployment #4 is active. [View Log](#)
created 14 hours ago

Deployment	Status
#4 (latest)	 Active, 1 replica
#3	 Complete
#2	 Complete
#1	 Complete

Rollback

sample-webapp-docker created 15 hours ago

app sample-webapp-docker

History Configuration Environment Events

🔄 Deployment #4 is active. [View Log](#)
created 14 hours ago

Filter by label

Deployment	Status
#4 (latest)	🔄 Active, 1 replica
#3	✅ Complete
#2	✅ Complete
#1	✅ Complete

sample-webapp-docker-3 created 14 hours ago

app sample-webapp-docker openshift.io/deployment-config.name sample-webapp-docker

Details Environment Logs Events

Status: ✅ Complete Roll Back

Deployment Config: sample-webapp-docker

Status Reason: image change

Selectors:
deployment=sample-webapp-docker-3
deploymentconfig=sample-webapp-docker

Replicas: 0 current / 0 desired

0 pods

Template

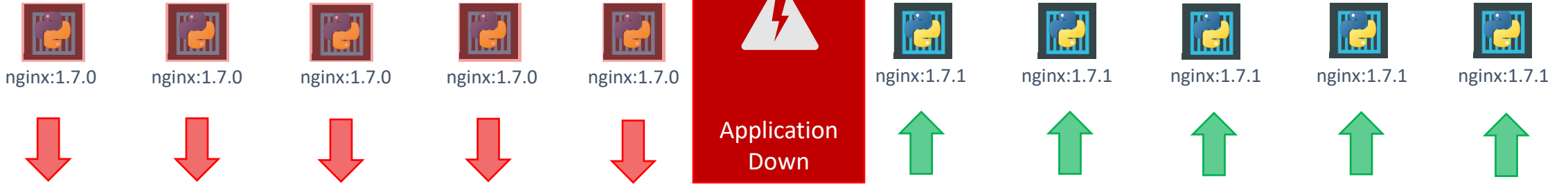
Containers

sample-webapp-docker

- 📦 Image: myproject/sample-webapp-docker 4560011 212.3 MiB
- 🔄 Build: sample-webapp-docker, #4
- 📄 Source: Update app.py a706365 authored by Administrator
- 🔌 Ports: 8080/TCP

Deployment Strategies

Recreate

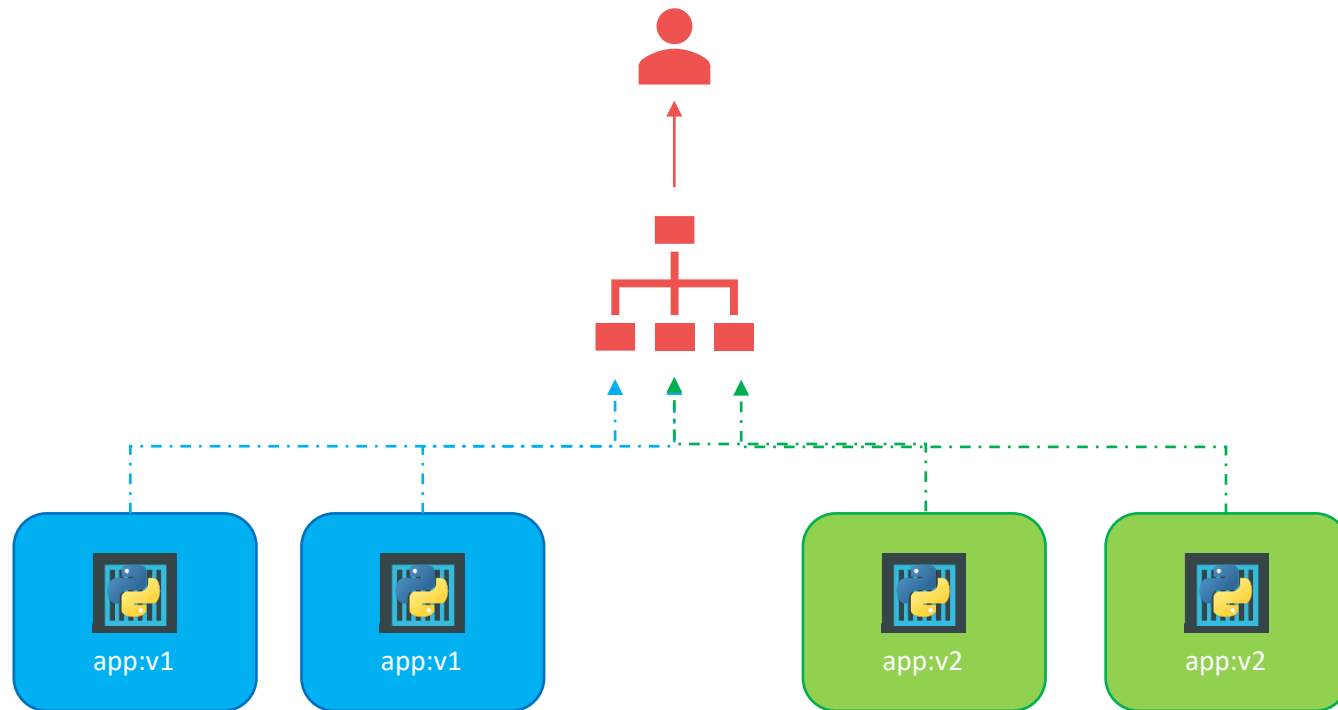


Rolling Update

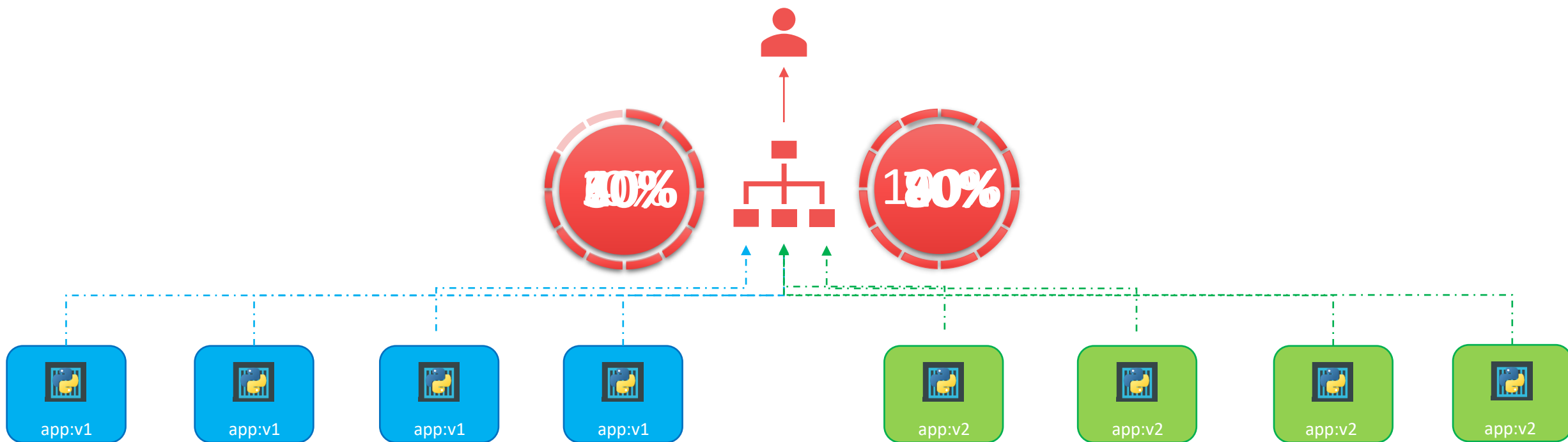


 Application Up

Advanced Strategies – Blue/Green



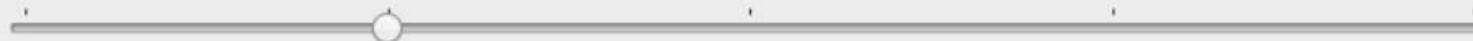
Advanced Strategies – A/B



Service Weights

nodejs-ex 25%

75% mongodb



Percentage of traffic sent to each service. Drag the slider to adjust the values or [edit weights as integers](#).

Commands - Review

```
> oc rollout latest dc/simple-webapp-docker
```

```
> oc rollout history dc/simple-webapp-docker
```

```
> oc rollout describe dc simple-webapp-docker
```

```
> oc rollout undo dc/simple-webapp-docker
```



