



Lab 4

Portable Technologies in Cloud

Professor: Irina Geiman

CODE: CLO835

Name: Niluxsi Puvanenthiran

Student Number: 119163228

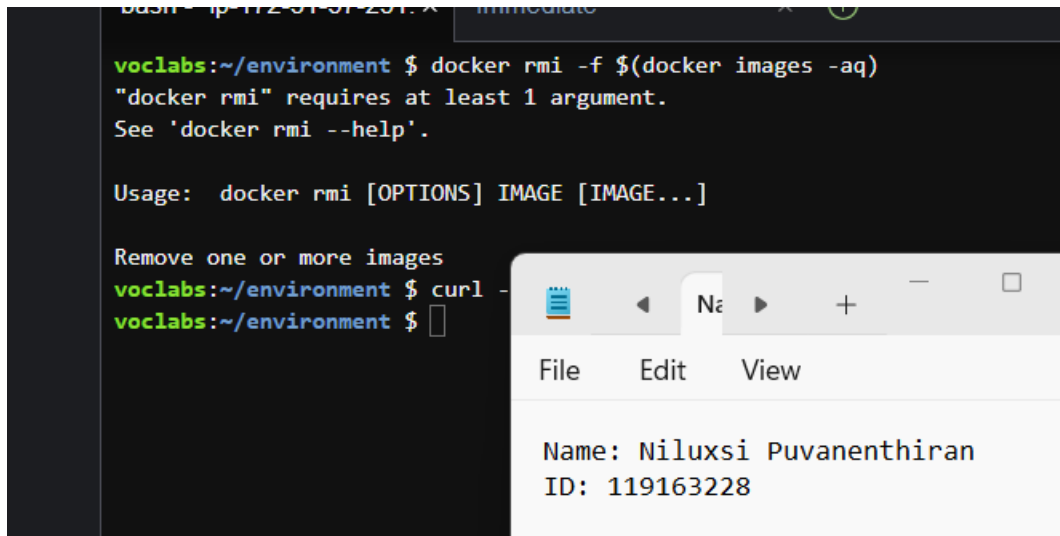
CONTENTS

Pre-requisites	4
Step1: remove all the docker images in Cloud9	4
Step2: Install Kind	4
Step 3: Install kubectl - important! It should match cluster version 1.21	5
Create Cluster Description and Deploy Cluster	6
Step1: Create a config file for the kind cluster	6
Step 2: Create the cluster	6
Deploy our first application	7
Step1: Deploy nginx application	7
Step2: List the pods	7
Step 3: Expose Your application	7
Step 4: List Services	7
Access the application, Scale the pods, Get more information on running pods and nodes, create a deploy a new version	8
Step1: Run docker inspect to retrieve container IP	8
Step 2: Run "kubectl get services" to retrieve the random port that port 80 is mapped to	8
Step 3 : Curl	9
References	10

Screenshot 1 Removing all pre-existing docker images	4
Screenshot 2 Kind package installed.....	4
Screenshot 3 download link.....	5
Screenshot 4 Kubectl was created and version was found.....	5
Screenshot 5 Config file was created.	6
Screenshot 6 Cluster was created.....	6
Screenshot 7 pod nginx created	7
Screenshot 8 Listing the pods	7
Screenshot 9 Application was exposed	7
Screenshot 10 Services were listed.....	7
Screenshot 11 Container id was found and docker inspect was executed	8
Screenshot 12 Container IP was found	8
Screenshot 13 Random port mapped to Port 80 was found	8
Screenshot 14 Output received	9

