
KODEKLOUD

Red Hat

OPENSHIFT

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

Kubernetes for the Absolute Beginners -...

Mumshad Mannambeth

4.5 (161)

Docker Advanced - SWARM - Hands-on

Mumshad Mannambeth

4.6 (139)

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?

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

KODEKLOUD

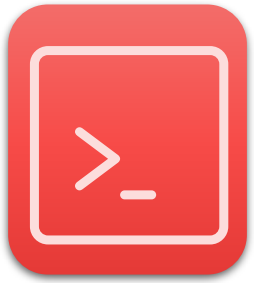

KODEKLOUD

Red Hat

OPENSHIFT

Introduction

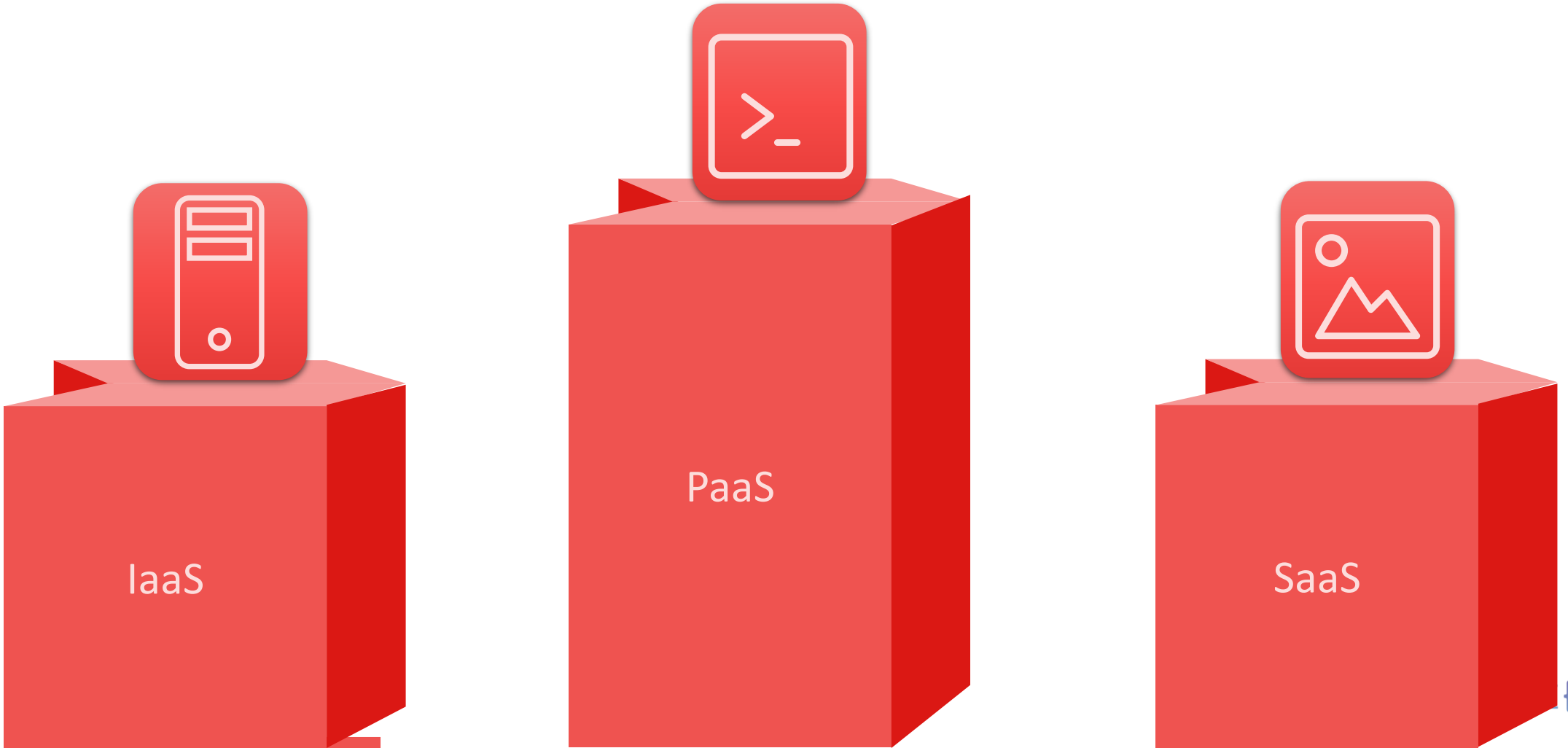
KODEKLOUD

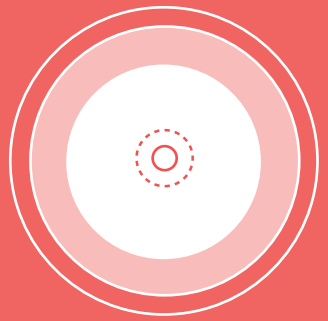



KODEKLOUD

OPENSHIFT
by Red Hat



KLOUD



OPENSHIFT
Origin

Open source application container platform

OPENSHIFT
Online

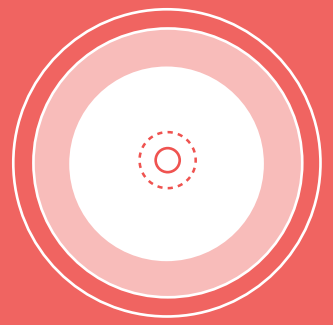
Public Application Development hosting service

OPENSHIFT
Dedicated

Managed private cluster on AWS/Google Clouds

OPENSHIFT
Enterprise

On-Premise private PaaS



OPENSHIFT

Origin

Open source application container platform

OPENSHIFT

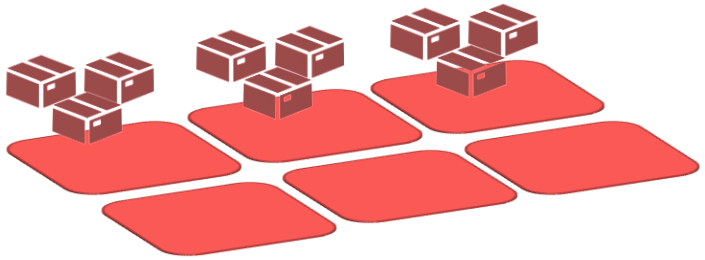
Origin

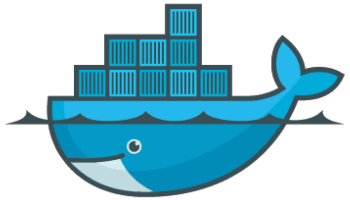
Open source application
container platform

OPENSHIFT 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

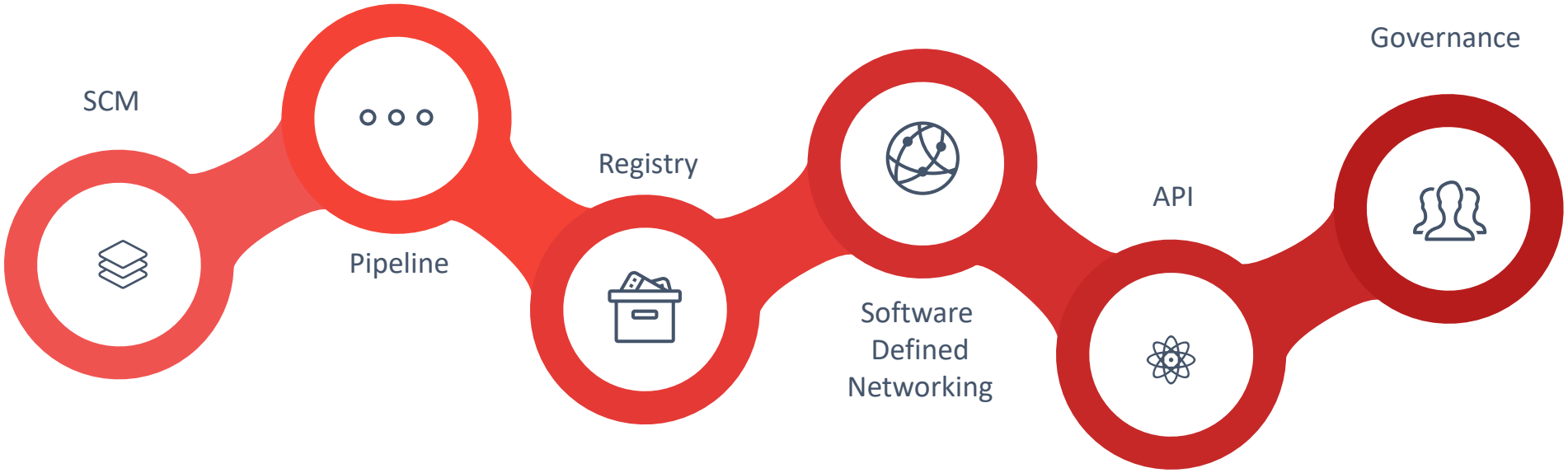
Kubernetes





Docker

Tools



Kubernetes for the Absolute Beginners -...

Mumshad Mannambeth

4.5 (136)

Docker Advanced - SWARM - Hands-on

Mumshad Mannambeth

4.6 (136)

Docker for the Absolute Beginner - Hands On

Mumshad Mannambeth

4.6 (968)

KODEKLOUD


KODEKLOUD

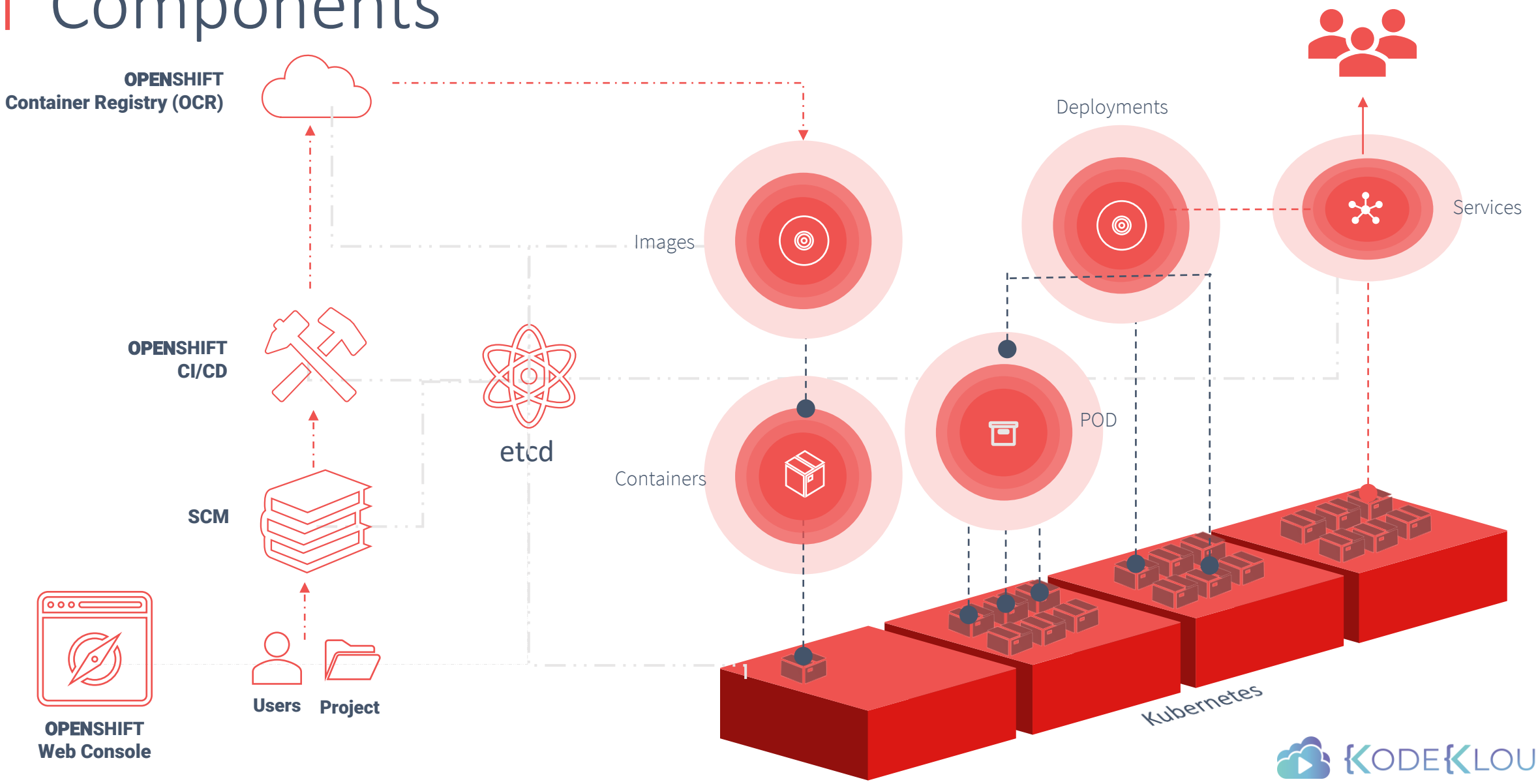

KODEKLOUD

Red Hat

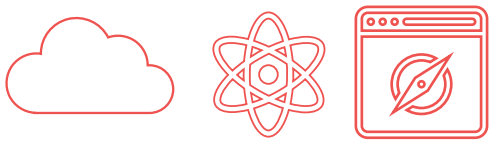
OPENSHIFT

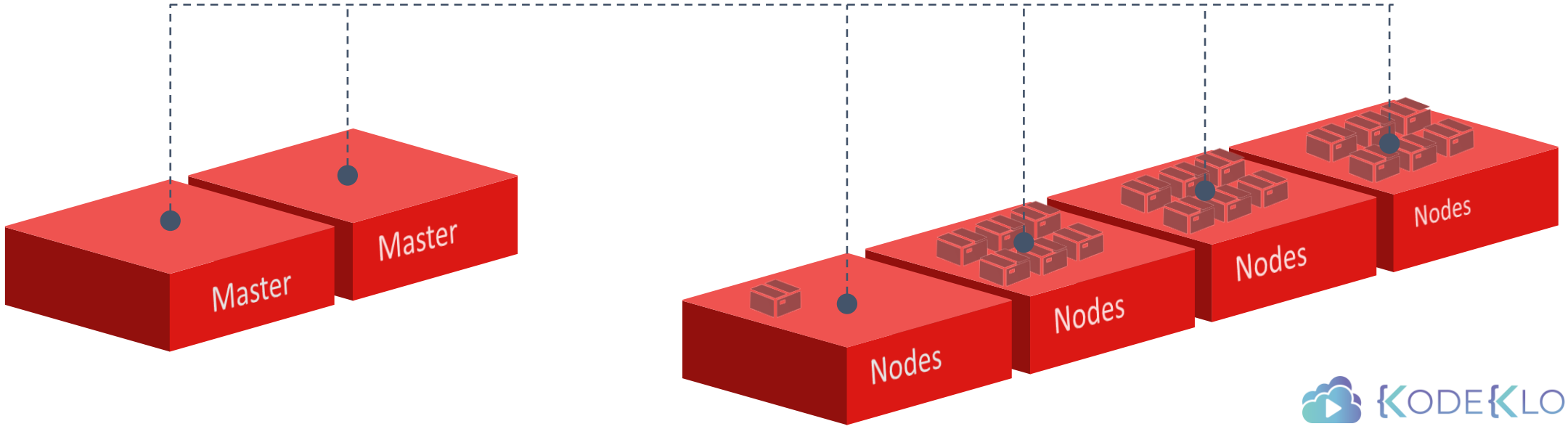
Architecture

Components


OPENSHIFT
Container Registry (OCR)
OPENSHIFT
CI/CD
SCM
OPENSHIFT
Web Console
Users
Project
etcd
Images
Containers
POD
Deployments
Services
Kubernetes

Master - Node





KODEKLOUD


KODEKLOUD

Red Hat

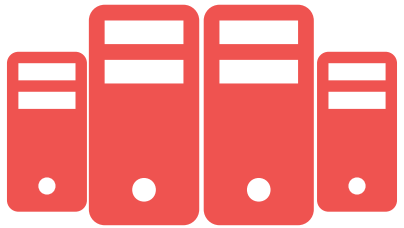
OPENSHIFT

Setup - Minishift

Setup **OPEN**SHIFT

All-in-One

Single Master
Multiple Nodes

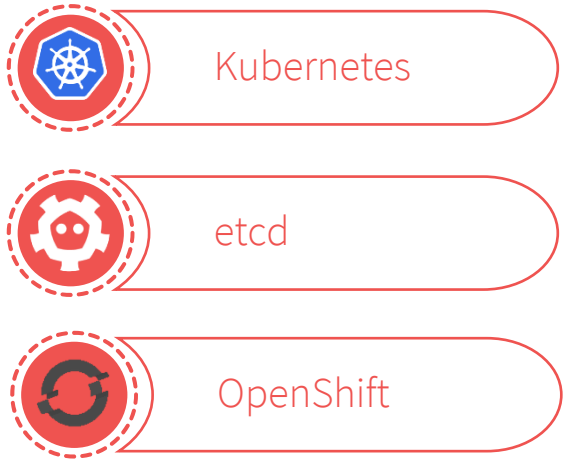


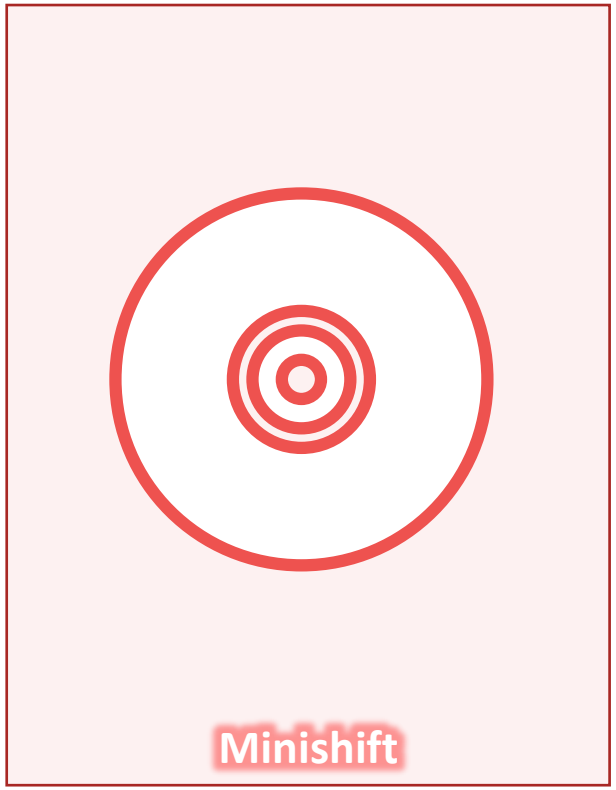
Multiple Master
Multiple Nodes

- **On-Premise**
- **Cloud**
- **Package Manager (RPM)**
- **Containerized (Docker)**

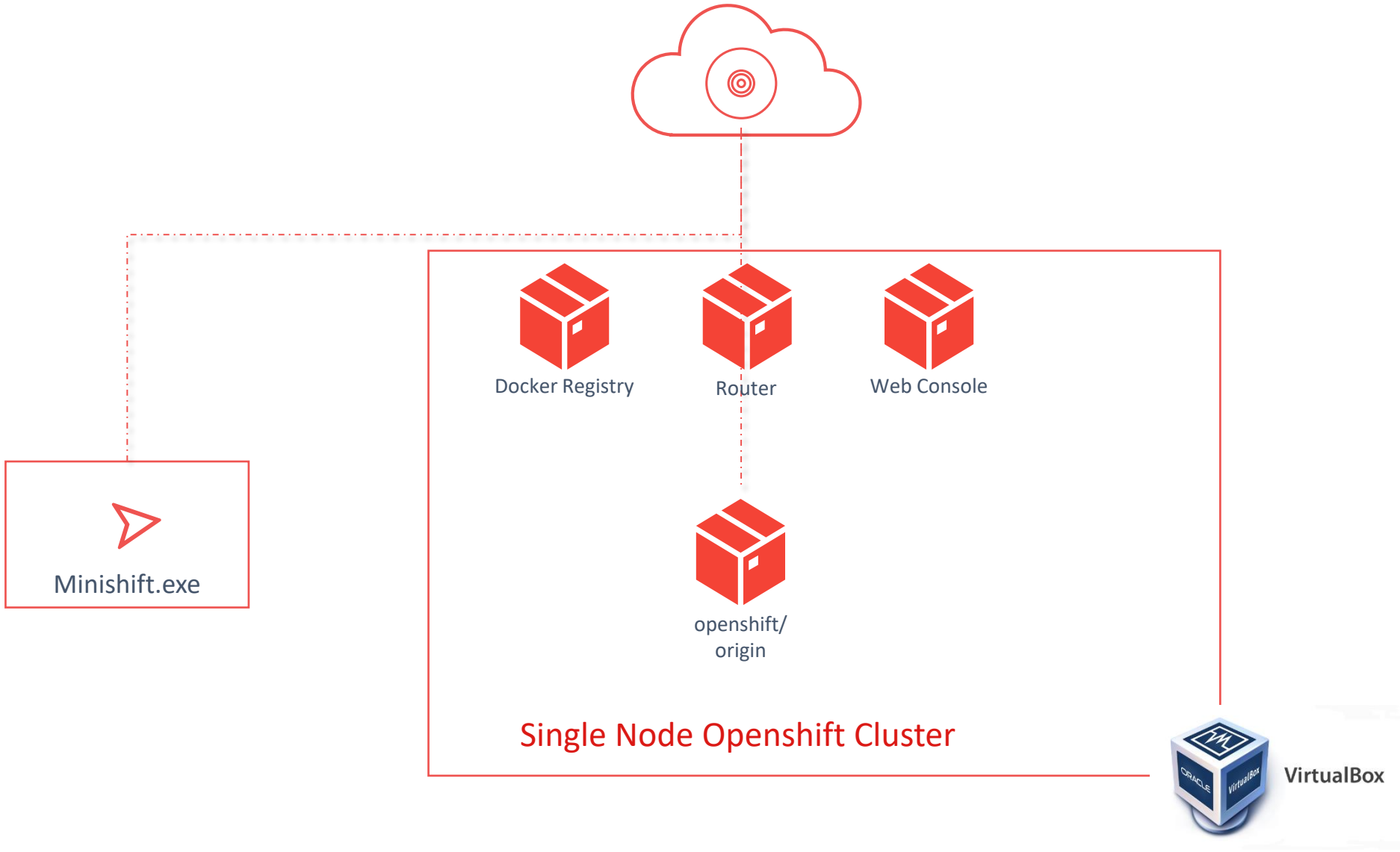
KODEKLOUD

All-in-One





KODEKLOUD


Docker Registry
Router
Web Console
Minishift.exe
openshift/
origin
Single Node Openshift Cluster
VirtualBox

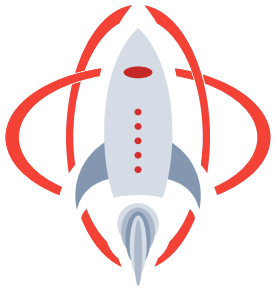

KODEKLOUD

Red Hat

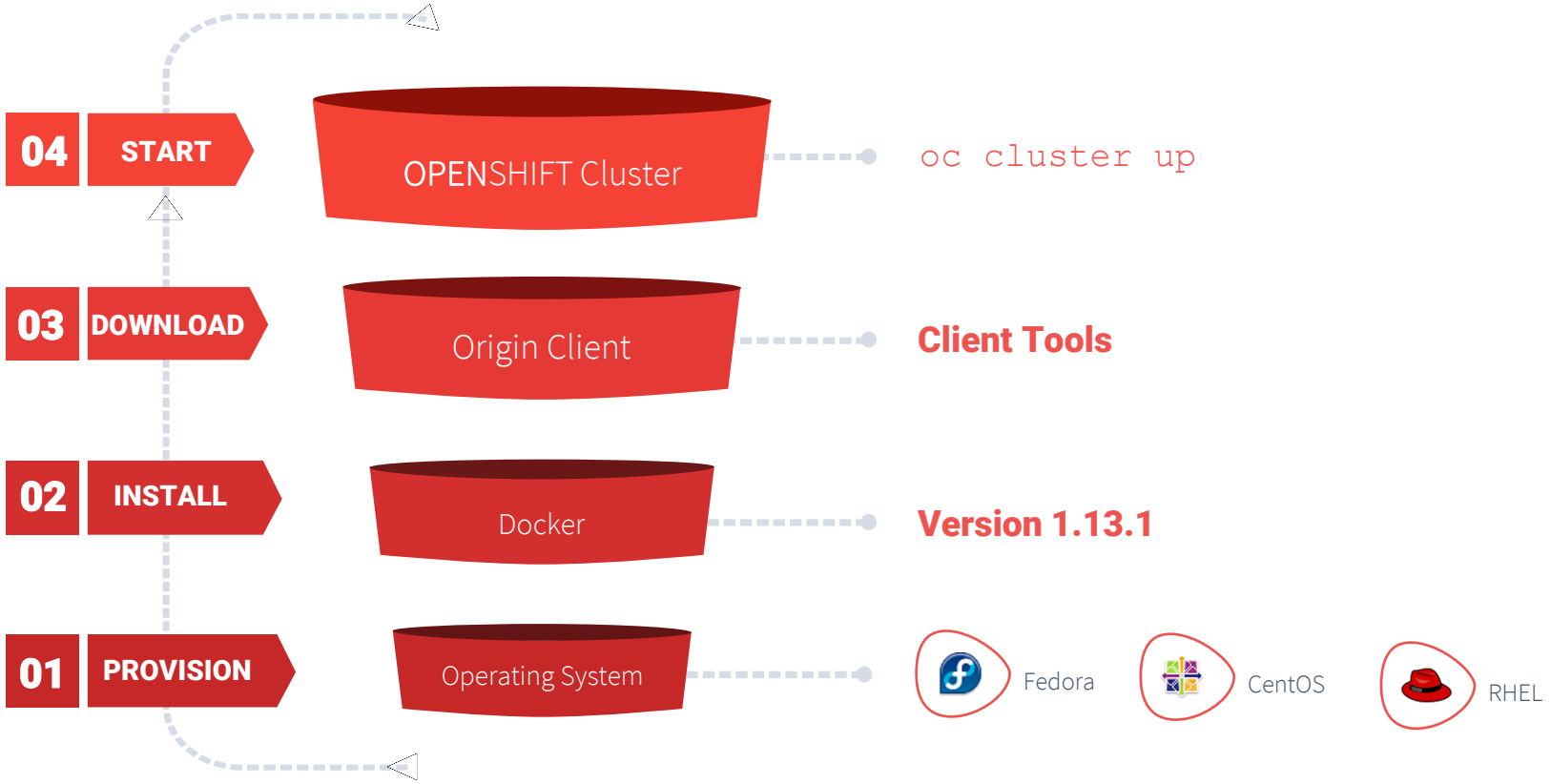
OPENSHIFT

Setup – Using Docker

All-in-One



without Minishift



Containerized (Docker)