{KODE{KLOUD

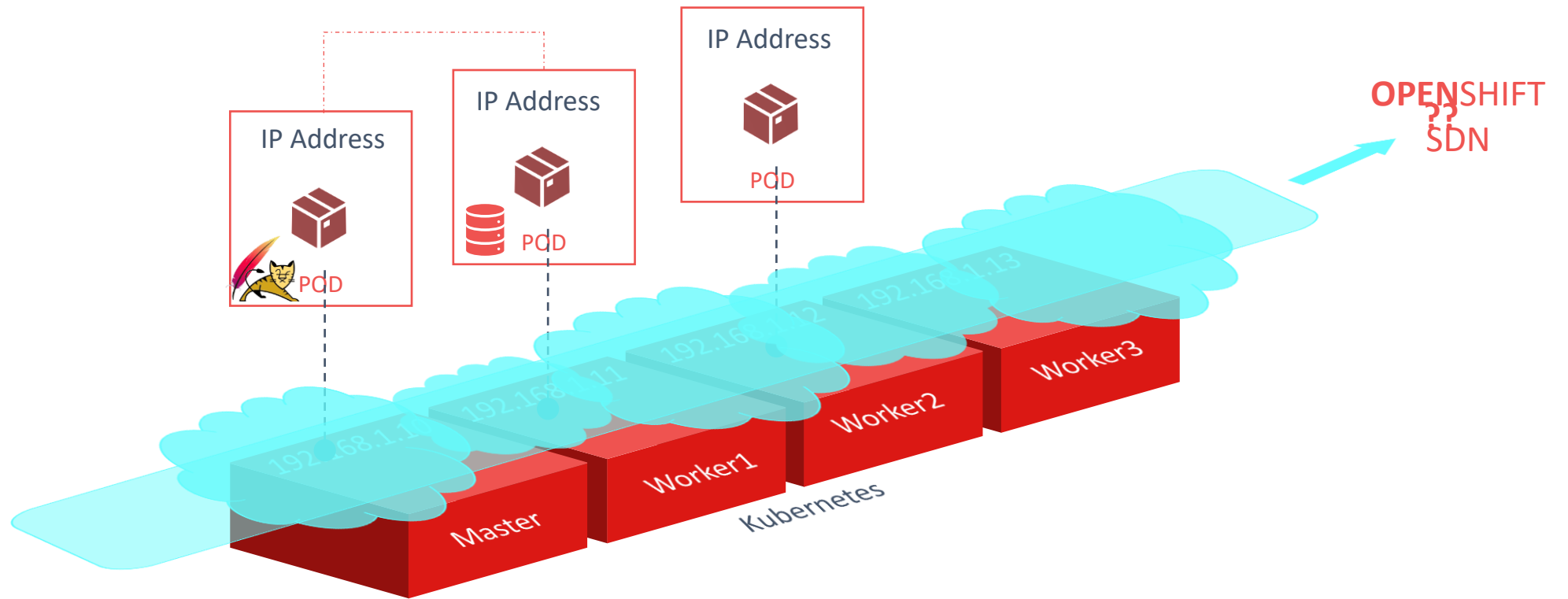


Red Hat

OPENSSHIFT

Networking

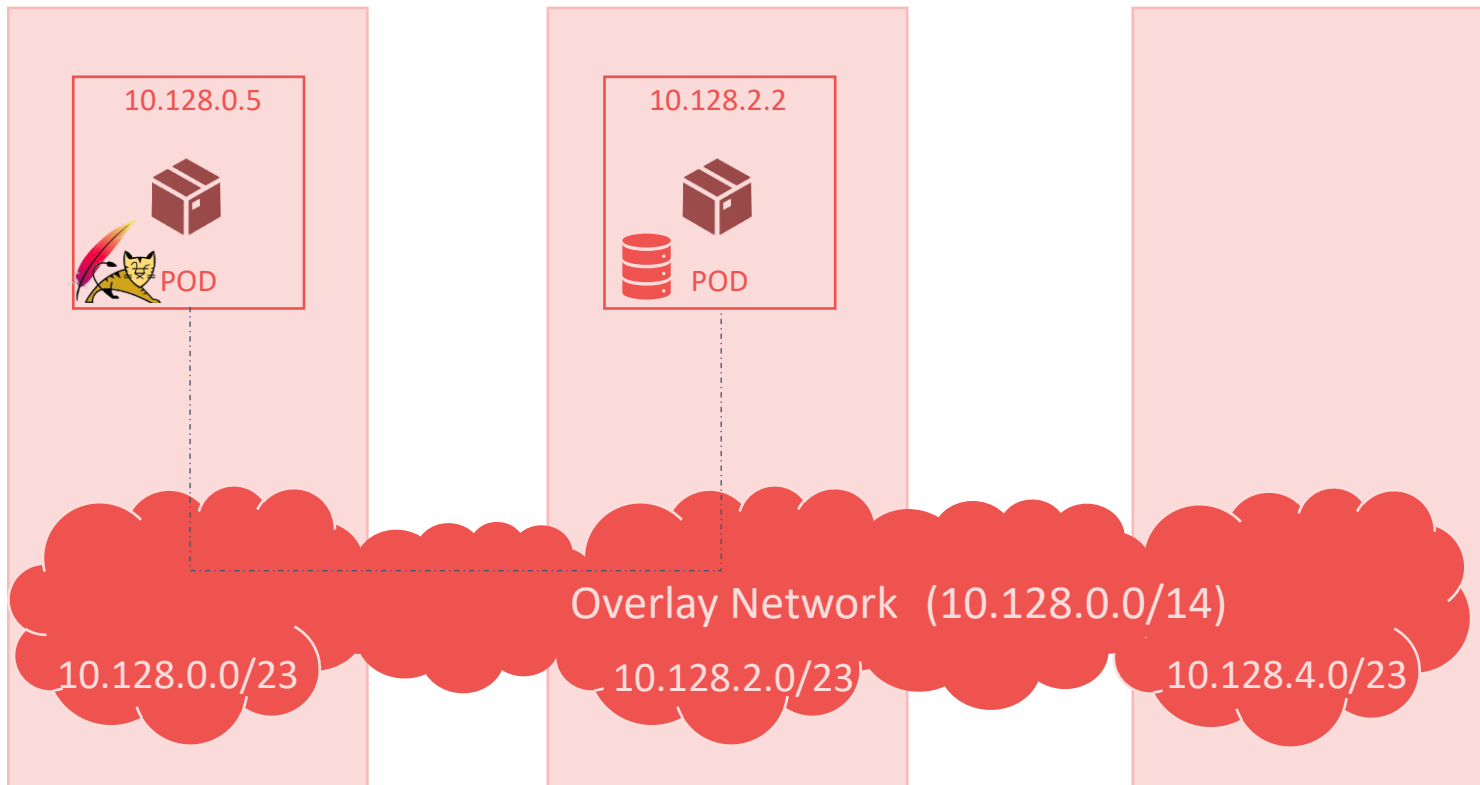
Challenge



OPENSIFT SDN

```
> oc get pods -o wide
```

my-web-app	1/1	Running	0	2d	10.128.0.5	localhost
my-sql-db	1/1	Running	0	1d	10.128.2.2	localhost