PRE-REQUISITES

Step1: remove all the docker images in Cloud9



```

voclabs:~/environment $ docker rmi -f $(docker images -aq)
"docker rmi" requires at least 1 argument.
See 'docker rmi --help'.

Usage:  docker rmi [OPTIONS] IMAGE [IMAGE...]

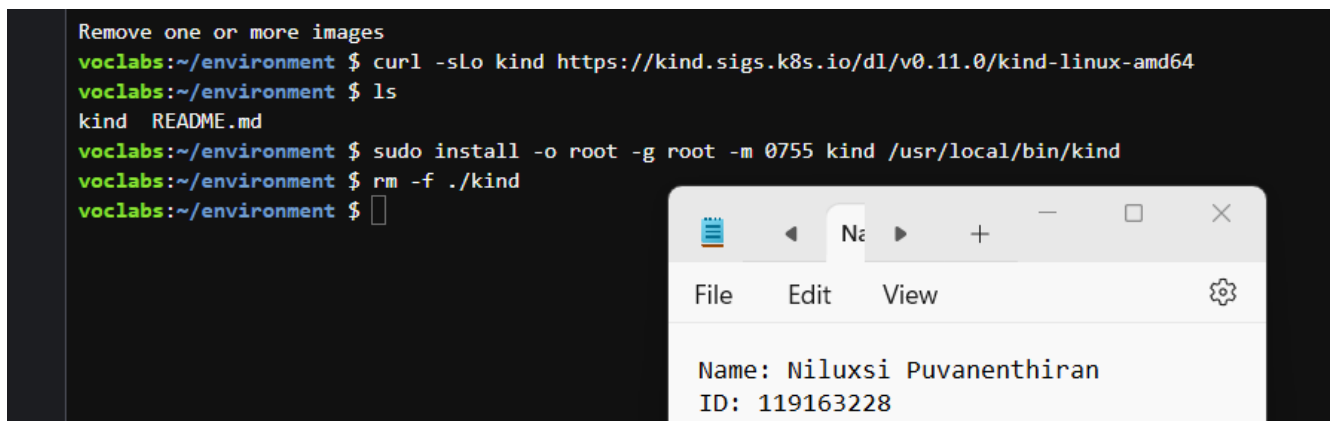
Remove one or more images
voclabs:~/environment $ curl -
voclabs:~/environment $
  
```

Name: Niluxsi Puvanenthiran
ID: 119163228

Screenshot 1 Removing all pre-existing docker images

As there is no image the below error received

Step2: Install Kind



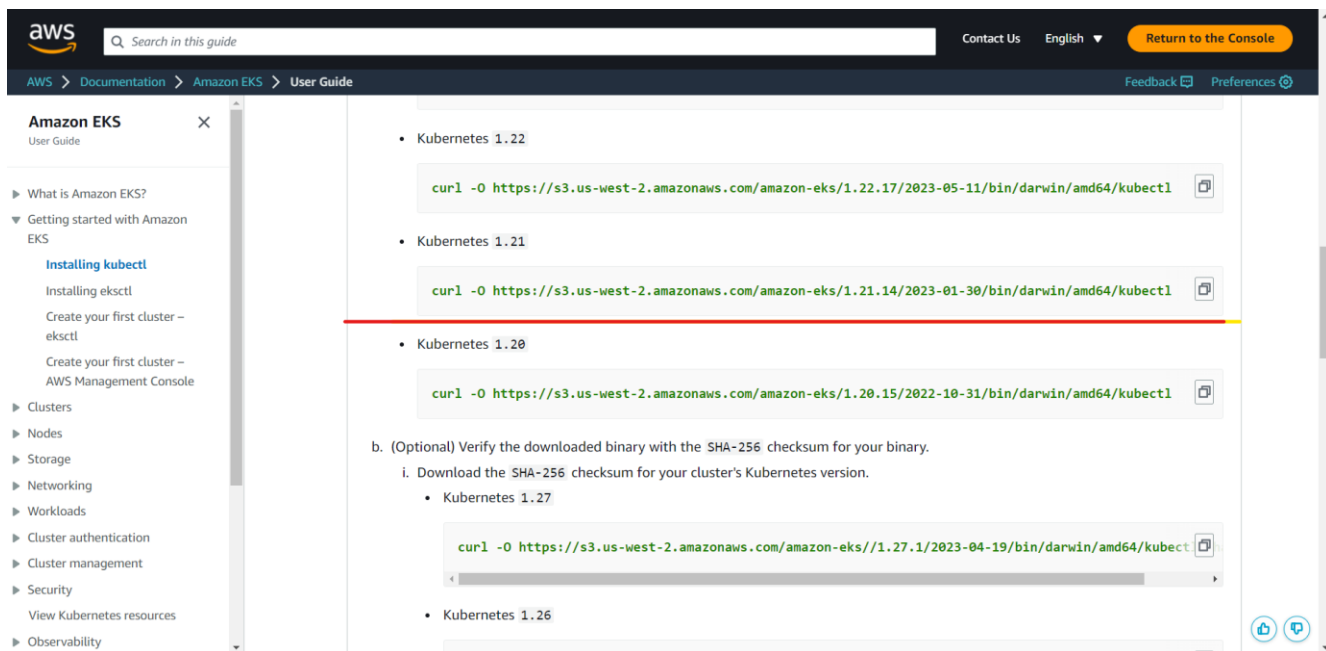
```

Remove one or more images
voclabs:~/environment $ curl -slo kind https://kind.sigs.k8s.io/dl/v0.11.0/kind-linux-amd64
voclabs:~/environment $ ls
kind  README.md
voclabs:~/environment $ sudo install -o root -g root -m 0755 kind /usr/local/bin/kind
voclabs:~/environment $ rm -f ./kind
voclabs:~/environment $
  
