HPE Container Platform Kubernetes Application Image Development

Lab Solution Worksheet

Convert Standard EPIC Image to KubeDirector App (H20) via Web Terminal 2

Build H20 Docker Image 2

Create Dockerfile 2

Build Docker Image 3

Push Docker Image to Docker Hub 3

Deploy KubeDirector application(H20) in HPE Container Platform 5

Login 5

Create KubeDirectorApp file 5

Launch & access the application 8

# Convert Standard EPIC Image to KubeDirector App (H20) via Web Terminal

This lab is to convert a standard EPIC image to Kubernetes Application. Using it, pods can be spun up.

## Build H20 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 H20

[root@dev335]# cd H20/

1. Obtain the required tgz file

(s3 link for appconfig.tgz - <https://bluedata-srujan.s3.amazonaws.com/dev/bins/appconfig.tgz>)

[root@dev335 H20]# wget <https://bluedata-srujan.s3.amazonaws.com/dev/bins/appconfig.tgz>

1. Create Dockerfile with the given contents

# image for RHEL/CentOS

FROM bluedata/centos7:latest

#

# Update and then install Shellinabox

#

RUN yum install epel-release -y

RUN yum install -y shellinabox.x86\_64

#

#Install Oracle jdk

#

RUN yum install -y java-11-openjdk.x86\_64

#

#

#

RUN mkdir -p /usr/lib/hadoop/conf

#

# Install aws jars

#

RUN wget -q http://s3.amazonaws.com/bluedata-catalog/thirdparty/aws-jars/aws-java-sdk-1.7.4.jar -P /opt/bluedata/

RUN wget -q http://s3.amazonaws.com/bluedata-catalog/thirdparty/aws-jars/hadoop-aws-2.7.1.jar -P /opt/bluedata/

# Install H2O

RUN mkdir -p /usr/lib/h2o

RUN wget http://h2o-release.s3.amazonaws.com/h2o/rel-yates/2/h2o-3.24.0.2.zip

RUN unzip h2o-3.24.0.2.zip -d /usr/lib/h2o

RUN ln -s /usr/lib/h2o/h2o-3.24.0.2 /usr/lib/h2o/latest

RUN mkdir /usr/lib/h2o/latest/conf

RUN useradd -r h2o

RUN chown -Rf h2o /usr/lib/h2o\*

## Download thirdparty aws jars

RUN wget -q http://s3.amazonaws.com/bluedata-catalog/thirdparty/aws-jars/aws-java-sdk-1.7.4.jar -P /opt/bluedata/

RUN wget -q http://s3.amazonaws.com/bluedata-catalog/thirdparty/aws-jars/hadoop-aws-2.7.1.jar -P /opt/bluedata/

# make spark bin dir accessible to all

RUN echo "export PATH=$PATH:/usr/lib/h2o/latest/bin/" > /etc/profile.d/updatePath.sh

#

# Update and then install Shellinabox

#

RUN yum install epel-release -y

RUN yum install -y shellinabox.x86\_64

RUN mkdir /opt/configscripts/

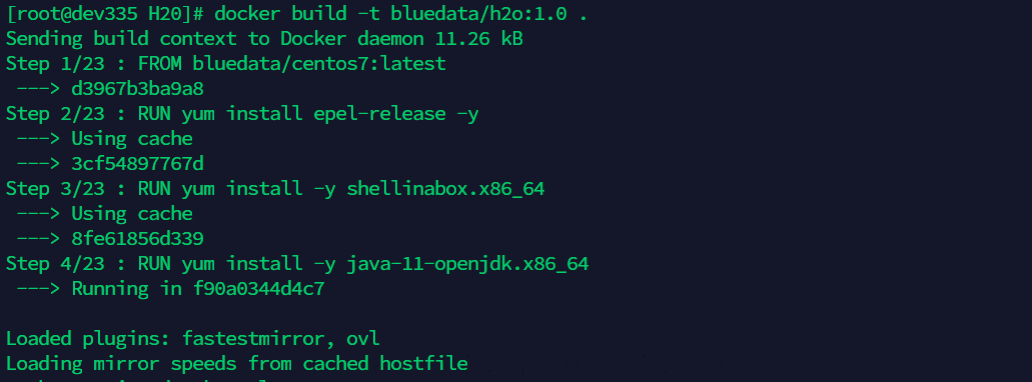
COPY appconfig.tgz /opt/configscripts/

### Build Docker Image

Follow the below procedure to build Docker image

1. Execute the command to build Docker image

[root@dev335 H20]# docker build -t bluedata/h20:1.0 .