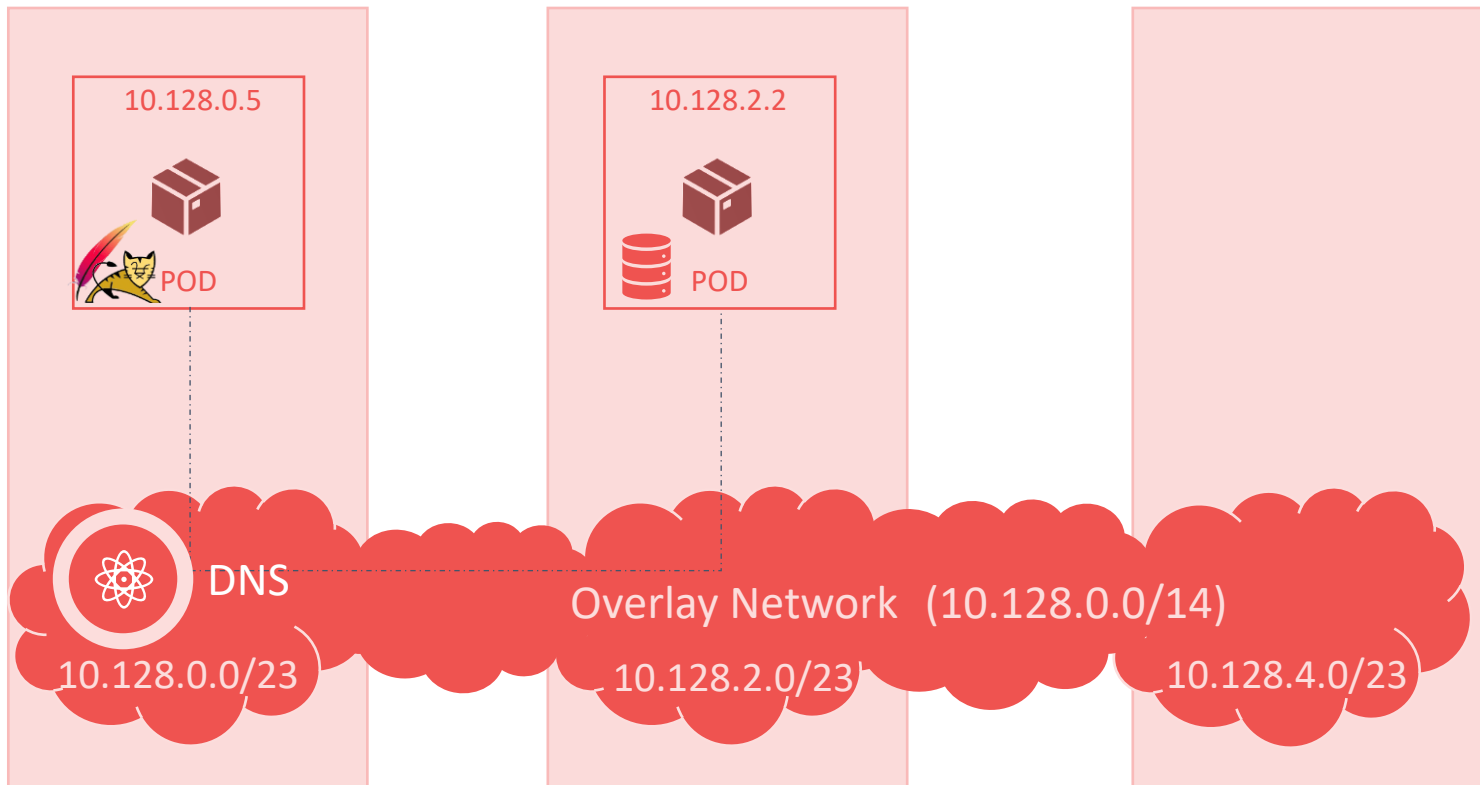
Open vSwitch

- VLAN tagging
- Trunking
- LACP
- Port Mirroring

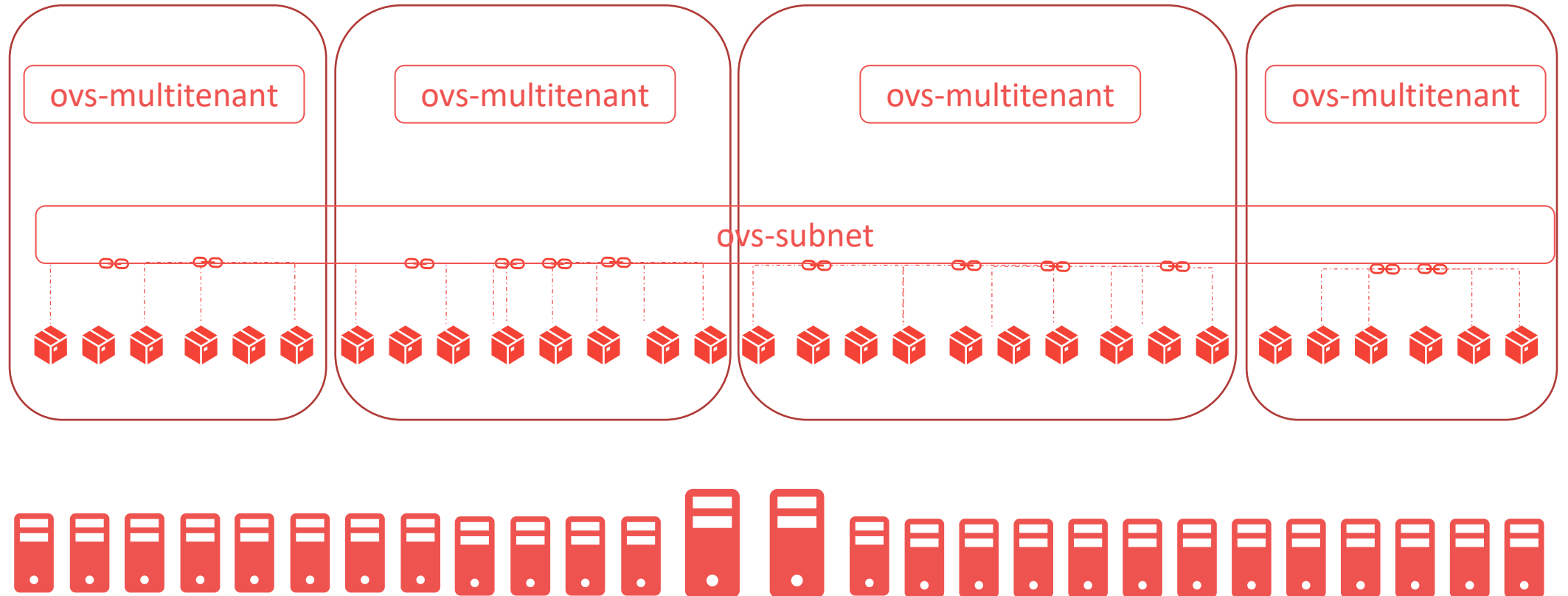
OPENSSHIFT DNS

```
mysql.connect(mysql)
```

