First Use case Deploy any application Using Helm on Kubernetes:

Helm:

helm is the package manager for Kubernetes (like yum, apt and home brew) that allows easily package, configure, and deploy applications onto Kubernetes clusters.
Introduced first time in 2015

helm helps you manage k8s application with helm charts which help you define,install and upgrade even the most complex kubernetes application.

Helmchart:

A Helm chart is a package that contains all the necessary Kubernetes resources and configurations required to deploy an application or service in a Kubernetes cluster. Helm, often referred to as the "package manager for Kubernetes," uses charts to simplify the deployment and management of applications within Kubernetes.

A Helm chart typically includes:

1. **Chart.yaml**:
   - This file contains metadata about the chart, such as its name, version, description, and other details.
2. **values.yaml**:
   - This file contains the default configuration values for the chart. Users can override these values by providing their own values file or specifying values directly on the command line.
3. **templates/**:
   - This directory contains the Kubernetes resource templates (YAML files) that Helm uses to generate the final manifest files. These templates use the Go templating language to allow dynamic configuration.
4. **charts/**:
   - This directory can contain dependencies, which are other charts that this chart depends on.
5. **README.md**:

- A Markdown file providing an overview of the chart, how to use it, and other pertinent information.

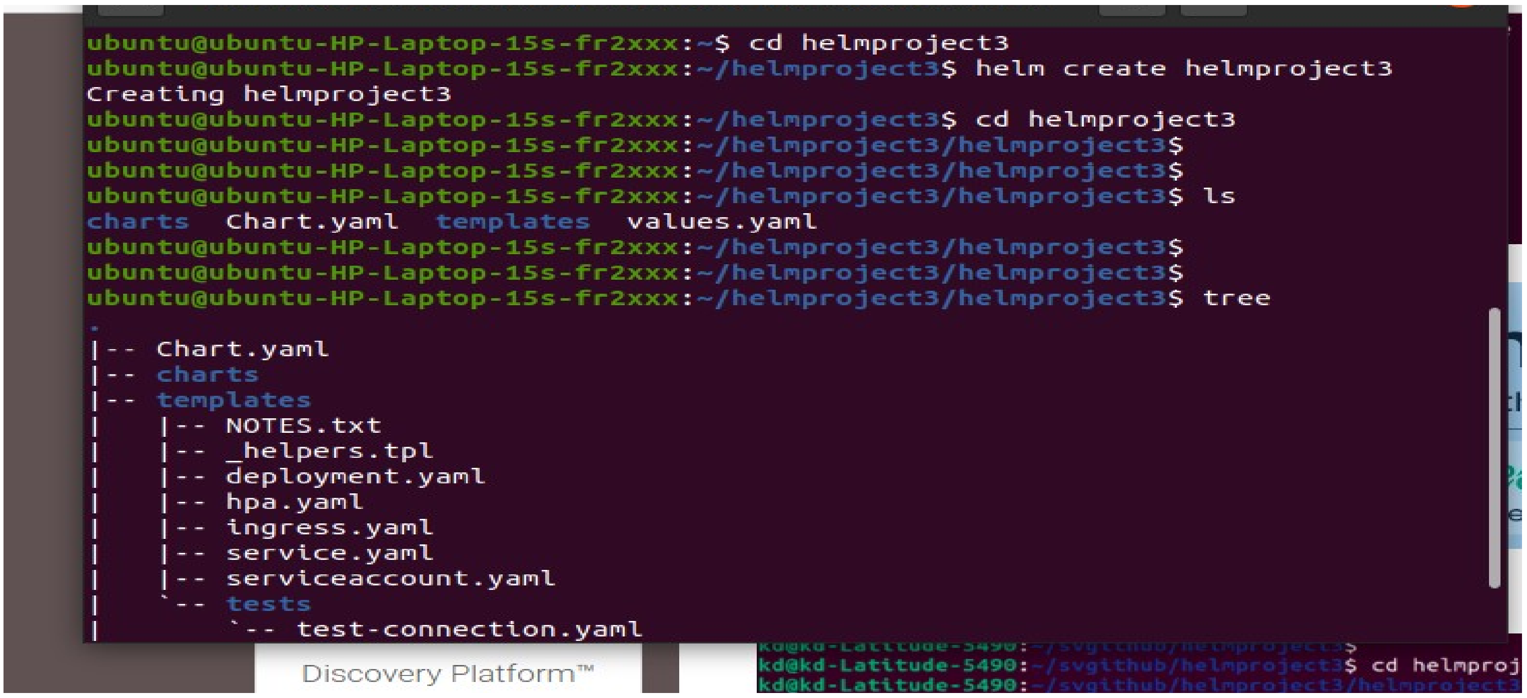
6. **_helpers.tpl**:
    - A file containing helper templates that can be used in other templates for reusability.

