|  |  |
| --- | --- |
|  |  |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*YAML Files used in demo\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EXAMPLE OF LABELS

kind: Pod

apiVersion: v1

metadata:

name: delhipod

labels:

env: development

class: pods

spec:

containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-Bhupinder; sleep 5 ; done"]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

NODE SELECTOR EXAMPLE

kind: Pod

apiVersion: v1

metadata:

name: nodelabels

labels:

env: development

spec:

containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-Bhupinder; sleep 5 ; done"]

nodeSelector:

hardware: t2-medium

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EXAMPLE OF REPLICATION CONTROLLER

kind: ReplicationController

apiVersion: v1

metadata:

name: myreplica

spec:

replicas: 2

selector:

myname: Bhupinder Rajput

template:

metadata:

name: testpod6

labels:

myname: Bhupinder

spec:

containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-Bhupinder; sleep 5 ; done"]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EXAMPLE OF REPLICA SET

kind: ReplicaSet

apiVersion: apps/v1

metadata:

name: myrs

spec:

replicas: 2

selector:

matchExpressions: # these must match the labels

- {key: myname, operator: In, values: [Bhupinder, Bupinder, Bhopendra]}

- {key: env, operator: NotIn, values: [production]}

template:

metadata:

name: testpod7

labels:

myname: Bhupinder

spec:

containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Technical-Guftgu; sleep 5 ; done"]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A piece of paper with writing on it

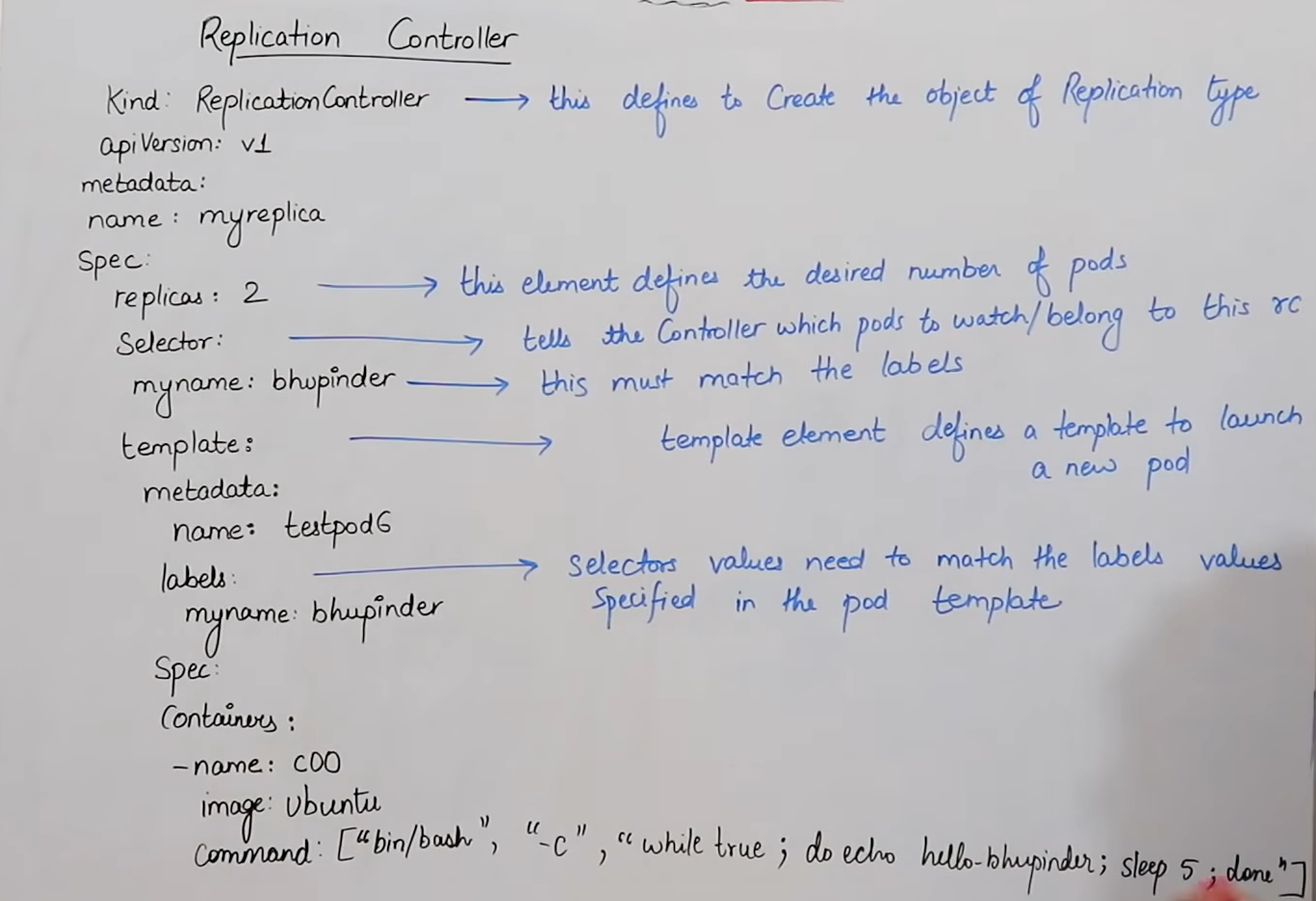
Description automatically generated with medium confidence