SkyDNS



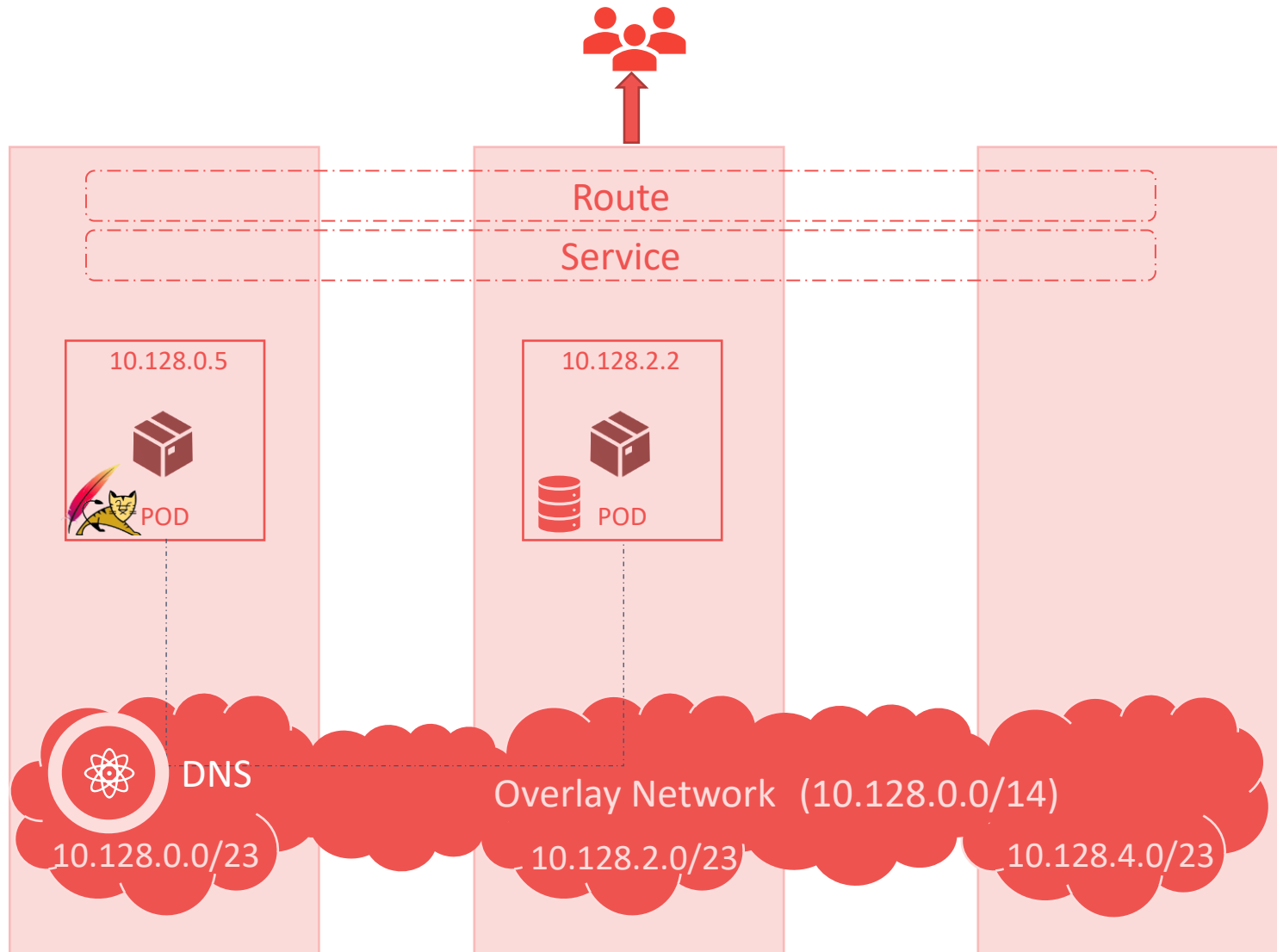
SDN Plugins





Additional Plugins

External Connectivity





{KODE{KLOUD

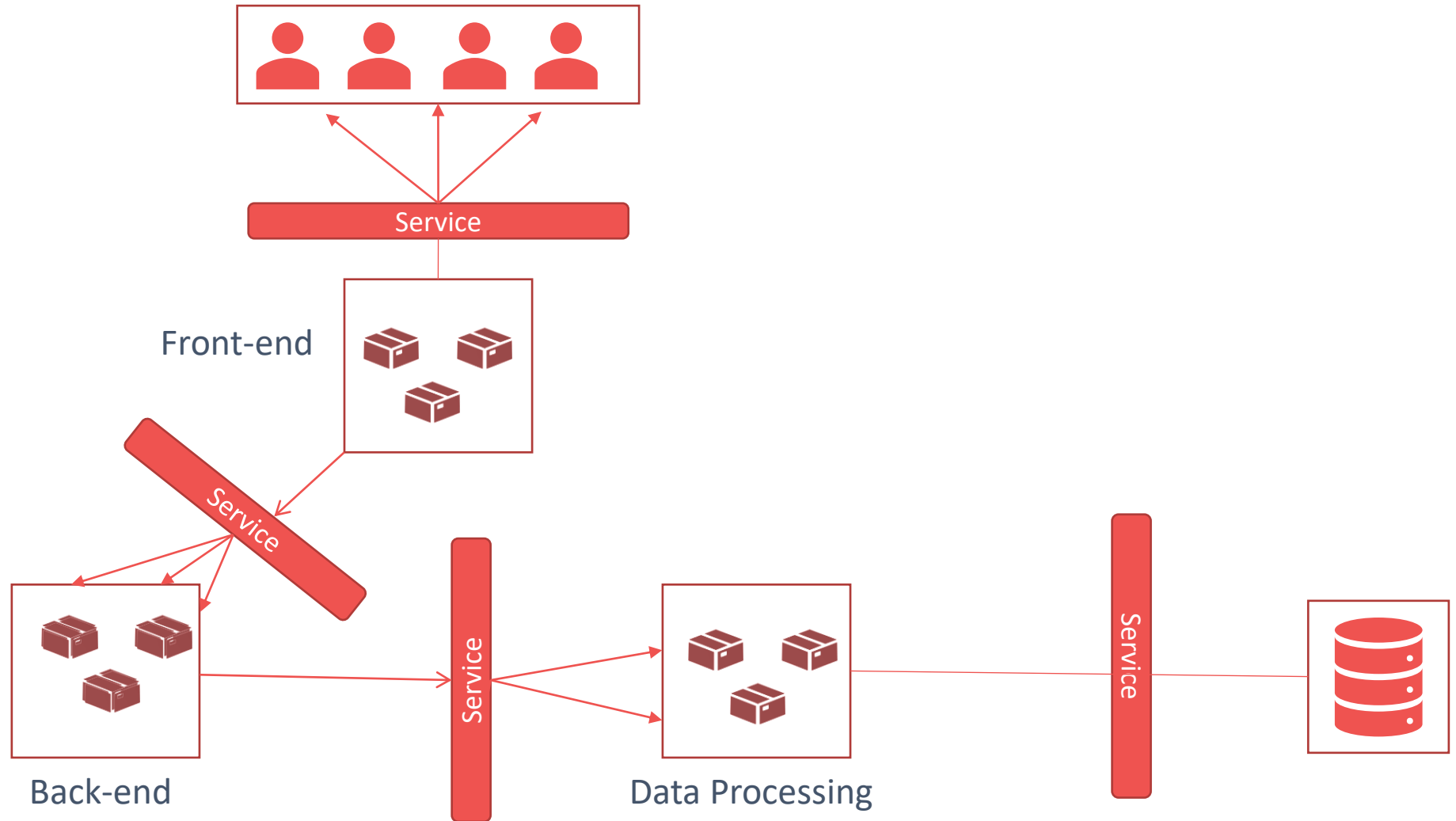


Red Hat

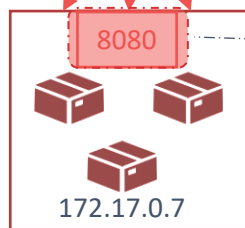
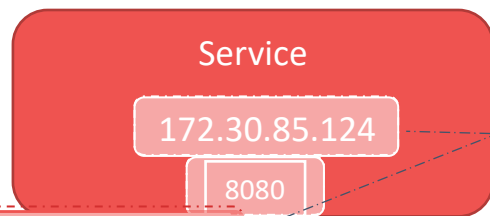
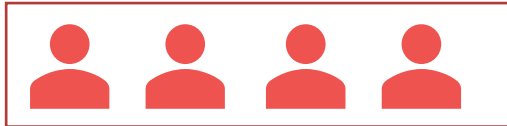
OPENS SHIFT

Services and Routes

Service



Service



simple-webapp-docker

simple-webapp-docker created 2 days ago

app simple-webapp-docker

Details Events

Selectors: deploymentconfig=simple-webapp-docker

Type: ClusterIP

IP: 172.30.85.124

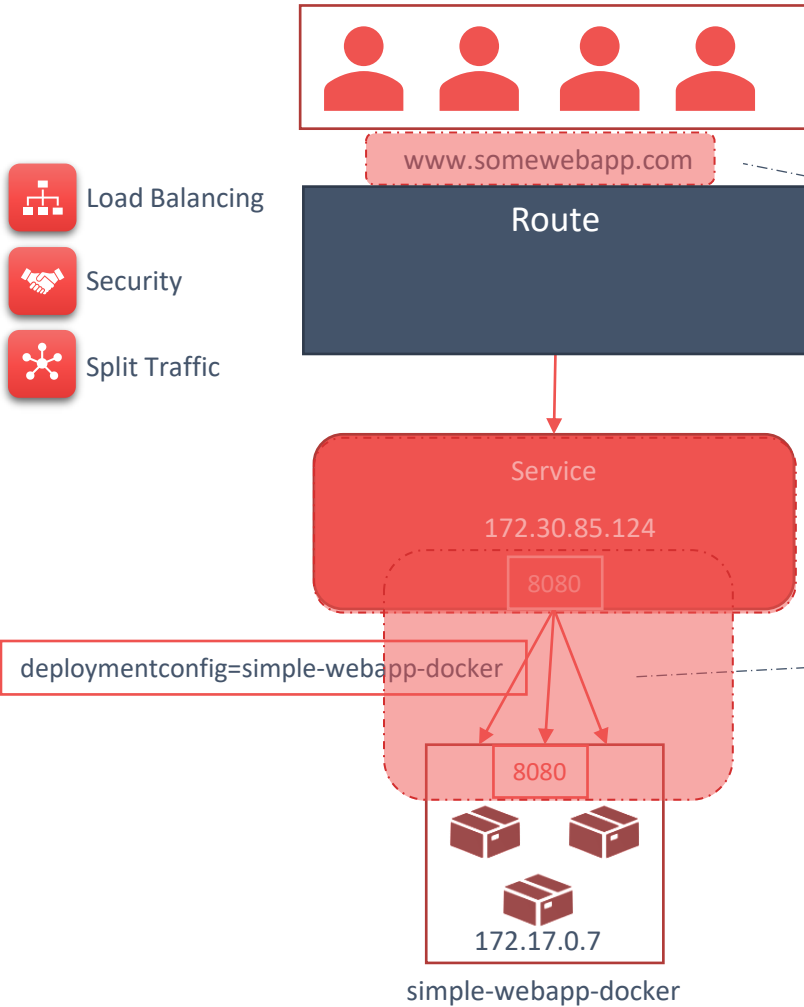
Hostname: simple-webapp-docker.test-project.svc ⓘ

Session affinity: None

Traffic

Route	Service Port	Target Port
simple-webapp-docker	8080/TCP (8080-tcp)	8080

Route



Create Route

Routing is a way to make your application publicly visible.

*** Name**
my-route
A unique name for the route within the project.

Hostname
www.example.com
Public hostname for the route. If not specified, a hostname is generated.

Path
/
Path that the router watches to route traffic to the service.

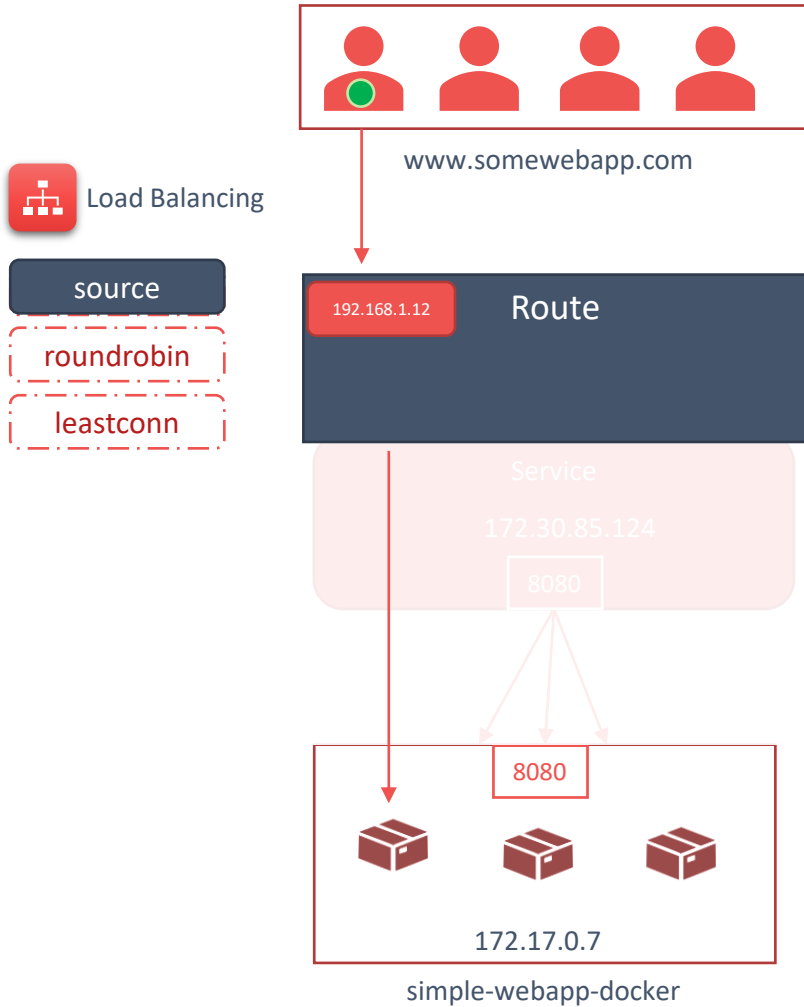
***Service**
django-ex
Service to route to.

Target Port
8080 → 8080 (TCP)
Target port for traffic.

