HPE Container Platform Kubernetes Application Image Development

Lab Solution Worksheet

Deploy “Springboot” Kubernetes Application (KubeDirector) via Web Terminal 2

Build Springboot Docker Image 2

Create Dockerfile 2

Build Docker Image 2

Push Docker Image to Docker Hub 2

Deploy KubeDirector application in HPE Container Platform 4

Login 4

Create KubeDirectorApp file 4

Launch & access the application 6

# Deploy “Springboot” Kubernetes Application (KubeDirector) via Web Terminal

This lab is to create an application and registering it in the Kubernetes Application page. Using it, pods can be spun up.

## Build Springboot Docker Image

This section is to:

1. Create Dockerfile
2. Build Docker Image
3. Push Docker Image to Docker Hub

### Create Dockerfile

Follow the below procedure to create Dockerfile:

1. SSH to a host
2. Create a directory & navigate to it

[root@dev335 ]# mkdir springboot

[root@dev335 ]# cd springboot/

1. Obtain the required jar file

[root@dev335 springboot]# wget <https://bluedata-srujan.s3.amazonaws.com/dev/bins/gs-actuator-service-0.1.0.jar>

1. Create Dockerfile with the given contents

FROM java:8

WORKDIR /home

RUN wget https://bluedata-srujan.s3.amazonaws.com/dev/bins/gs-actuator-service-0.1.0.jar -P /home/

EXPOSE 9000

CMD java -jar gs-actuator-service-0.1.0.jar

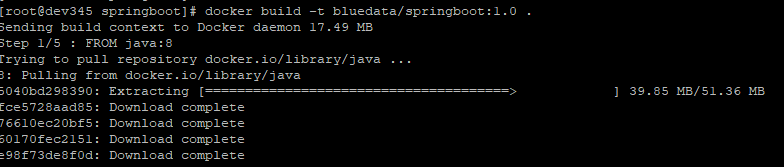


### Build Docker Image

Follow the below procedure to build Docker image

1. Execute the command to build Docker image

[root@dev335 springboot]# docker build -t bluedata/springboot:1.0 .



### Push Docker Image to Docker Hub

Follow the below procedure to push Docker image to Docker Hub

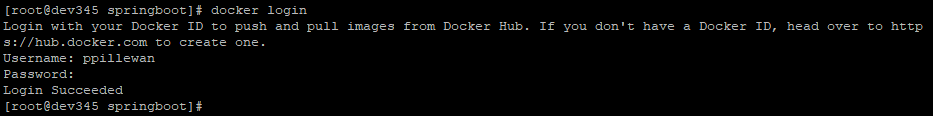
1. View the created Docker Image

[root@dev345 springboot]# docker images



1. Login to Docker

[root@dev345 springboot]# docker login



1. Tag the Docker image

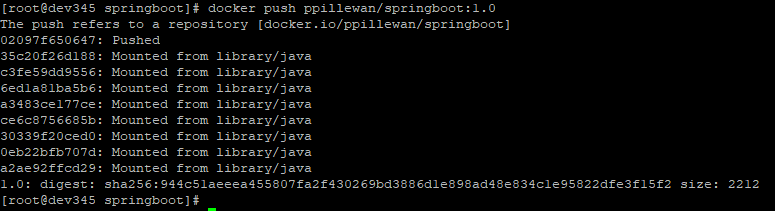
[root@dev345 springboot]# docker tag c6529c4466e8 ppillewan/springboot:1.0