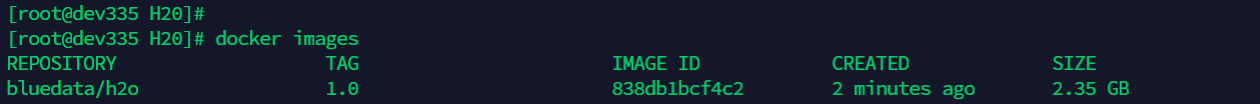
Note: If internet is unavailable then need to modify command to build docker image and add the proxy with docker build command

### Push Docker Image to Docker Hub

Follow the below procedure to push Docker image to Docker Hub

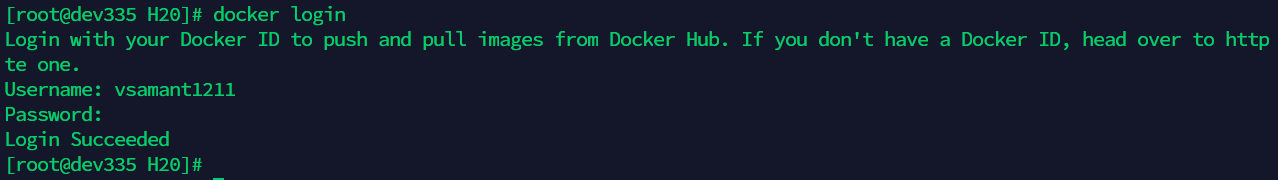
1. View the created Docker Image

[root@dev335 H20]# docker images



1. Login to Docker

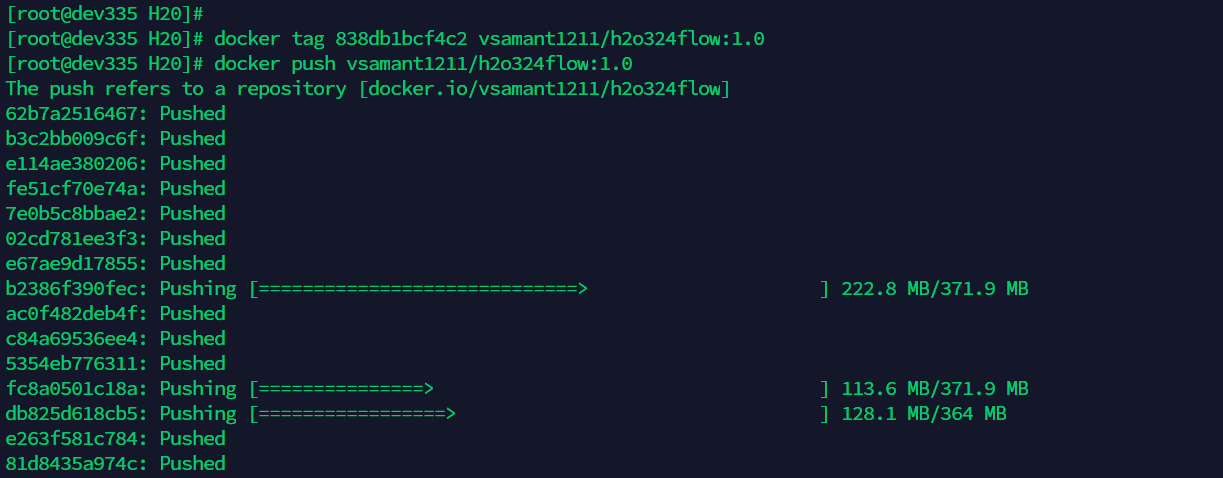
[root@dev335 H20]# docker login



1. Tag the Docker image

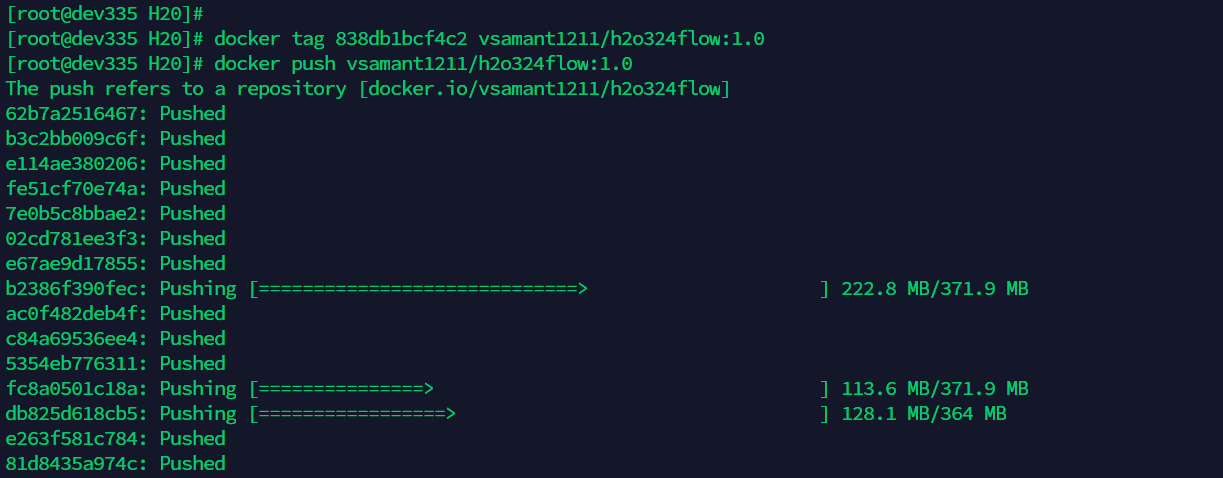
[root@dev335 H20]# docker tag 838db1bcf4c2 vsamant1211/h2o324flow:1.0