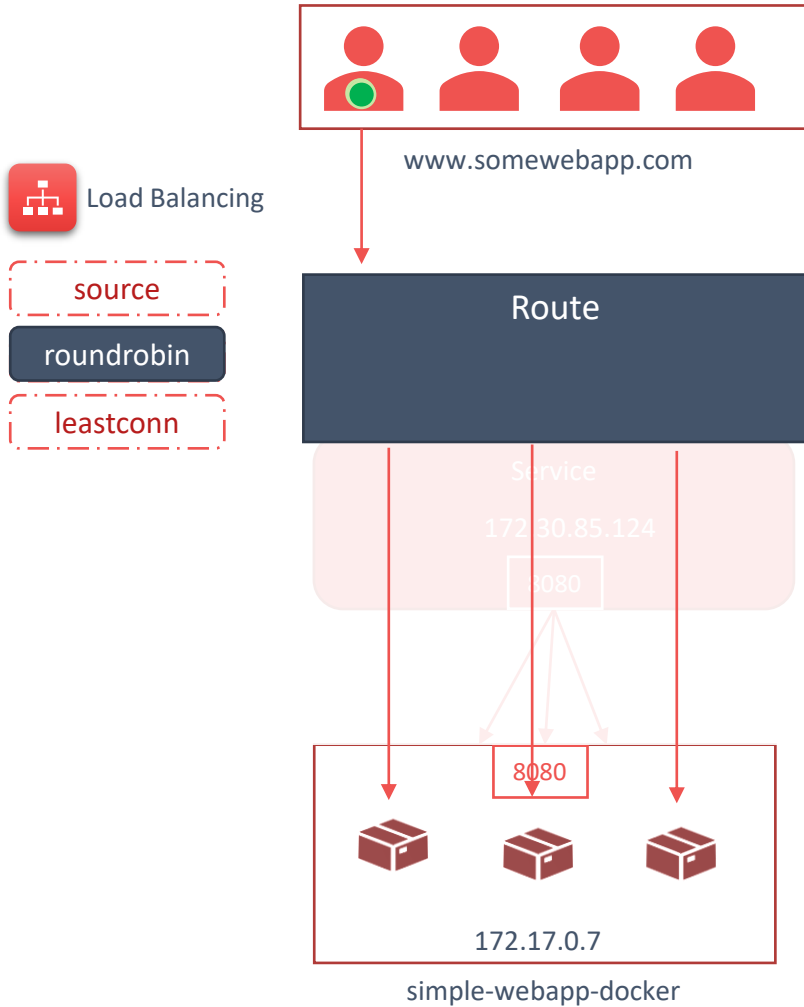
Alternate Services
 Split traffic across multiple services
Routes can direct traffic to multiple services for A/B testing. Each service has a weight controlling how much traffic it gets.

Security
 Secure route
Routes can be secured using several TLS termination types for serving certificates.

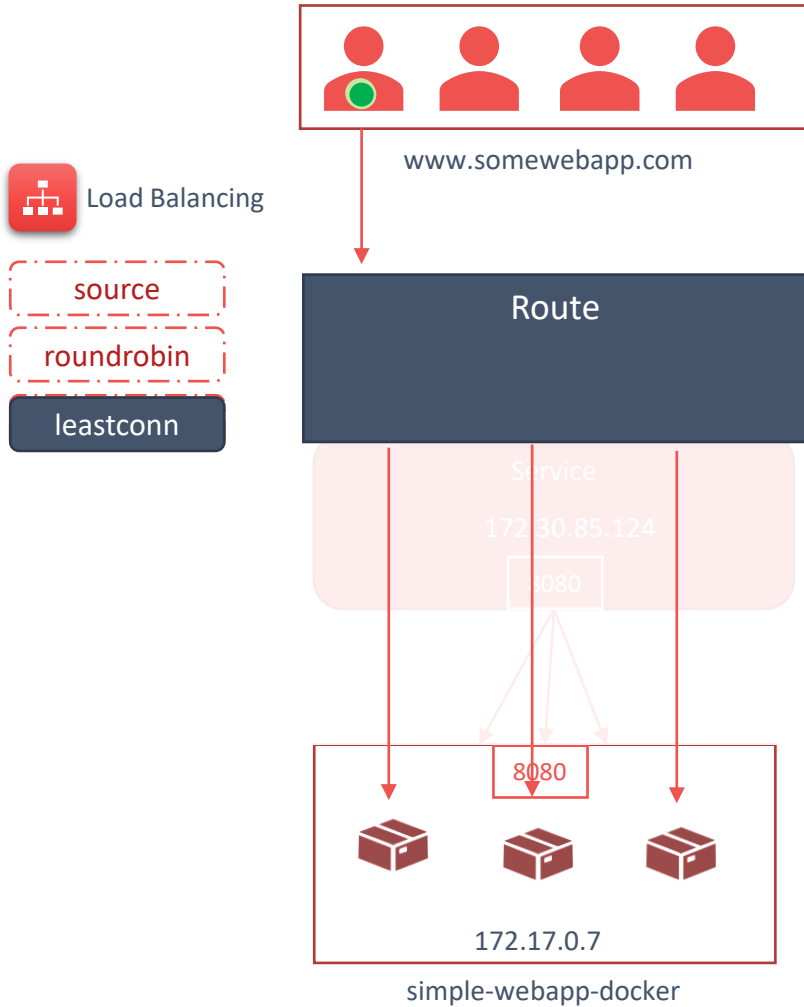
Route – Load Balancing



Route – Load Balancing



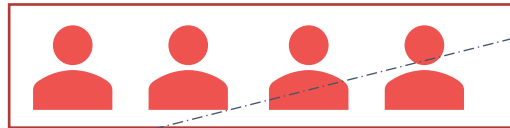
Route – Load Balancing



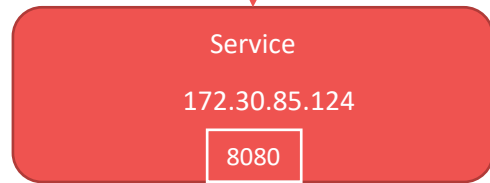
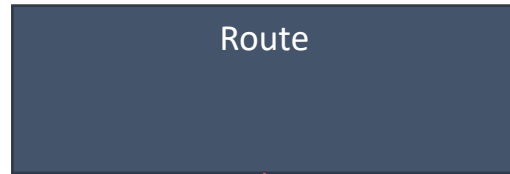
Route - Security



Security

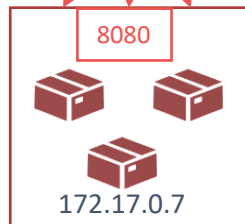


https://www.somewebapp.com



8080

deploymentconfig=simple-webapp-docker



172.17.0.7

simple-webapp-docker

Security

Secure route

Routes can be secured using several TLS termination types for serving certificates.

TLS Termination

Edge

[Learn More ↗](#)

Insecure Traffic

None

None

Allow

Redirect

Certificate

The PEM format certificate. Upload file by dragging & dropping, selecting it, or pasting from the clipboard.

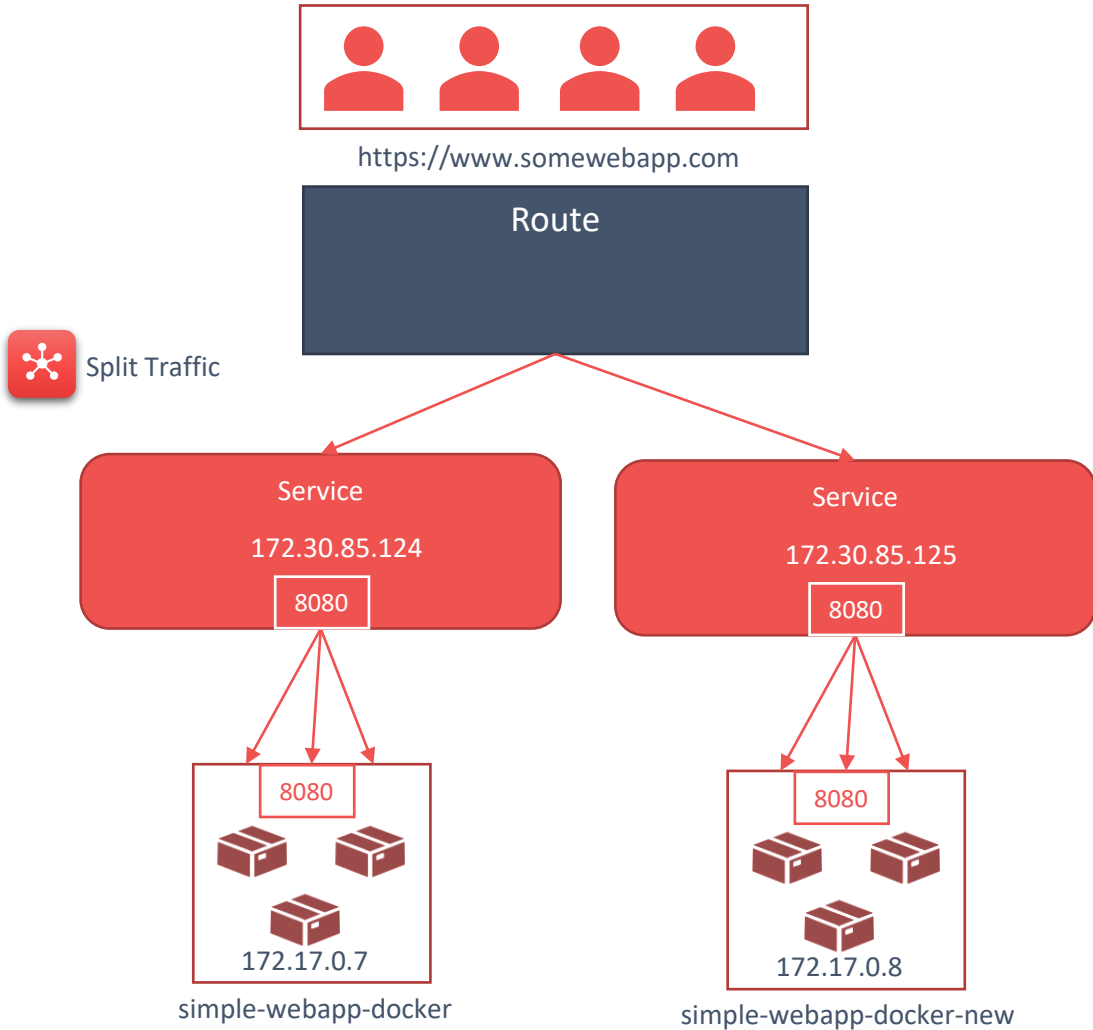
Private Key

The PEM format key. Upload file by dragging & dropping, selecting it, or pasting from the clipboard.

CA Certificate

The PEM format CA certificate chain. Upload file by dragging & dropping, selecting it, or pasting from the clipboard.