What We need to done before starting project:
A. Ubuntu Operating system
B. 8GB RAM

1. Install minikube on machine
2. Create project repository on GitHub
3. Install helm on machine
4. Clone Github Repository on machine

Most imp Kubectl Commands:
Minikube status
minikube start
minikube start –force
minikube start –driver=docker
minikube start --driver=none
Kubectl get pod
kubectl get svc
kubectl get ns
kubectl get all --all-namespaces
kubectl deployment
kubectl describe deployment releasename -n namespace
kubectl describe pod podname -n namespaceminikube service releasename –url
minikube ip
kubectl port-forward my-pod 8080:80
kubectl delete pod podname
kubectl apply -f service.yaml

Create new custome helm chart by using
$ helm create helmproject3



```
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~$ cd helmproject3
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject3$ helm create helmproject3
Creating helmproject3
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject3$ cd helmproject3
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject3/helmproject3$
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject3/helmproject3$
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject3/helmproject3$ ls
charts  Chart.yaml  templates  values.yaml
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject3/helmproject3$
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject3/helmproject3$
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject3/helmproject3$ tree
.
|-- Chart.yaml
|-- charts
|-- templates
|   |-- NOTES.txt
|   |-- _helpers.tpl
|   |-- deployment.yaml
|   |-- hpa.yaml
|   |-- ingress.yaml
|   |-- service.yaml
|   |-- serviceaccount.yaml
|   `-- tests
|       `-- test-connection.yaml
```

```
kd@kd-Latitude-5490:~/svgithub/helmproject3$ cd helmproj
kd@kd-Latitude-5490:~/svgithub/helmproject3/helmproject3
```

then tree command
tree .

Create Dockerfile for your application .. I am creating for tomcat application..

in Dockerfile we can write

FROM tomcat:9.0.20

Then Edite Chart.yaml , deployment.yaml, service.yaml, values.yaml file as per your project requirement.

Chart.yaml file: edit version 9.0.20

```
apiVersion: v2
name: helmproject3
description: A Helm chart for Kubernetes

# A chart can be either an 'application' or a 'library' chart.
#
# Application charts are a collection of templates that can be packaged into v>
# to be deployed.
#
# Library charts provide useful utilities or functions for the chart developer>
# a dependency of application charts to inject those utilities and functions i>
# pipeline. Library charts do not define any templates and therefore cannot be>
type: application
```