**Note:** vsamant1211 is the username here, you can replace it with those you used in the previous step.

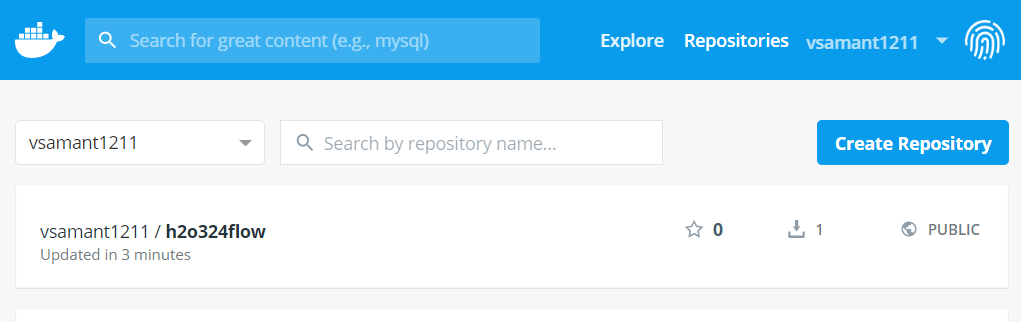


1. Push the Docker image

[root@dev335 H20]# docker push vsamant1211/h2o324flow:1.0



1. Verify the image in Docker Hub

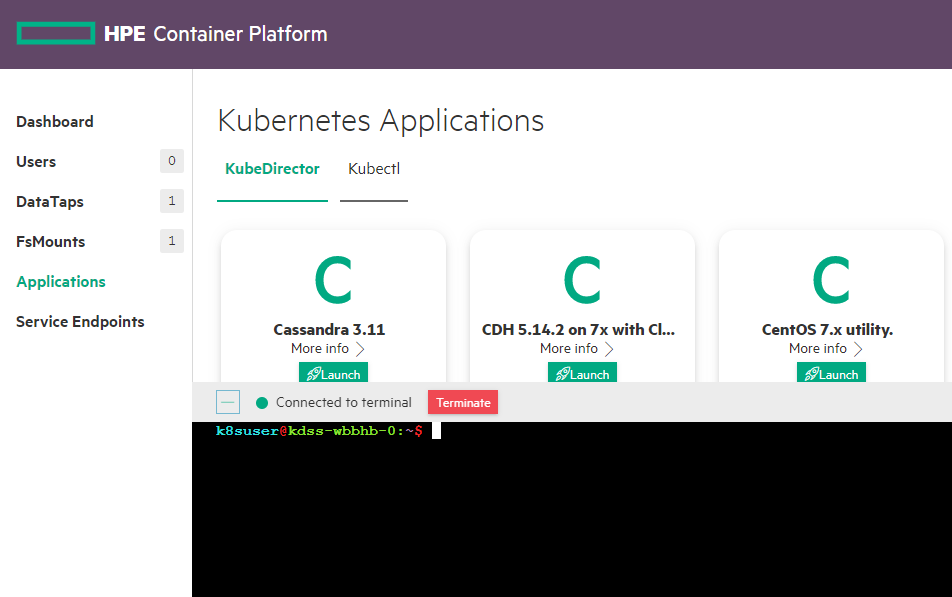


## Deploy KubeDirector application (H20) in HPE Container Platform

### 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

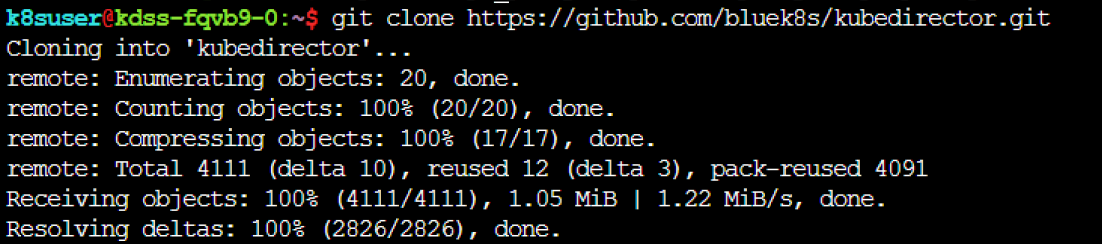


### Create KubeDirectorApp file

Follow the below procedure

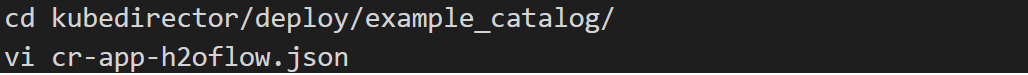
1. Clone the KubeDirector project

k8suser@kdss-fqvb9-0:~$ git clone <https://github.com/bluek8s/kubedirector.git>

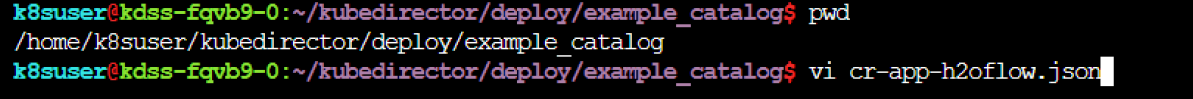


1. Navigate to the given location

k8suser@kdss-fqvb9-0:~$ cd kubedirector/deploy/example\_catalog/



1. Create cr-app-h2oflow.json file with the given content



{

"kind": "KubeDirectorApp",

"spec": {

"defaultImageRepoTag": "docker.io/vsamant1211/h2o324flow:1.0",