Route – Split Traffic



Alternate Services

Split traffic across multiple services

Routes can direct traffic to multiple services for A/B testing. Each service has a weight controlling how much traffic it gets.

* Service

simple-webapp

Alternate service for route traffic.

[Remove Service](#) | [Add Another Service](#)

Service Weights

simple-webapp-docker 50% 50% simple-webapp-docker-new

Percentage of traffic sent to each service. Drag the slider to adjust the values or [edit weights as integers](#)



{KODE {KLOUD

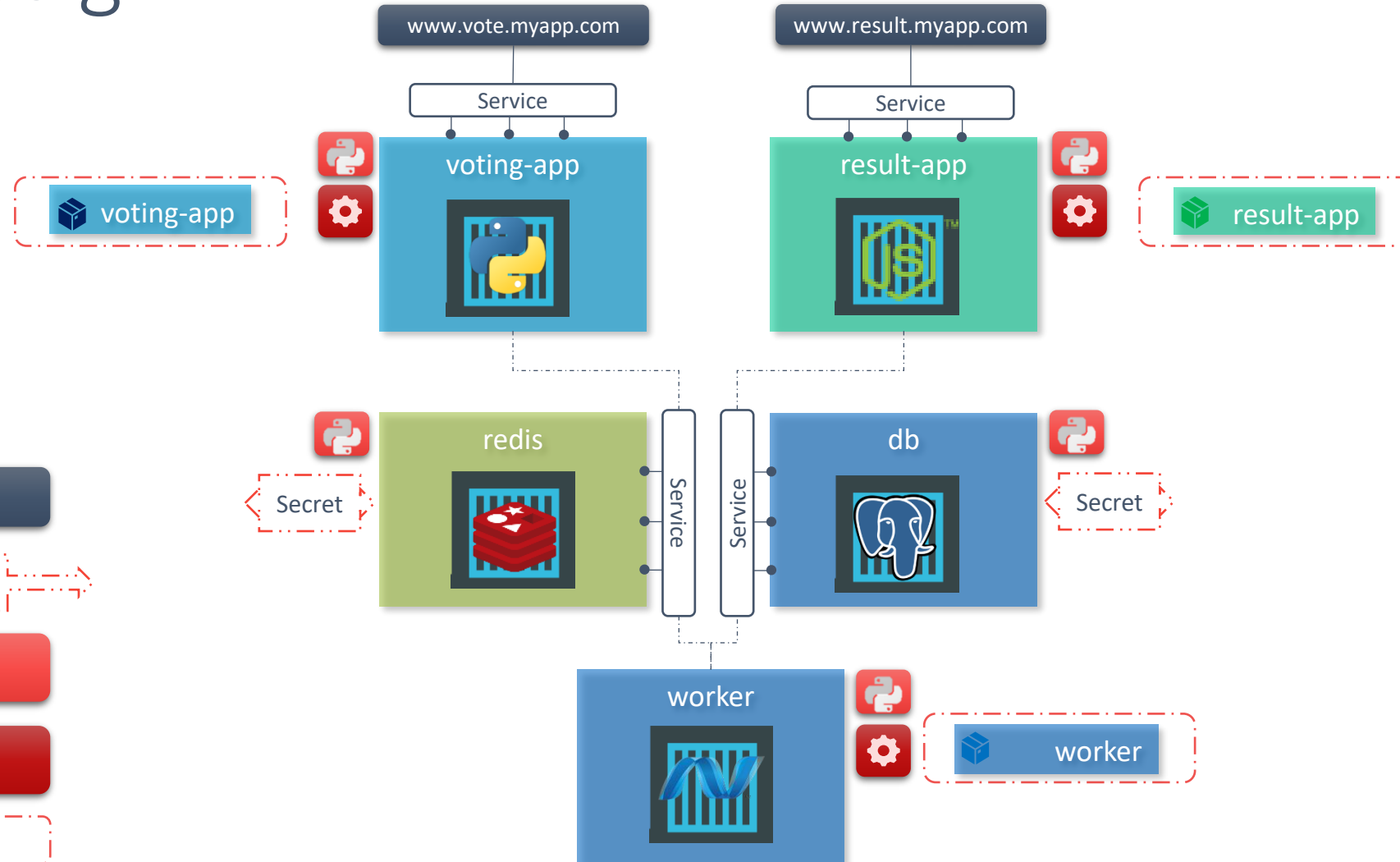


Red Hat

OPENSIFT

Example Voting Application

Design



Route

Secrets
DB Credentials

Deployment

Build

Image Stream

Service



{KODE{KLOUD

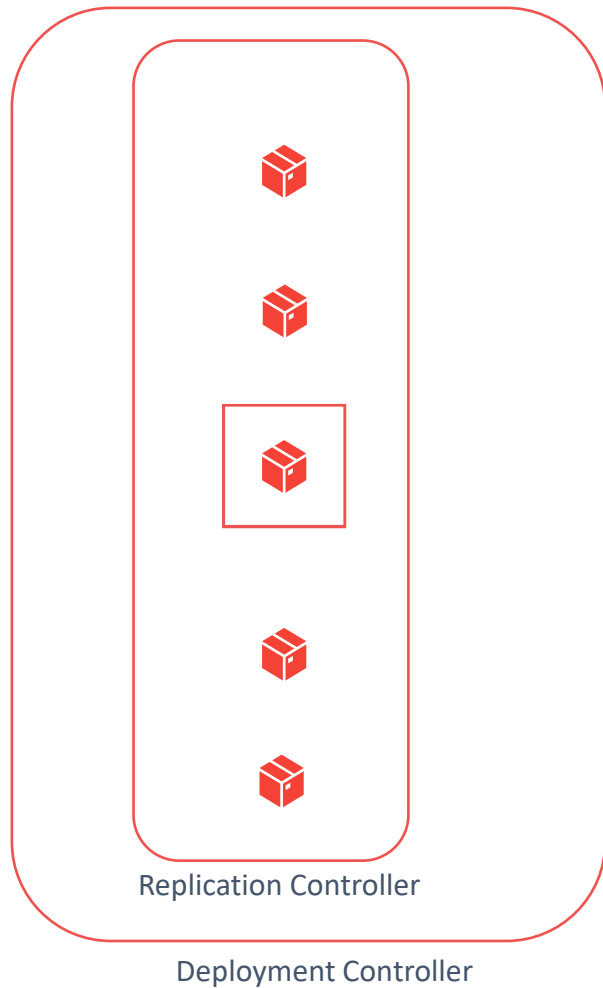


Red Hat

OPENS SHIFT

Scaling

Scale Deployment



APPLICATION
simple-webapp <https://simple-webapp-docker-my-webapplication.192.168.99.100.nip.io>

DEPLOYMENT CONFIG
simple-webapp, #1

CONTAINERS

simple-webapp

- Image: my-webapplication/simple-webapp 64ad4d7 212.2 MiB
- Build: simple-webapp, #1
- Source: Add new file 1c42c7e
- Ports: 8080/TCP

5 pod

NETWORKING

Service - Internal Traffic
simple-webapp
8080/TCP (8080-tcp) → 8080

Routes - External Traffic
<https://simple-webapp-docker-my-webapplication.192.168.99.100.nip.io>
Route simple-webapp-docker, target port 8080-tcp