```

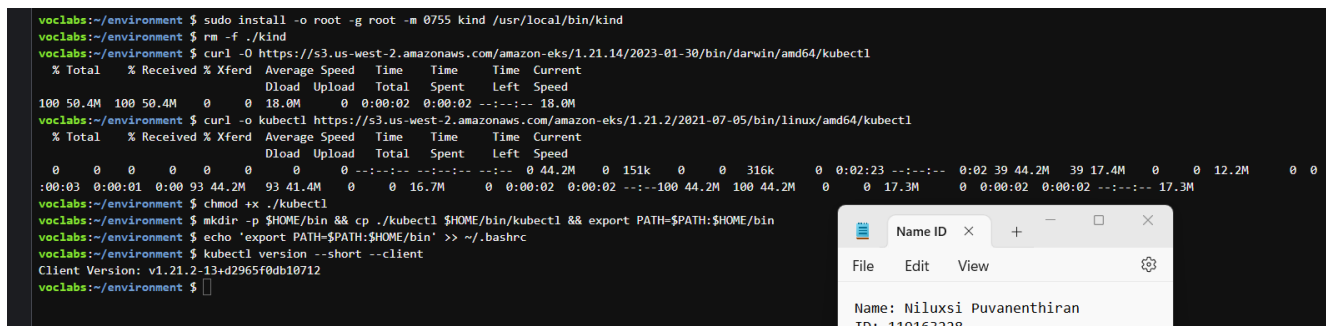
Name: Niluxsi Puvanenthiran
ID: 119163228

