

SWE645 - HW2: Containerize survey form and setup Kubernetes cluster using Rancher

Team Member: Aniket Pandey, Siddhanth Kalyanpur, Tanush Singh

Step 1: Build Docker image and containerize survey form using Docker

- Write Dockerfile as below

```
Dockerfile > ...
1  # Base image.
2  FROM tomcat:8.5.73
3
4  # Author/Maintainer; This will be visible from docker inspect with the other labels.
5  LABEL author.name="Aniket Pandey"
6
7  # Copying war file to tomcat server.
8  ADD Survey.war /usr/local/tomcat/webapps/
```

- Build docker image with the below command. My Dockerfile is inside infra folder and therefore I've specified path as infra/Dockerfile. As a standard practice all the infrastructure setup related files should be kept inside infra folder for easy segregation.
docker build -f infra/Dockerfile --tag survey:amd64-v1.0 .
- Tag the docker image and push it on dockerhub using the below command
docker tag survey:amd64-v1.0 aniket414/survey:amd64-v1.0
docker push aniket414/survey:amd64-v1.0

Step 2: Create two EC2 instance, one for deployment and the other for Rancher setup

- Select the ubuntu ami from AWS marketplace and create two instances
- Allow inbound traffic using security group from following ports

| Inbound rules (9) | | | | | | | | Manage tags | Edit inbound rules |
|-----------------------------|------------|-------------|----------|------------|-----------|--|--|-------------|--------------------|
| Filter security group rules | | | | | | | | < 1 > | ⚙ |
| Security group rule... | IP version | Type | Protocol | Port range | Source | | | | |
| sgr-0225625f9a78f1e54 | IPv4 | HTTP | TCP | 80 | 0.0.0.0/0 | | | | |
| sgr-04640b85a1c4f2a9e | IPv6 | Custom TCP | TCP | 8080 | ::/0 | | | | |
| sgr-06d9bba398e7fabd4 | IPv4 | All traffic | All | All | 0.0.0.0/0 | | | | |
| sgr-024dd4f67097e3ba3 | IPv4 | HTTPS | TCP | 443 | 0.0.0.0/0 | | | | |
| sgr-0246fd54e55a3a1e6 | IPv6 | HTTP | TCP | 80 | ::/0 | | | | |
| sgr-02bd1b51428929... | IPv6 | HTTPS | TCP | 443 | ::/0 | | | | |
| sgr-021b81b98f23fd7f6 | IPv4 | SSH | TCP | 22 | 0.0.0.0/0 | | | | |
| sgr-0365140e4d19cf91b | IPv4 | Custom TCP | TCP | 8080 | 0.0.0.0/0 | | | | |
| sgr-0d96147c2f4eb03f3 | IPv6 | All traffic | All | All | ::/0 | | | | |

Step 3: Setup Rancher on one of the instance

- SSH into the instance using pem file
`ssh -i key-value.pem ubuntu@public-dns`
- Install docker after updating
`sudo apt-get update`
`sudo apt install docker.io`
- Install Rancher using the following command
`sudo docker run --privileged -d --restart=unless-stopped -p 80:80 -p 443:443 rancher/rancher`
- After some time open instance public dns which will display Rancher login page where we'll setup username and password
- Login inside and click on create cluster
- Choose custom option from the list of available of available choices and click next
- Enter the cluster name and leave everything as default and click on next
- In the next section select control plane, etcd, and worker and copy the command generated

Cluster Options

Customize Node Run Command

Editing node options will update the command you will run on your existing machines

1

Node Options

Choose what roles the node will have in the cluster.