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

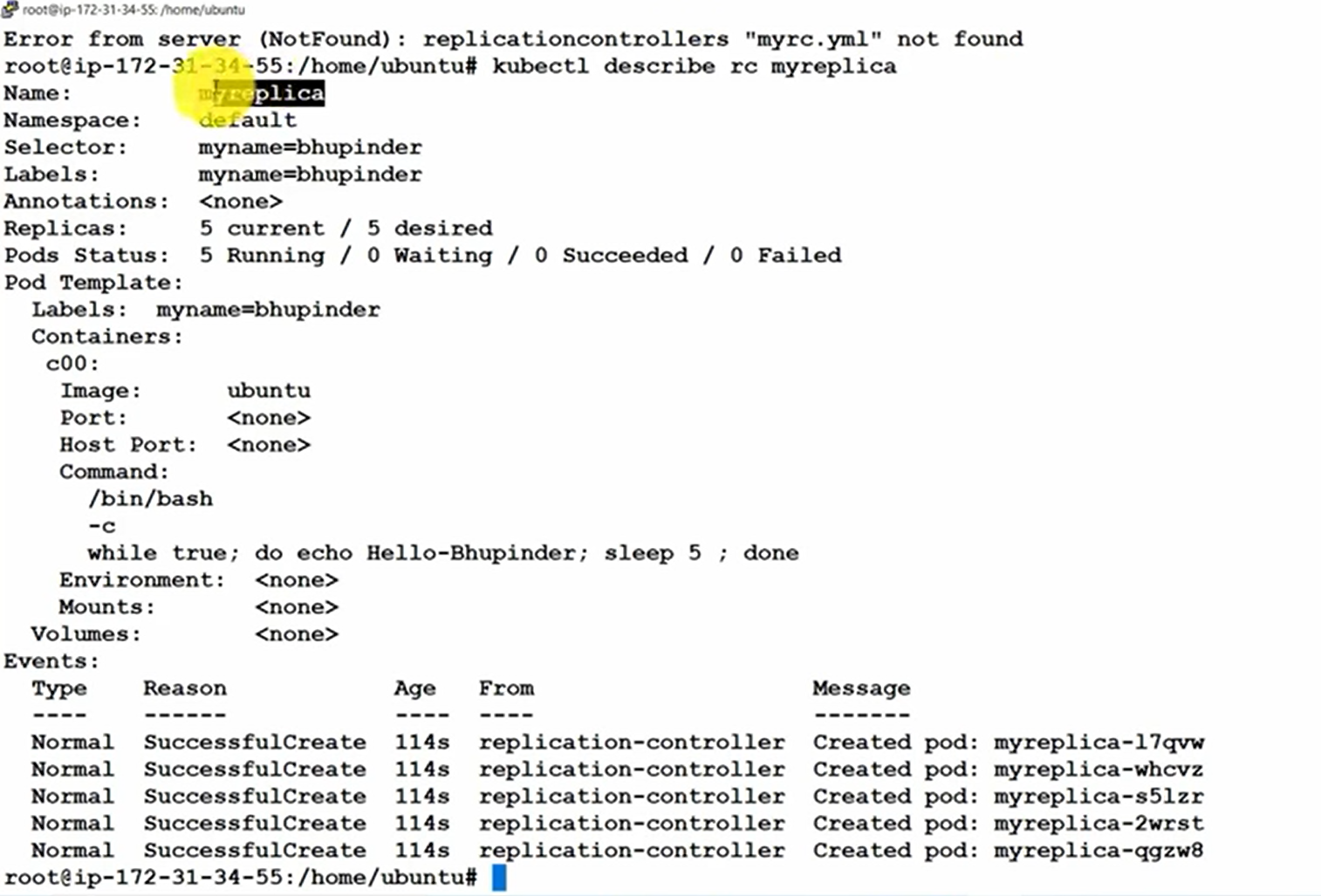
|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |



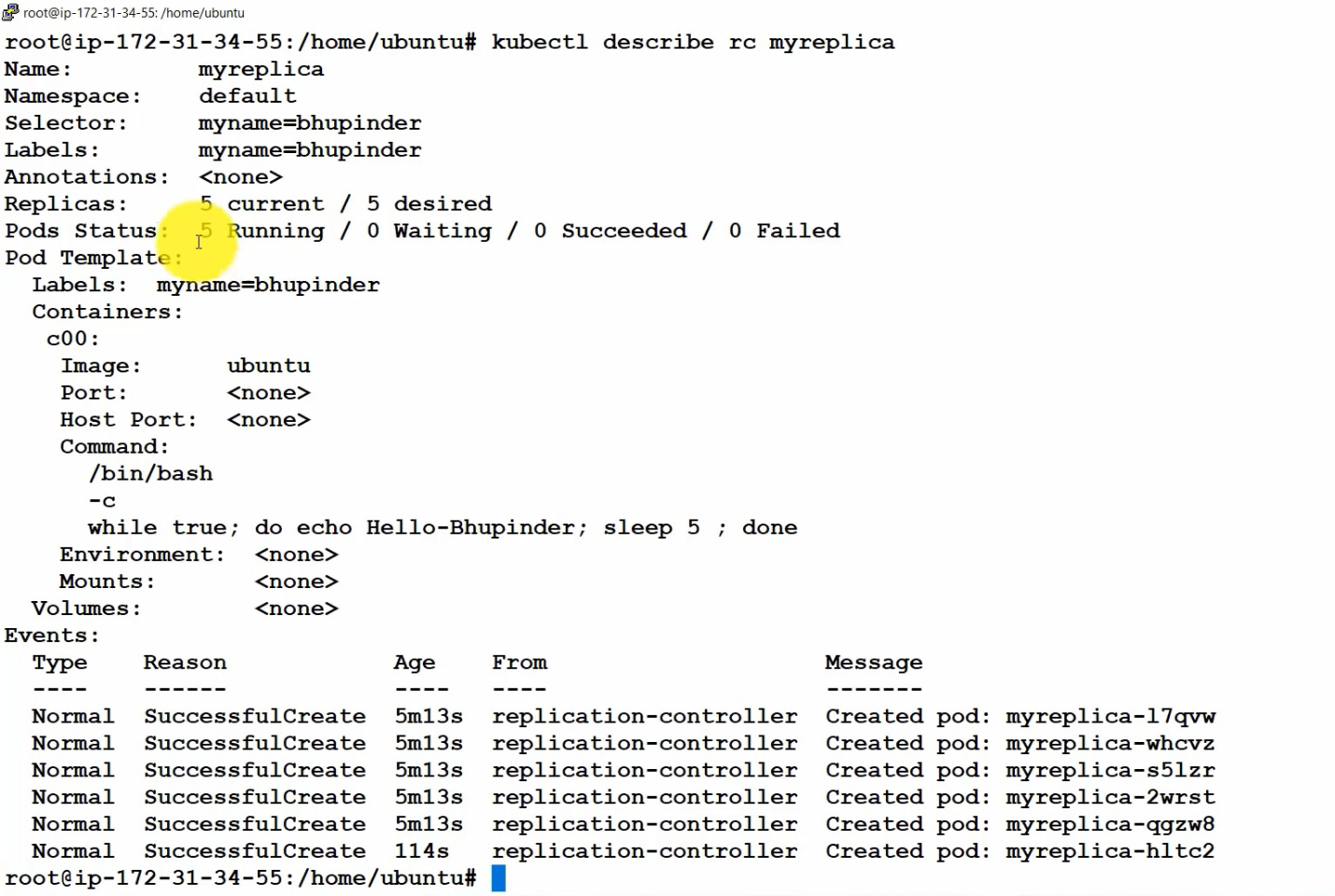
Text

Description automatically generated



Table

Description automatically generated



Table

Description automatically generated

Table

Description automatically generated