Screenshot 2 Kind package installed

Step 3: Install kubectl – important! It should match cluster version 1.21



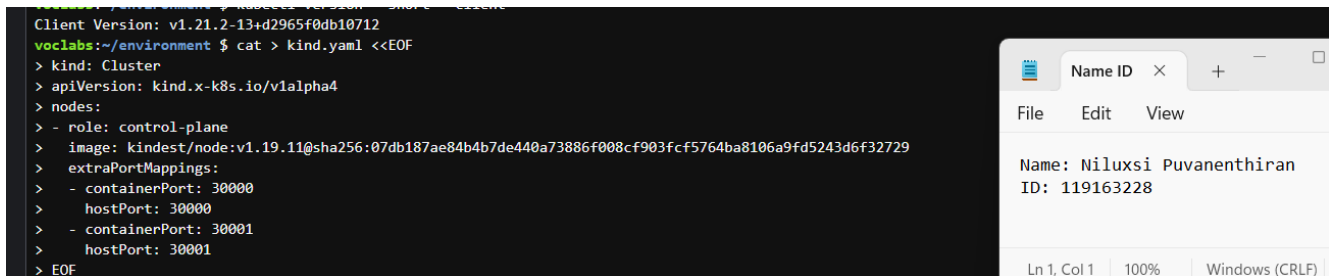
Screenshot 3 download link



Screenshot 4 Kubectl was created and version was found

CREATE CLUSTER DESCRIPTION AND DEPLOY CLUSTER

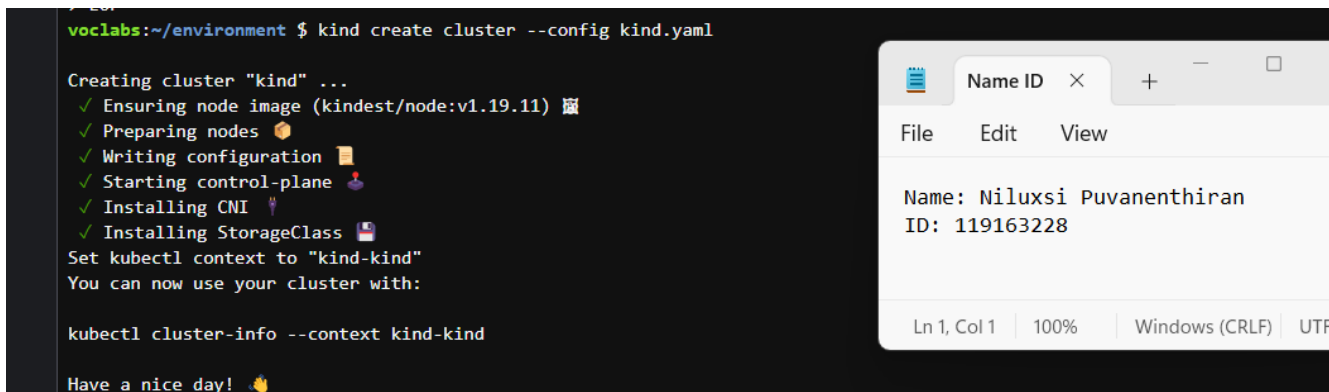
Step1: Create a config file for the kind cluster



```
Client Version: v1.21.2-13+d2965f0db10712
voclabs:~/environment $ cat > kind.yaml <<EOF
> kind: Cluster
> apiVersion: kind.x-k8s.io/v1alpha4
> nodes:
> - role: control-plane
>   image: kindest/node:v1.19.11@sha256:07db187ae84b4b7de440a73886f008cf903fcf5764ba8106a9fd5243d6f32729
>   extraPortMappings:
>     - containerPort: 30000
>       hostPort: 30000
>     - containerPort: 30001
>       hostPort: 30001
> EOF
```

Screenshot 5 Config file was created.

Step 2: Create the cluster



```
voclabs:~/environment $ kind create cluster --config kind.yaml

Creating cluster "kind" ...
 ✓ Ensuring node image (kindest/node:v1.19.11)
 ✓ Preparing nodes
 ✓ Writing configuration
 ✓ Starting control-plane
 ✓ Installing CNI
 ✓ Installing StorageClass
Set kubectl context to "kind-kind"
You can now use your cluster with:

kubectl cluster-info --context kind-kind

Have a nice day! 🍀
```

Screenshot 6 Cluster was created

DEPLOY OUR FIRST APPLICATION

Step1: Deploy nginx application

```
pod "nginx" deleted
voclabs:~/environment $ kubectl run nginx --image=nginx --port=80
pod/nginx created
voclabs:~/environment $ kubectl get pods
```

Screenshot 7 pod nginx created

Step2: List the pods

```
voclabs:~/environment $ rm pods
rm: cannot remove 'pods': No such file or directory
voclabs:~/environment $ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   0           115s
voclabs:~/environment $ kubectl delete pod nginx
pod "nginx" deleted
voclabs:~/environment $ kubectl run nginx --image=nginx --port=80
pod/nginx created
voclabs:~/environment $ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   0           8s
voclabs:~/environment $
```

Screenshot 8 Listing the pods

What are pods? (What are Kubernetes pods, n.d.)