Node Role

☒ etcd

☒ Control Plane

☒ Worker

Show advanced options

2

Run this command on one or more existing machines already running a supported version of Docker.

```
sudo docker run -d --privileged --restart=unless-stopped --net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run rancher/rancher-agent:v2.6.4 --server https://ec2-54-227-210-92.compute-1.amazonaws.com --token 7msdjxpf7glcfxhxp8bt9rwsftq6zbn5t5ttclfmqgfcz4xdjcjv --ca-checksum d331501ac4fac75407cf5f13372487c001b5fb9775d73b6f27a0f663378b812d --etcd --controlplane --worker
```

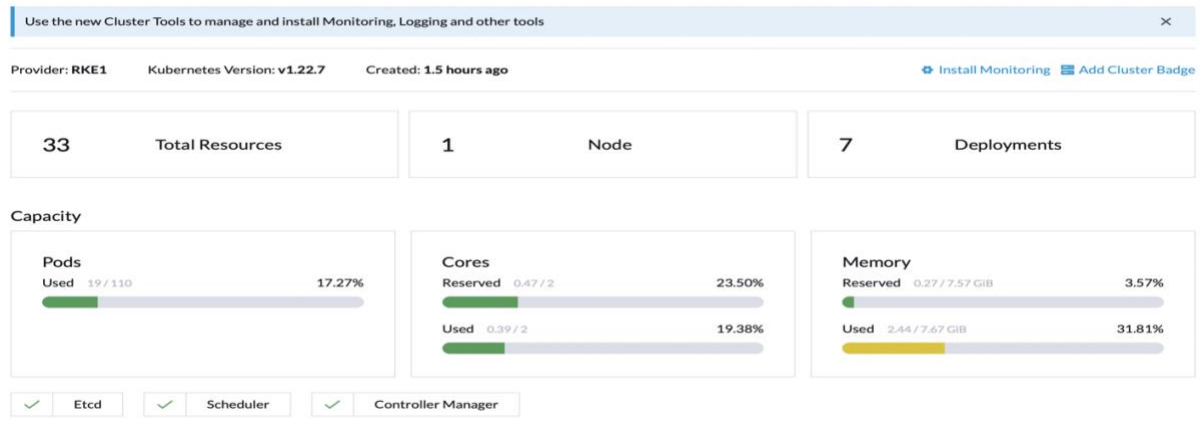
Done

Step 4: Setting up another instance for orchestration

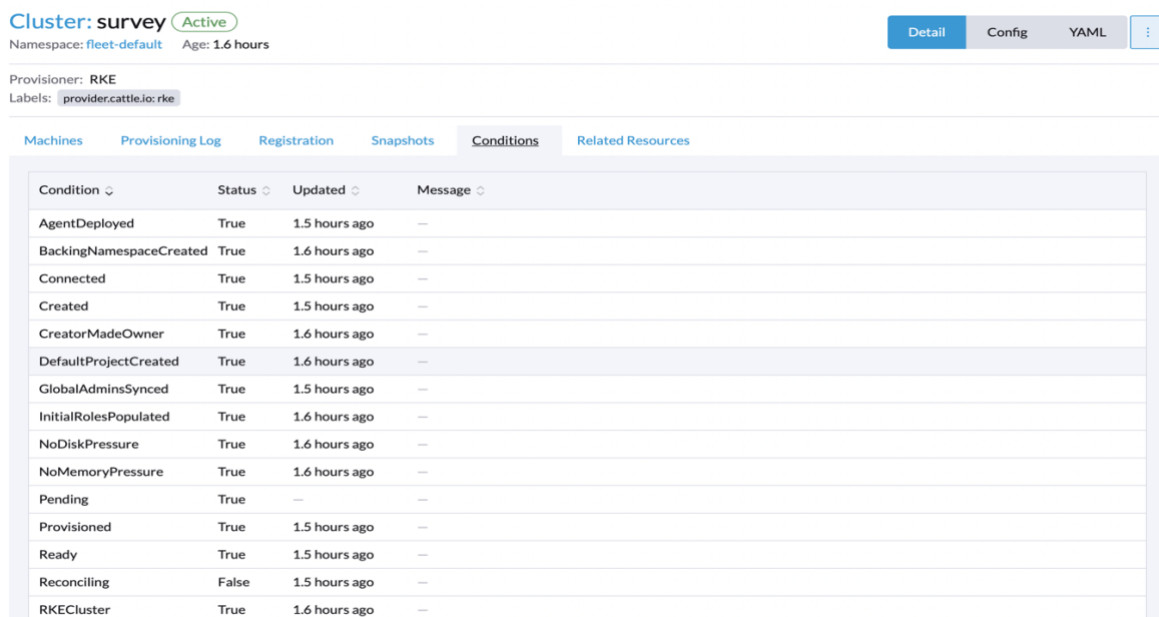
- SSH into the instance using pem file
`ssh -i key-value.pem ubuntu@public-dns`
- Install docker after updating
`sudo apt-get update`
`sudo apt install docker.io`