KODEKLOUD

Red Hat

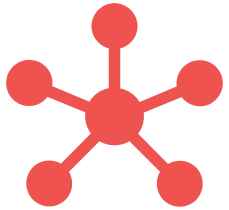
OPENSHIFT

Web Console and CLI

Management Tools

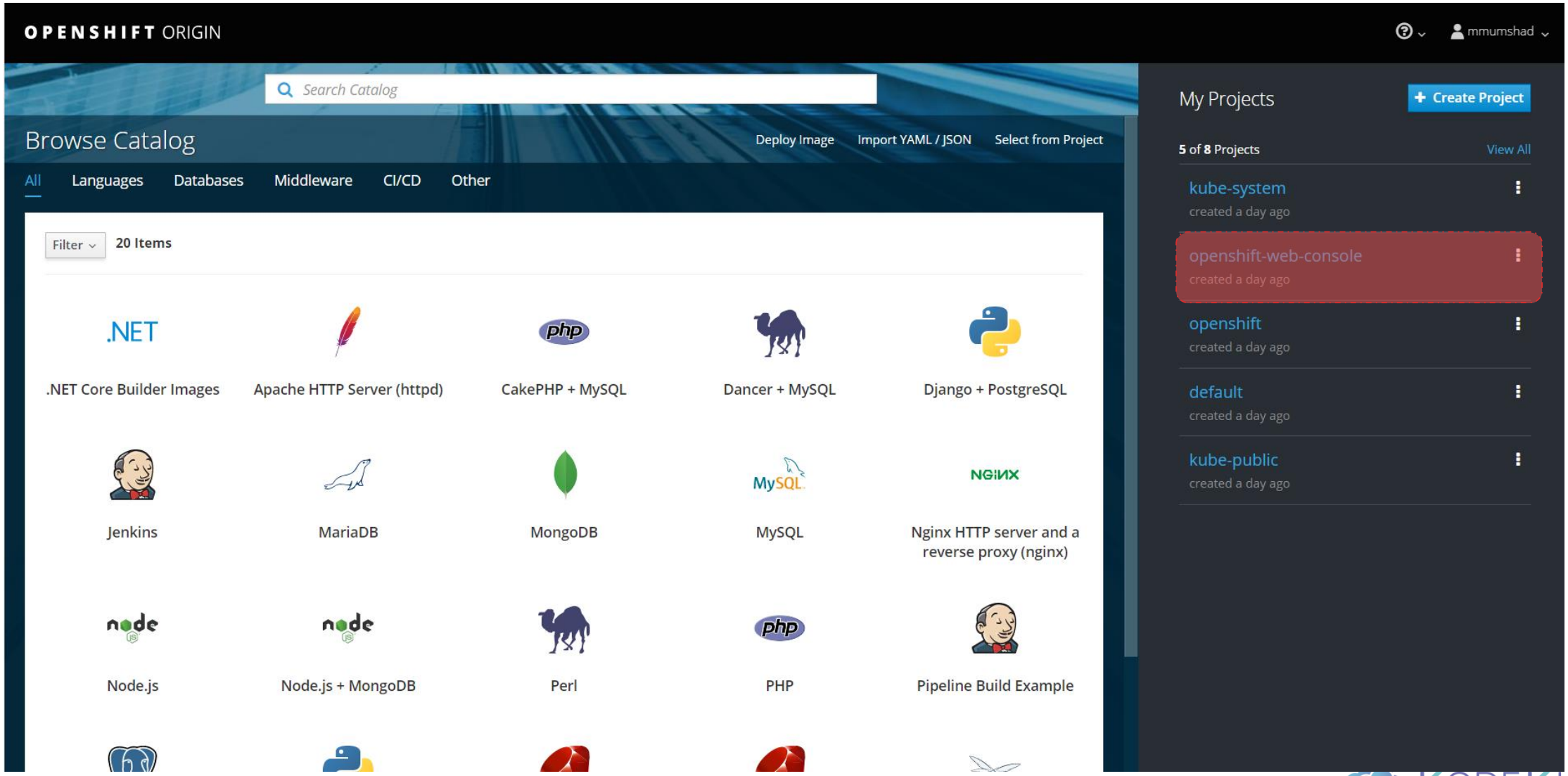
Web Console

CLI



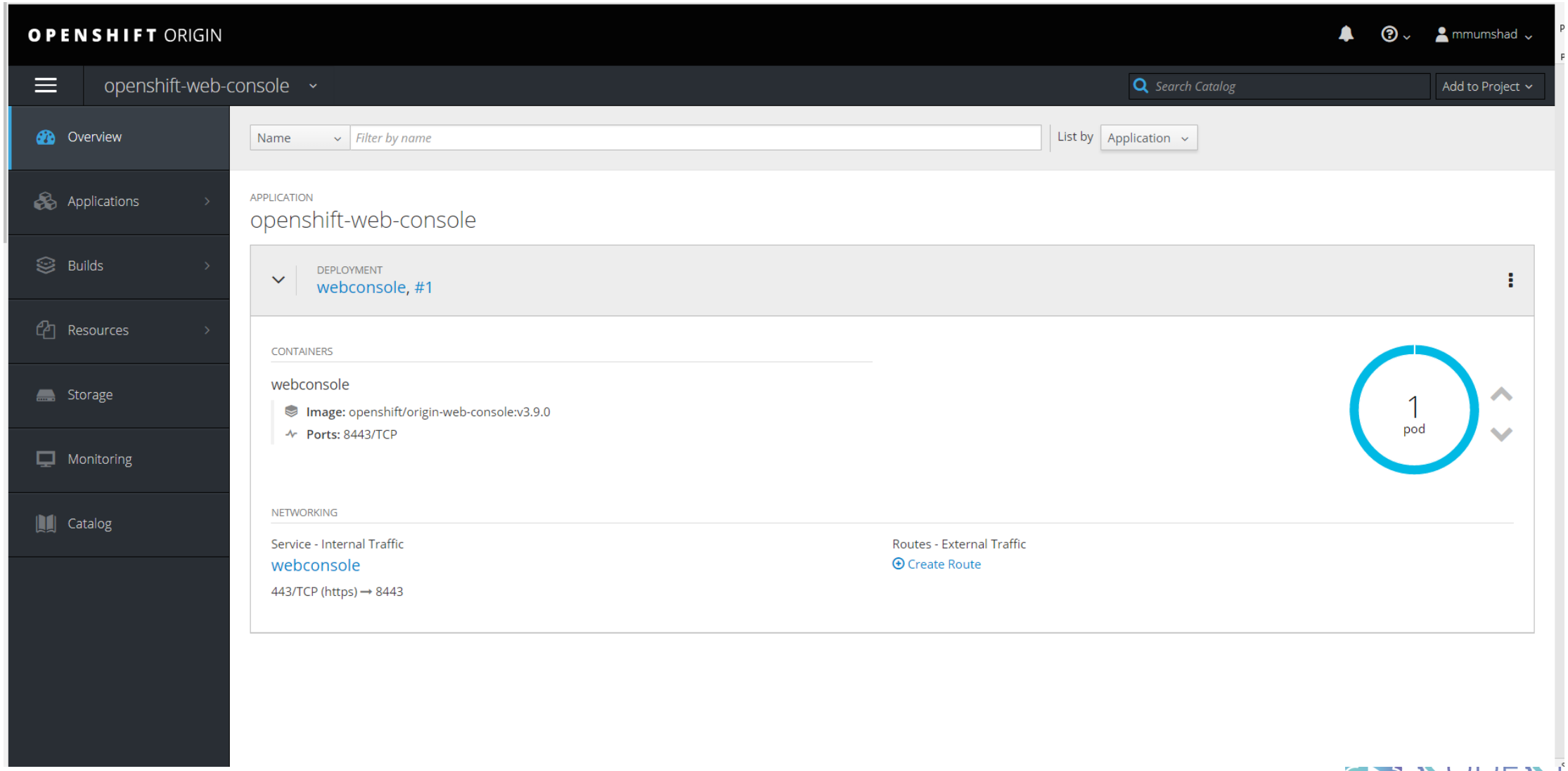
REST API

Web Console - Homepage



KODEKLOUD

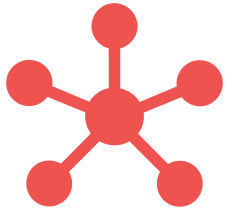
Web Console - Project View



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
```

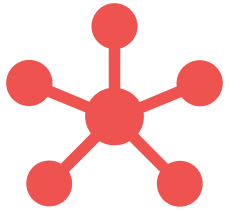
```
User, developer, logged out of https://openshift.example.com
```

KODEKLOUD

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
```


27pZjHWn9GU5B5De4H3ICP6ZBd_7txQJMvz79

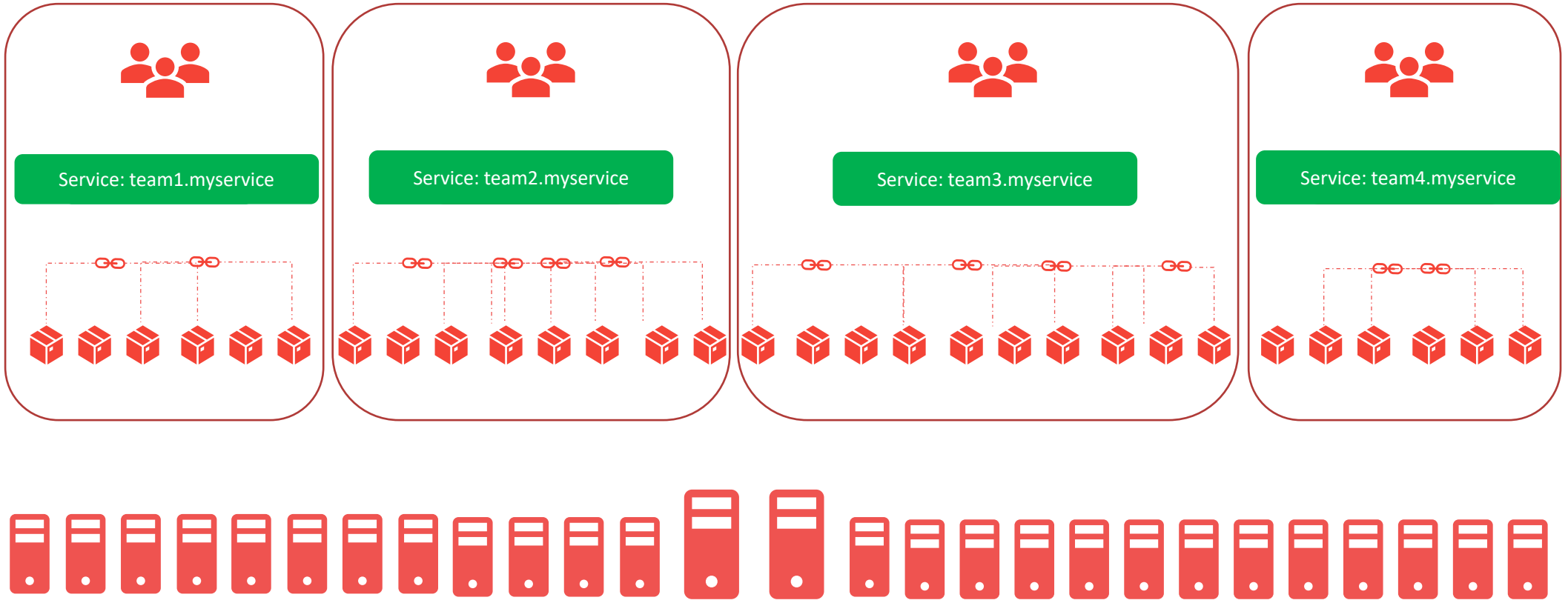

KODEKLOUD

Red Hat

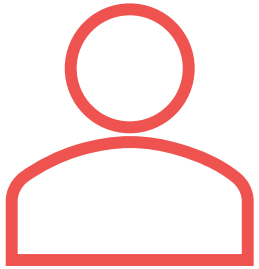
OPENSHIFT

Projects and Users

Projects



Users



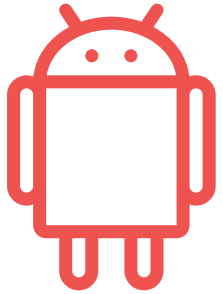
Regular

developer

System

system:admin

system:master



Service

system:serviceaccount:proj1:db_user

KODEKLOUD

OAuth Server

Allow All

Deny All

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


KODEKLOUD

Red Hat **OPEN**SHIFT

Pre-Requisite

Source Code Management

Source Code Management

Application

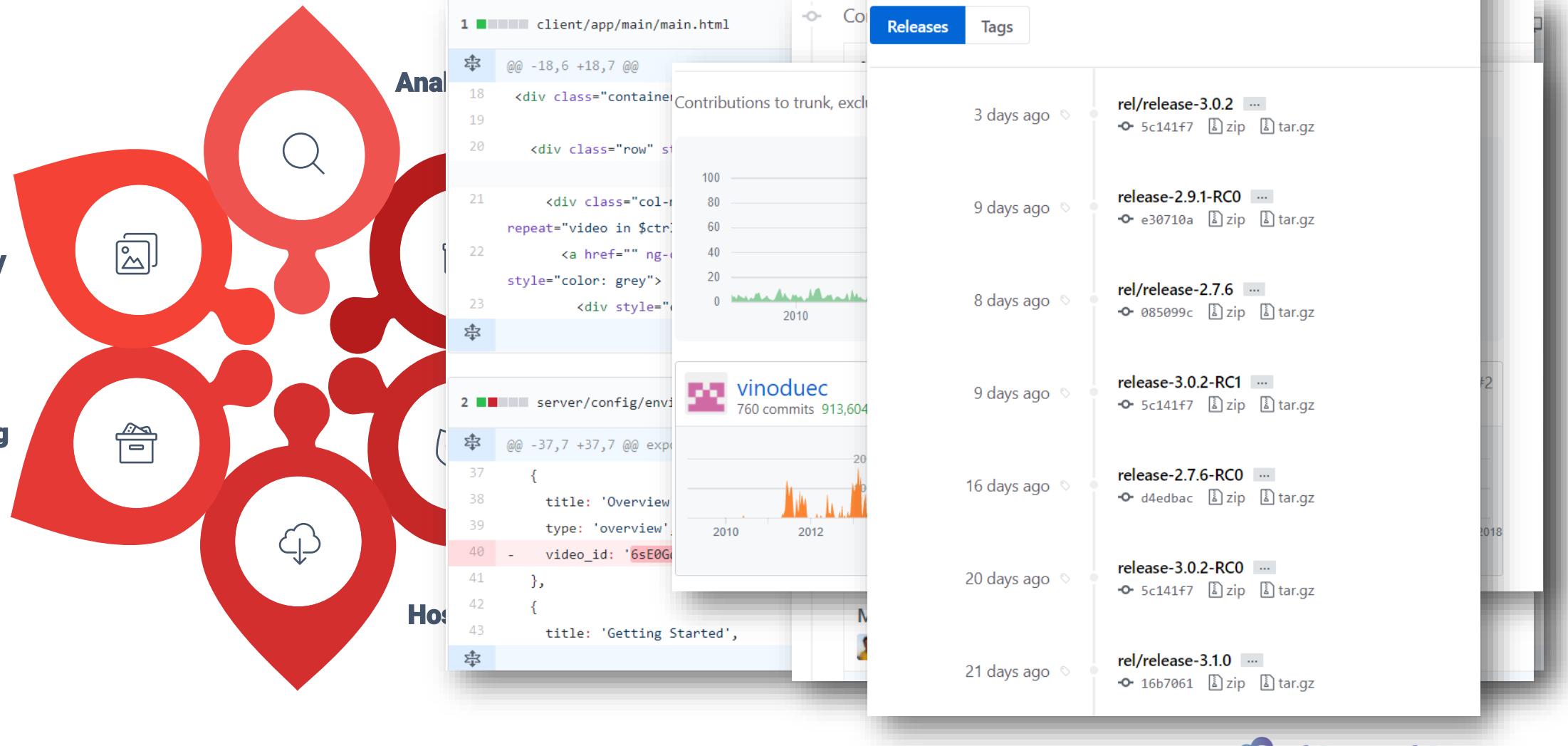
Web Console

API Backend


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

KODEKLOUD

Source Code Management



KODEKLOUD

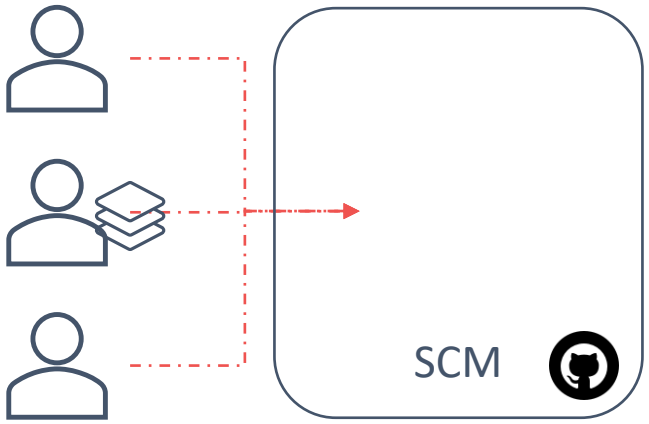

KODEKLOUD

Red Hat **OPEN**SHIFT

Pre-Requisite

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
```

KODEKLOUD

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
```

01 Dockerfile

Install Platform

```
> docker build Dockerfile
```

03

Install and Configure Pre-requisites

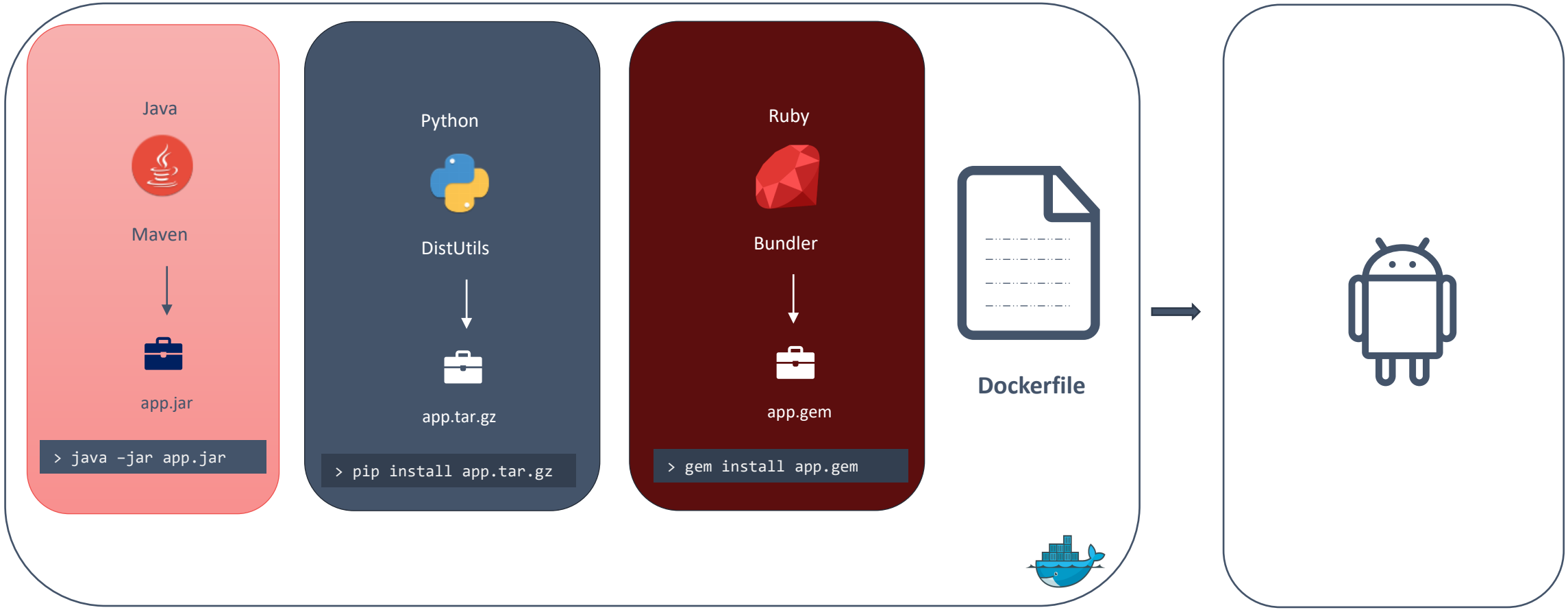
Configure Services

04

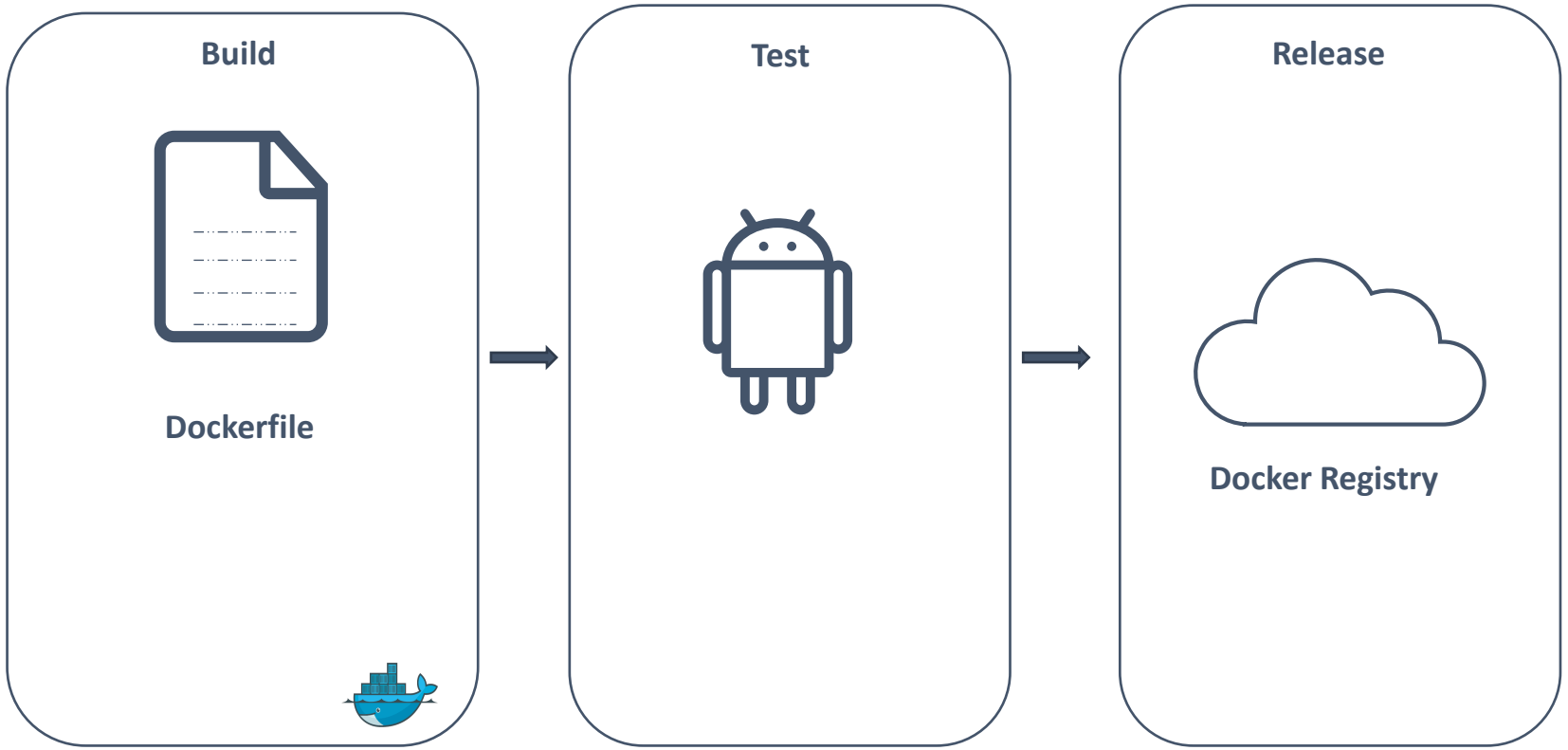
Start Services

```
> docker run app
```