A pod is the smallest execution unit in Kubernetes. A pod encapsulates one or more applications. Pods are ephemeral by nature, if a pod (or the node it executes on) fails, Kubernetes can automatically create a new replica of that pod to continue operations.

Step 3: Expose Your application

```
nginx 1/1 Running 0 8s
voclabs:~/environment $ kubectl expose pod nginx --type=NodePort --name nginx-http
service/nginx-http exposed
voclabs:~/environment $ kubectl get services
```

Screenshot 9 Application was exposed

Step 4: List Services

```
voclabs:~/environment $ kubectl expose pod nginx --type=NodePort --name nginx-http
service/nginx-http exposed
voclabs:~/environment $ kubectl get services
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes ClusterIP  10.96.0.1     <none>         443/TCP          10m
nginx-http NodePort   10.96.214.56 <none>         80:32534/TCP     9s
voclabs:~/environment $
```

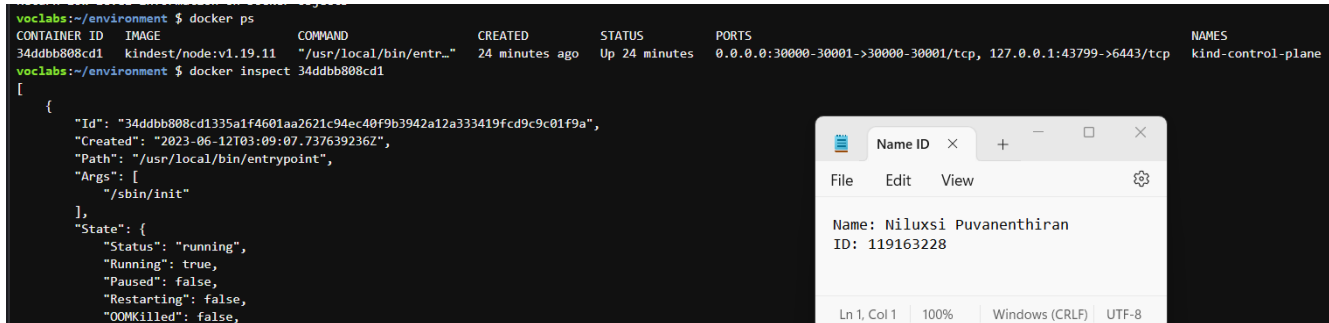
Screenshot 10 Services were listed

What are Services?

A Kubernetes service defines a logical set of pods and a policy for accessing them. It acts as a stable network endpoint (usually with a cluster IP address) that represents a group of pods providing the same functionality