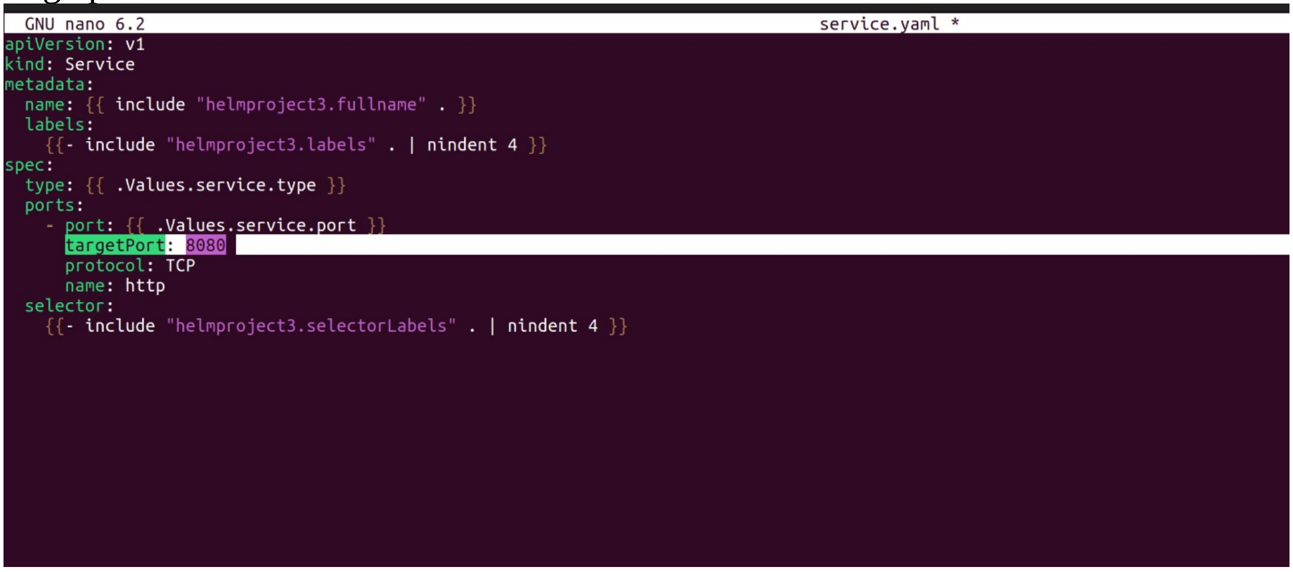
# This is the chart version. This version number should be incremented each ti>
# to the chart and its templates, including the app version.
# Versions are expected to follow Semantic Versioning (https://semver.org/)
version: 9.0.20

values.yaml:

image:
repository: pramila188/tomcat
pullPolicy: IfNotPresent
# Overrides the image tag whose default is the chart appVersion.
Tag: 9.0.20

Service.yaml:

service:
type: NodePort
port: 8029
targetport: 8080



```
  GNU nano 6.2                                              service.yaml *
apiVersion: v1
kind: Service
metadata:
  name: {{ include "helmproject3.fullname" . }}
  labels:
    {{- include "helmproject3.labels" . | nindent 4 }}
spec:
  type: {{ .Values.service.type }}
  ports:
    - port: {{ .Values.service.port }}
      targetPort: 8080
      protocol: TCP
      name: http
  selector:
    {{- include "helmproject3.selectorLabels" . | nindent 4 }}
```