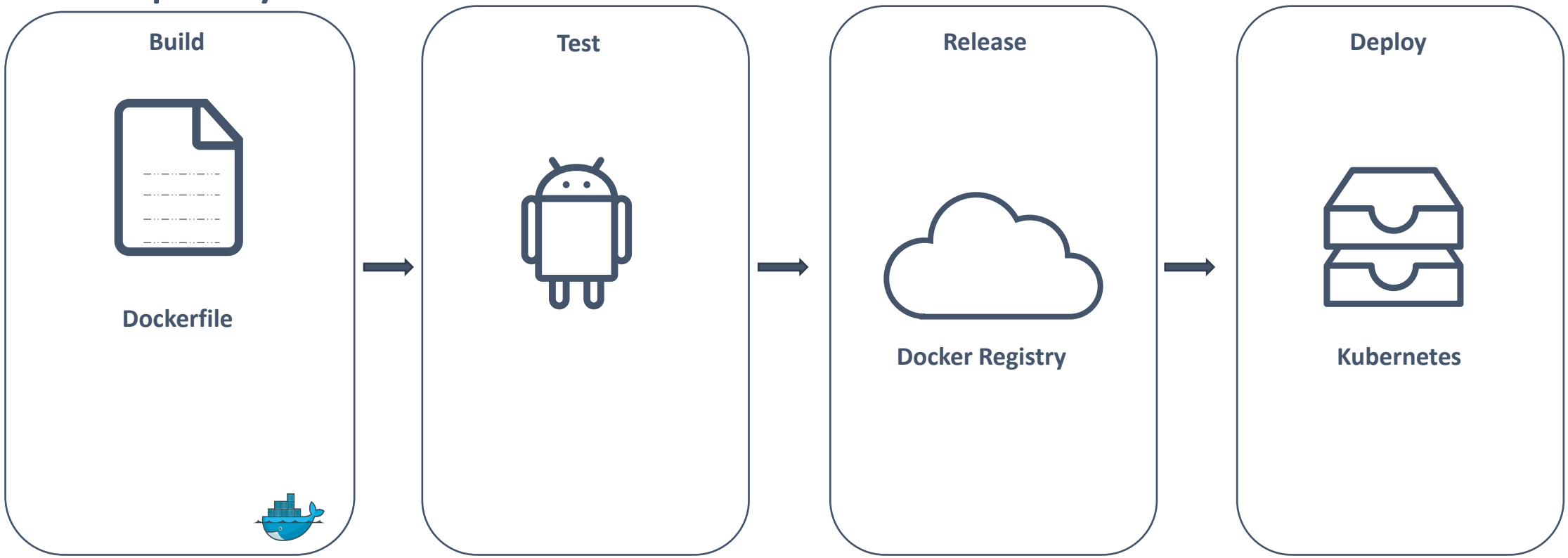
Test


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

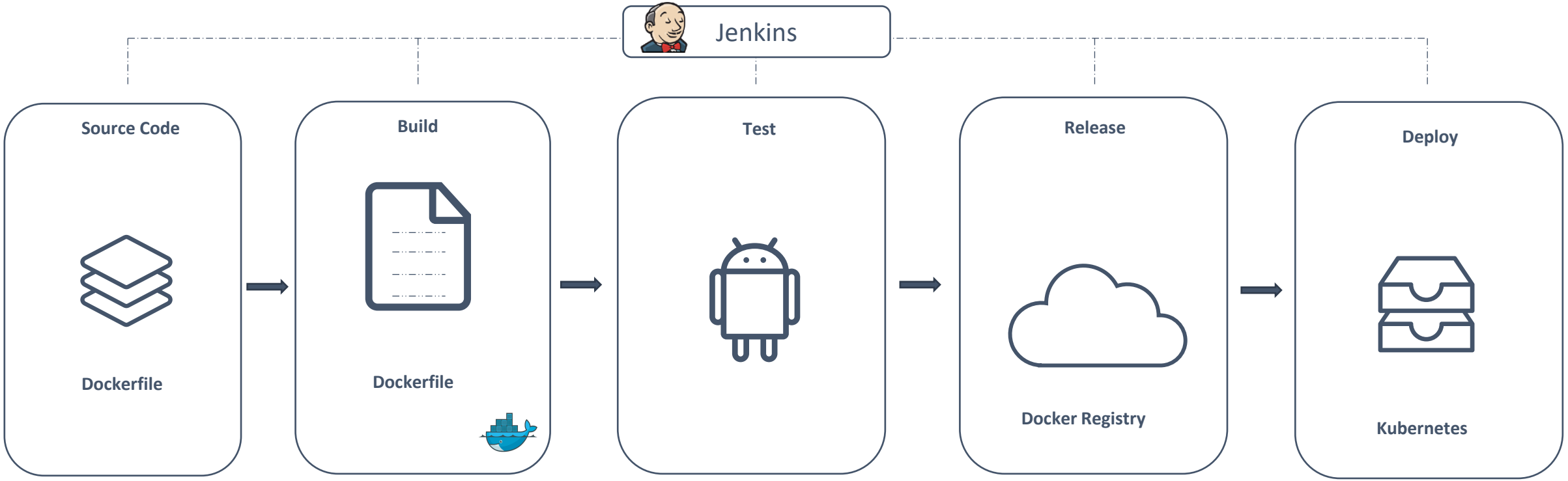
Release



Deploy



Build Pipeline



Continuous Integration

Continuous Delivery

Continuous Deployment


KODEKLOUD

Red Hat

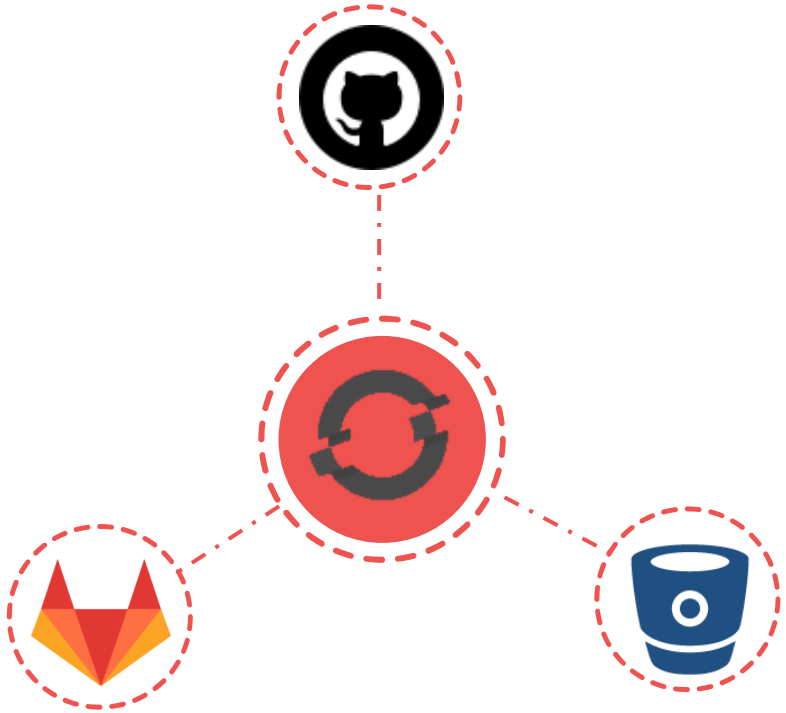
OPENSHIFT

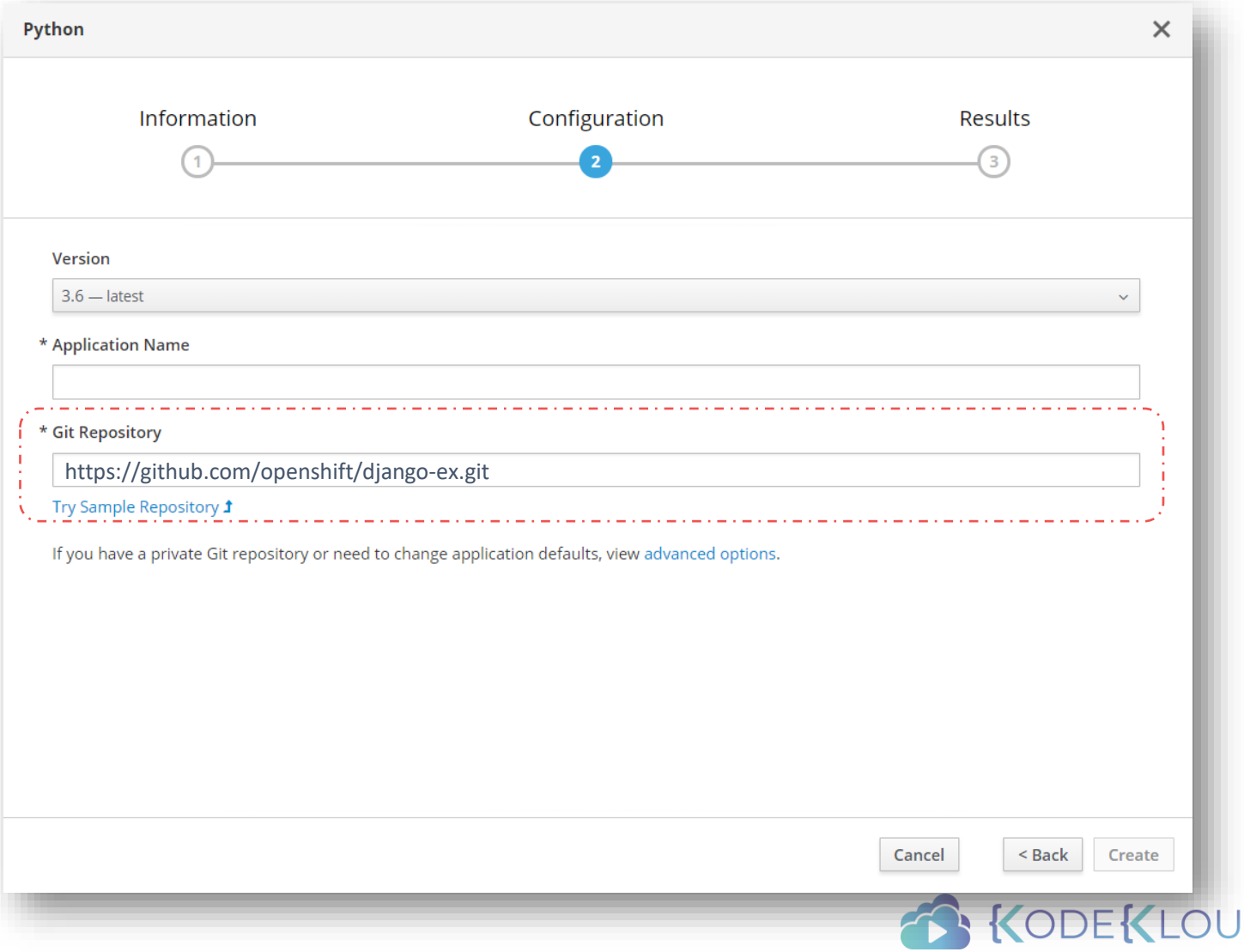
SCM, Builds and Deployments in **OPEN**SHIFT

Pre-Requisites

- Docker
- Docker Image
- Docker Registry

Source Code Management

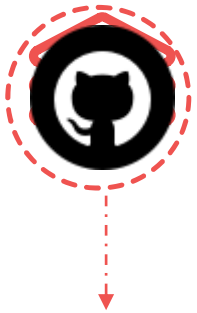




KODEKLOUD

Build

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



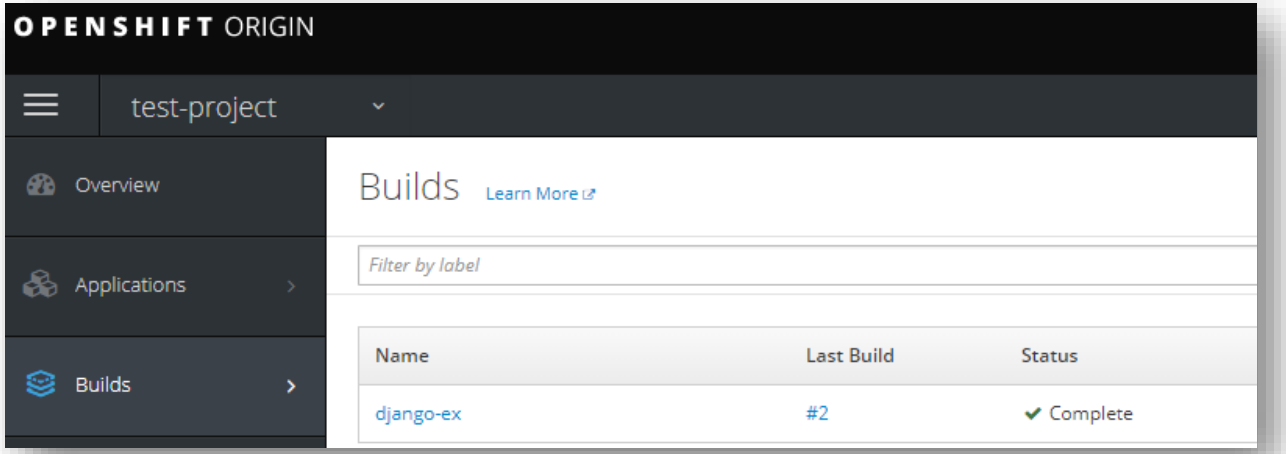
01 Create Build

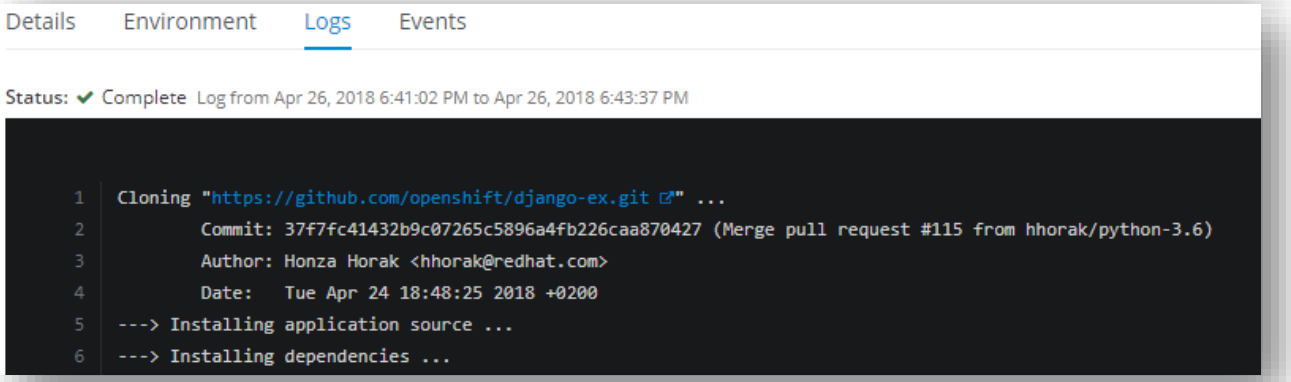
02 Download Source

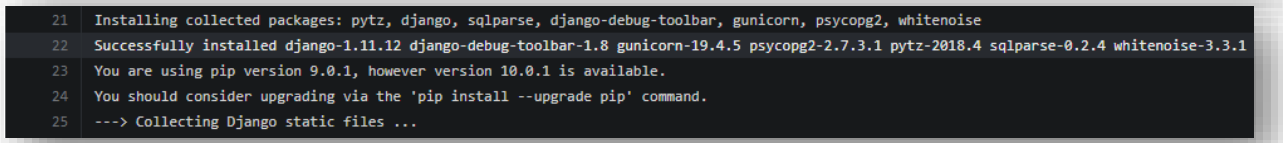
03 Build Image

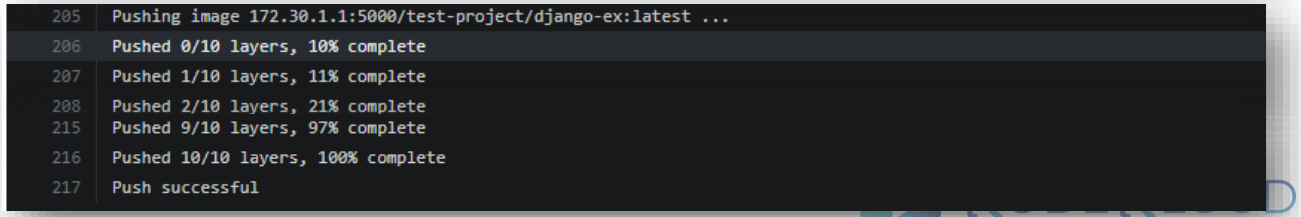
04 Push to Registry

Docker Registry







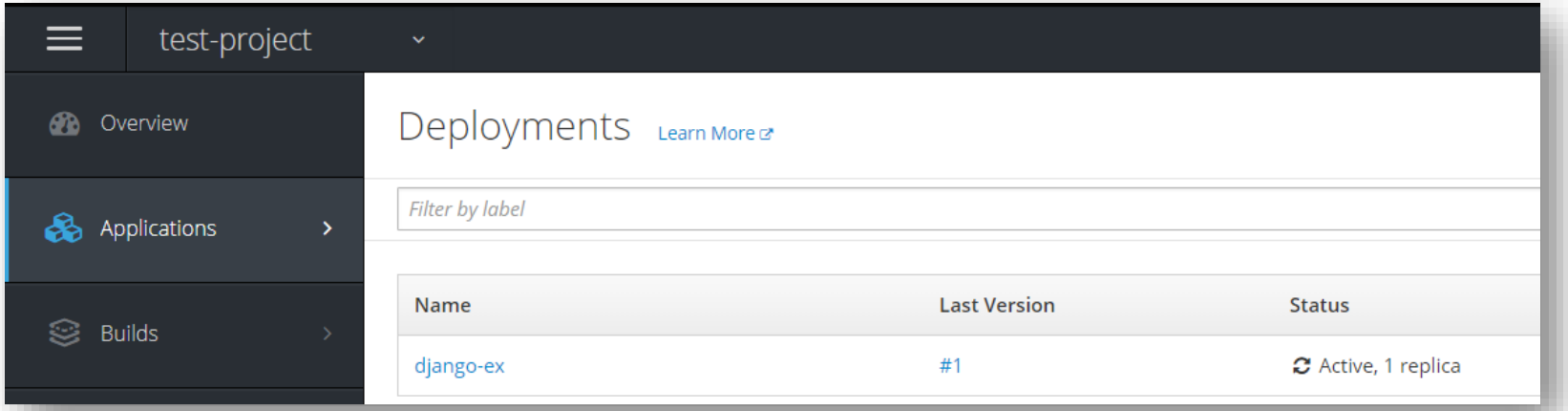


Deployment

04 **Push to Registry**

Docker Registry

05 **Deploy**




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

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

KODEKLOUD

Deployment

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

01 Create Build

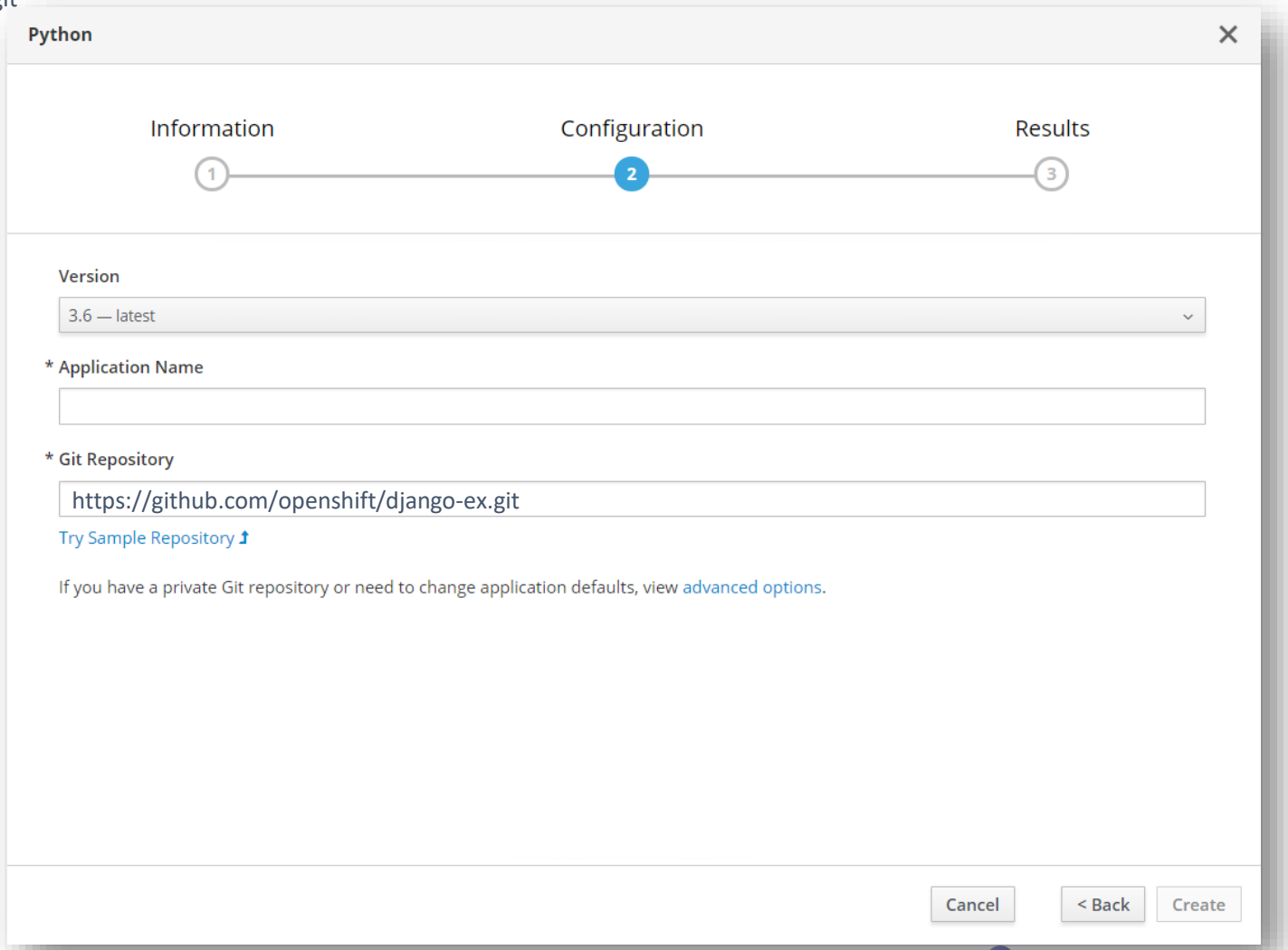
02 Download Source

03 Build Image

04 Push to Registry

Docker Registry

05 Deploy



KODEKLOUD


KODEKLOUD

Red Hat

OPENSHIFT

Builds in **OPEN**SHIFT

Objectives

- Build Strategies
- Create a new Build Configuration

Pre-Requisites

- YAML Files

Build Strategy

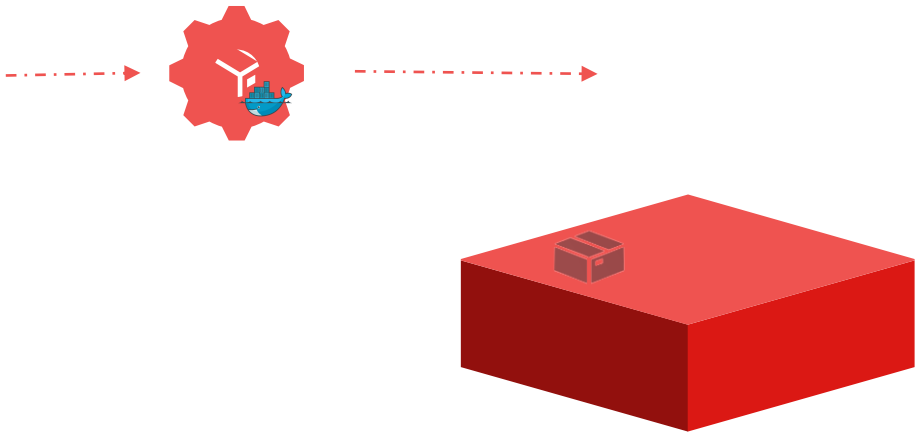
1 Docker Build

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)
```



KODEKLOUD

Build Strategy

2 Source-To-Image (S2I)

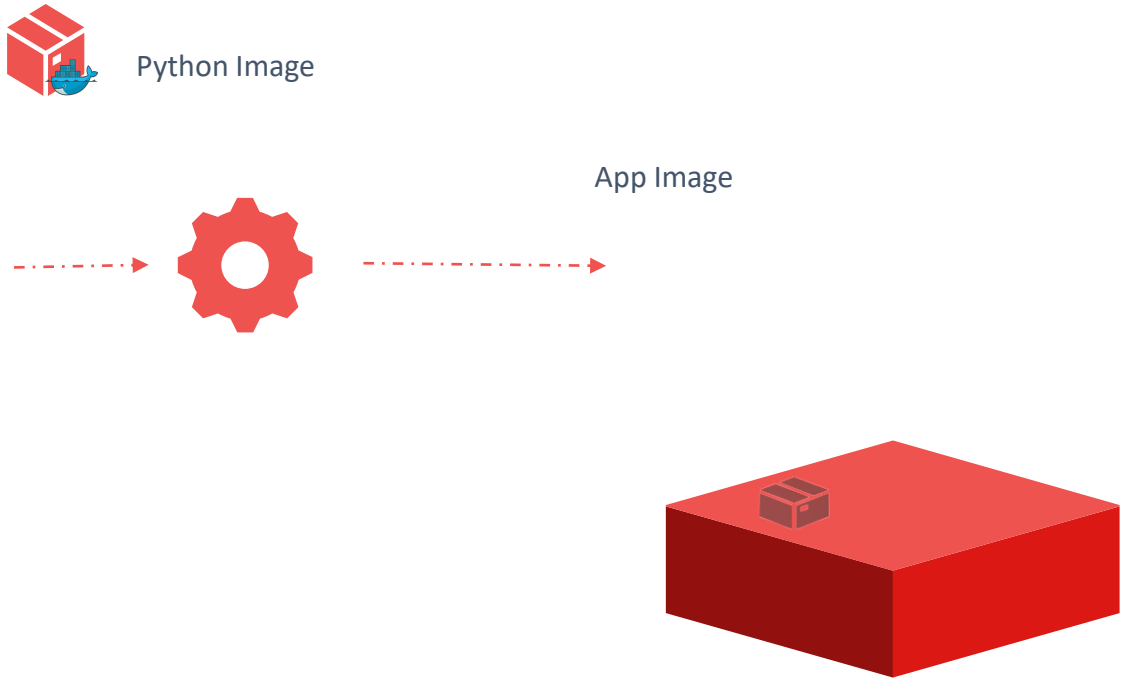
app.py

```python
import os
from flask import Flask
app = Flask(__name__)

@app.route("/")
def main():
    return "Welcome!"

@app.route('/how are you')
def hello():
    return 'I am good, how about you?'

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8080)
```



Build Strategy

3 Custom Build

Java

app.jar

Python

app.tar.gz

Ruby

app.gem

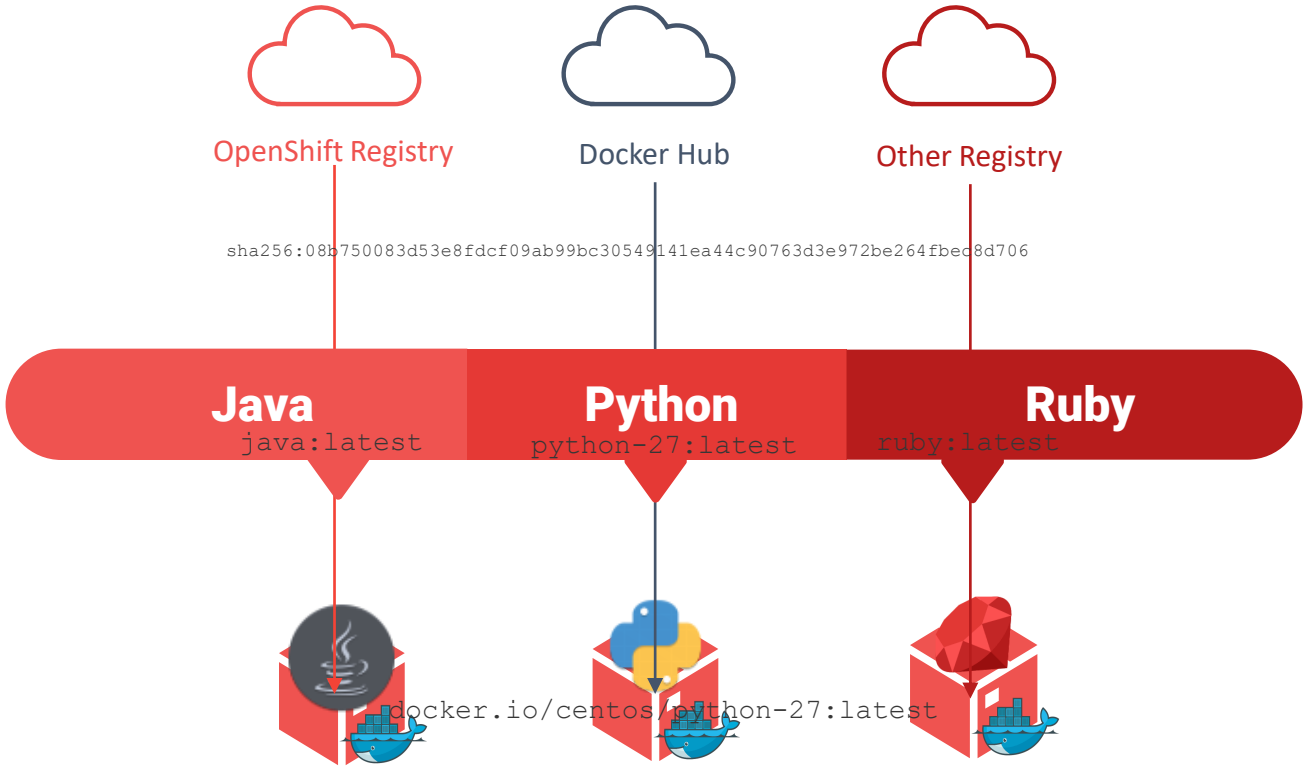
Image Stream

Python Image

Code

Application Image

Image Streams

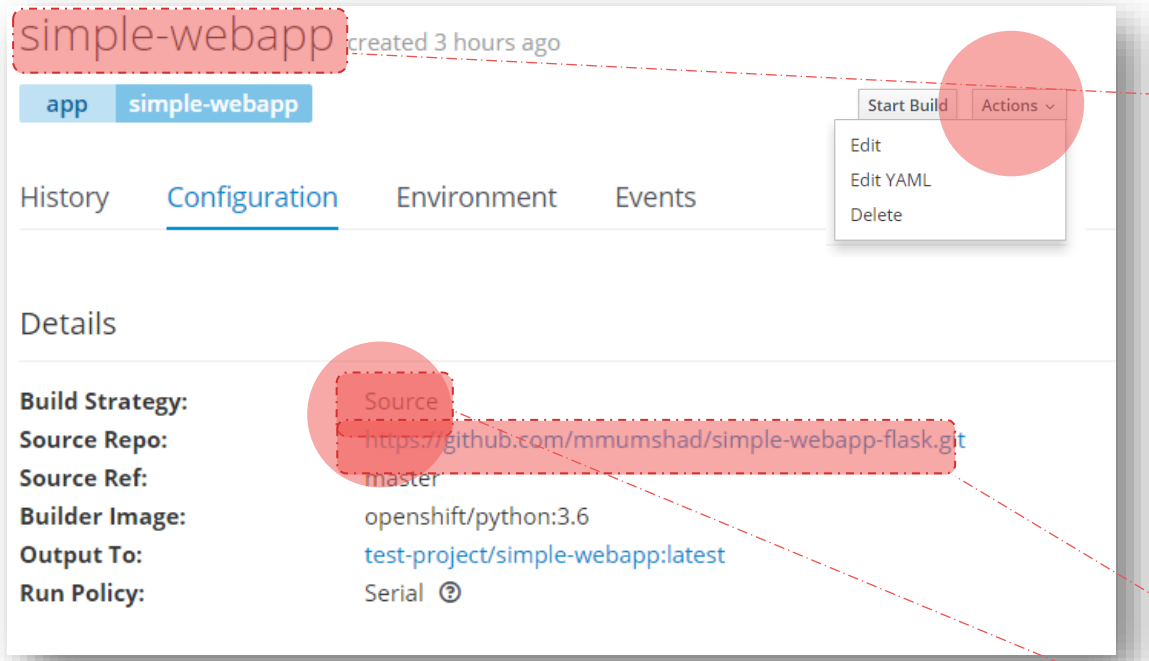


172.30.1.1:5000/myproject/java:latest

Other-registry/ruby/ruby:2.0

KODEKLOUD

View Build Config



S2i-build-config.yaml