**Note:** ppillewan is the username here, you can replace it with those you used in the previous step.

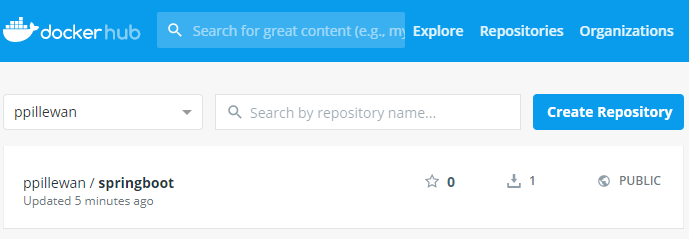


1. Push the Docker image

[root@dev345 springboot]# docker push ppillewan/springboot:1.0



1. Verify the image in Docker Hub

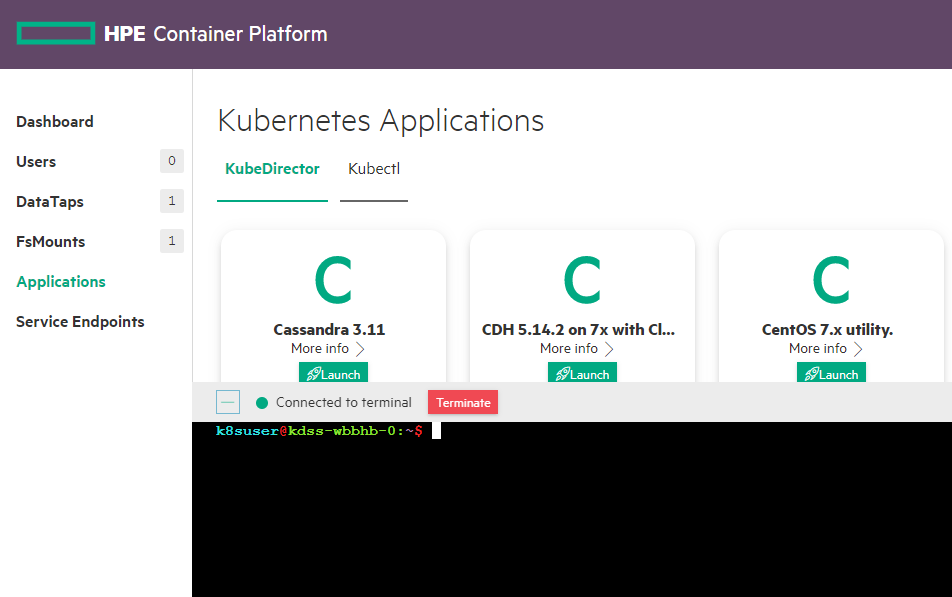


## Deploy KubeDirector application in HPE Container Platform

### Login

Follow the procedure

1. Login to HPE Container Platform Web UI
2. From the left-hand menu, click on **Kubernetes** -> **Tenants**
3. Enter any Kubernetes Tenants
4. Click on **Applications**
5. Open the Kubernetes Web Terminal from below

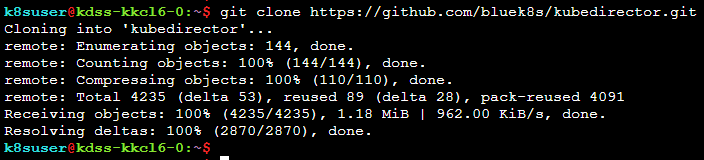


### Create KubeDirectorApp file

Follow the below procedure

1. Clone the KubeDirector project

k8suser@kdss-kkcl6-0:~$ git clone <https://github.com/bluek8s/kubedirector.git>



1. Navigate to the given location

k8suser@kdss-kkcl6-0:~$ cd kubedirector/deploy/example\_catalog/



1. Create cr-app-primews-springboot.json file with the given content

{

"apiVersion": "kubedirector.hpe.com/v1beta1",

"kind": "KubeDirectorApp",

"metadata": {

"name": "prime-springboot-app-svc"

},

"spec": {

"systemdRequired": true,

"defaultPersistDirs": ["/home"],

"config": {

"roleServices": [

{

"serviceIDs": [

"ssh",

"prime-service"

],

"roleID": "springboot"

}

],

"selectedRoles": [

"springboot"

]

},

"label": {

"name": "Springboot Webservice For Primes",

"description": "Web service to detect primenumbers in Sprintboot"

},

"distroID": "bluedata/PrimeWS",

"version": "1.1",

"configSchemaVersion": 7,

"services": [

{

"endpoint": {

"port": 22,

"isDashboard": false

},

"id": "ssh",

"label": {

"name": "SSH"

}

},

{

"endpoint": {

"urlScheme": "http",

"path": "/prime",

"isDashboard": true,

"port": 9000

},

"id":"prime-service",

"label": {

"name": "Prime service"

}

}

],

"defaultImageRepoTag": "docker.io/ppillewan/springboot:1.0",

"defaultConfigPackage": null,

"roles": [

{

"cardinality": "1",

"id": "springboot"

}

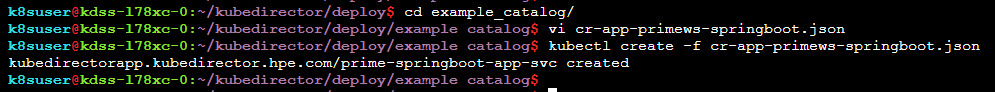
]