"version": "1.0",

"roles": [

{

"cardinality": "1",

"id": "H2O\_Master"

},

{

"cardinality": "2+",

"id": "H2O\_Slave"

}

],

"distroID": "bluedata/h2o324flow",

"label": {

"name": "H2O-324flow",

"description": "H2O Standalone 324flow"

},

"systemdRequired": true,

"configSchemaVersion": 7,

"defaultConfigPackage": {

"packageURL": "file:///opt/configscripts/appconfig.tgz"

},

"services": [

{

"endpoint": {

"port": 22,

"isDashboard": false

},

"id": "ssh",

"label": {

"name": "SSH"

}

},

{

"endpoint": {

"path": "/",

"urlScheme": "http",

"port": 54321,

"isDashboard": true

},

"id": "h2o",

"label": {

"name": "H2O"

}

},

{

"endpoint": {

"path": "/",

"urlScheme": "https",

"port": 4200,

"isDashboard": true

},

"id": "webssh",

"label": {

"name": "Web ssh"

}

}

],

"defaultPersistDirs": [

"/usr",

"/opt",

"/var",

"/data"

],

"config": {

"selectedRoles": [

"H2O\_Master",

"H2O\_Slave"

],

"roleServices": [

{

"roleID": "H2O\_Master",

"serviceIDs": [

"ssh",

"h2o",

"webssh"

]