```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

```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

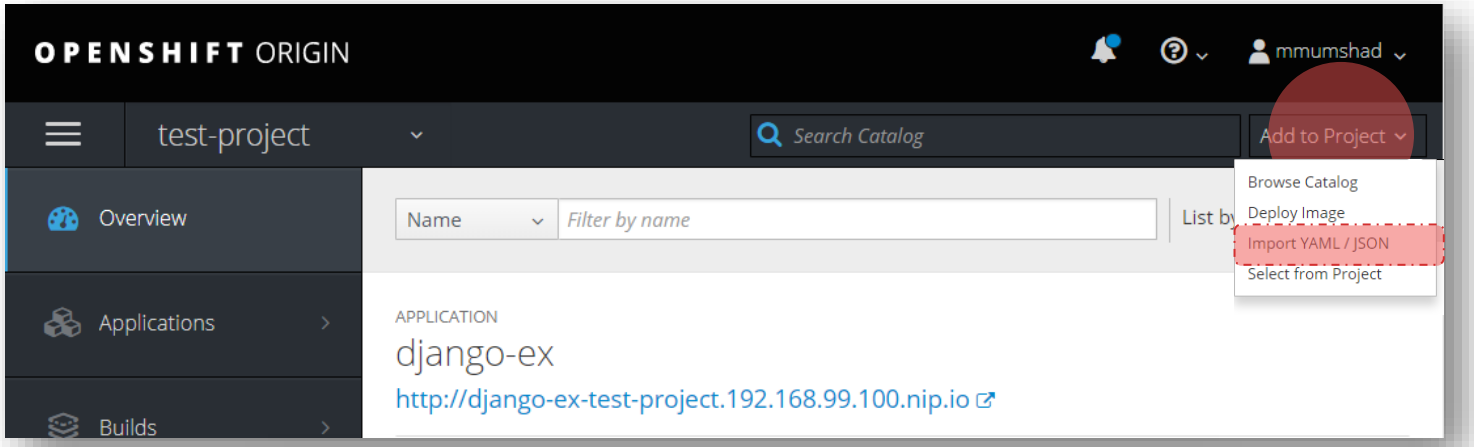
```yaml
kind: "BuildConfig"
apiVersion: "v1"
metadata:
  name: "simple-webapp-docker"
spec:
  runPolicy: "Serial"
  triggers:
    -
      type: "GitHub"
      github:
        secret: "b5e471d57f79f52e"
    - type: "Generic"
      generic:
        secret: "4be5b473f9985dcf"
    -
      type: "ImageChange"
  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"
```

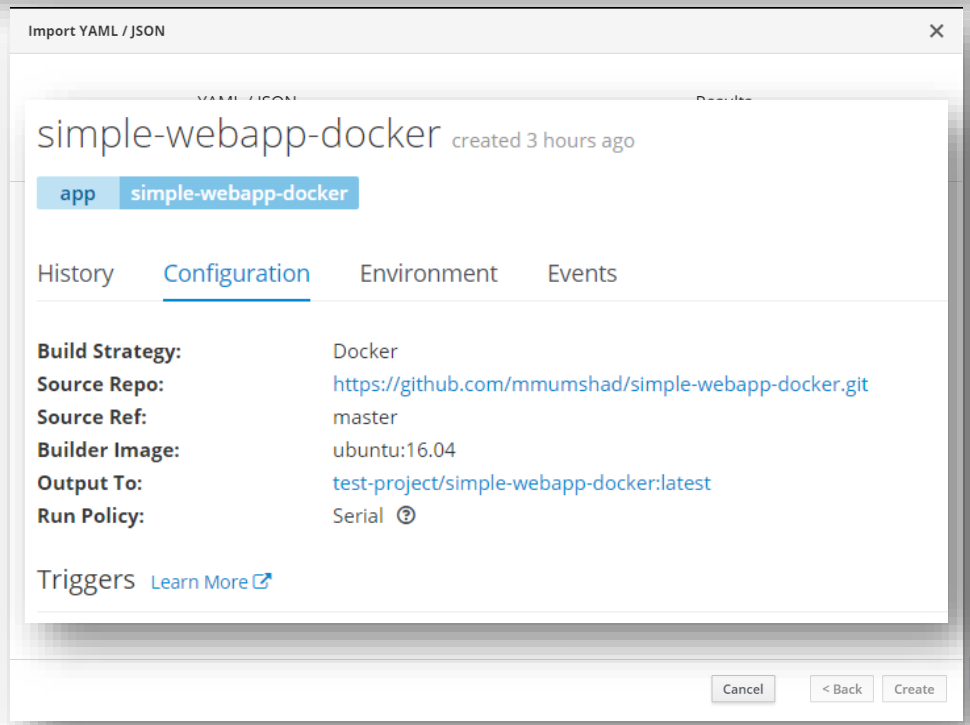
Create Build Configuration

Docker-build-config.yaml

```yaml
kind: "BuildConfig"
apiVersion: "v1"
metadata:
  name: "simple-webapp-docker"
spec:
  runPolicy: "Serial"
  triggers:
    -
      type: "GitHub"
      github:
        secret:
    - type: "Ge
      generic:
        secret:
    -
      type: "Im
  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"
```

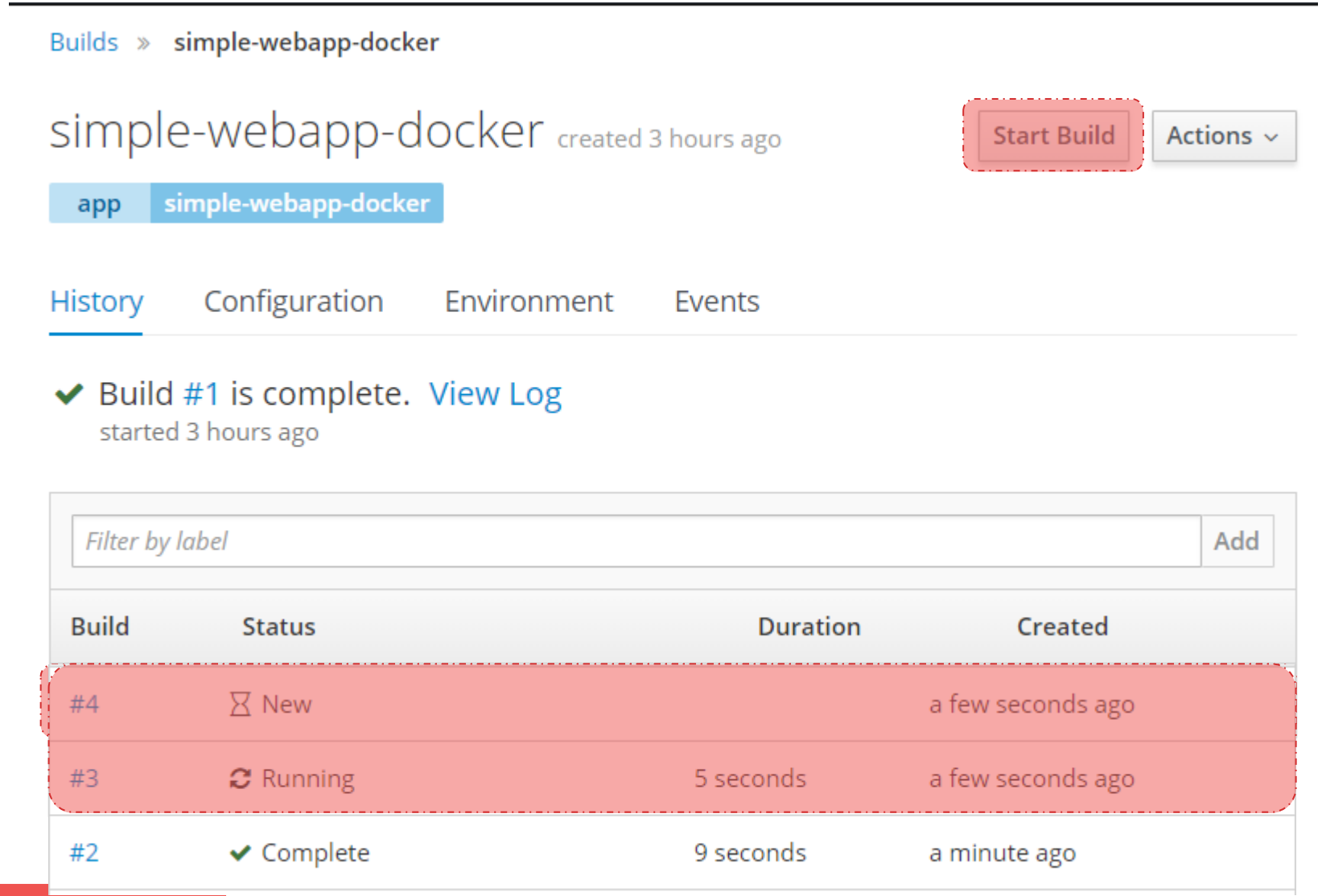
```yaml
apiVersion: build.openshift.io/v1
kind: BuildConfig
metadata:
  name: simple-webapp-docker-2
spec:
  output:
    to:
      kind: ImageStreamTag
      name: 'simple-webapp-docker:latest'
  runPolicy: Serial
  source:
    git:
      ref: master
      uri: 'https://github.com/mmumshad/simple-webapp-docker.git'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





DEKLOUD

Start a build



KODEKLOUD

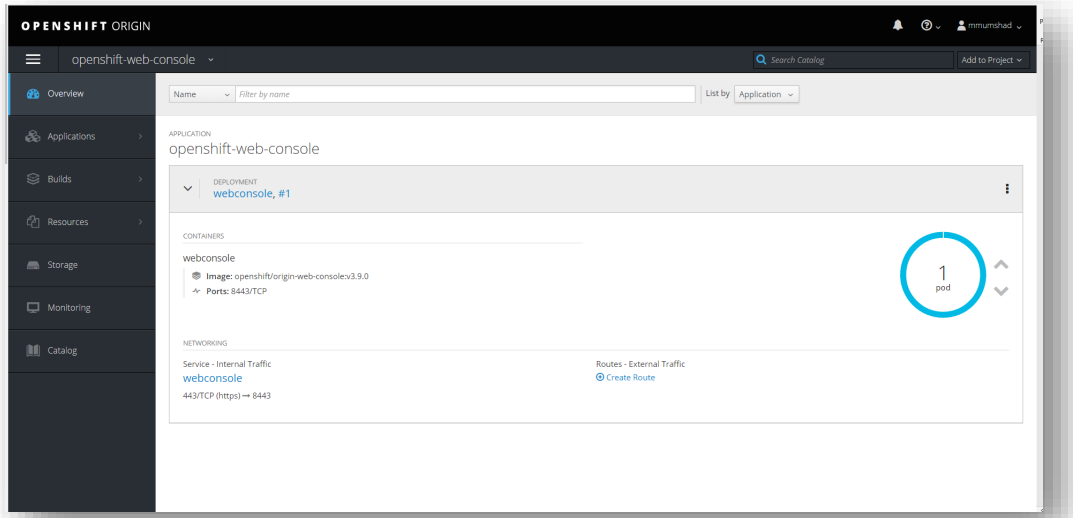

KODEKLOUD

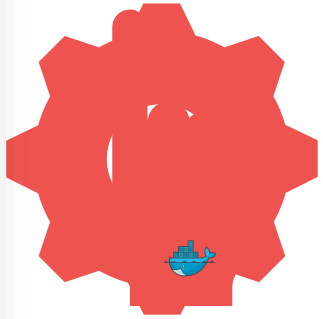
Red Hat

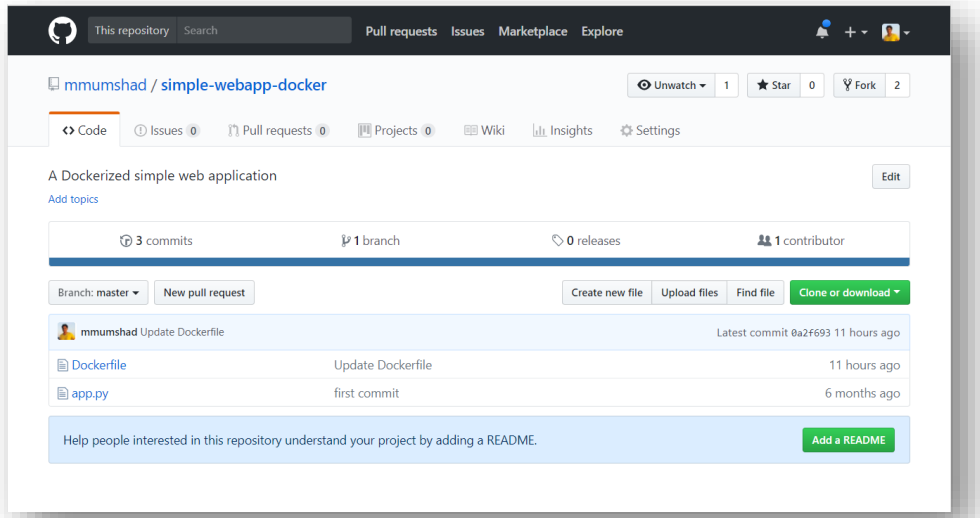
OPENSHIFT

Build Triggers

Trigger Build







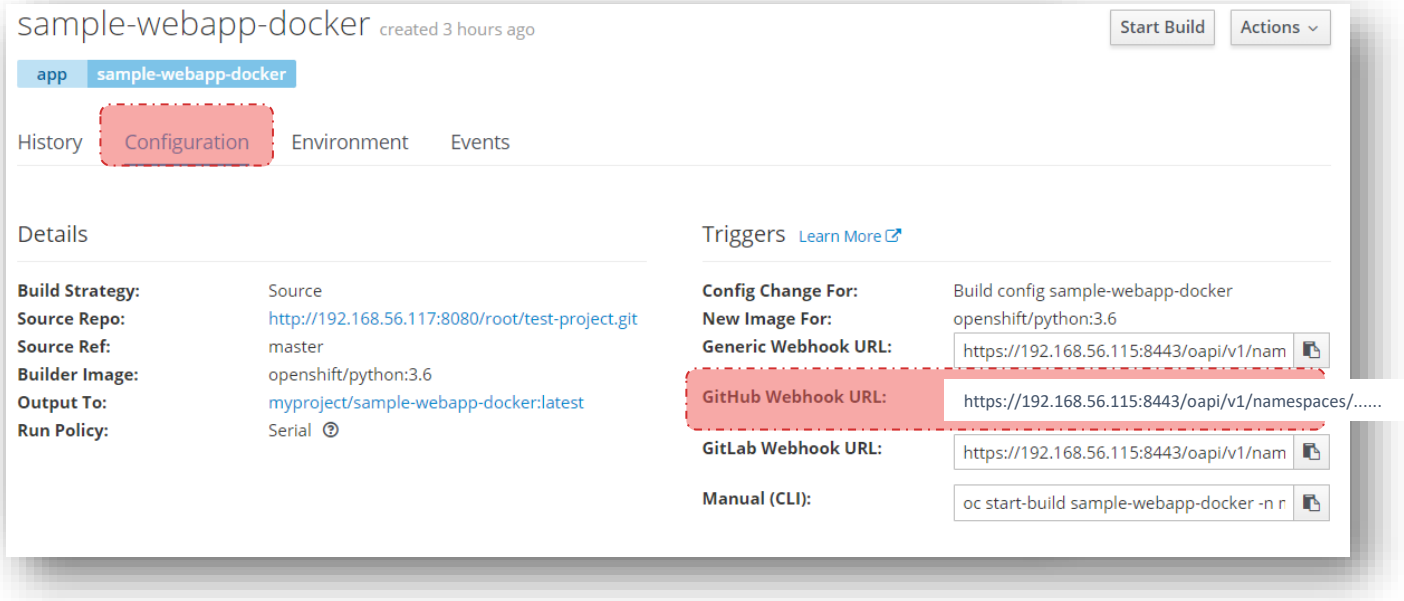
Openshift

Automated Build

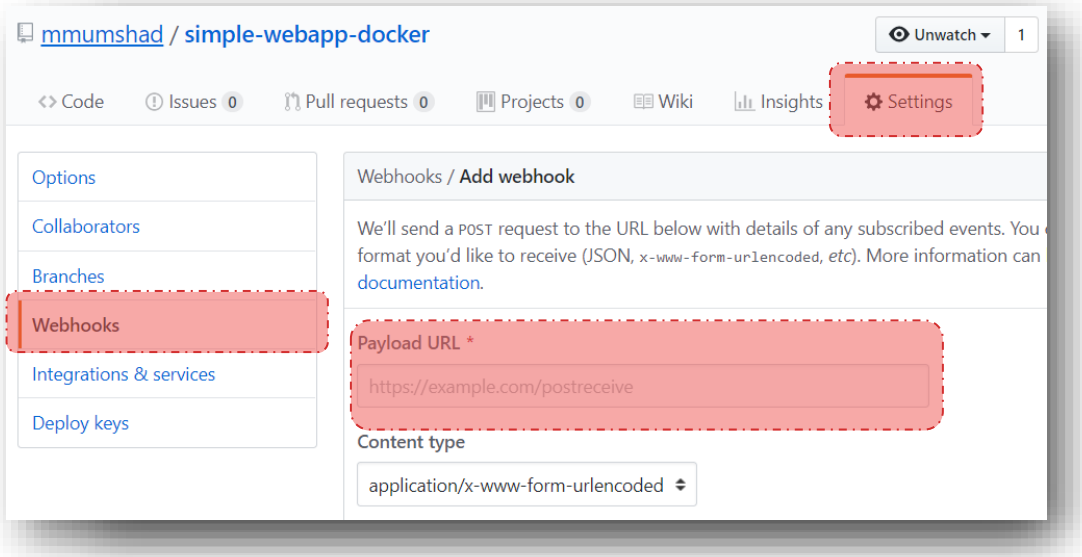
Code Repository

KODEKLOUD

Webhook



Openshift



Code Repository

KODEKLOUD

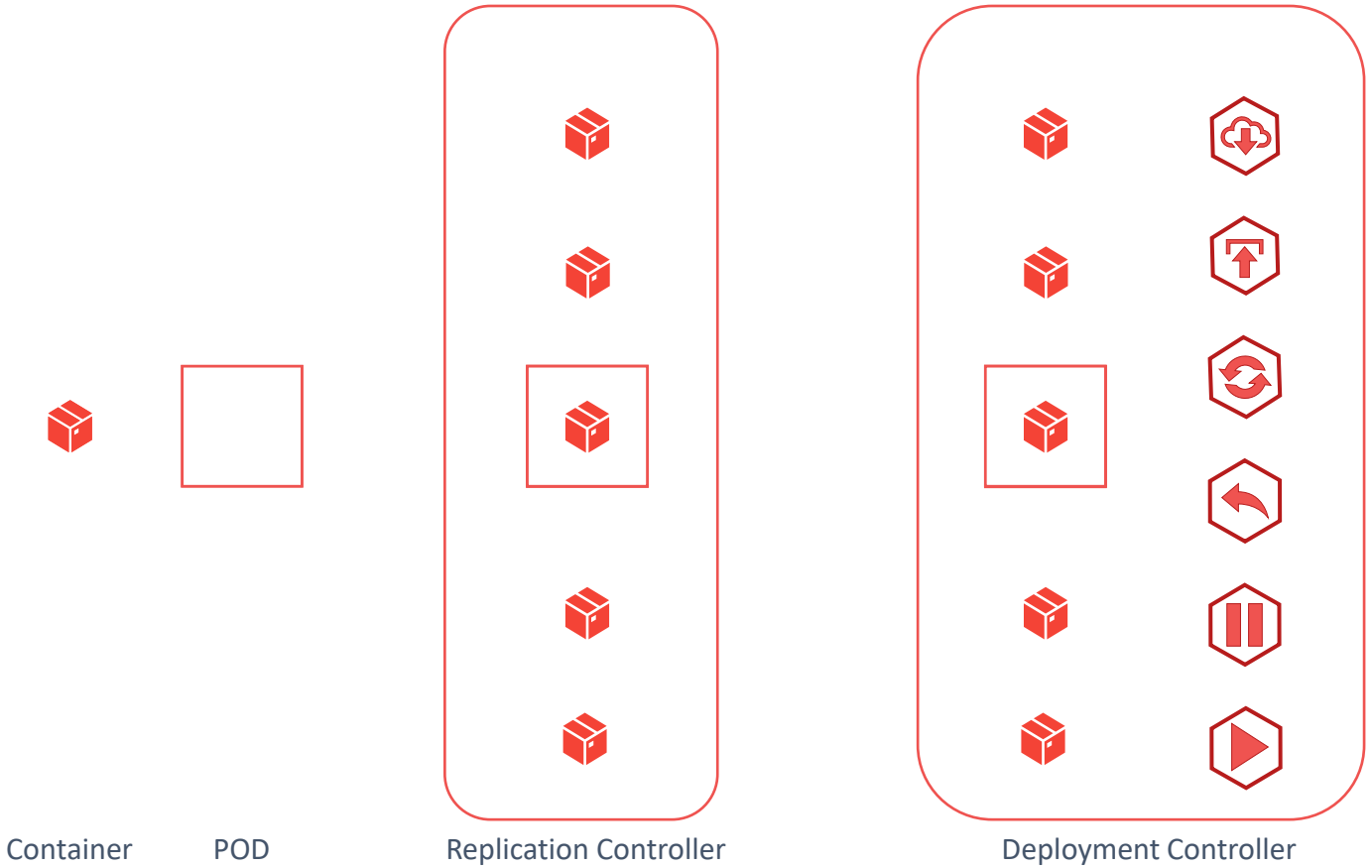

KODEKLOUD

Red Hat

OPENSHIFT

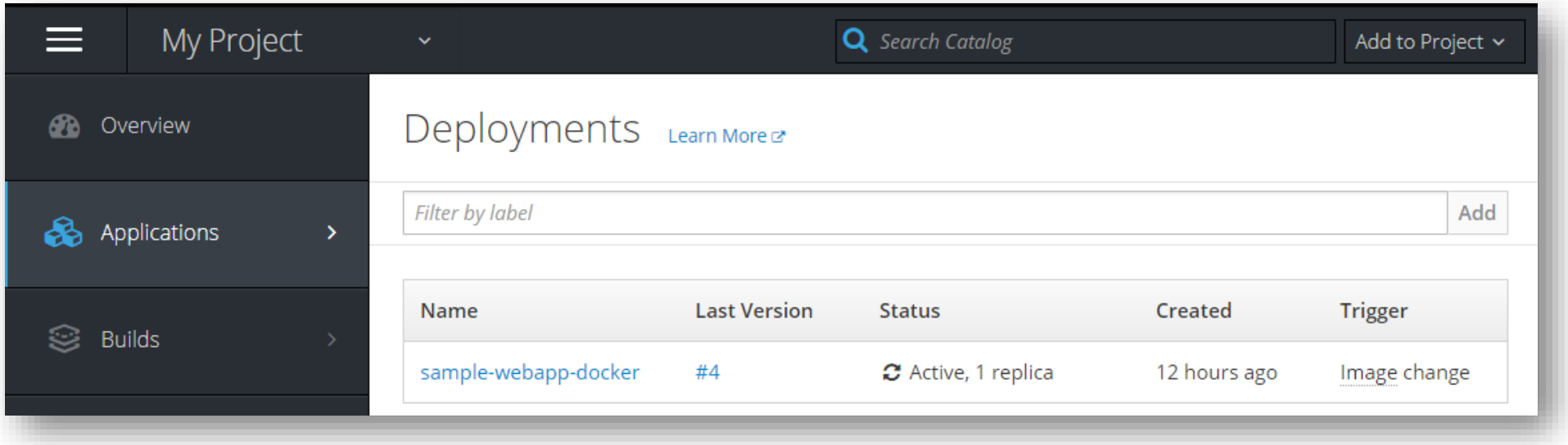
Deployments

Deployment Controller



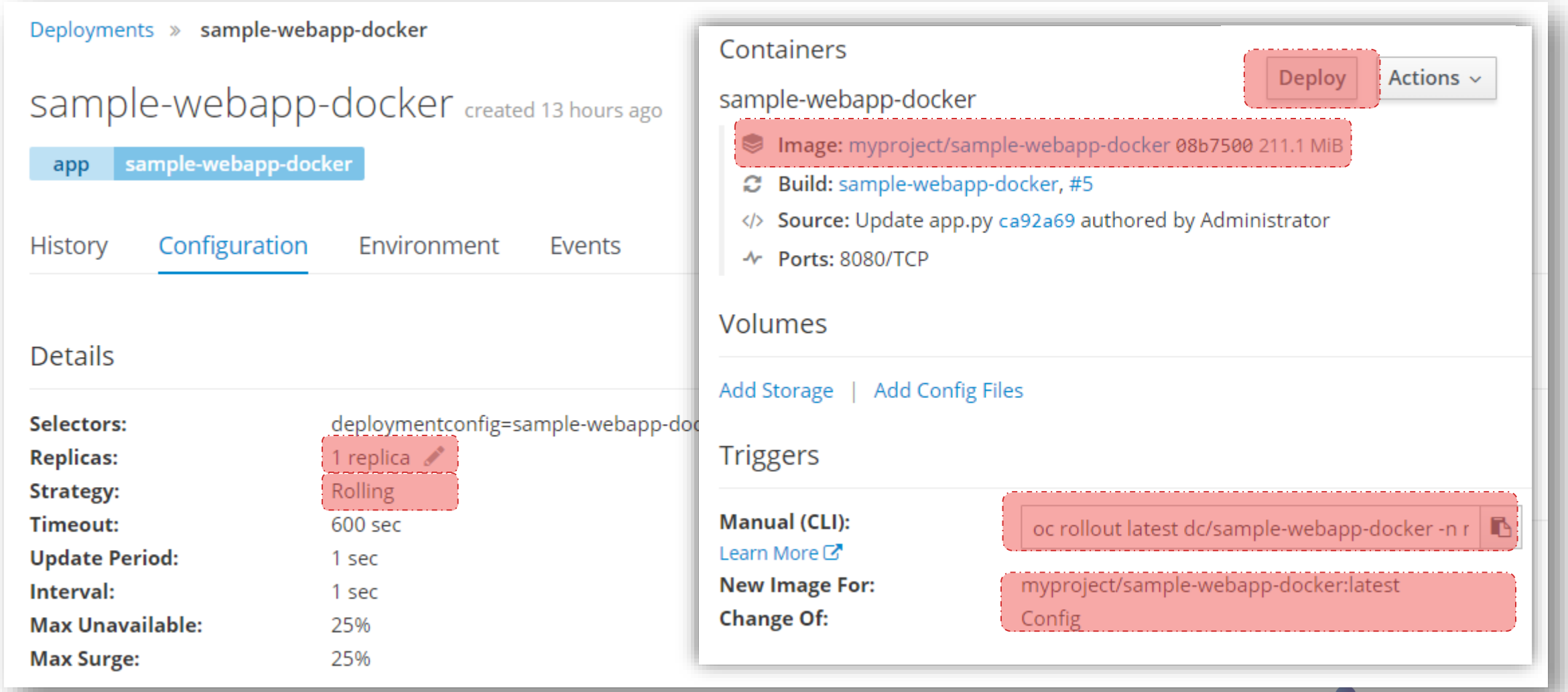
KODEKLOUD

Deployment



KODEKLOUD

Deployment



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): `oc rollout latest dc/sample-webapp-docker -n r`