You can build your Docker image using

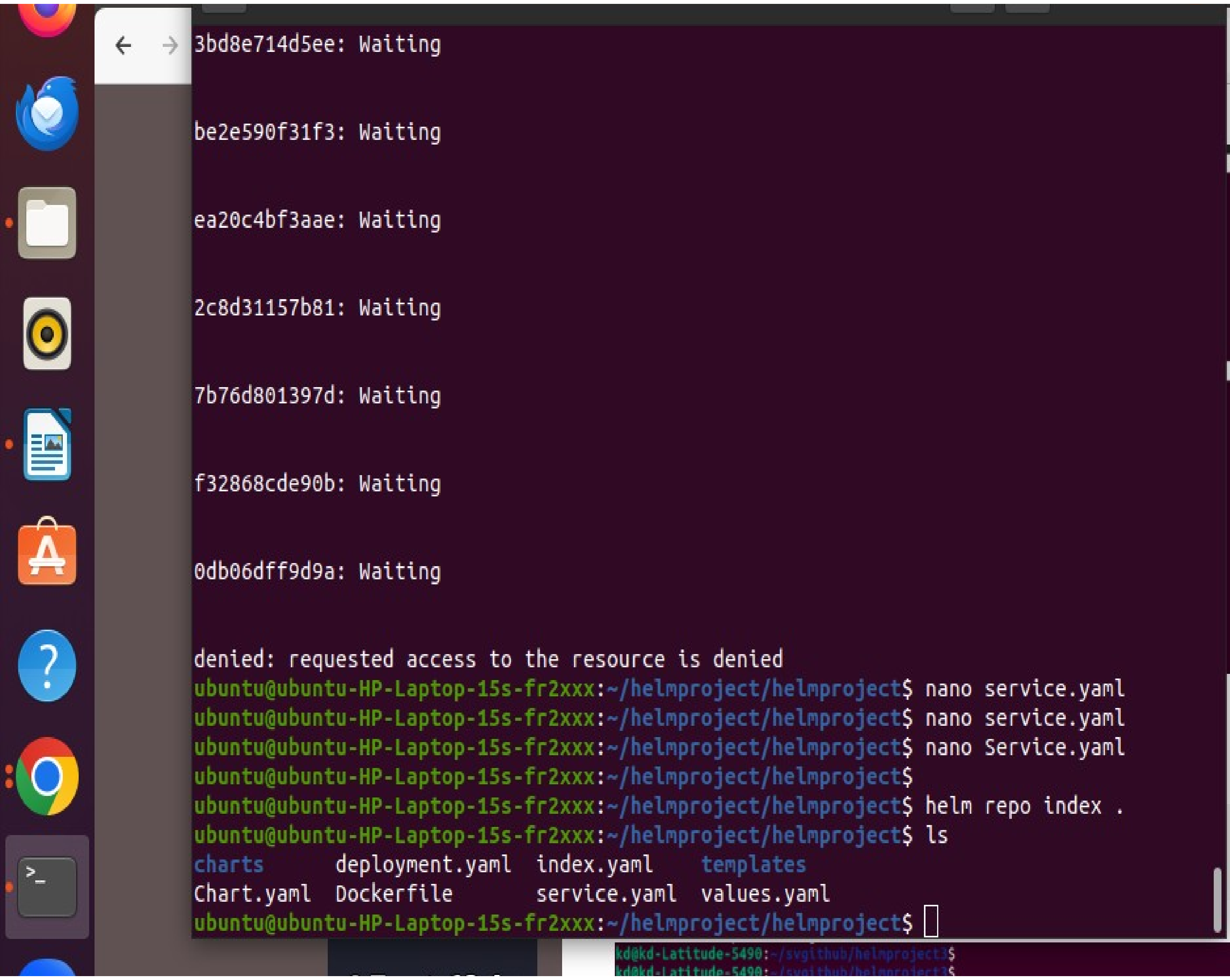
$ docker build -t pramila188/tomat:9.0.20 .

