

Automation Deployment Use Jenkins In Kubernetes

Yogyakarta Kubernetes Cloud Native at Praxis: Jenkins & Cloud-Native App





Hello!

I am Huda Ridwan

DevOps Engineer @mamikos

Full Time Prayer, Part Time Engineer



mamikos.com

Outline

- Dockerize simple nodejs app
- Manually deployment kubernetes use YAML
- Create service
- Ingress
- CI / CD use Jenkins



History

- Migrate from monolithic system to microservices architecture
- Agile



Why Kubernetes ?

- Open Source as Project of Cloud Native Computing Foundation
- Container Orchestration
- Horizontal Scaling
- Self Healing
- More....



Dockerfile



```
# Get the Node.js base Docker image - shared!
FROM node:11.13.0-alpine
# Set the directory to run our Docker commands in
WORKDIR /app
# Copy your application source to this directory
COPY . ./
RUN npm install
EXPOSE 8081
CMD ["node", "index.js"]
```



Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hogwart-school-deployment
  labels:
    app: hogwarts-school
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hogwarts-school
      track: stable
  template:
    metadata:
      labels:
        app: hogwarts-school
        track: stable
    spec:
      containers:
        - name: hogwarts-school
          image: ${DOCKER_REGISTRY}/${DOCKER_IMAGE_NAME}:${BUILD_NUMBER}
          ports:
            - containerPort: 8081
      restartPolicy: Always
```



Service

```
kind: Service
apiVersion: v1
metadata:
  name: hogwarts-school-service
spec:
  ports:
    - port: 8081
      protocol: TCP
      #type: LoadBalancer
  selector:
    app: hogwarts-school
```



Ingress

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: hogwarts-school-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
spec:
  rules:
    - host: hogwarts.belajarlinux.web.id
      http:
        paths:
          #- path: /
          - backend:
              serviceName: hogwarts-school-service
              servicePort: 8081
```



Ingress

- An API object that manages external access to the services in a cluster, typically HTTP.
- Ingress can provide load balancing, SSL termination and name-based virtual hosting.
- We Use Nginx Ingress, you can use another like : Traefic



Ingress AWS

- `kubectl apply -f`

<https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.27.1/deploy/static/mandatory.yaml>

- `kubectl apply -f`

<https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.27.1/deploy/static/provider/aws/service-l4.yaml>

- `kubectl apply -f`

<https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.27.1/deploy/static/provider/aws/patch-configmap-l4.yaml>



CI / CD

- Why we use CI / CD ?
- Automate Everything
- Kubernetes is compatible with the majority of CI/CD tools



Why Jenkins ?

- Open source and free
- One of the most popular CI/CD tools
- User Friendly
- Easy to install and rich plugins ecosystem
- Available for all platforms and different operating systems, Windows and OS X

Pipeline

- Checkout / Pull code
- Dockerize App
- Push dockerized app to Docker Registry
- Deploy kubernetes deployment



Kubernetes Continuous Deploy^{2.3.0}

Minimum Jenkins requirement: 2.60.3

ID: kubernetes-cd

Installs: 3999

[GitHub](#) →

Last released: 2 months ago

Maintainers

Azure DevOps Team

Dependencies

Pipeline: Step API ≥ 2.10

Azure Commons ≥ 1.0.4

Credentials Binding ≥ 1.16

Credentials ≥ 2.1.14

Docker Commons ≥ 1.10

SSH Credentials ≥ 1.13

Command Agent Launcher v.1.0 (implied)(what's this?)

Oracle Java SE Development Kit Installer v.1.0 (implied)

(what's this?)

JAXB v.2.3.0 (implied)(what's this?)

Trilead API v.1.0.4 (implied)(what's this?)

A Jenkins plugin to deploy resource configurations to a Kubernetes cluster.

It provides the following features:

- Fetch the cluster credentials from the master node via SSH. You may also configure it manually.
- Variable substitution for the resource configurations, allowing you to do dynamic resource deployment.
- Docker login credentials management for the private Docker registry.
- No need to install the `kubectl` tool on the Jenkins slave nodes.



Scope

ID

Description

Kubeconfig ☒ Enter directly

Content

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURStLS0tCk1JSUMwekNDQWJ1Z0F3SUJBZ0lNRmZjaVFU2FjUUUpEbFNMck1BMEdDU3FHU0liM0RRRUJ
Dd1VBTUJVeEV6QVIKQmdOVkBTVRDbXQxWW1WeWJtVjBaWE13SGhjTk1qQXaNakkxTURNd016UXhXaGNOTXpBd01qSTBNRE13TXpReA
pXakFWTVJNd0VRWURWUWFERXdwcmRXSmxjbTVsZEdWek1JSUJJaFQmdrcWhraUc5dzBCQVFFRkFBT0NBUThBCK1JSUJDZ0tDQVFFQ
TNKZ3RuNzBmcGptLyt1cE5raWtHMxhnWE1pUVEwdlJRdHdvU1M2dCtoM3dnN0ZvMlVUclQKV21hZHpuVE5Tjd1eTZkMEpuQmJLcko1aitidnVu
bkNiMUxGcWpVeEJGejM5QktyRFdGcWwwNEpvWVRBUS91MQphdGlzL3AvaUUrUU8wUklSUkdJR1RMeFRXbkdkjKzBVsjZPTi8rMmZTY3BSbjJK
ajl0UUhZwKRMNlXltQm9UZE9lcmijR3pUUURxTGQvRXFnbnGR6YIE3RmJCK2U4WUZJQkVtU3ZROXdqOEZBVnZ0QVR5ZENhUE5NYkxiYVpob
W9SZzIk1hQQW1NWWRBdHJMejN1ZWlVWRFdXd2Z29HS3RqOGdyaXVpZzZ0NjEvUE5sZ0dReStkMUy0VE1Q1RYRfJlawnJc2U3Tzh4N1F0
SEcrSjJkU1poU0RPaDNHL2M3WnNhTnJ3SURBUUFCb3lnd0lUQU9CZ05WSE4QkFmOEVCQU1DCKFRWXdEd1IEVlIwVEFRSC9CQVV3QXdf
```



Preparation

- Install Kubernetes Continuous Deploy Plugin
- Docker Login
- Add jenkins public key into repository
- Copy your .kube/config into credentials



```
env.DOCKER_REGISTRY = 'elqahtani'
env.DOCKER_IMAGE_NAME = 'hogwarts-school'
node('master') {
    stage('HelloWorld') {
        echo 'Hello World'
    }
    stage('Git Pull from Github') {
        git credentialsId: 'github_password', url: 'git@github.com:elqahtani/hogwarts-api.git'
    }
    stage('Build Docker Image') {
        sh "docker build --build-arg APP_NAME=hogwarts-school -t
$DOCKER_REGISTRY/$DOCKER_IMAGE_NAME:${BUILD_NUMBER} ."
    }
    stage('Push Docker Image to Dockerhub') {
        sh "docker push $DOCKER_REGISTRY/$DOCKER_IMAGE_NAME:${BUILD_NUMBER}"
    }
    stage('DeployTo Kubernetes Cluster') {
        kubernetesDeploy(
            kubeconfigId: 'kubeconfigbelajarlinux',
            configs: 'hogwarts-school-deployment.yml',
            enableConfigSubstitution: true
        )
    }
}
```

Demo

Perfection

- Run unit tests
- Integrate with ansible
- Use private repository
- Use webhook trigger
- Store credential use secret in kubernetes
- Delete old images build
- Etc.



Thanks!

Huda Ridwan

@elqahtani

<https://github.com/elqahtani/hogwarts-api>