},

{

"roleID": "H2O\_Slave",

"serviceIDs": [

"ssh",

"h2o"

]

}

]

}

},

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

"metadata": {

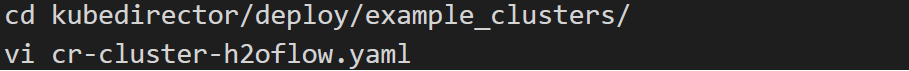
"name": "h2o324standalone"

}

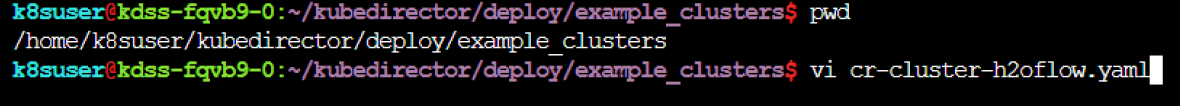
}

1. Navigate to the given localtion

k8suser@kdss-fqvb9-0:~$ cd kubedirector/deploy/example\_cluster/



1. Create cr-cluster-h2oflow.yaml file with the given content



apiVersion: "kubedirector.hpe.com/v1beta1"

kind: "KubeDirectorCluster"

metadata:

name: "h2o321standalone"

spec:

app: h2o324standalone

roles:

- id: H2O\_Master

resources:

requests:

memory: "4Gi"

cpu: "2"

limits:

memory: "4Gi"

cpu: "2"

- id: H2O\_Slave

members: 2

resources:

requests:

memory: "4Gi"

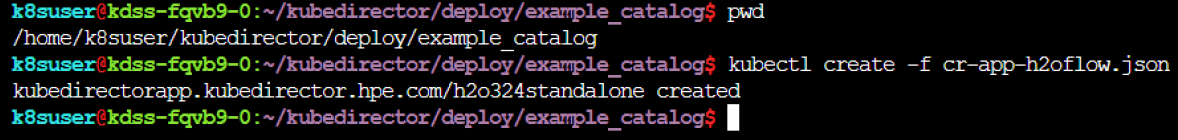
cpu: "2"

limits:

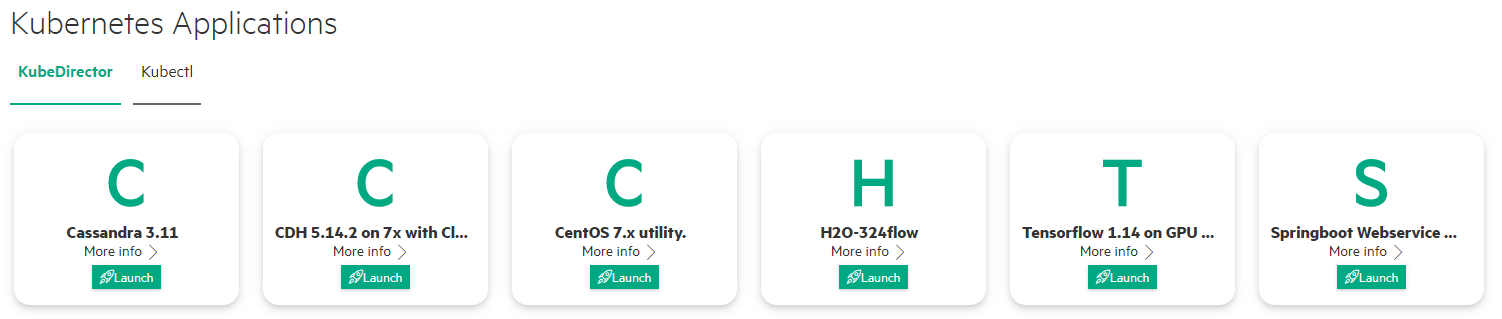
memory: "4Gi"

cpu: "2"