Traffic Split

simple-webapp	90%
simple-webapp-docker	10%



{KODE}{KLOUD



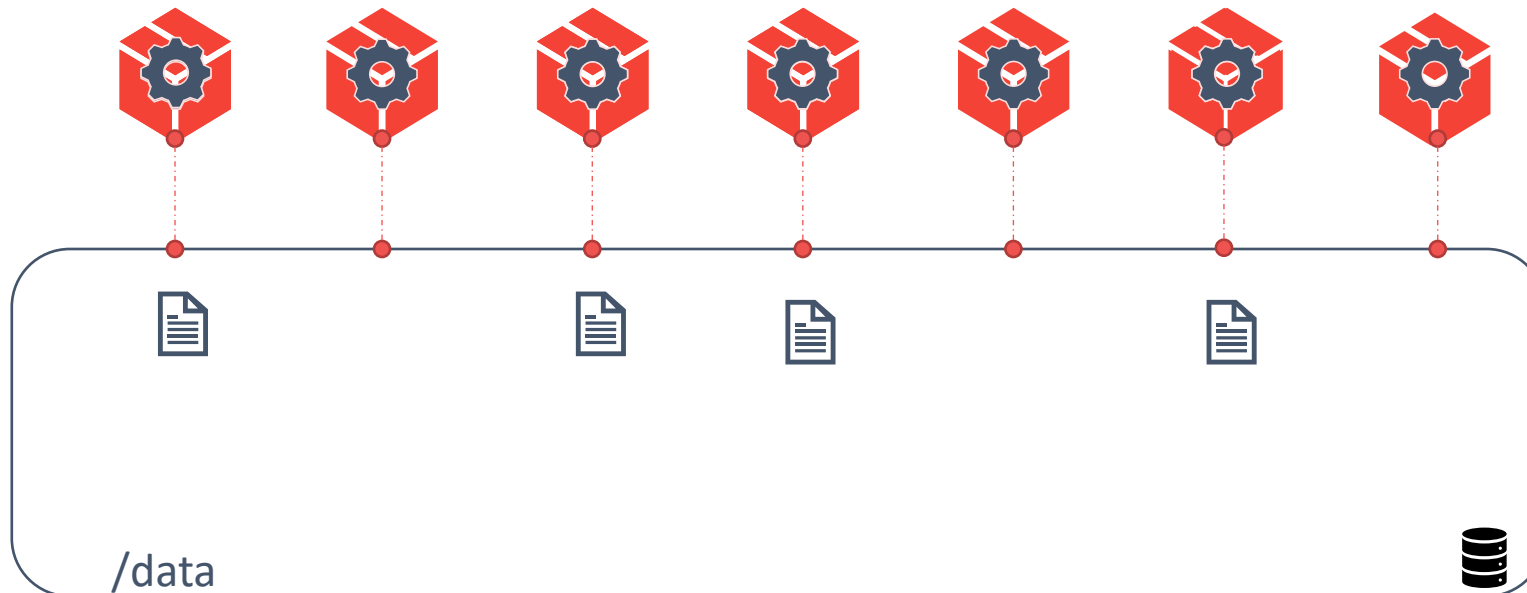
Red Hat

OPENSSHIFT

Storage

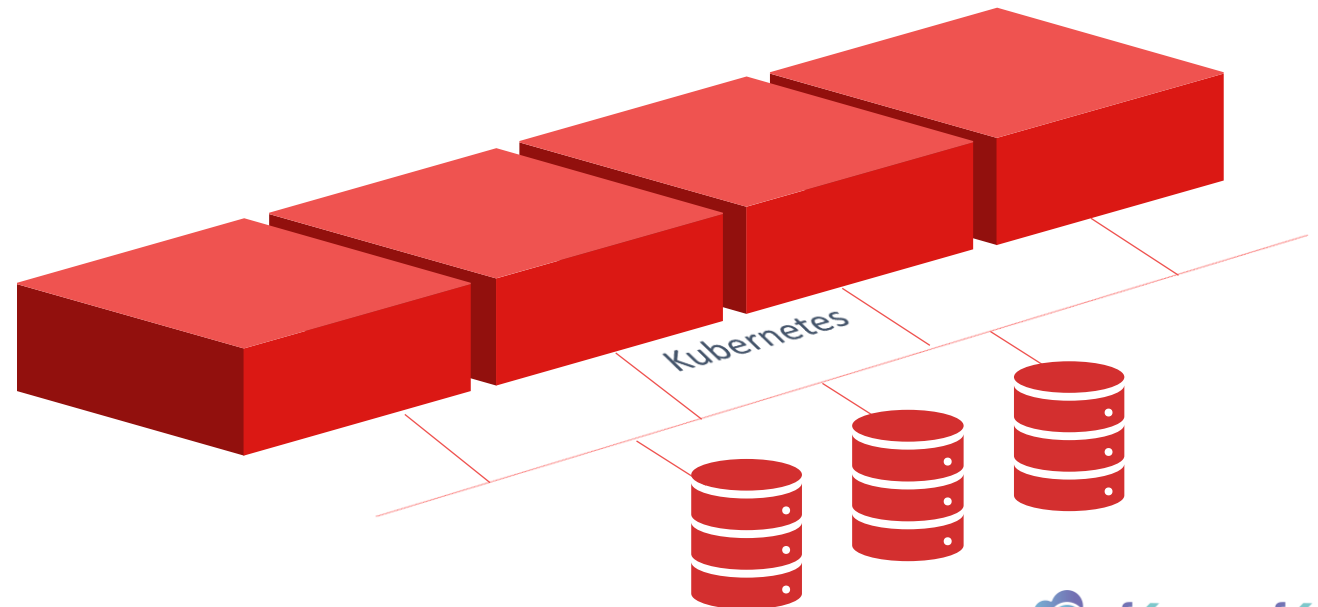


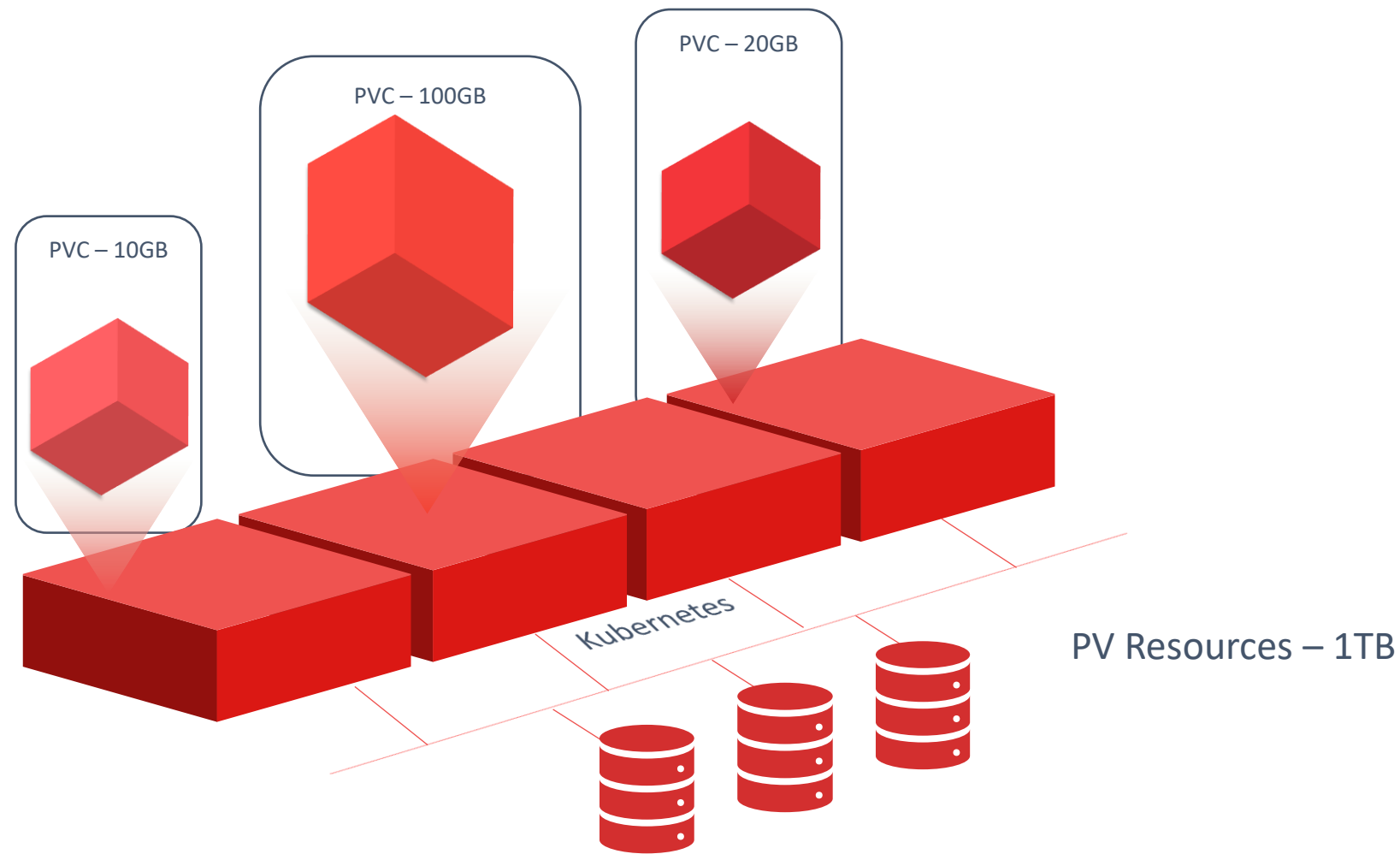
Persistent Volume



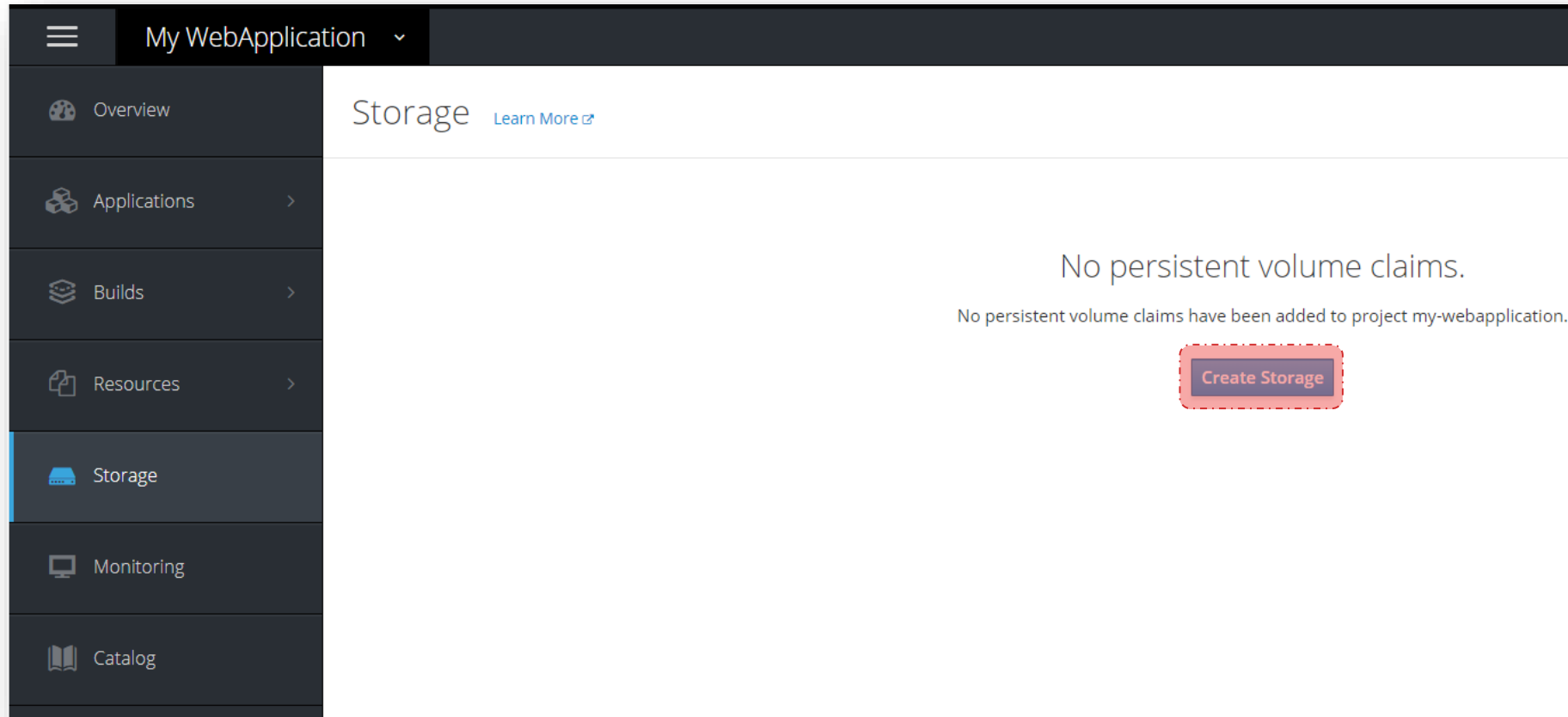
Plugins

- Local
- iSCSI
- Fibre Channel
- NFS
- GlusterFS
- Ceph RDB
- OpenStack Cinder
- AWS Elastic Block Store
- GCE Persistent Disk
- Azure Disk
- Azure File
- VMWare vSphere





Create Storage



Create Storage

Storage » Create Storage

Create Storage

Create a request for an administrator-defined storage asset by specifying size and permissions for a best fit. [Learn More](#)

*** Name**

A unique name for the storage claim within the project.

*** Access Mode**

Single User (RWO) Shared Access (RWX) Read Only (ROX)

Permissions to the mounted volume.

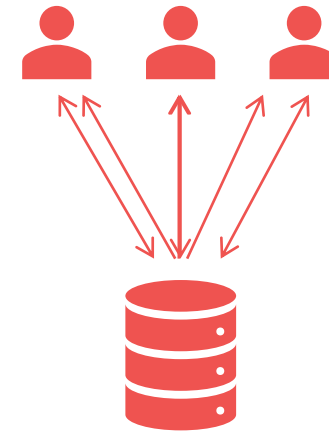
*** Size**

Desired storage capacity.

[What are GiB?](#)

Use label selectors to request storage

[Learn More](#)




Deployments - Volumes

[Deployments](#) » [simple-webapp-docker](#)

simple-webapp-docker created 17 hours ago

[History](#) [Configuration](#) [Environment](#) [Events](#)

Details

Selectors:	name=simple-webapp-docker
Replicas:	2 replicas 
Strategy:	Rolling
Timeout:	600 sec
Update Period:	1 sec
Interval:	1 sec
Max Unavailable:	25%
Max Surge:	25%

Volumes

volume-q7h2m [Remove](#)

Type:	persistent volume claim (reference to a persistent volume claim)
Claim name:	db-storage
Mode:	read-write

[Add Storage](#) | [Add Config Files](#)



{KODE{KLOUD



Red Hat

OPENSIFT





















Templates and Catalog

Catalog

Browse Catalog Deploy Image Import YAML / JSON Select from Project

[All](#) [Languages](#) [Databases](#) [Middleware](#) [CI/CD](#) [Other](#)

Filter ▾ 20 Items

 .NET .NET Core Builder Images	 Apache HTTP Server (httpd)	 CakePHP + MySQL	 Dancer + MySQL	 Django + PostgreSQL	 Jenkins
 MariaDB	 MongoDB	 MySQL	 Nginx HTTP server and a reverse proxy (nginx)	 Node.js	 Node.js + MongoDB
 Perl	 PHP	 Pipeline Build Example	 PostgreSQL	 Python	 Rails + PostgreSQL