Learn More

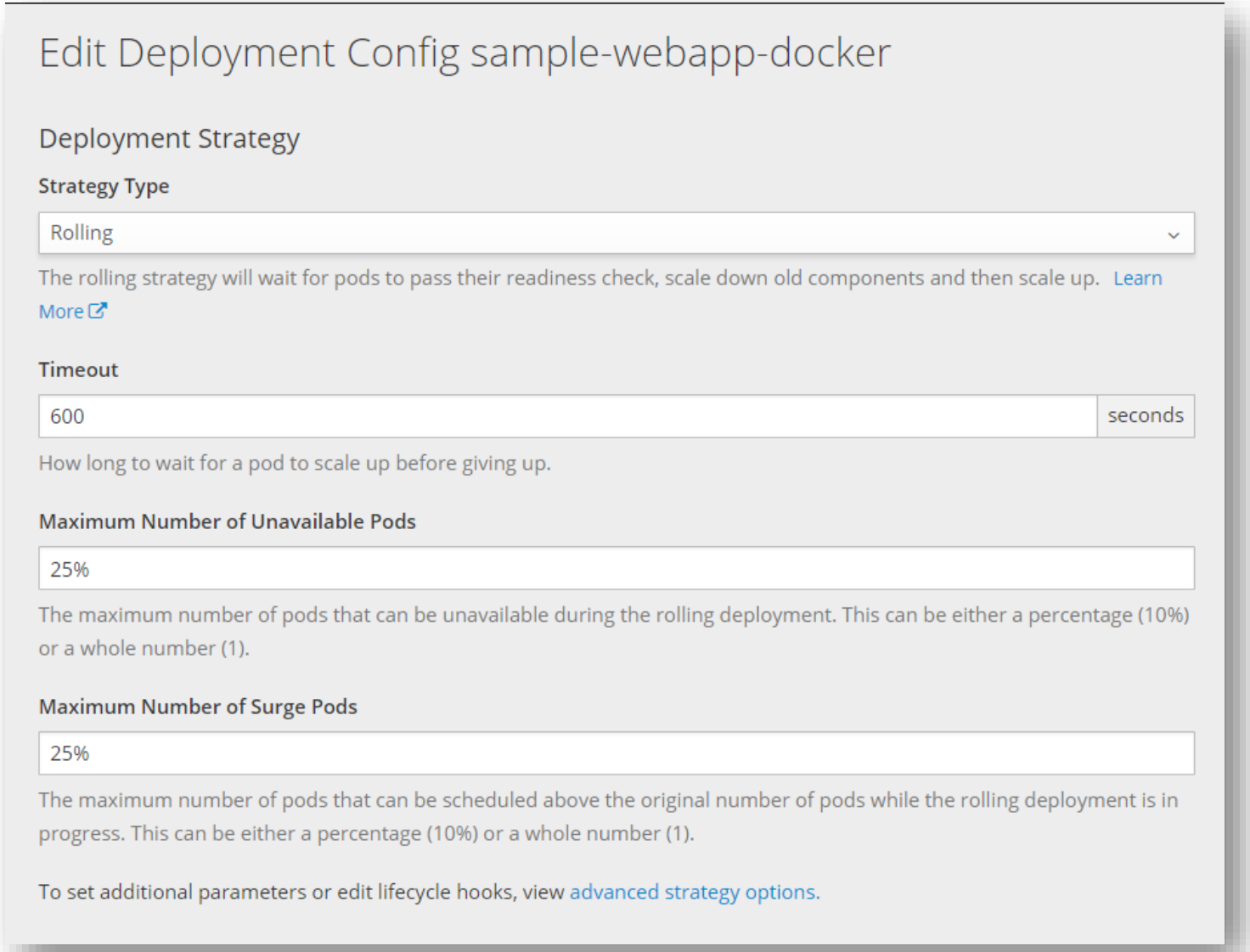
New Image For: myproject/sample-webapp-docker:latest

Change Of: Config

deployment-config.yaml

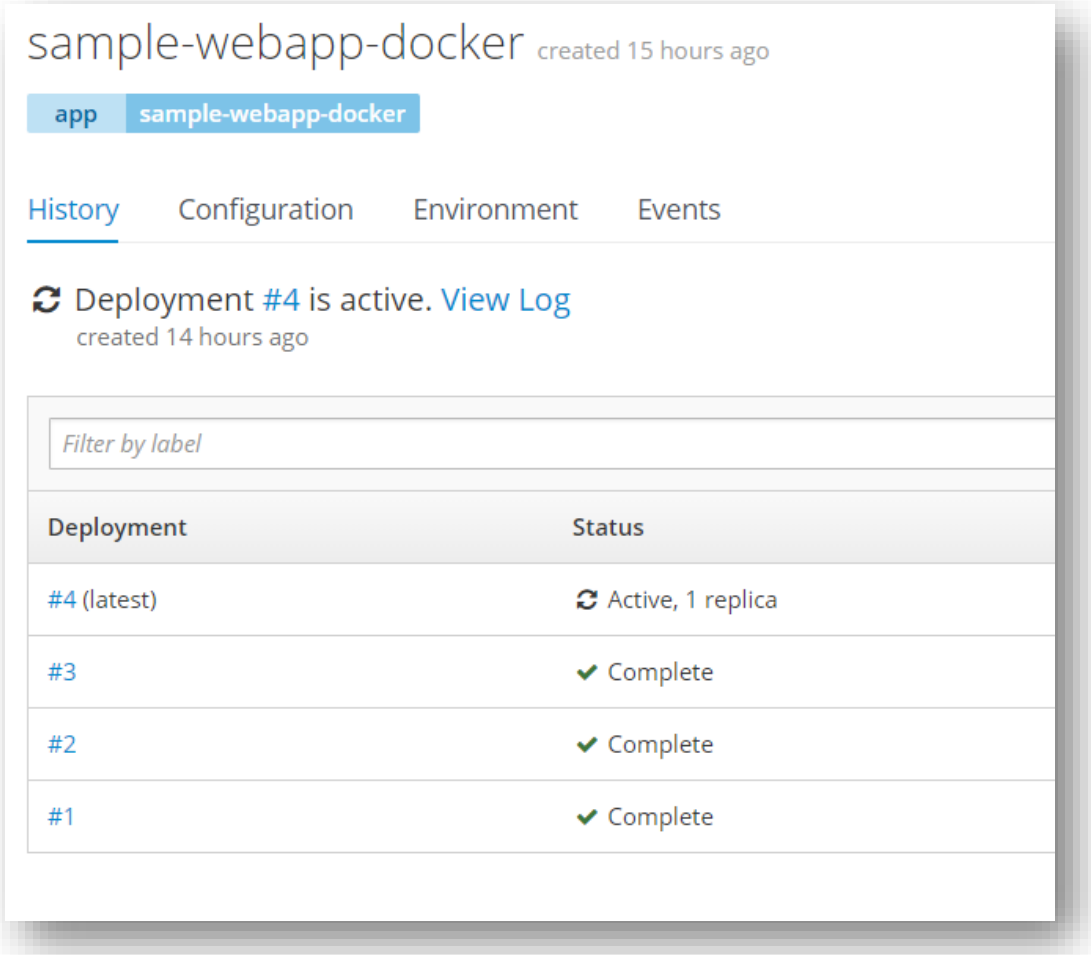
```yaml
apiVersion: apps.openshift.io/v1
kind: DeploymentConfig
metadata:
  name: sample-webapp-docker
spec:
  replicas: 1
  selector:
    deploymentconfig: sample-webapp-docker
  strategy:
    type: Rolling
  template:
    metadata:
      labels:
        app: sample-webapp-docker
        deploymentconfig: 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@sha256:08b750083d53e8fdcf09ab99bc30549141ea44c90763d3e972be264fbec8d706
      type: ImageChange
    - type: ConfigChange
```

Edit Deployment Configuration



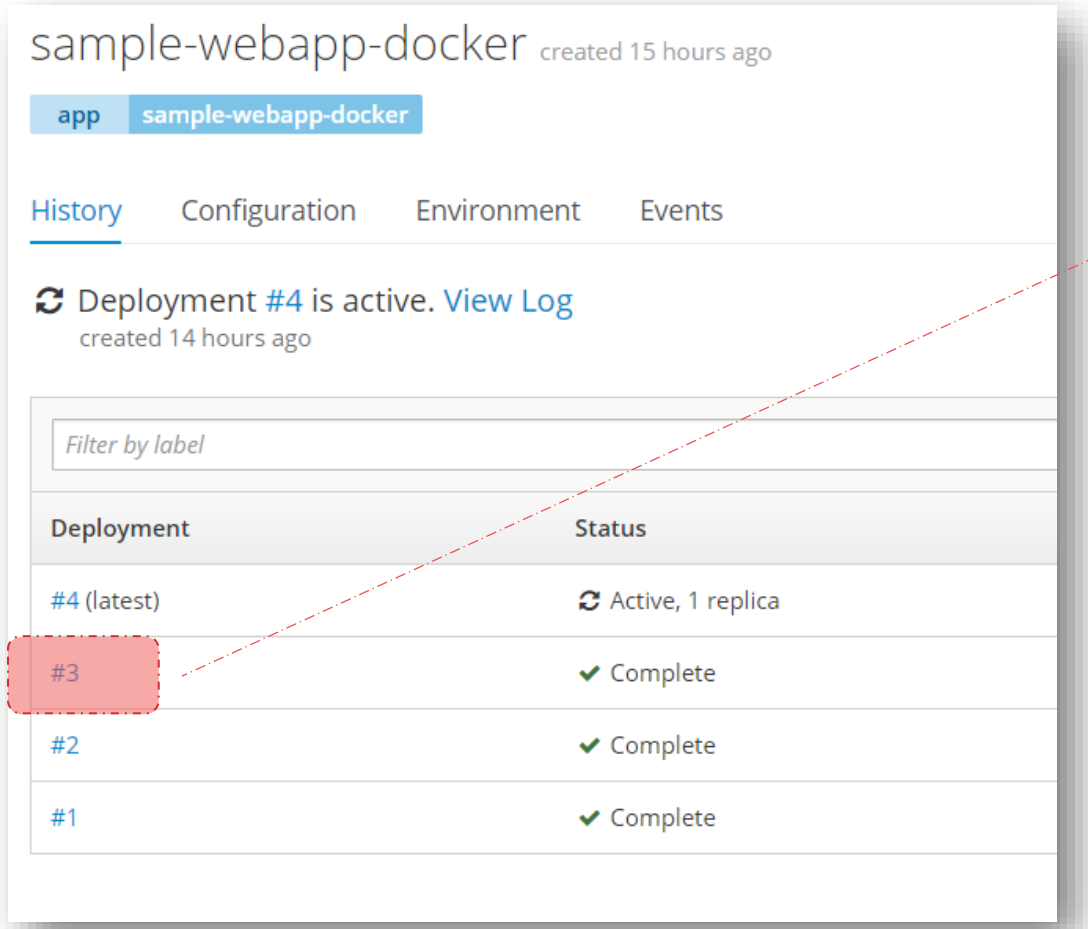
KODEKLOUD

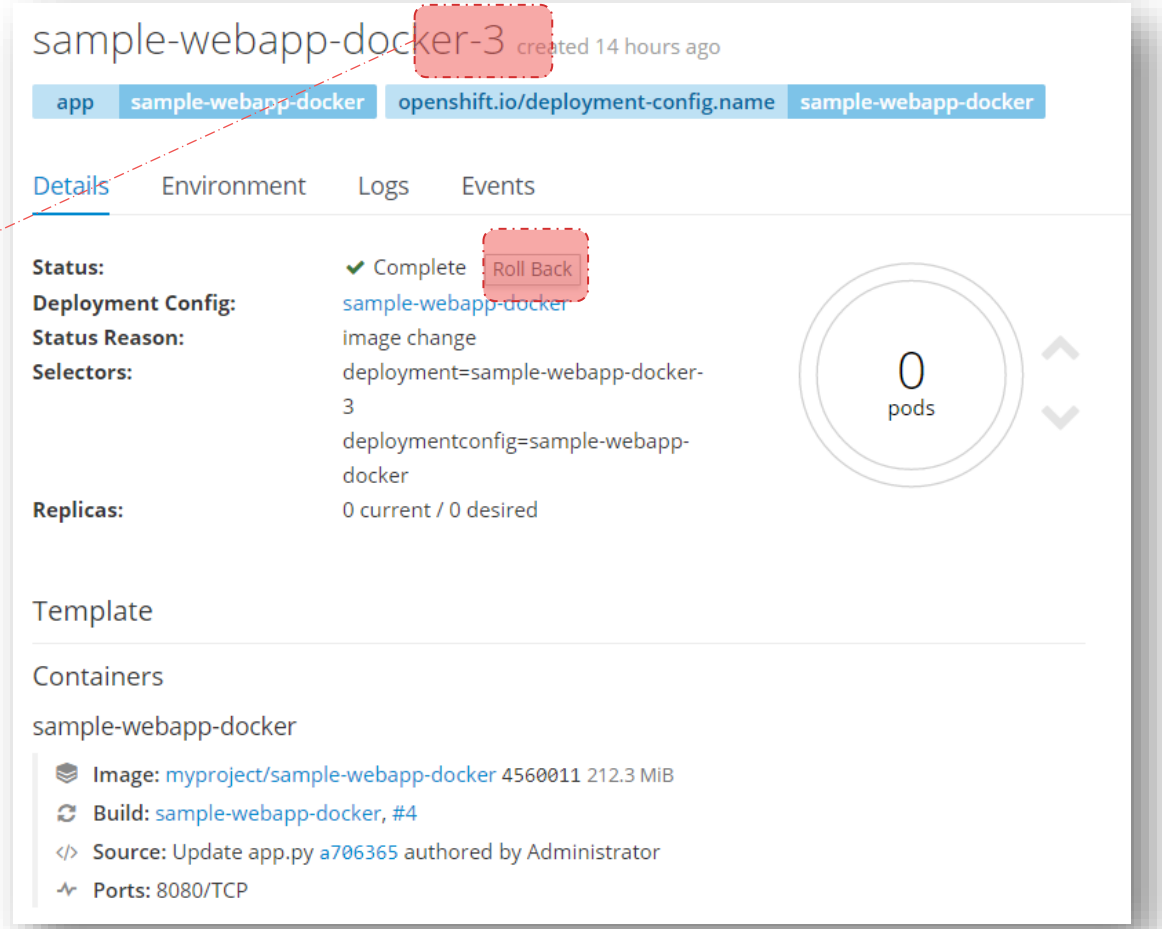
History



KODEKLOUD

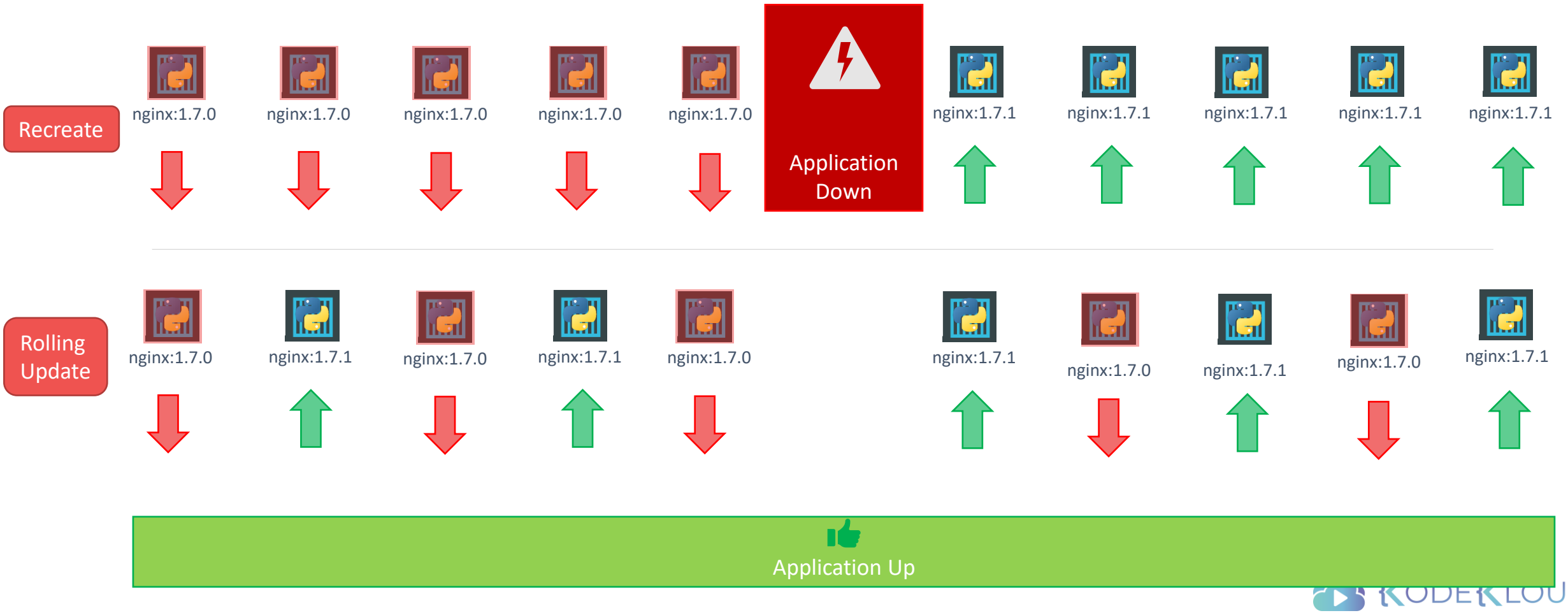
Rollback



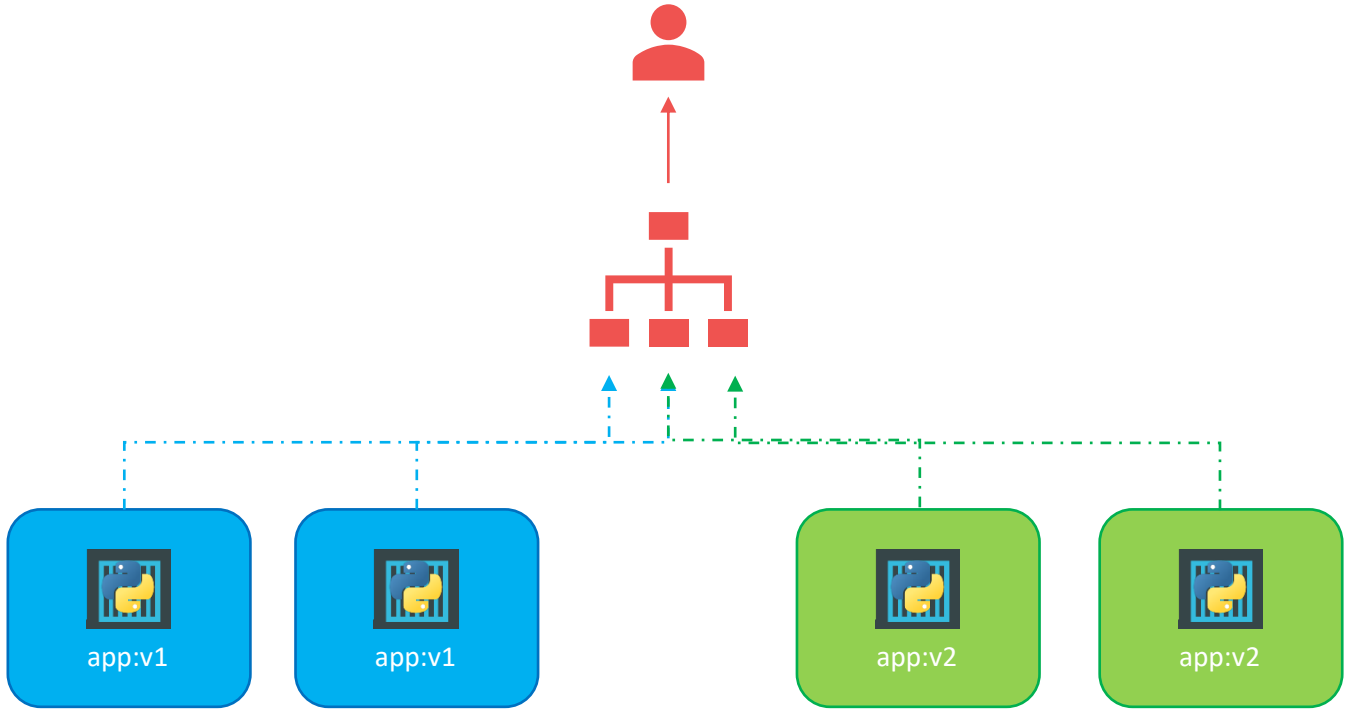


KODEKLOUD

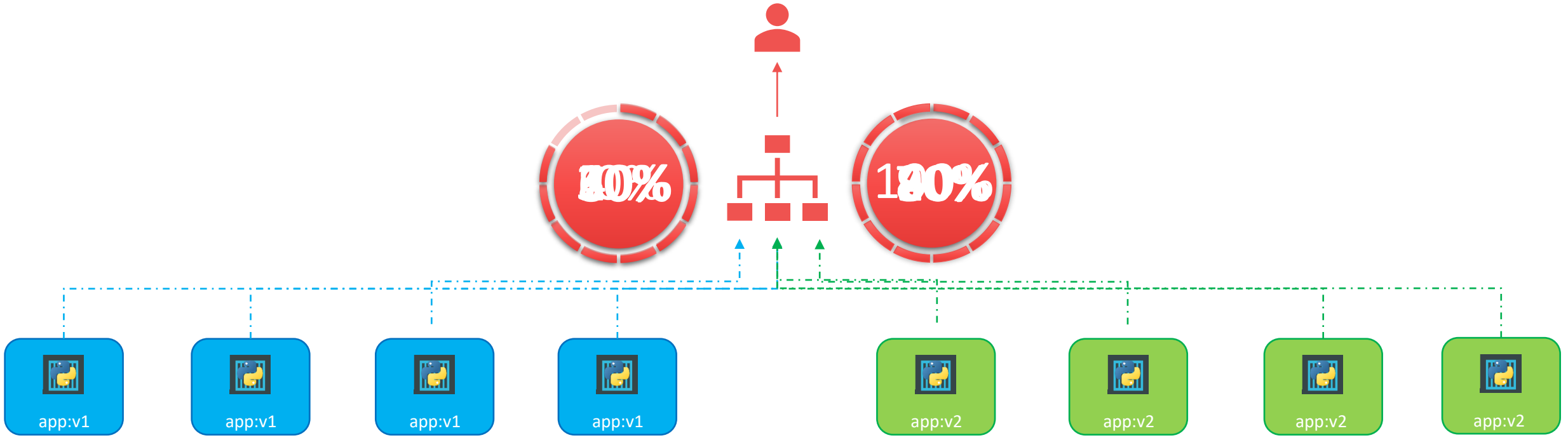
Deployment Strategies



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.

KODEKLOUD

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
```

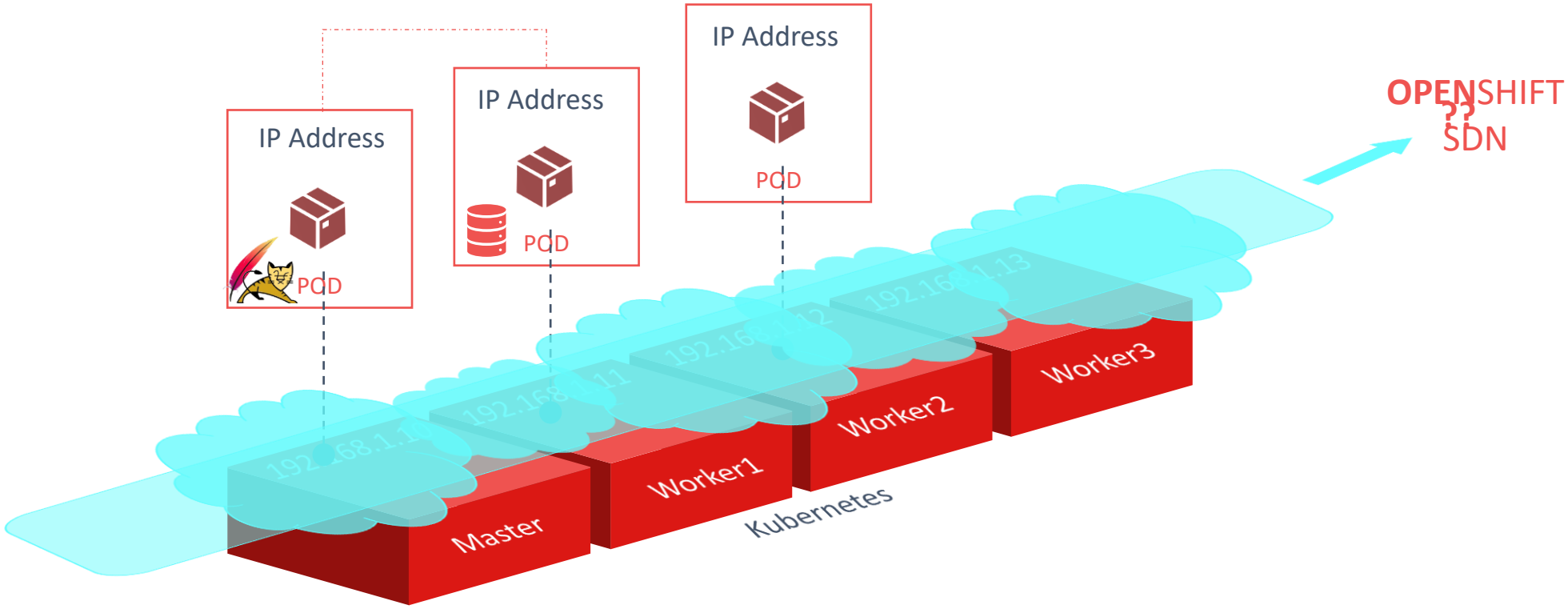

KODEKLOUD

Red Hat

OPENSHIFT

Networking

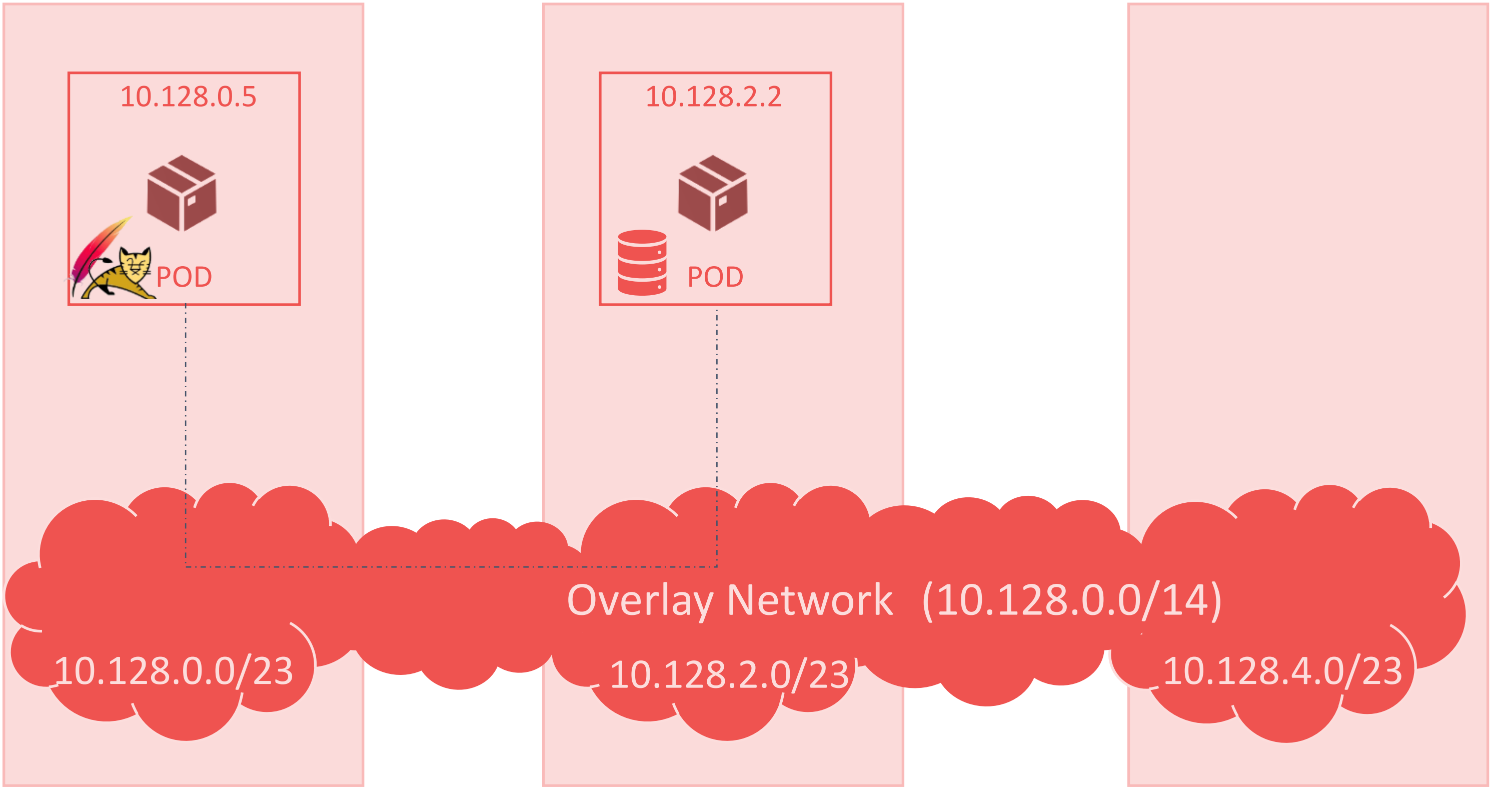
Challenge



KODEKLOUD

OPENSHIFT 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
```