|  |  |
| --- | --- |
|  |  |

Graphical user interface, text, application

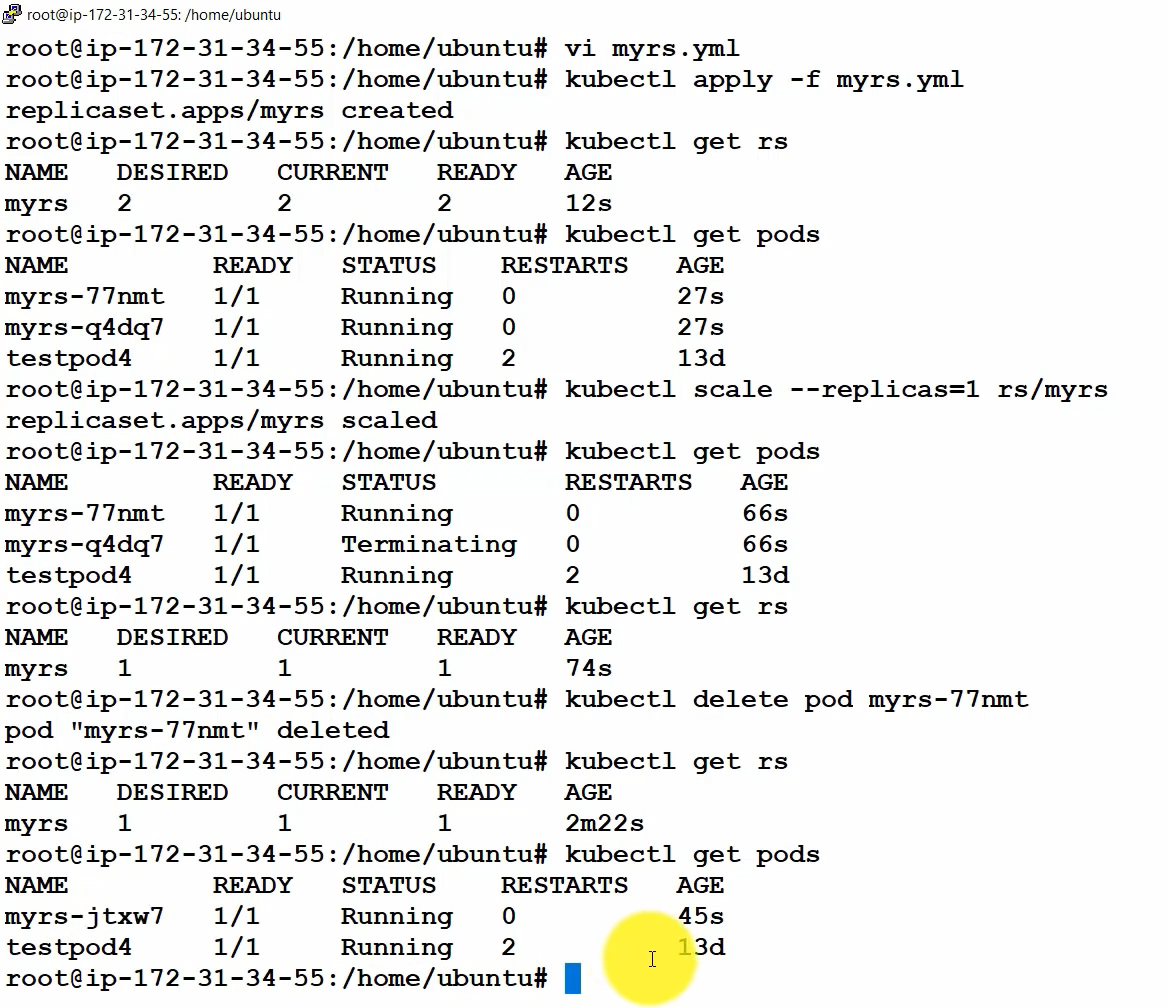
Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated with medium confidence