$ docker login

$ docker push pramila188/tomcat:9.0.20

navigate project dir and run

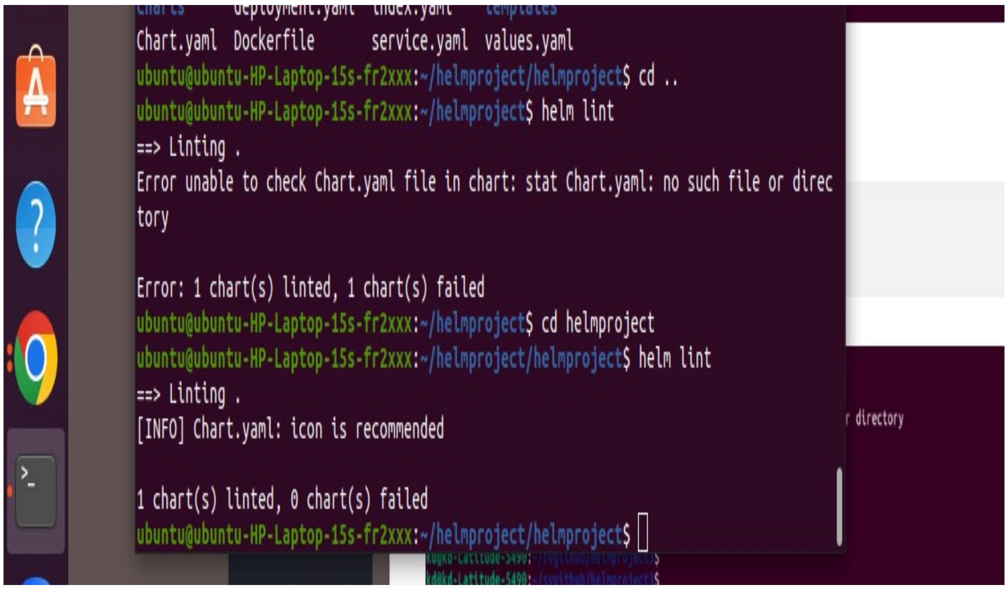
$ helm repo index .



```
3bd8e714d5ee: Waiting

be2e590f31f3: Waiting

ea20c4bf3aae: Waiting

2c8d31157b81: Waiting

7b76d801397d: Waiting

f32868cde90b: Waiting

0db06dff9d9a: Waiting

denied: requested access to the resource is denied
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$ nano service.yaml
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$ nano service.yaml
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$ nano Service.yaml
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$ helm repo index .
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$ ls
charts      deployment.yaml  index.yaml    templates
Chart.yaml  Dockerfile       service.yaml  values.yaml
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$
```

To check any errors run command

$ helm lint



```
Chart.yaml  Dockerfile      service.yaml  values.yaml
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$ cd ..
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject$ helm lint
==> Linting .
Error unable to check Chart.yaml file in chart: stat Chart.yaml: no such file or direc
tory

Error: 1 chart(s) linted, 1 chart(s) failed
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject$ cd helmproject
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$ helm lint
==> Linting .
[INFO] Chart.yaml: icon is recommended

1 chart(s) linted, 0 chart(s) failed
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject/helmproject$
```