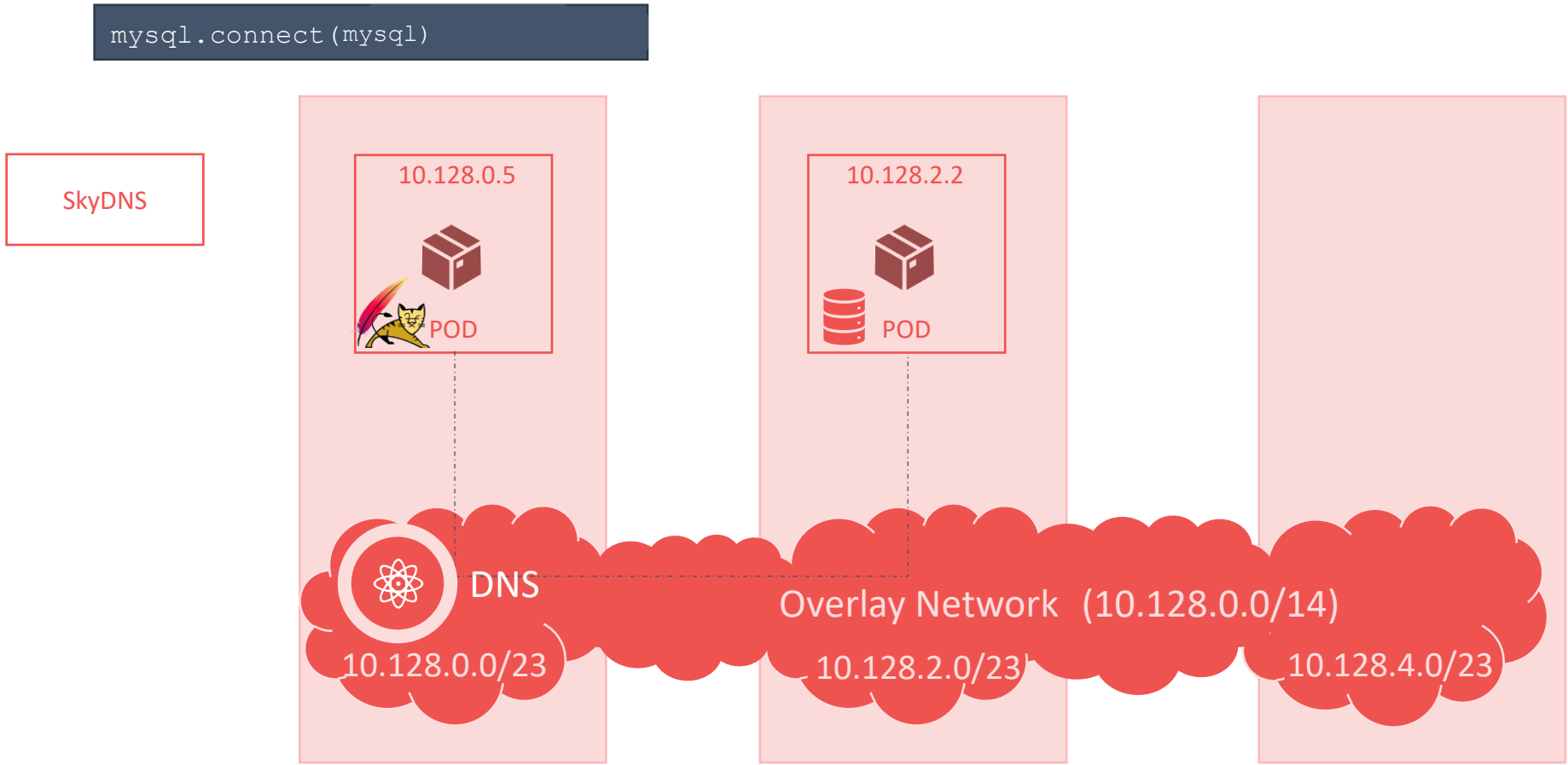


Open vSwitch

- VLAN tagging
- Trunking
- LACP
- Port Mirroring

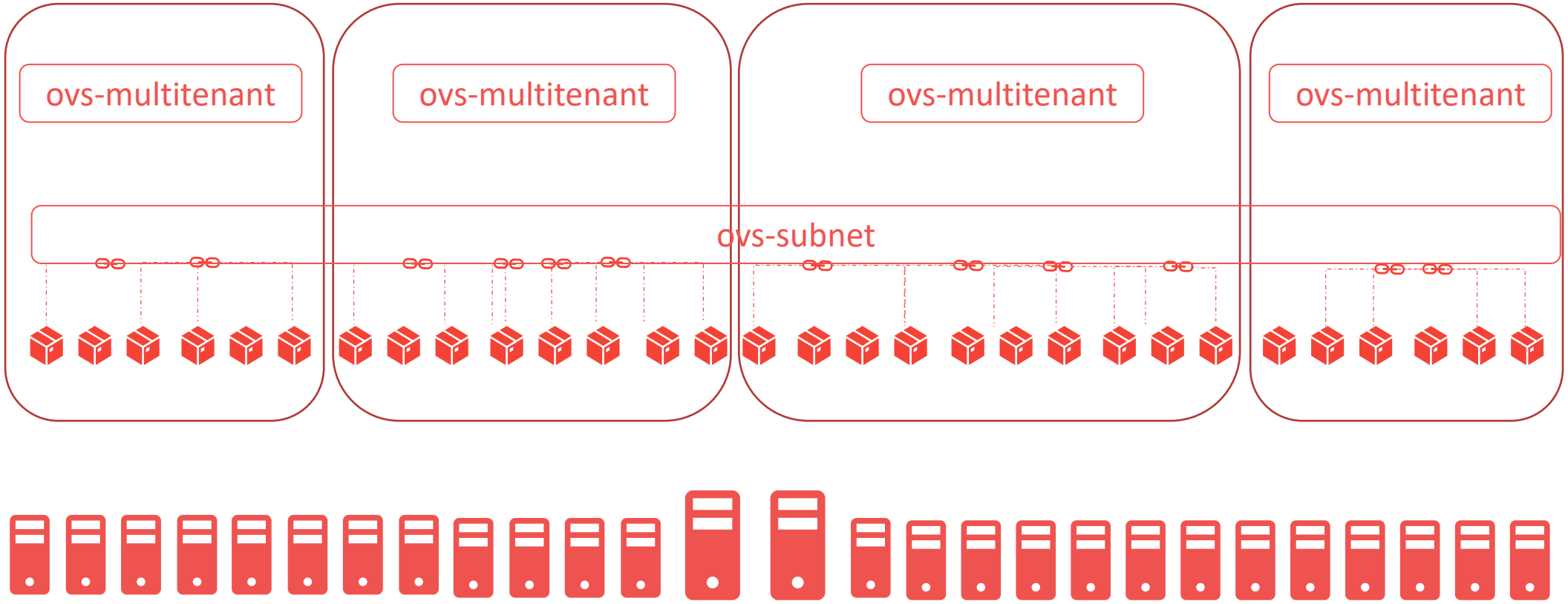
KODEKLOUD

OPENSHIFT DNS



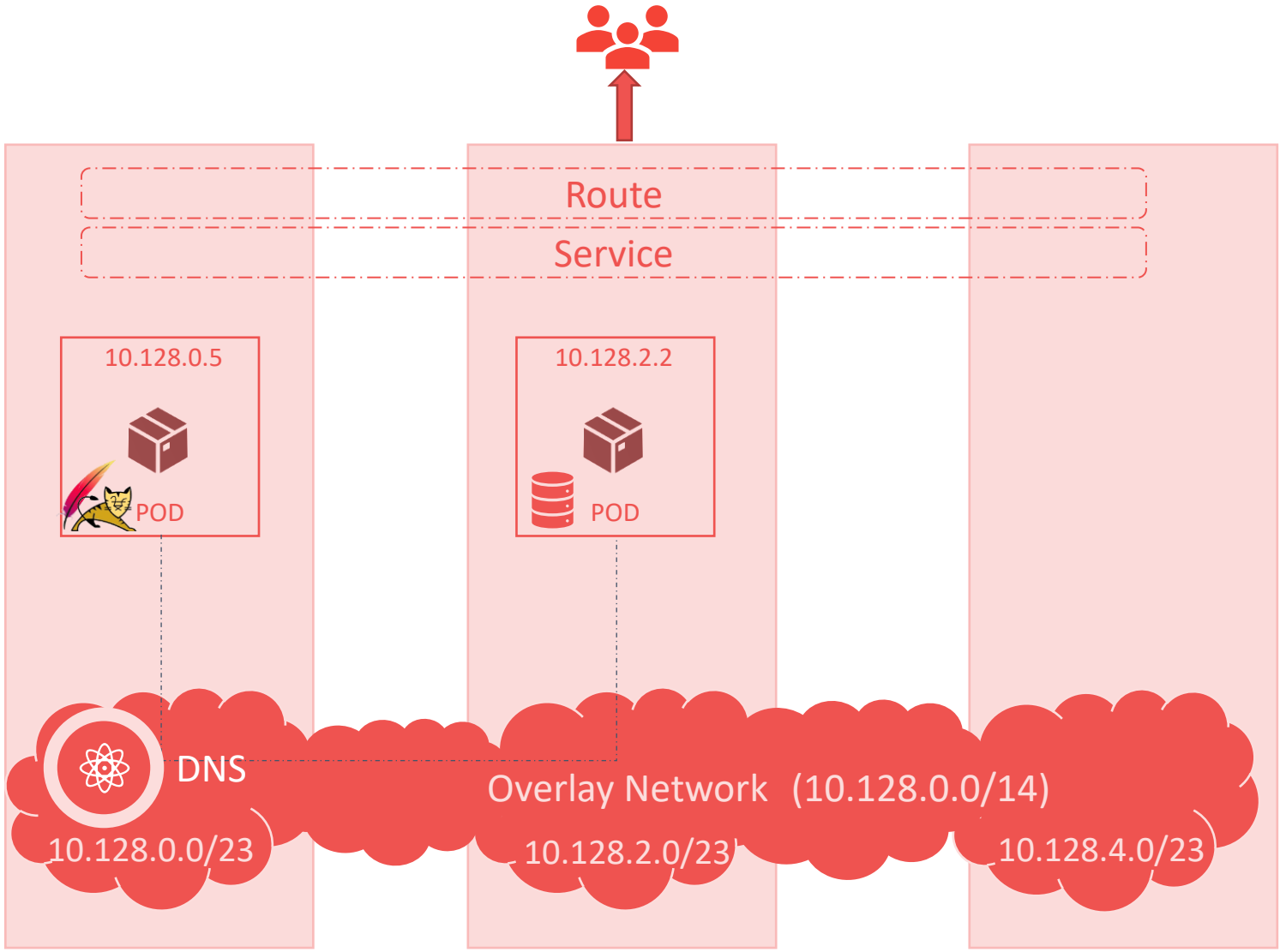
KODEKLOUD

SDN Plugins



Additional Plugins

External Connectivity



KODEKLOUD

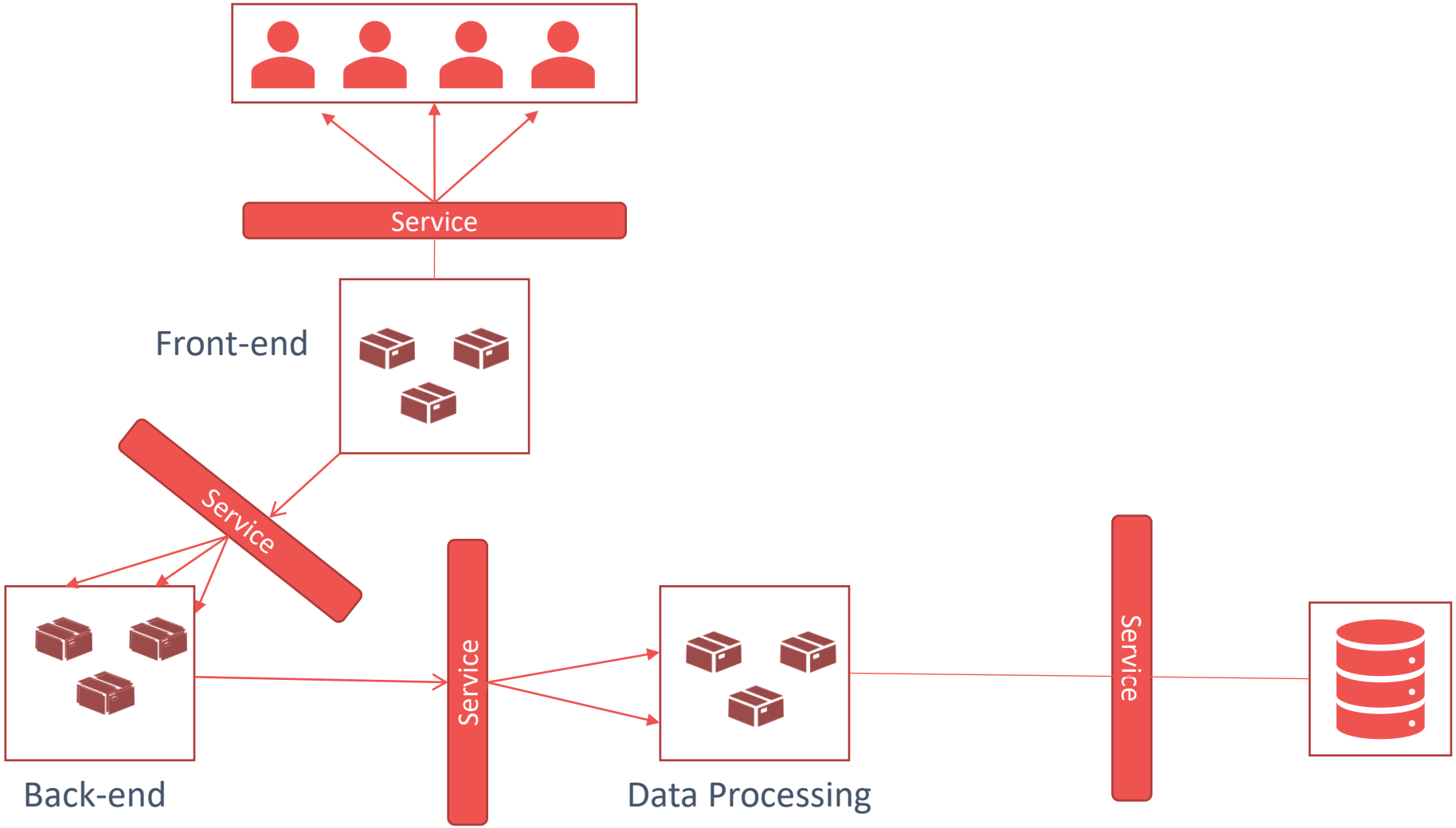

KODEKLOUD

Red Hat

OPENSHIFT

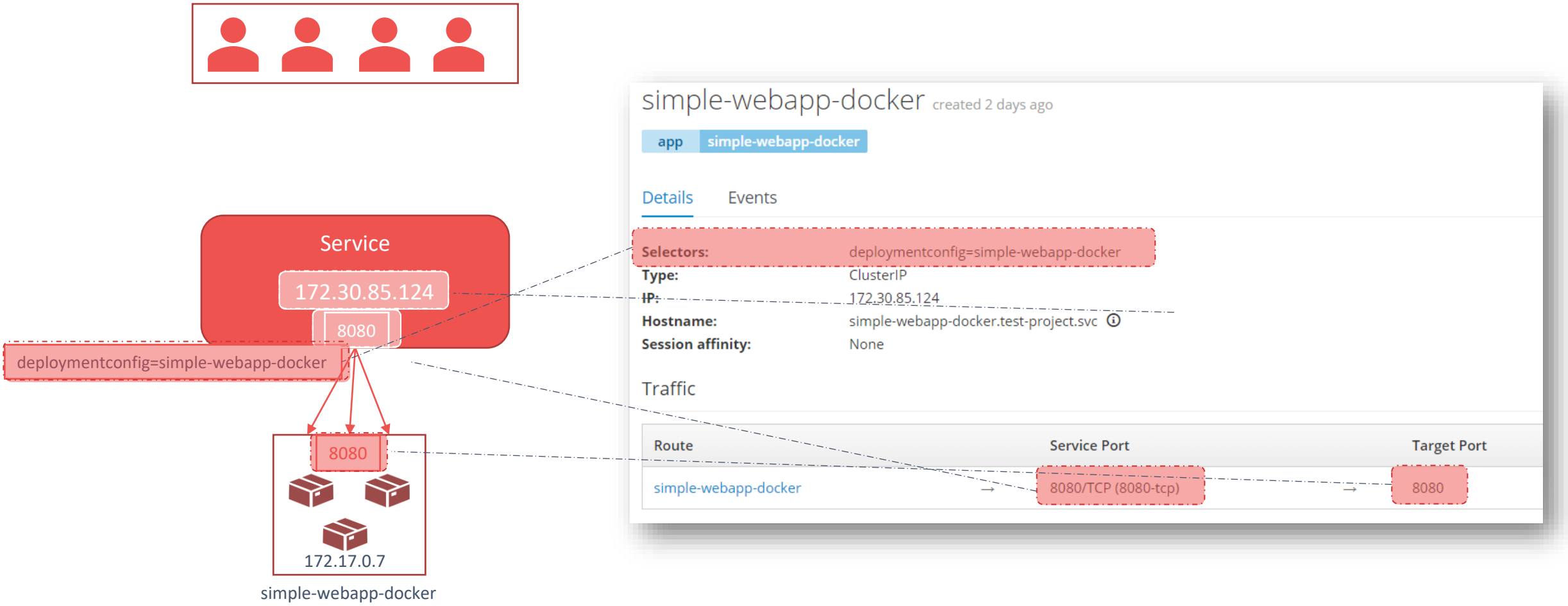
Services and Routes

Service

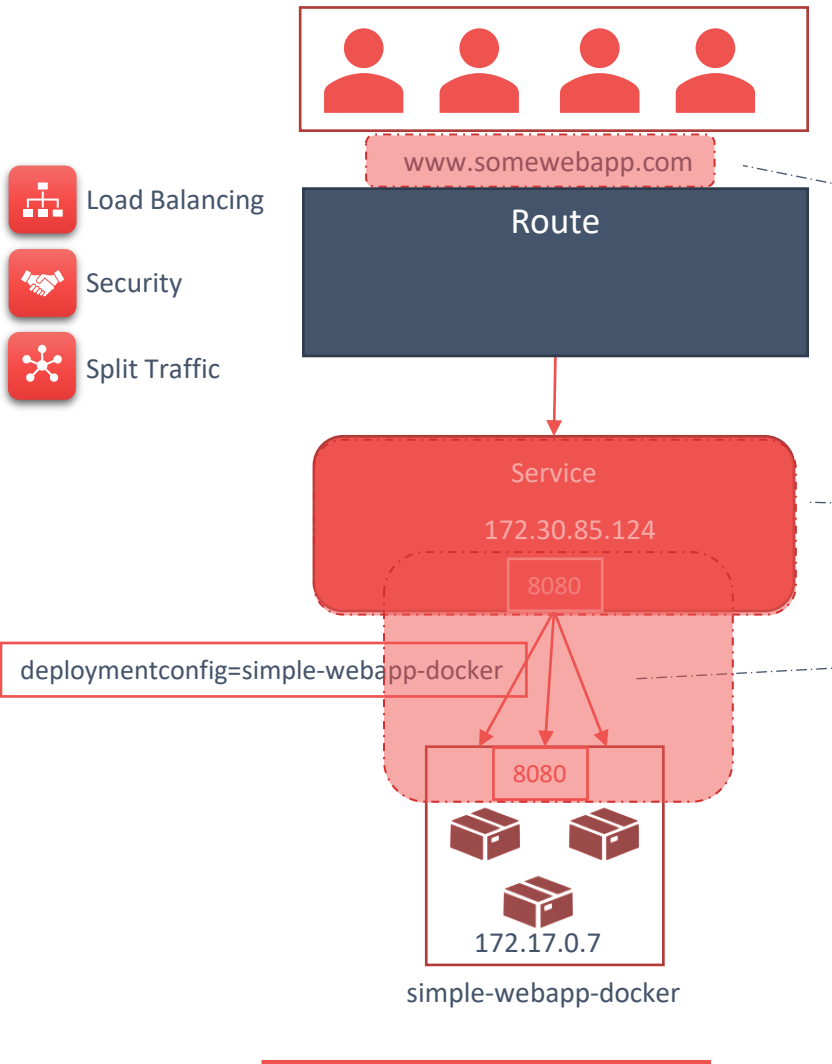


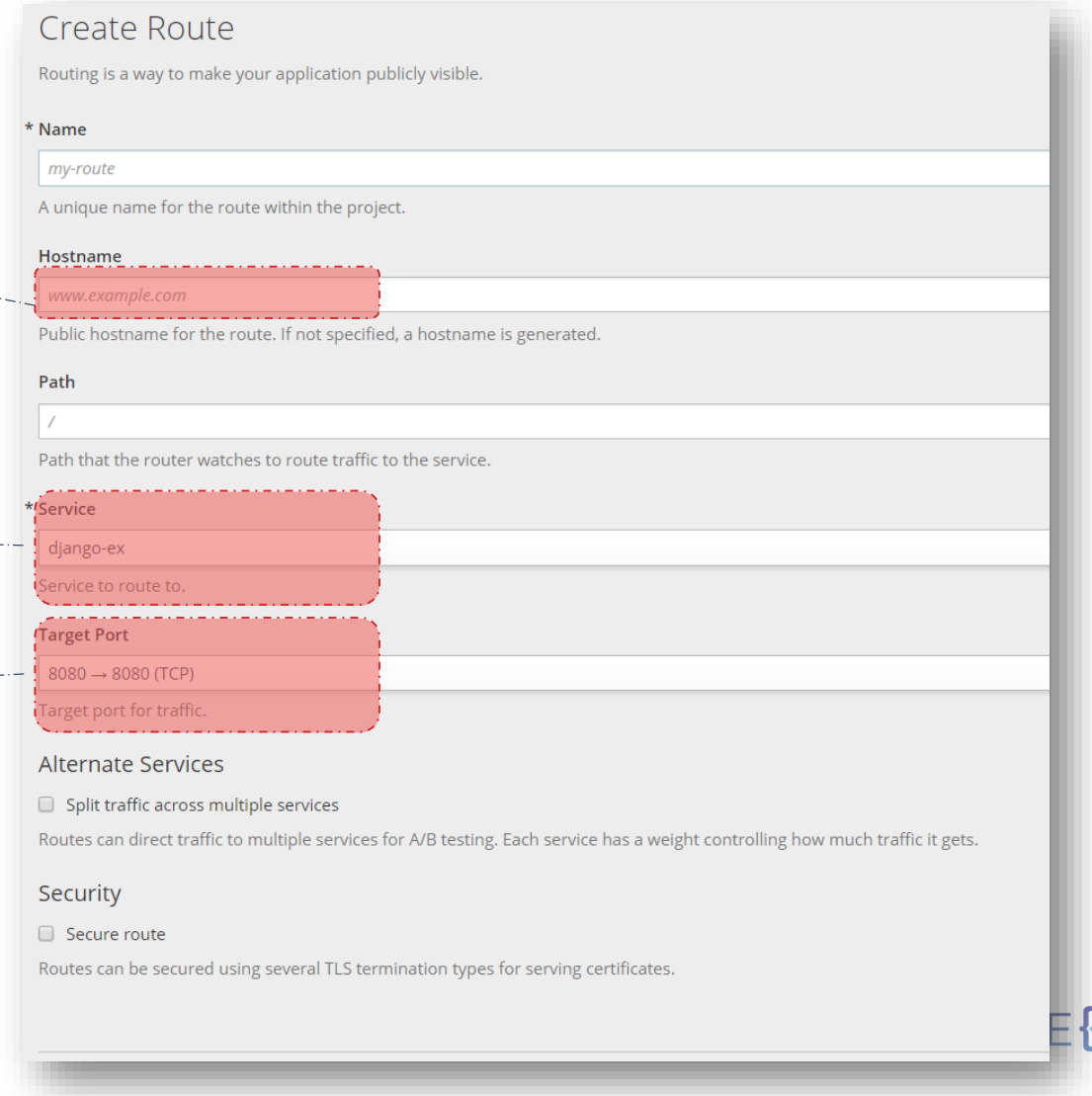
KODEKLOUD

Service

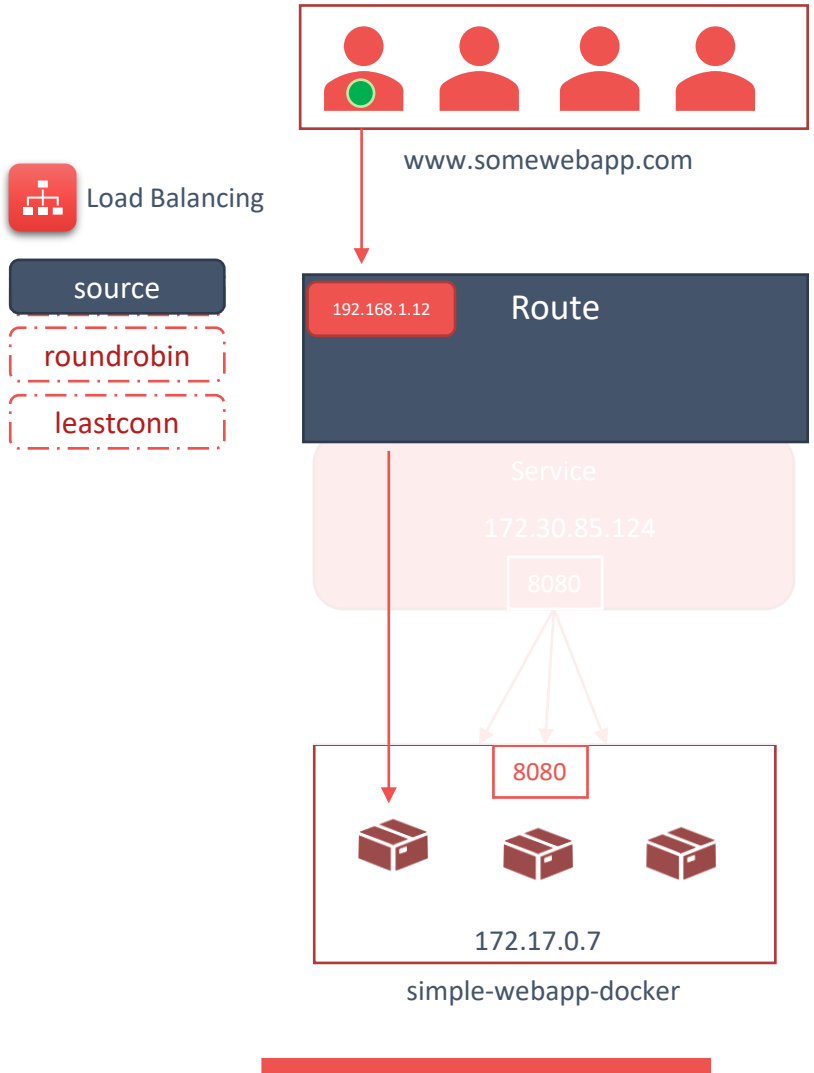

Service
172.30.85.124
8080
deployment=simple-webapp-docker
8080
172.17.0.7
simple-webapp-docker
simple-webapp-docker created 2 days ago
app
simple-webapp-docker
Details
Events
Selectors:
deployment=simple-webapp-docker
Type:
ClusterIP
IP:
172.30.85.124
Hostname:
simple-webapp-docker.test-project.svc
Session affinity:
None
Traffic
Route
Service Port
Target Port
simple-webapp-docker
8080/TCP (8080-tcp)
8080