Text

Description automatically generated

Kubernetes Networking

==============================================

LEC-51 DEVOPS TECHNICAL GUFTGU NOTES

====================

Install Docker

$ sudo apt update && apt -y install docker.io

Install kubectl

$ curl -LO https://storage.googleapis.com/kubern... -s https://storage.googleapis.com/kubern... && chmod +x ./kubectl && sudo mv ./kubectl /usr/local/bin/kubectl

Install Minikube

$ curl -Lo minikube https://storage.googleapis.com/miniku... && chmod +x minikube && sudo mv minikube /usr/local/bin/

Start Minikube

$ apt install conntrack

$ minikube start --vm-driver=none

$ minikube status

=================================

KUBERNETES NETWORKING

=================================

kind: Pod

apiVersion: v1

metadata:

name: testpod

spec:

containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Hello-Bhupinder; sleep 5 ; done"]

- name: c01

image: httpd

ports:

- containerPort: 80

================

kind: Deployment

apiVersion: apps/v1

metadata:

name: mydeployments

spec:

replicas: 1

selector: # tells the controller which pods to watch/belong to

matchLabels:

name: deployment

template:

metadata:

name: testpod1

labels:

name: deployment

spec:

containers:

- name: c00

image: httpd

ports:

- containerPort: 80

====================

kind: Service # Defines to create Service type Object

apiVersion: v1

metadata:

name: demoservice

spec:

ports:

- port: 80 # Containers port exposed

targetPort: 80 # Pods port

selector:

name: deployment # Apply this service to any pods which has the specific label

type: ClusterIP # Specifies the service type i.e ClusterIP or NodePort

$ kubectl get svc

===========================

volume labs

===========================

apiVersion: v1

kind: Pod

metadata:

name: myvolemptydir

spec:

containers:

- name: c1

image: centos

command: ["/bin/bash", "-c", "sleep 15000"]

volumeMounts: # Mount definition inside the container

- name: xchange

mountPath: "/tmp/xchange"

- name: c2

image: centos

command: ["/bin/bash", "-c", "sleep 10000"]

volumeMounts:

- name: xchange

mountPath: "/tmp/data"

volumes:

- name: xchange

emptyDir: {}

========================

HOST PATH

========================

apiVersion: v1

kind: Pod

metadata:

name: myvolhostpath

spec:

containers:

- image: centos

name: testc

command: ["/bin/bash", "-c", "sleep 15000"]

volumeMounts:

- mountPath: /tmp/hostpath

name: testvolume

volumes:

- name: testvolume

hostPath:

path: /tmp/data

|  |  |
| --- | --- |
| A picture containing text, whiteboard  Description automatically generated |  |

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

|  |  |
| --- | --- |
|  |  |

Text

Description automatically generated

Text

Description automatically generated



Text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