- Run the above copied command and after some time go to the Rancher page and you should see cluster provisioning has started.

Cluster Dashboard



- Condition of the cluster is active and healthy



Step 5: Deploy the docker image on cluster on setup three pods running at all time

- Enter the cluster just created on Rancher and click on Deployment under the Workload section. Alternatively you can also copy the kubeconfig file and if you have kubectl setup done on your local then paste the kubeconfig file content in .kube/config and you can easily access the cluster and deployment using kubectl from your local.
- Enter the replica set as mentioned in assignment i.e. 3
- Enter the docker image uri

- Setup NodePort as a service

Namespace
default

Name *
survey-deployment

Description
Any text you want that better describes this resource

Replicas *
3

General

Health Check

Labels & Annotations

Networking

Node Scheduling

Pod Scheduling

Resources

Scaling and Upgrade Policy

Security Context

Storage

General

Container Name
container-0

Init Container

Standard Container

Image

Container Image *
aniket414/survey:amd64-v1.0

Pull Policy
Always

Pull Secrets

Ports

Service Type
Node Port

Name
survey-service

Private Container Port
8080

Protocol
TCP

Listening Port
3008

Command

Command
e.g. /bin/sh

Arguments
e.g. /usr/sbin/httpd -f httpd.conf

- Click on done and you should see your pods coming up.

Deployment: survey-deployment Active

Detail
Config
YAML

Namespace: default Age: 57 secs

Image: aniket414/survey:v1.0
Annotations: [Show 1 annotation](#)

Pods by State

3
Running

Pods
Conditions
Related Resources

| State | Name | Node | Image |
|----------------------|------------------------------------|------------------|-----------------------|
| Running | survey-deployment-65b84cfb9f-7knqj | ip-172-31-82-241 | aniket414/survey:v1.0 |
| Running | survey-deployment-65b84cfb9f-hmtrt | ip-172-31-82-241 | aniket414/survey:v1.0 |
| Running | survey-deployment-65b84cfb9f-nd2zg | ip-172-31-82-241 | aniket414/survey:v1.0 |

- You can also verify the status from local using kubectl command

- Following are the screenshot of service, node, deployment, and pods

```
(base) aniket@Anikets-MacBook-Pro Assignment-2 % kubectl get service
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes           ClusterIP     10.43.0.1      <none>          443/TCP          2d13h
survey-deployment    ClusterIP     10.43.84.201   <none>          8080/TCP          2d12h
survey-deployment-nodeport NodePort      10.43.181.229  <none>          8080:30654/TCP   2d12h
(base) aniket@Anikets-MacBook-Pro Assignment-2 % kubectl get nodes
NAME                STATUS    ROLES          AGE      VERSION
ip-172-31-82-241    Ready    controlplane,etcd,worker 2d13h    v1.22.7
(base) aniket@Anikets-MacBook-Pro Assignment-2 % kubectl get pods
NAME                READY    STATUS    RESTARTS   AGE
survey-deployment-667774c59b-fs75l 1/1      Running    0           43h
survey-deployment-667774c59b-rth9q 1/1      Running    0           43h
survey-deployment-667774c59b-t9zsh 1/1      Running    0           43h
(base) aniket@Anikets-MacBook-Pro Assignment-2 % kubectl get deployment
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
survey-deployment  3/3      3              3            2d12h
```