Route


Load Balancing
Security
Split Traffic
www.somewebapp.com
Route
Service
172.30.85.124
8080
deploymentconfig=simple-webapp-docker
8080
172.17.0.7
simple-webapp-docker

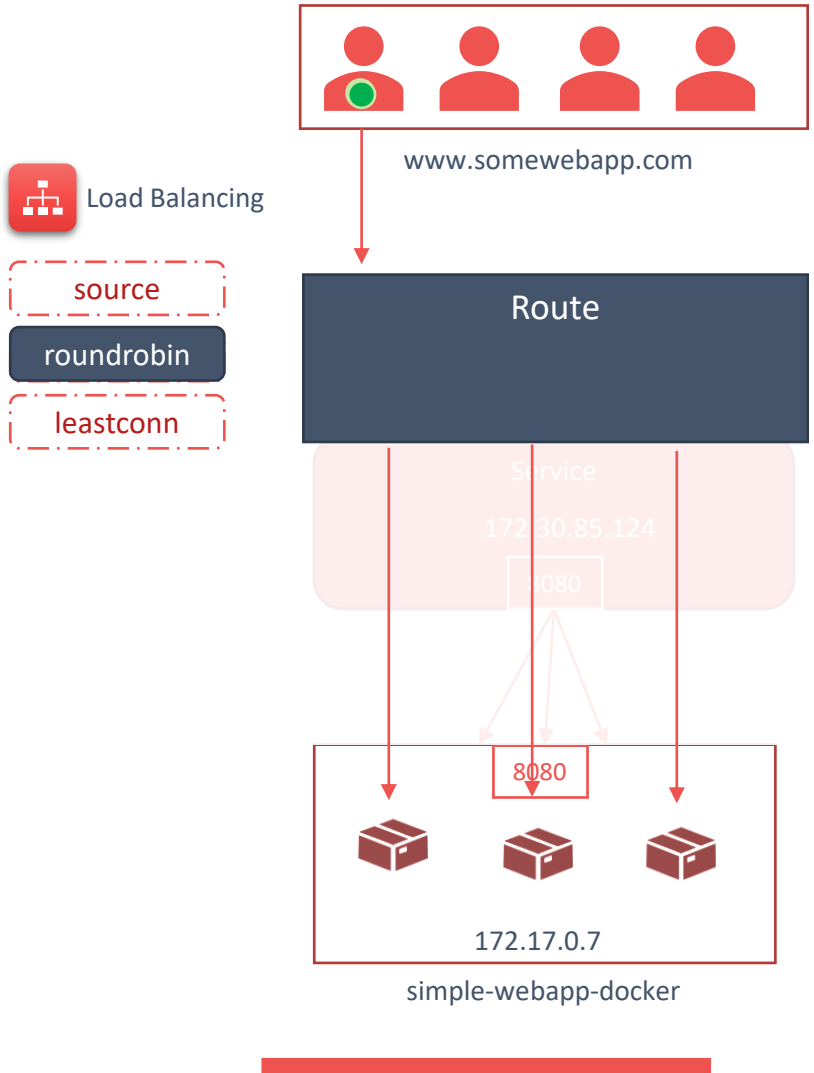

Create Route
Routing is a way to make your application publicly visible.
* Name
my-route
A unique name for the route within the project.
Hostname
www.example.com
Public hostname for the route. If not specified, a hostname is generated.
Path
/
Path that the router watches to route traffic to the service.
* 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