1. Create the application using JSON file to show up in HPECP UI:

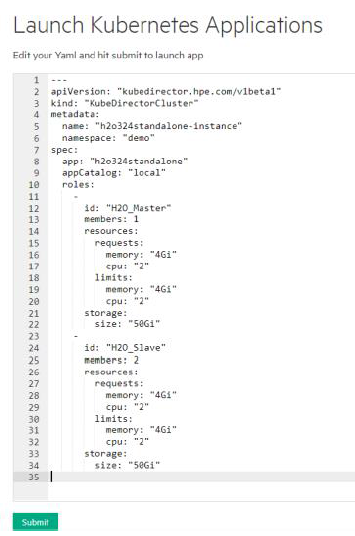


1. On the Kubernetes Application page, a new application will appear

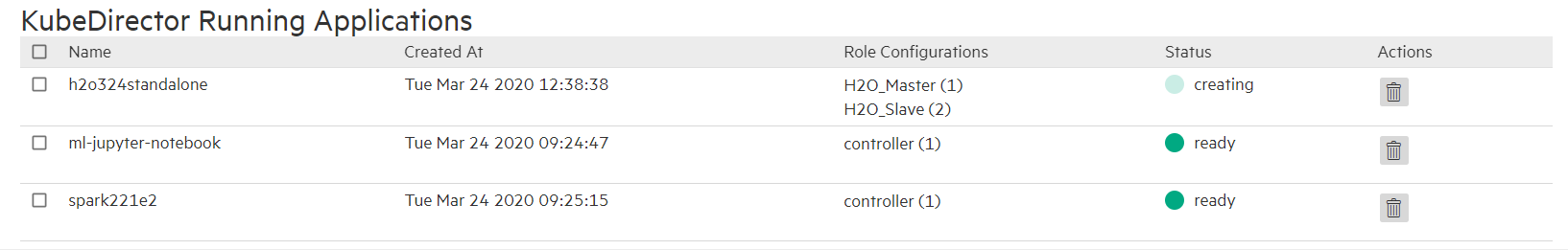


## Launch & access the application

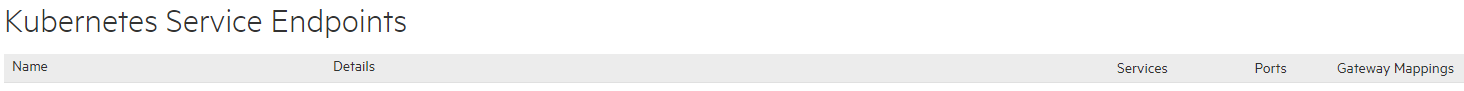
1. Click on the **Launch** button on the Application, a Launch Kubernetes Application page will come (make edits, if required) and click on **Submit**

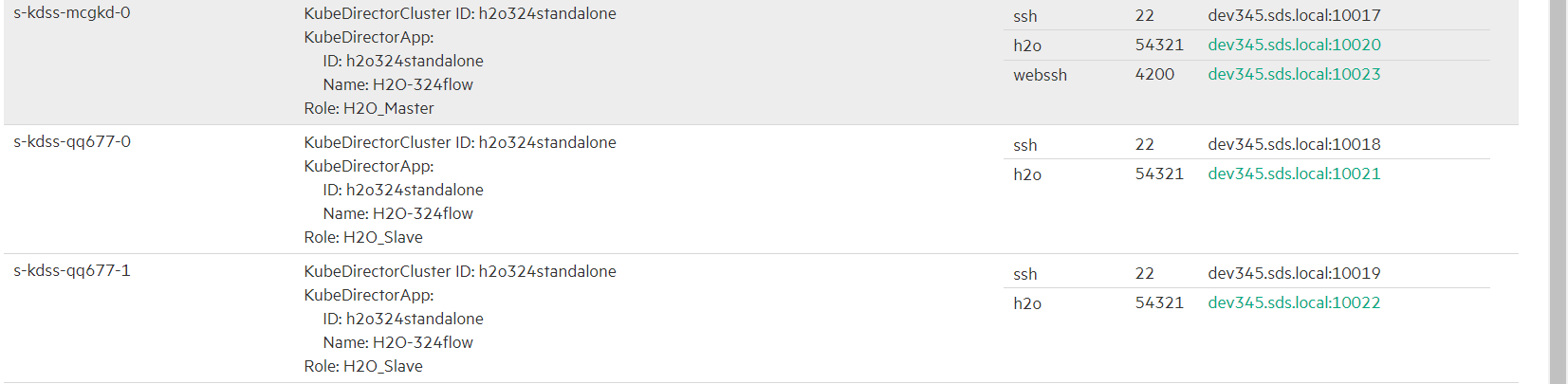


1. On the KubeDirector Running Applications section, the new application will start getting created (watch the Status column)



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, you can see the h20 web UI