}

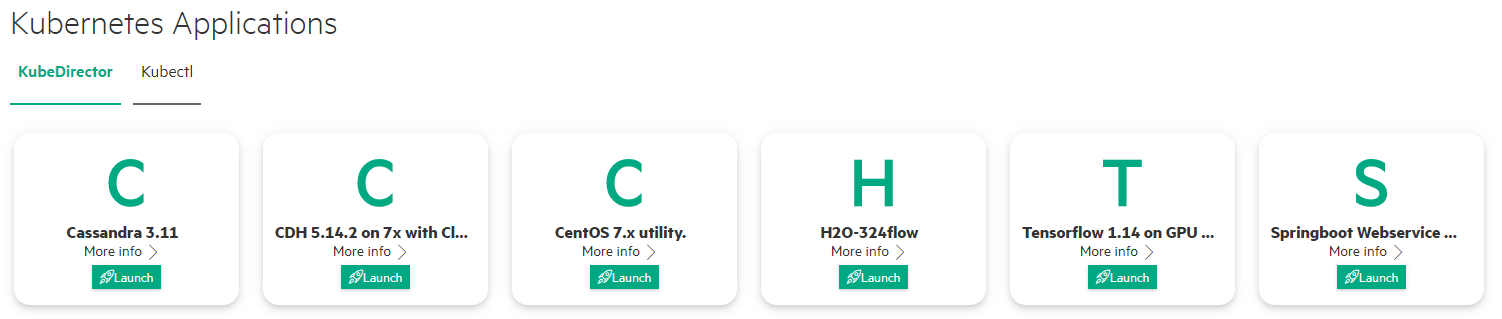
}

1. Create the JSON file

k8suser@kdss-kkcl6-0:~/kubedirector/deploy/example\_catalog$ kubectl create –f cr-app-primews-springboot.json

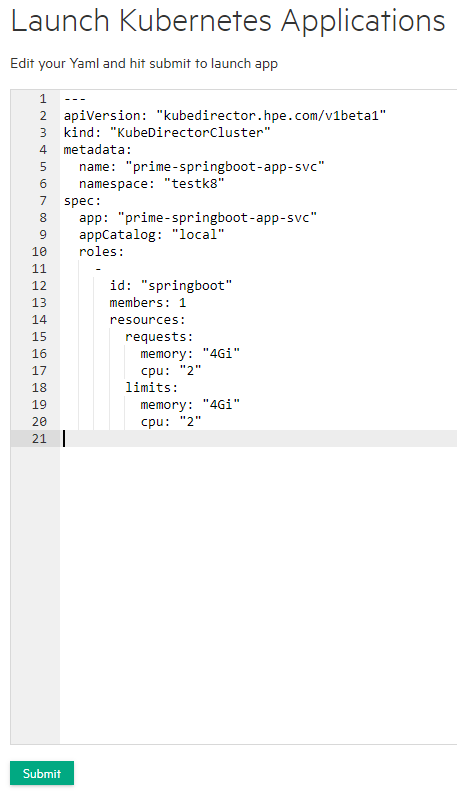


1. On the Kubernetes Application page, a new application will appear

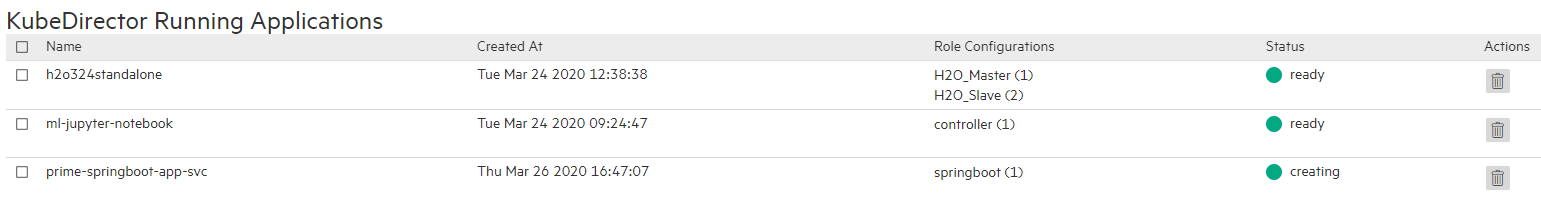


## Launch & access the application

1. Click on the **Launch** button on the Application, a Launch Kubernetes Application page will come (make edits, if required) and click on **Submit**

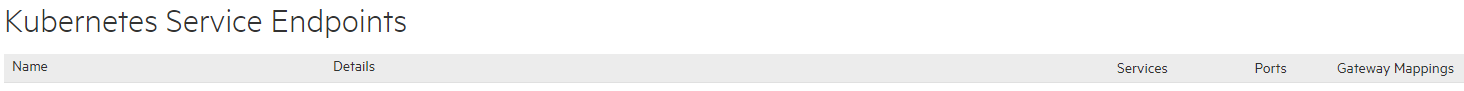


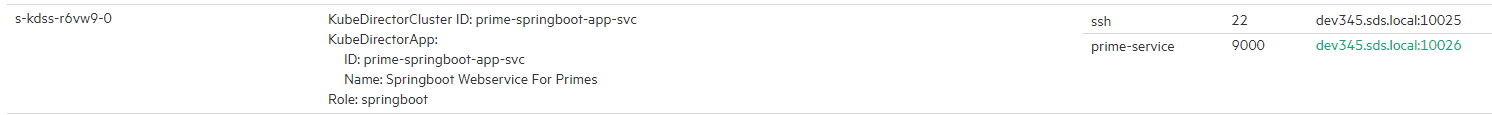
1. On the KubeDirector Running Applications section, the new application will start getting created (watch the Status column)





1. Once the application is in ready state, it can be accessible via service endpoints





1. Clicking on **Gateway mappings** entry will open-up a new tab, append **/prime?number=X** to test the application (Where X, can be any number to check if its prime or not)