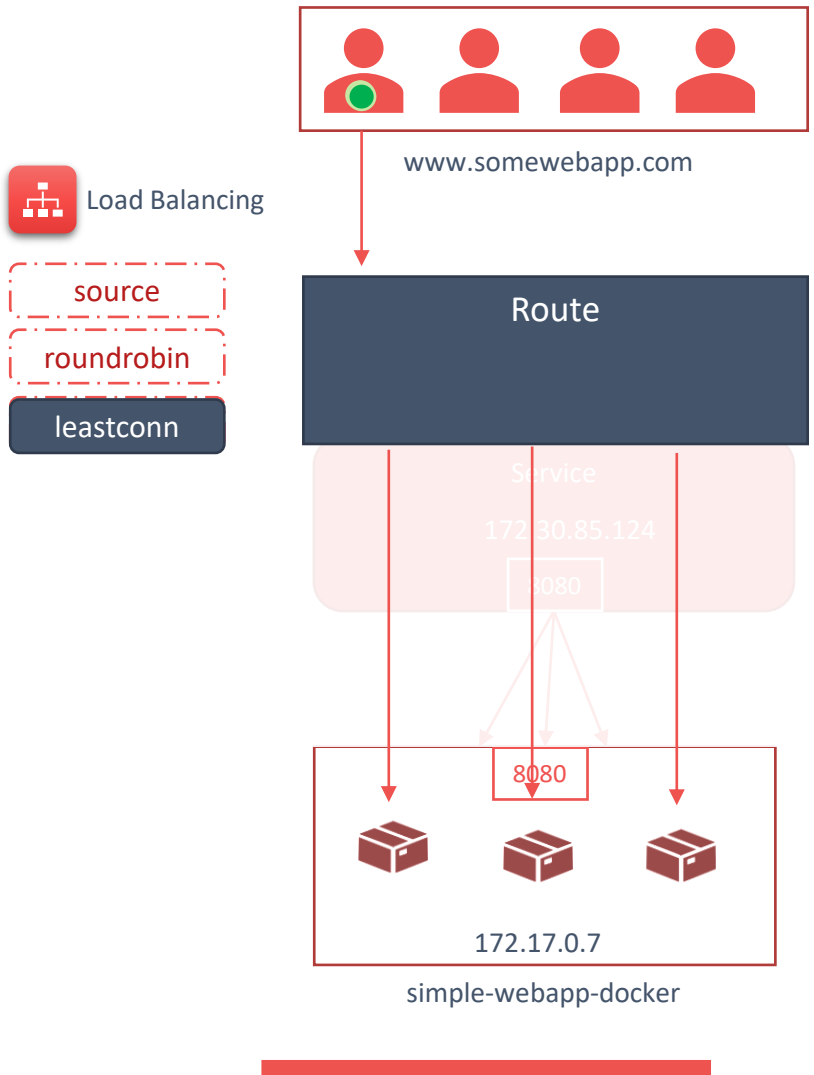
KODEKLOUD

Route – Load Balancing



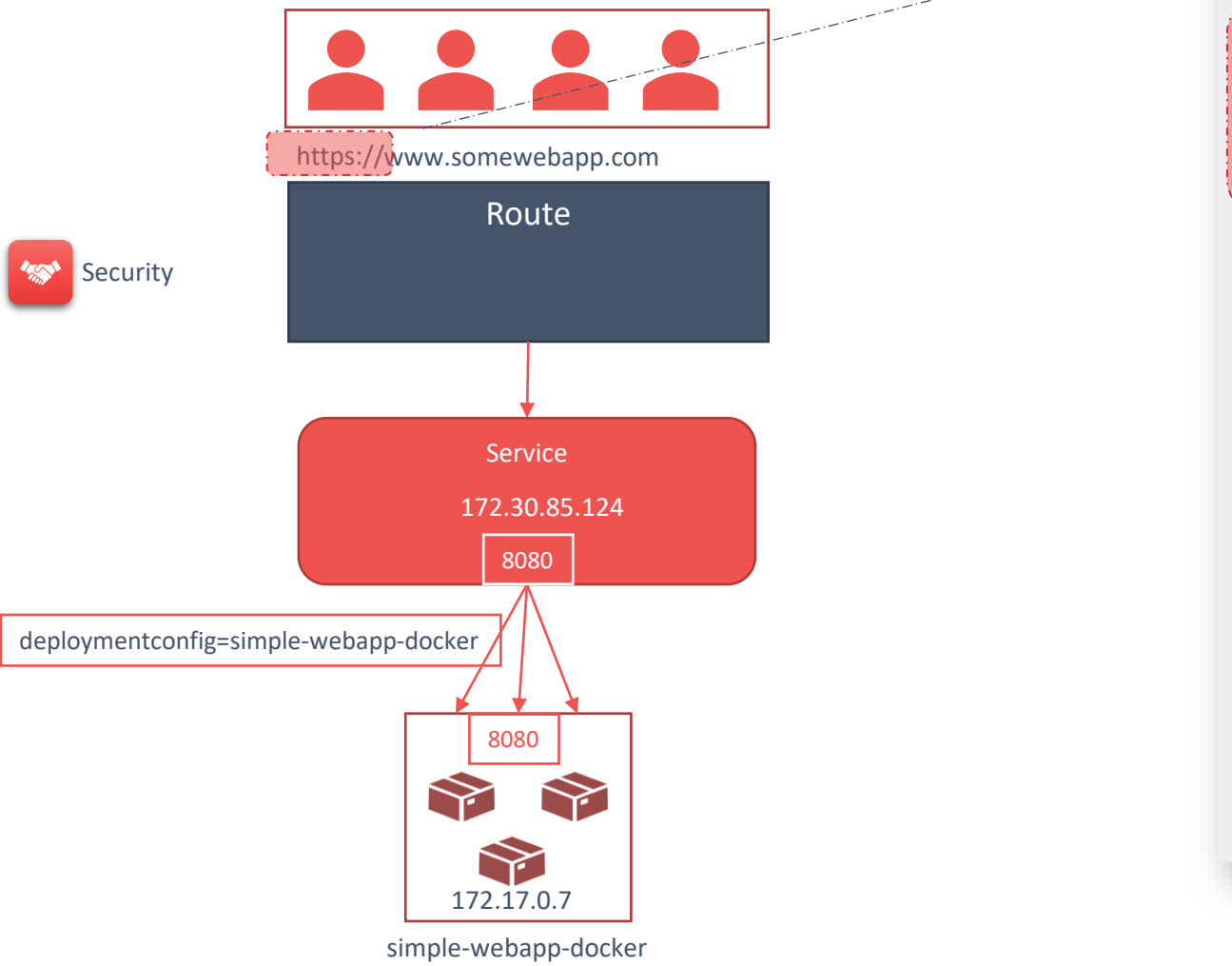
KODEKLOUD

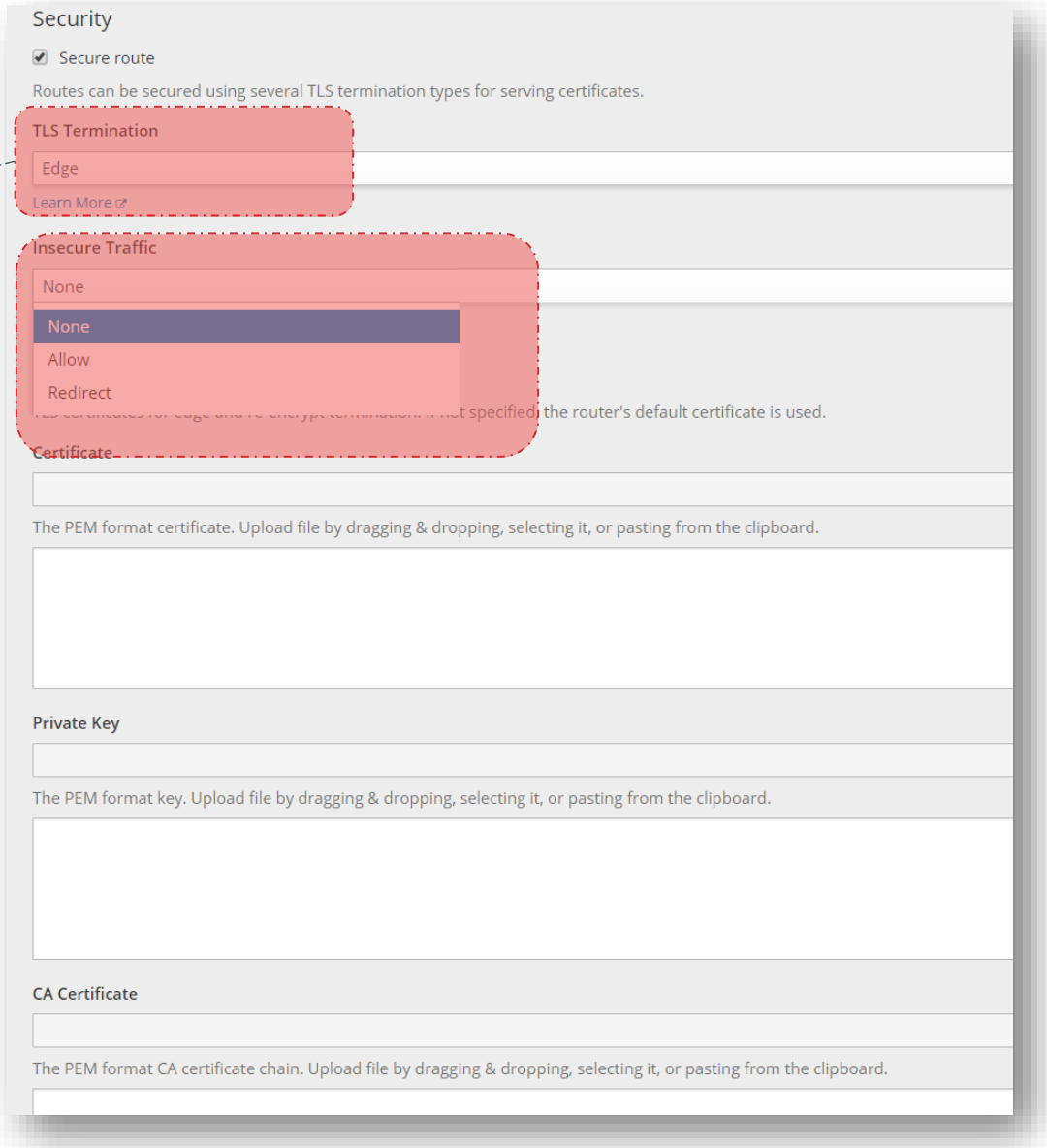
Route – Load Balancing



KODEKLOUD

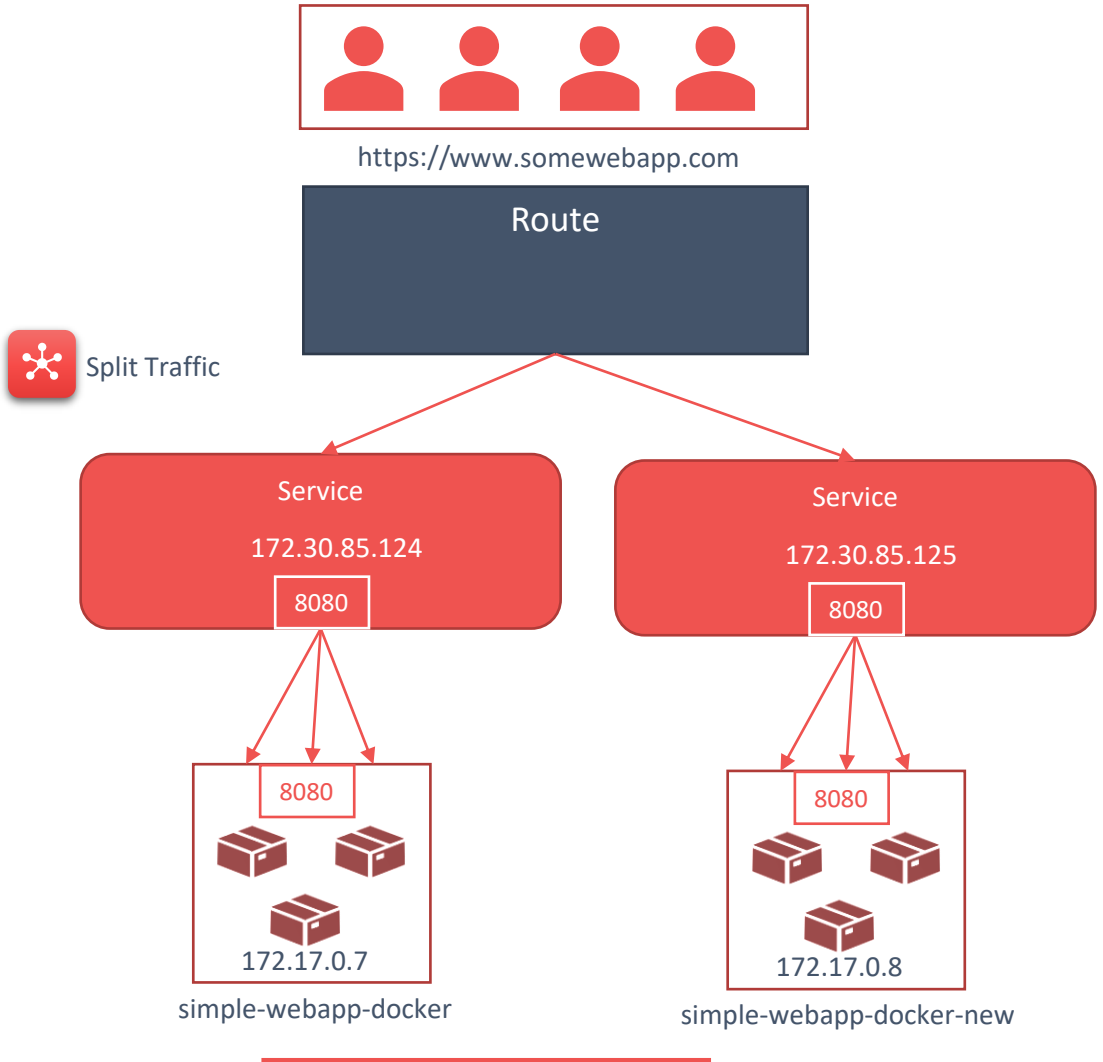
Route - Security

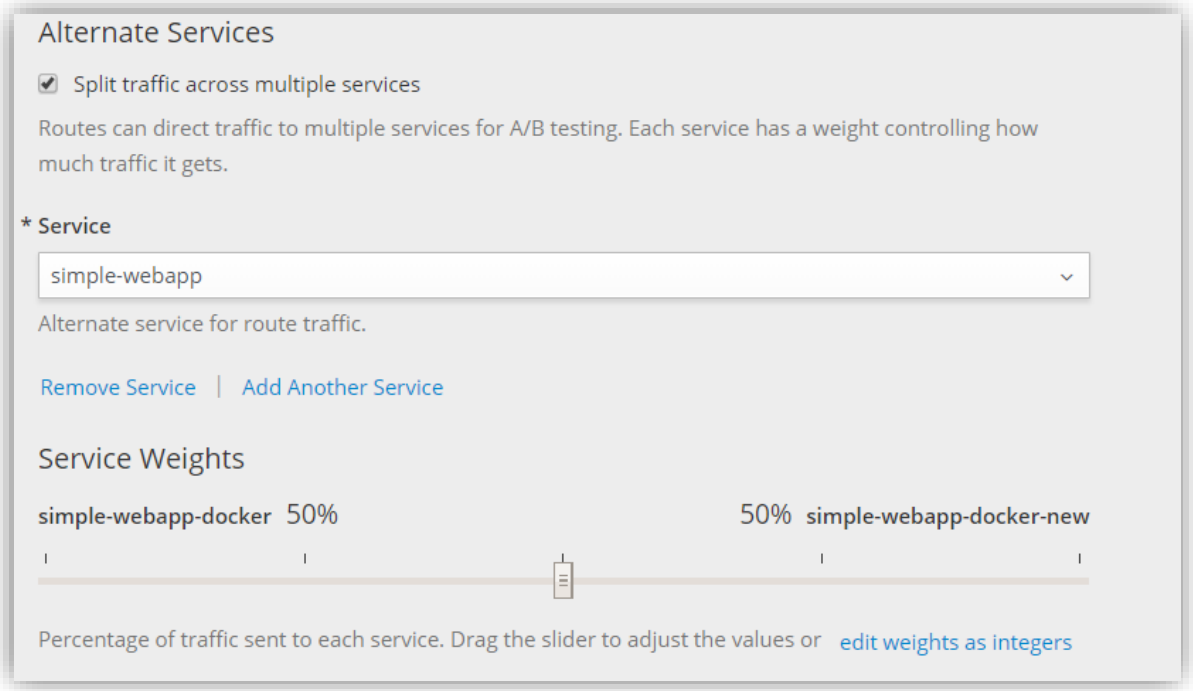




KODEKLOUD

Route – Split Traffic





KODEKLOUD

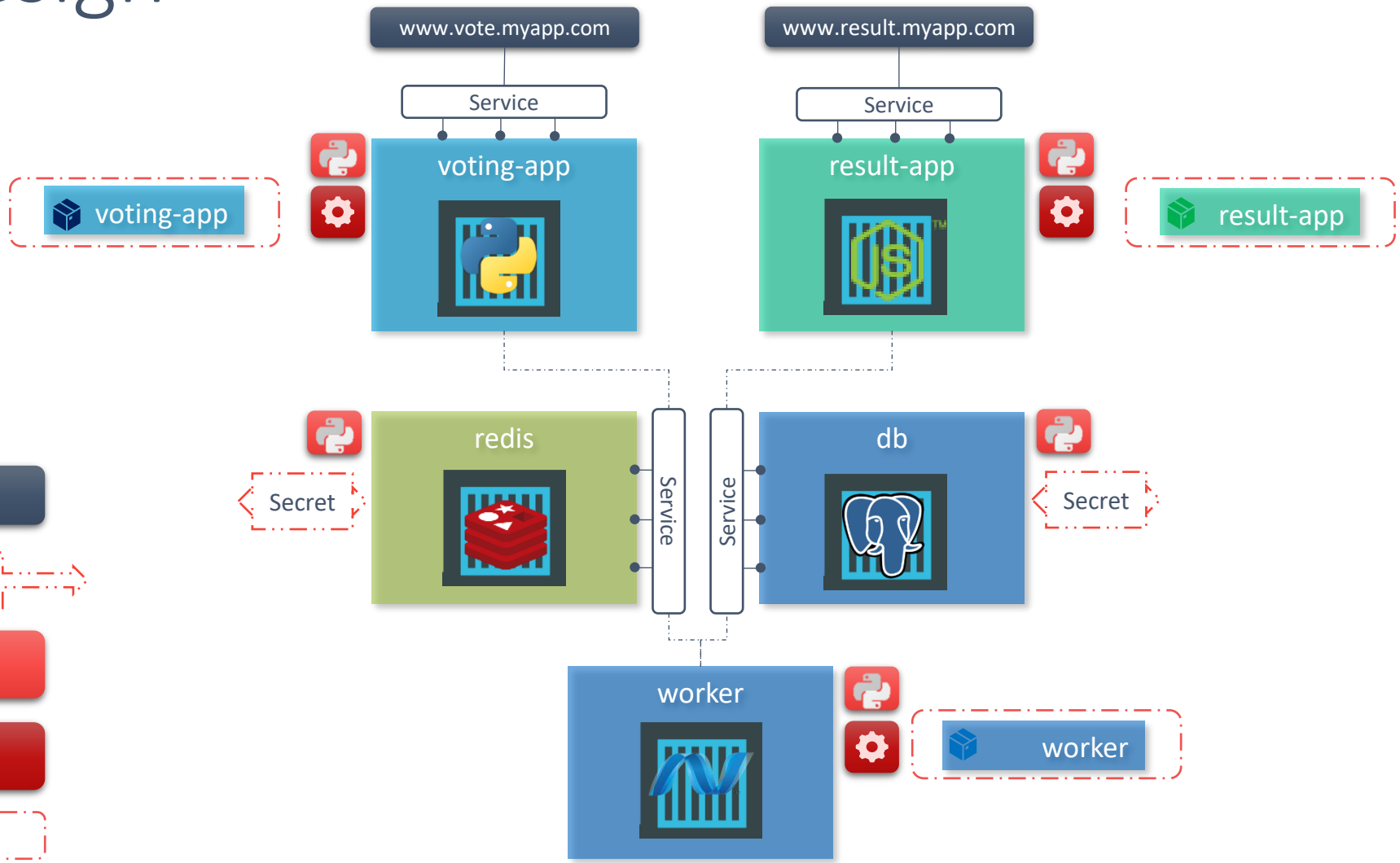

KODEKLOUD

Red Hat

OPENSHIFT

Example Voting Application

Design



KODEKLOUD

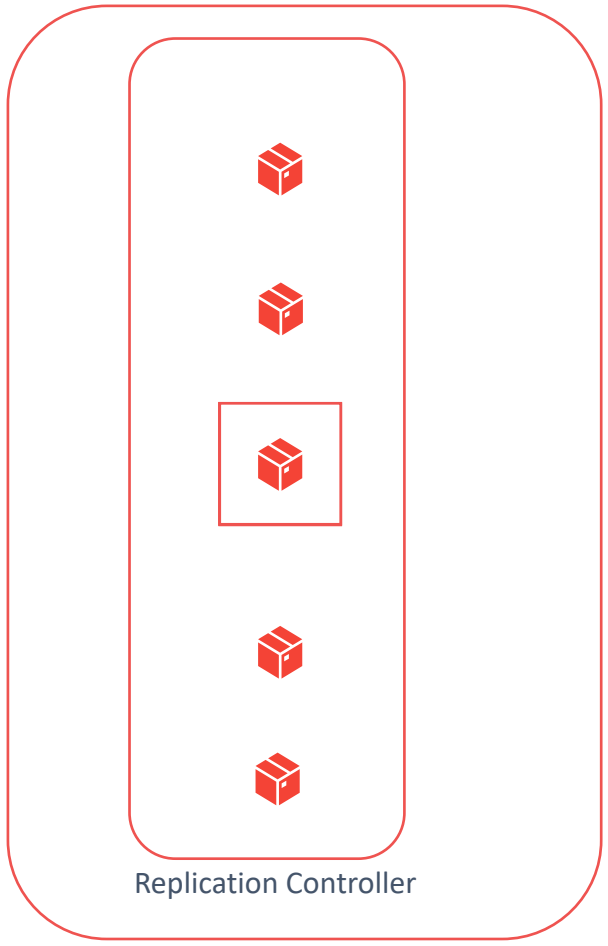

KODEKLOUD

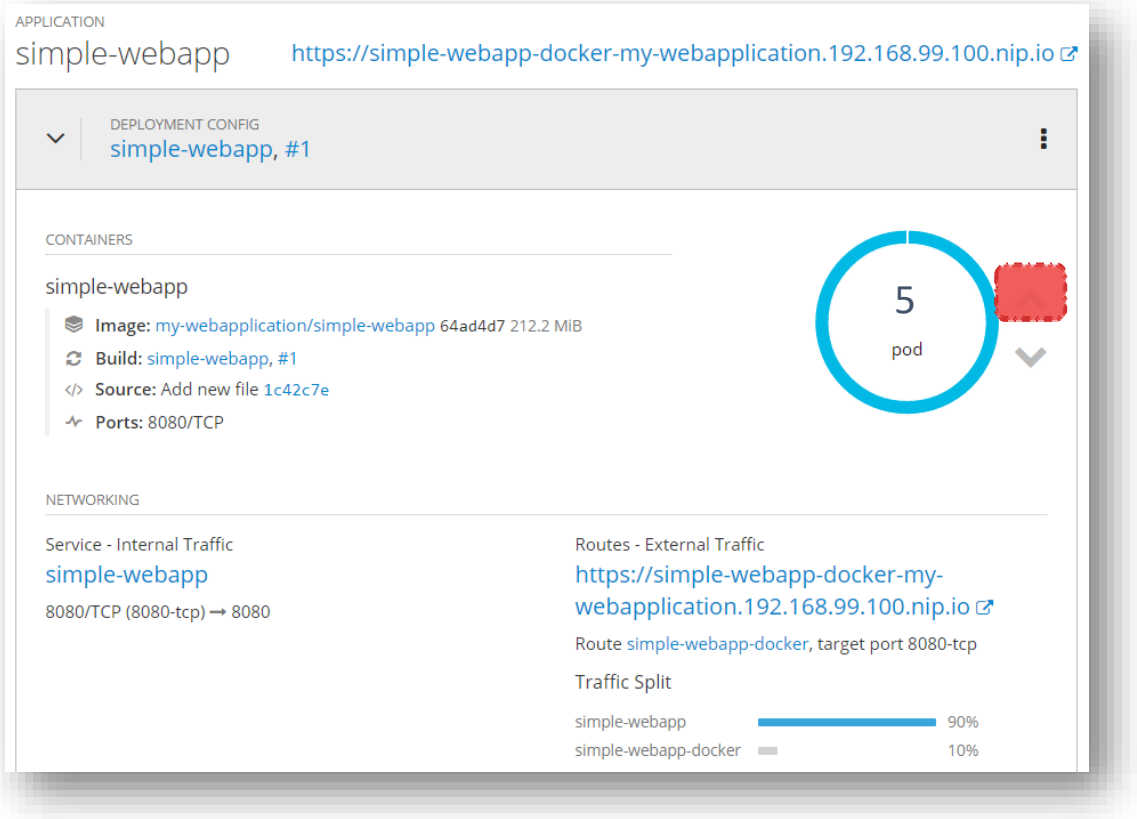
Red Hat

OPENSHIFT

Scaling

Scale Deployment





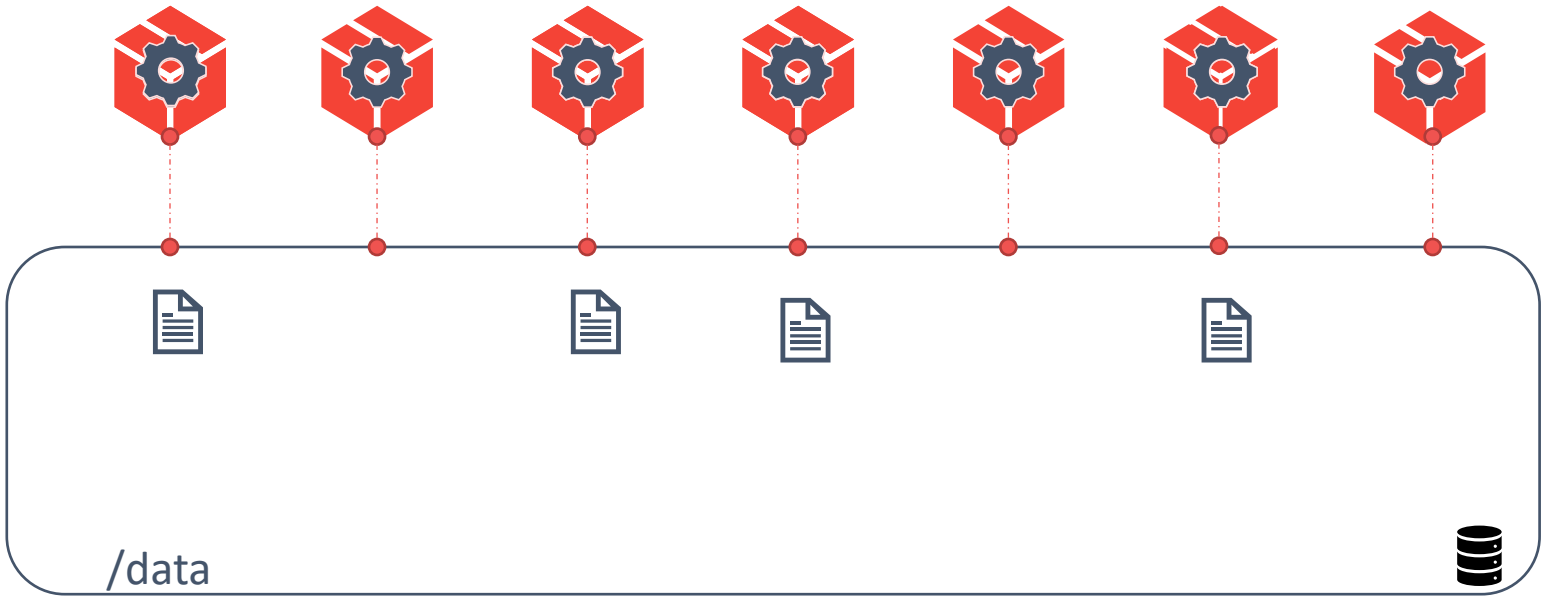

KODEKLOUD

Red Hat

OPENSHIFT

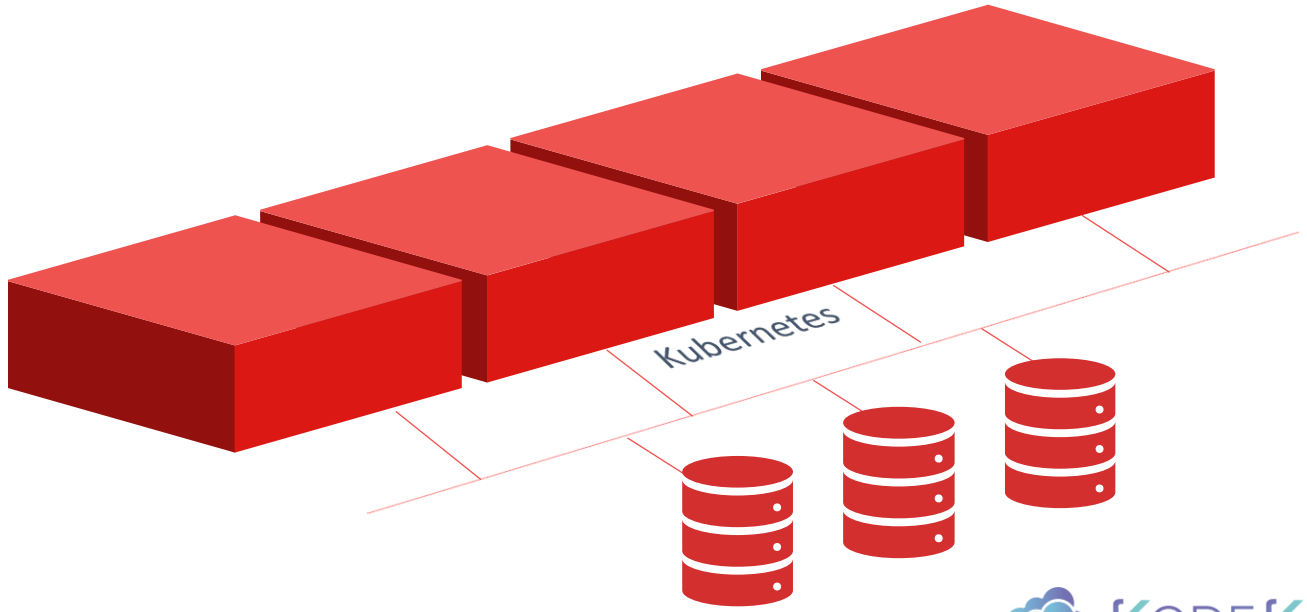
Storage

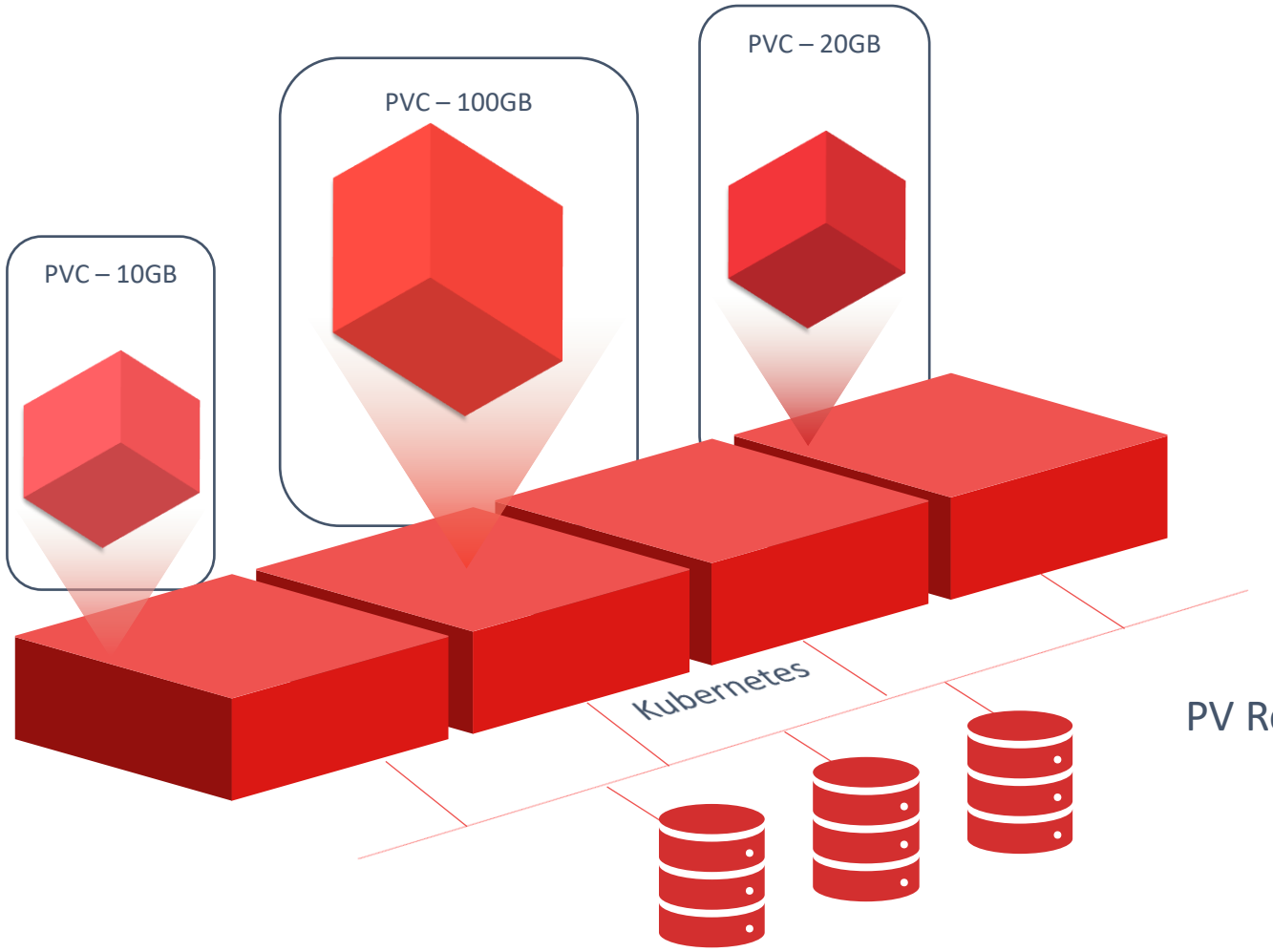
Persistent Volume



Plugins

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

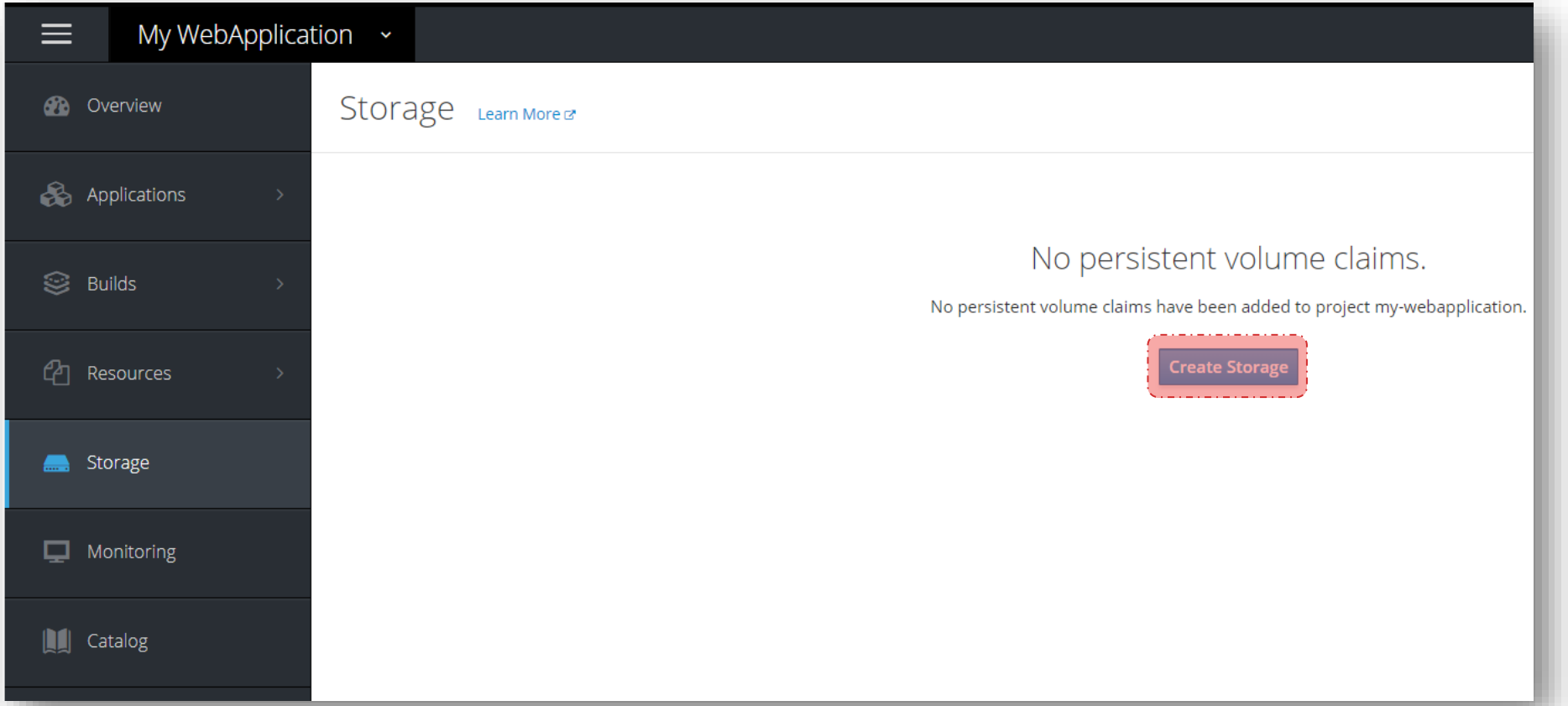




PV Resources – 1TB

KODEKLOUD

Create Storage



KODEKLOUD

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

my-storage-claim

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

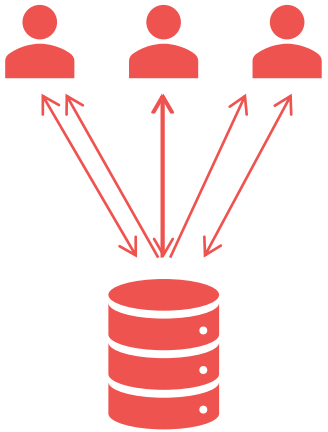
Desired storage capacity.

What are GiB?

Use label selectors to request storage

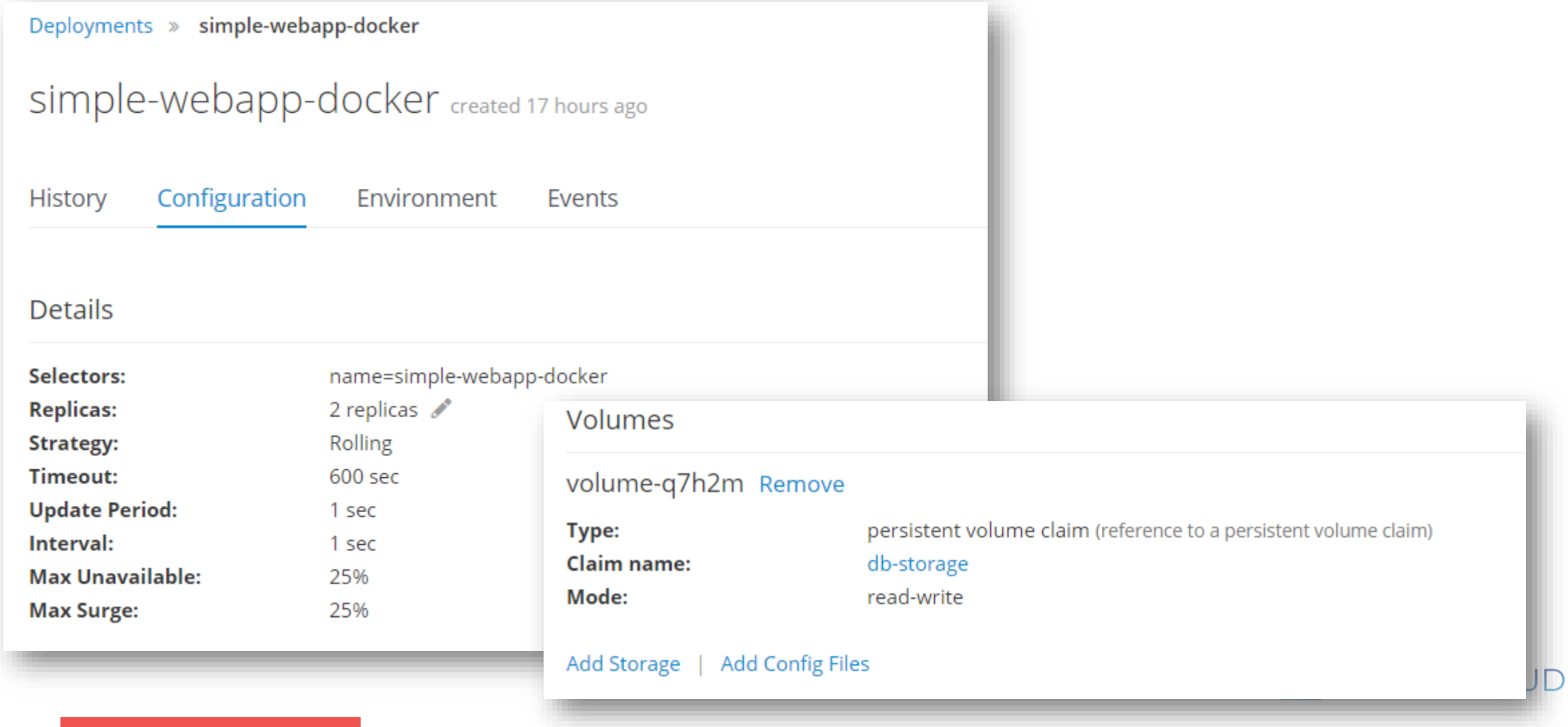
Learn More

Create Cancel



KODEKLOUD

Deployments - Volumes



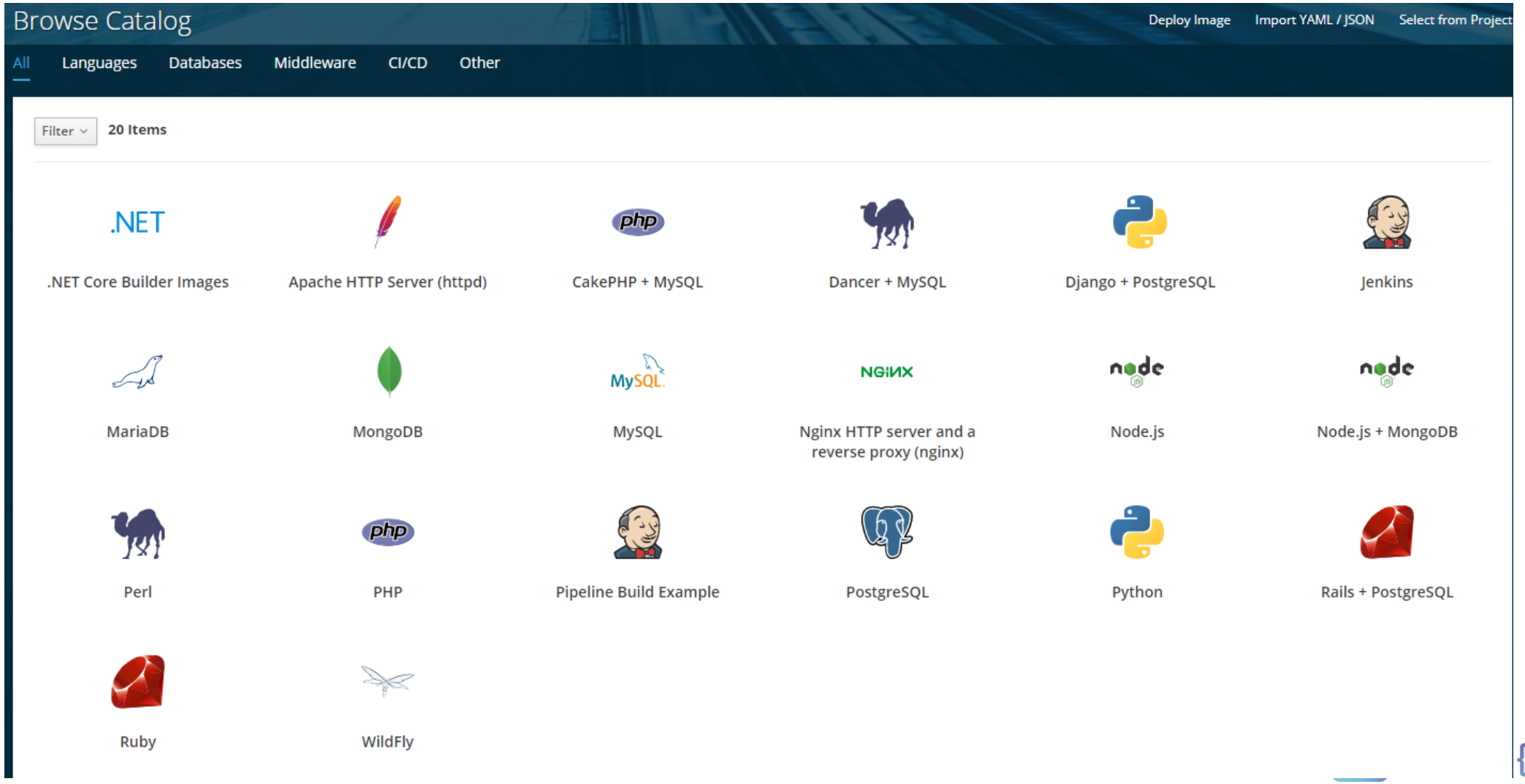

KODEKLOUD

Red Hat

OPENSHIFT

Templates and Catalog

Catalog



Te

Browse Catalog

Deploy Image | Import YAML / JSON | Select from Project

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

Filter | 20 Items

- .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
- Custom App

Image Stream
Application

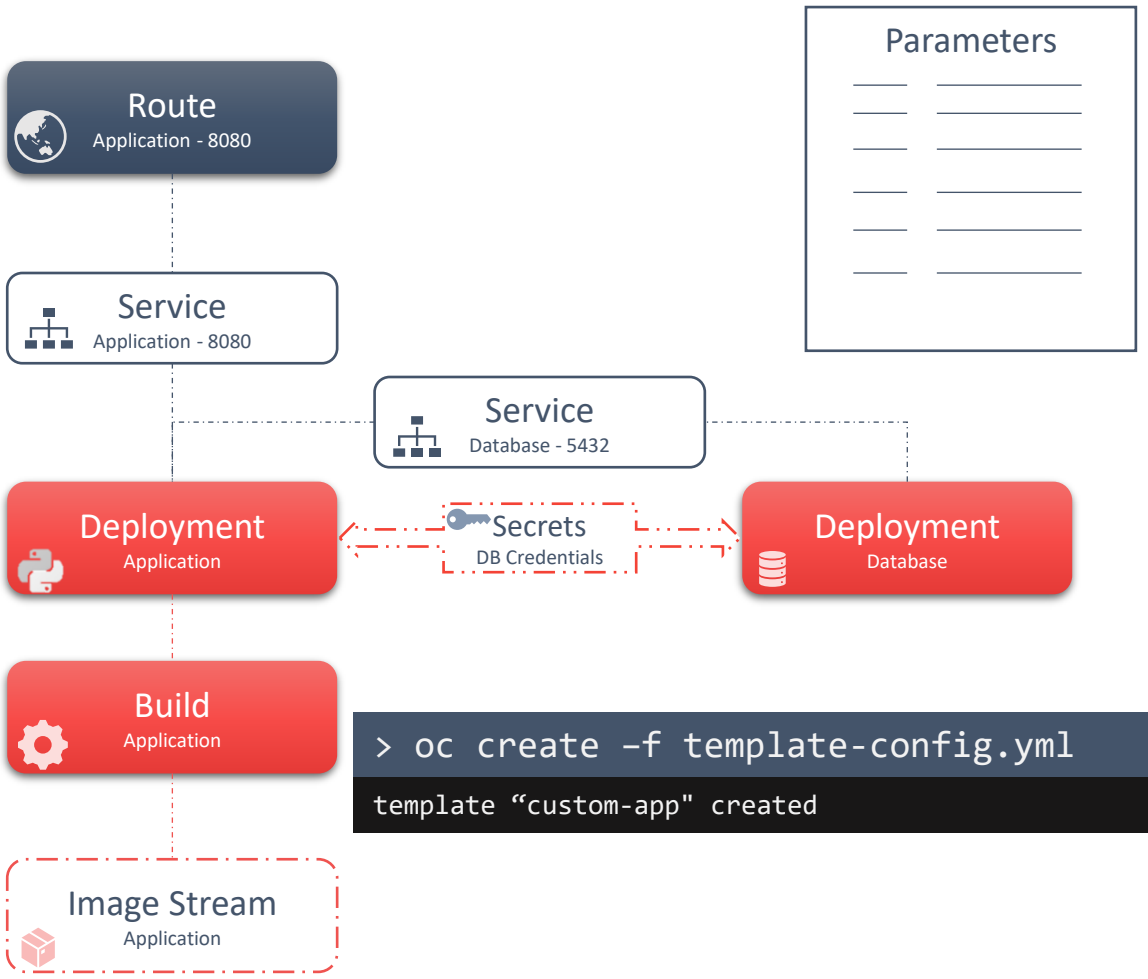
Maximum amount of memory the Django container can use.

* Memory Limit (PostgreSQL)

512Mi

Maximum amount of memory the PostgreSQL container can use.

Template




```
> oc create -f template-config.yml
template "custom-app" created
```

template-config.yaml

```yaml
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"
```

KLOUD

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: {}
```


KODEKLOUD

Conclusion

OpenShift Overview

Architecture

Projects and Users

Builds and Deployments

Networking

Services and Routes

YAML Definition Files

Example Voting Application