ACCESS THE APPLICATION, SCALE THE PODS, GET MORE INFORMATION ON RUNNING PODS AND NODES, CREATE A DEPLOY A NEW VERSION

Step1: Run docker inspect to retrieve container IP



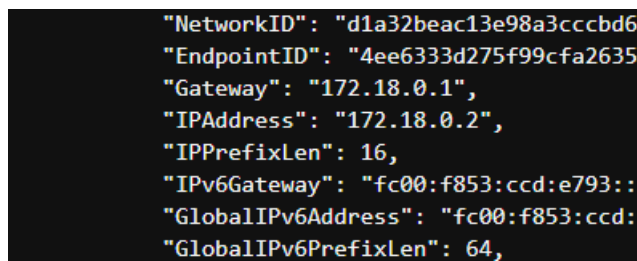
```

voclabs:~/environment $ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
34ddb808cd1   kindest/node:v1.19.11   "/usr/local/bin/entr..." 24 minutes ago Up 24 minutes   0.0.0.0:30000-30001->30000-30001/tcp, 127.0.0.1:43799->6443/tcp   kind-control-plane

voclabs:~/environment $ docker inspect 34ddb808cd1
[
  {
    "Id": "34ddb808cd1335a1f4601aa2621c94ec40f9b3942a12a333419fcd9c9c01f9a",
    "Created": "2023-06-12T03:09:07.737639236Z",
    "Path": "/usr/local/bin/entrypoint",
    "Args": [
      "/sbin/init"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,

```

Screenshot 11 Container id was found and docker inspect was executed



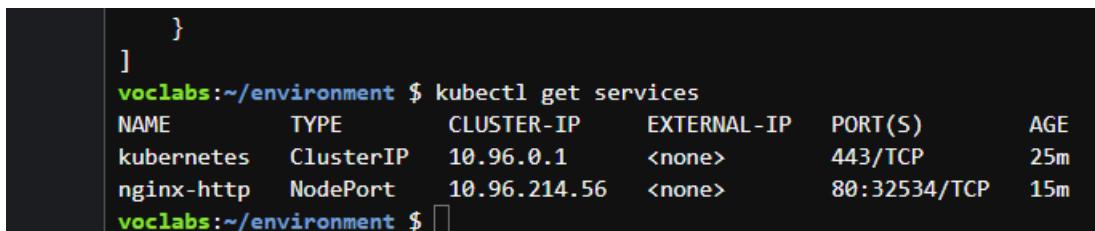
```

"NetworkID": "d1a32beac13e98a3cccbd64",
"EndpointID": "4ee6333d275f99cfa2635f",
"Gateway": "172.18.0.1",
"IPAddress": "172.18.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "fc00:f853:ccd:e793::1",
"GlobalIPv6Address": "fc00:f853:ccd:e793::1",
"GlobalIPv6PrefixLen": 64,

```

Screenshot 12 Container IP was found

Step 2: Run "kubectl get services" to retrieve the random port that port 80 is mapped to



```

}
]
voclabs:~/environment $ kubectl get services
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes    ClusterIP     10.96.0.1    <none>         443/TCP           25m
nginx-http    NodePort      10.96.214.56 <none>         80:32534/TCP      15m
voclabs:~/environment $

```

Screenshot 13 Random port mapped to Port 80 was found

Step 3 : Curl

```
curl: (7) Failed to connect to 172.18.0.2 port 80
curl: (7) Failed to connect to 0.0.127.22 port 80
voclabs:~/environment $ curl 172.18.0.2:32534/TC
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.25.0</center>
</body>
</html>
```

Name ID × + − □

File Edit View

Name: Niluxsi Puvanenthiran
ID: 119163228

Screenshot 14 Output received

REFERENCES

Geiman, I. (2023, Summer). Lectures and Slides, CLO835_Portable Technologies in cloud. Seneca Newham Campus, North York.

Installing or updating kubectl. (n.d.). Retrieved from www.docs.aws.amazon.com/:

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

Learner Lab. (2023). Retrieved from <https://awsacademy.instructure.com/>.

What are Kubernetes pods. (n.d.). Retrieved from

[https://www.vmware.com/topics/glossary/content/kubernetes-](https://www.vmware.com/topics/glossary/content/kubernetes-pods.html#:~:text=A%20pod%20is%20the%20smallest,that%20pod%20to%20continue%20operations.)

[pods.html#:~:text=A%20pod%20is%20the%20smallest,that%20pod%20to%20continue%20operations.](https://www.vmware.com/topics/glossary/content/kubernetes-pods.html#:~:text=A%20pod%20is%20the%20smallest,that%20pod%20to%20continue%20operations.)