 Ruby	 WildFly				

Browse Catalog

Deploy Image Import YAML / JSON Select from Project

All Languages Databases Middleware CI/CD Other

Filter 20 Items

.NET

.NET Core Builder Images



Apache HTTP Server (httpd)

php

CakePHP + MySQL



Dancer + MySQL



Django + PostgreSQL



Jenkins



MariaDB



MongoDB

MySQL

MySQL

NGINX

Nginx HTTP server and a reverse proxy (nginx)

node

Node.js

node

Node.js + MongoDB



Perl

php

PHP



Pipeline Build Example



PostgreSQL



Python



Rails + PostgreSQL



Ruby



WildFly



Custom App

Image Stream Application

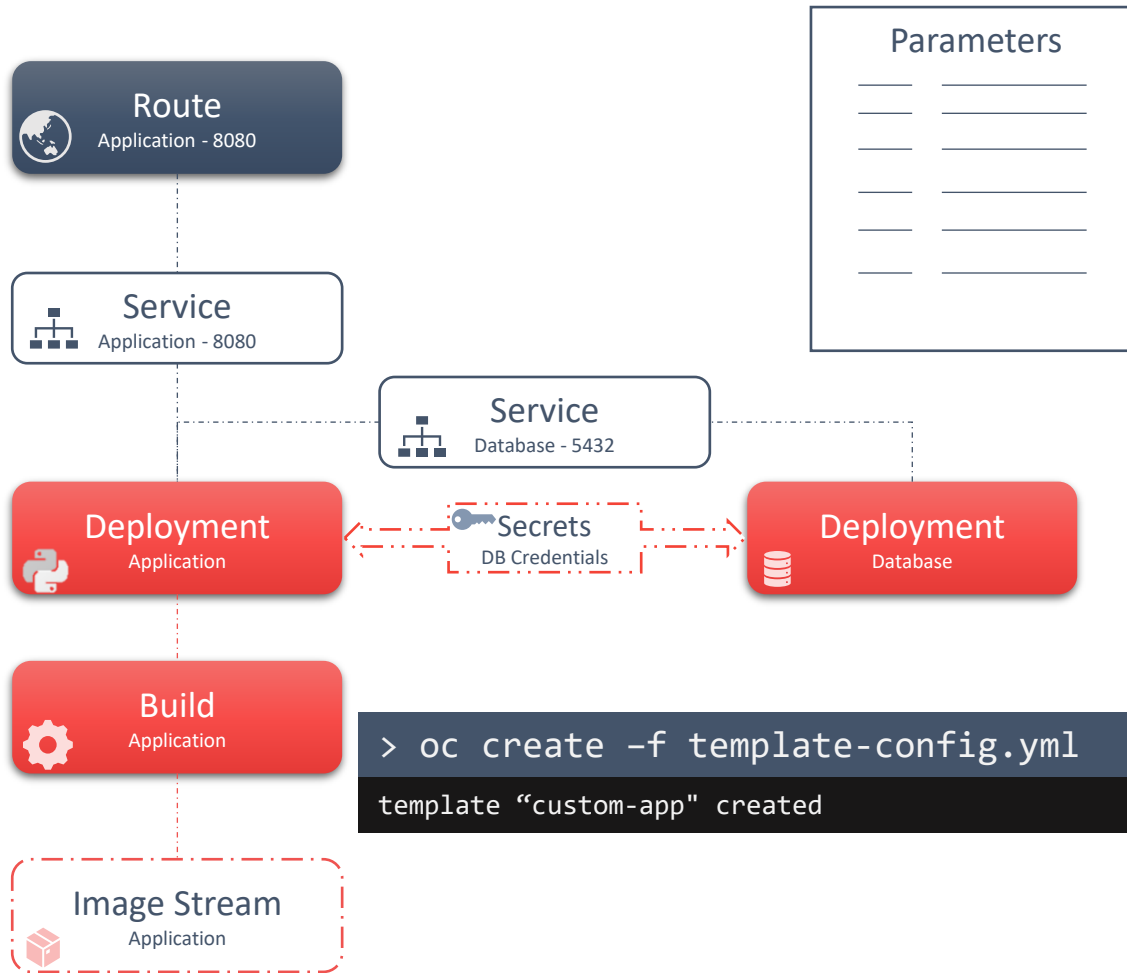
Maximum amount of memory the Django container can use.

* Memory Limit (PostgreSQL)

512Mi

Maximum amount of memory the PostgreSQL container can use.

Template



template-config.yml

```
apiVersion: v1
kind: Template
metadata:
  name: custom-app
objects:
- apiVersion: v1
  kind: Secret
  <.. code hidden ..>
- apiVersion: v1
  kind: Service
  <.. code hidden ..>
- apiVersion: v1
  kind: Service
  <.. code hidden ..>
- apiVersion: v1
  kind: Route
  <.. code hidden ..>
- apiVersion: v1
  kind: BuildConfig
  <.. code hidden ..>
- apiVersion: v1
  kind: DeploymentConfig
  <.. code hidden ..>
- apiVersion: v1
  kind: DeploymentConfig
  <.. code hidden ..>
- apiVersion: v1
  kind: ImageStream
  <.. code hidden ..>
parameters:
- displayName: "Namespace"
  name: "NAMESPACE"
```

Export

```
> oc export service db
```

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    template.openshift.io/expose-uri:
      postgres://{.spec.clusterIP}:{.spec.ports[?(.name=="postgresq
l")].port}
  creationTimestamp: null
  labels:
    app: postgresql-persistent
    template: postgresql-persistent-template
  name: db
spec:
  ports:
  - name: postgresql
    port: 5432
    protocol: TCP
    targetPort: 5432
  selector:
    name: db
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```



{KODE}{KLOUD

Conclusion

OpenShift Overview

Architecture

Projects and Users

Builds and Deployments

Networking

Services and Routes

YAML Definition Files

Example Voting Application