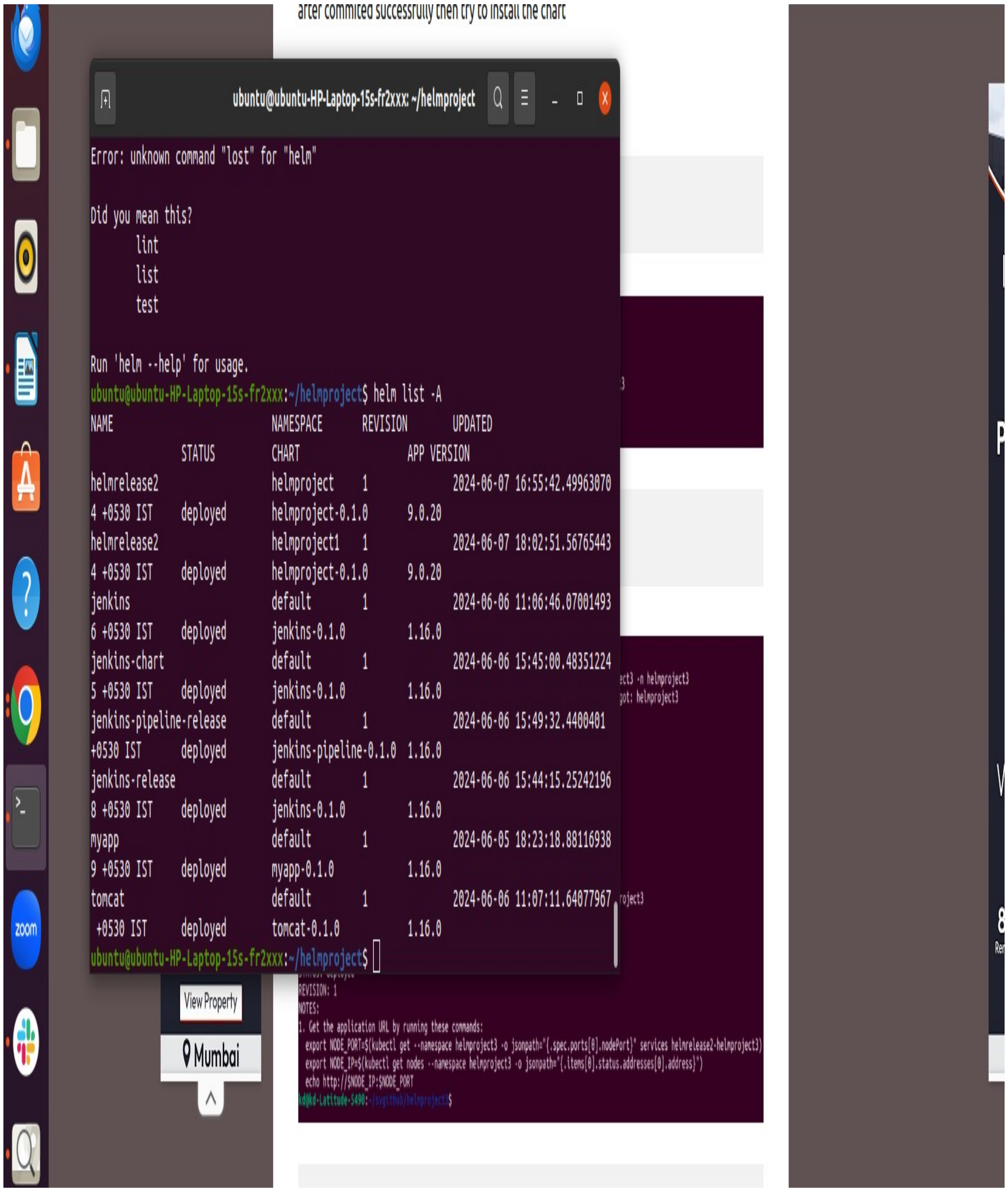
after commited successfully then try to install the chart

first create your own namespace

$ kubectl create ns helmproject3

$ helm install helmrelease2 helmproject3 -n helmproject3

$ helm list
$ helm list -A



after commited successfully then try to install the chart

ubuntu@ubuntu-HP-Laptop-15s-fr2xxx: ~/helmproject

```
Error: unknown command "lost" for "helm"

Did you mean this?
        lint
        list
        test

Run 'helm --help' for usage.
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject$ helm list -A
NAME                    NAMESPACE       REVISION        UPDATED
                STATUS          CHART                   APP VERSION
helmrelease2            helmproject     1               2024-06-07 16:55:42.49963070
4 +0530 IST     deployed        helmproject-0.1.0       9.0.20
helmrelease2            helmproject1    1               2024-06-07 18:02:51.56765443
4 +0530 IST     deployed        helmproject-0.1.0       9.0.20
jenkins                 default         1               2024-06-06 11:06:46.07001493
6 +0530 IST     deployed        jenkins-0.1.0           1.16.0
jenkins-chart           default         1               2024-06-06 15:45:00.48351224
5 +0530 IST     deployed        jenkins-0.1.0           1.16.0
jenkins-pipeline-release        default 1               2024-06-06 15:49:32.4400401
+0530 IST       deployed        jenkins-pipeline-0.1.0  1.16.0
jenkins-release         default         1               2024-06-06 15:44:15.25242196
8 +0530 IST     deployed        jenkins-0.1.0           1.16.0
myapp                   default         1               2024-06-05 18:23:18.88116938
9 +0530 IST     deployed        myapp-0.1.0             1.16.0
tomcat                  default         1               2024-06-06 11:07:11.64077967
 +0530 IST      deployed        tomcat-0.1.0            1.16.0
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject$
```

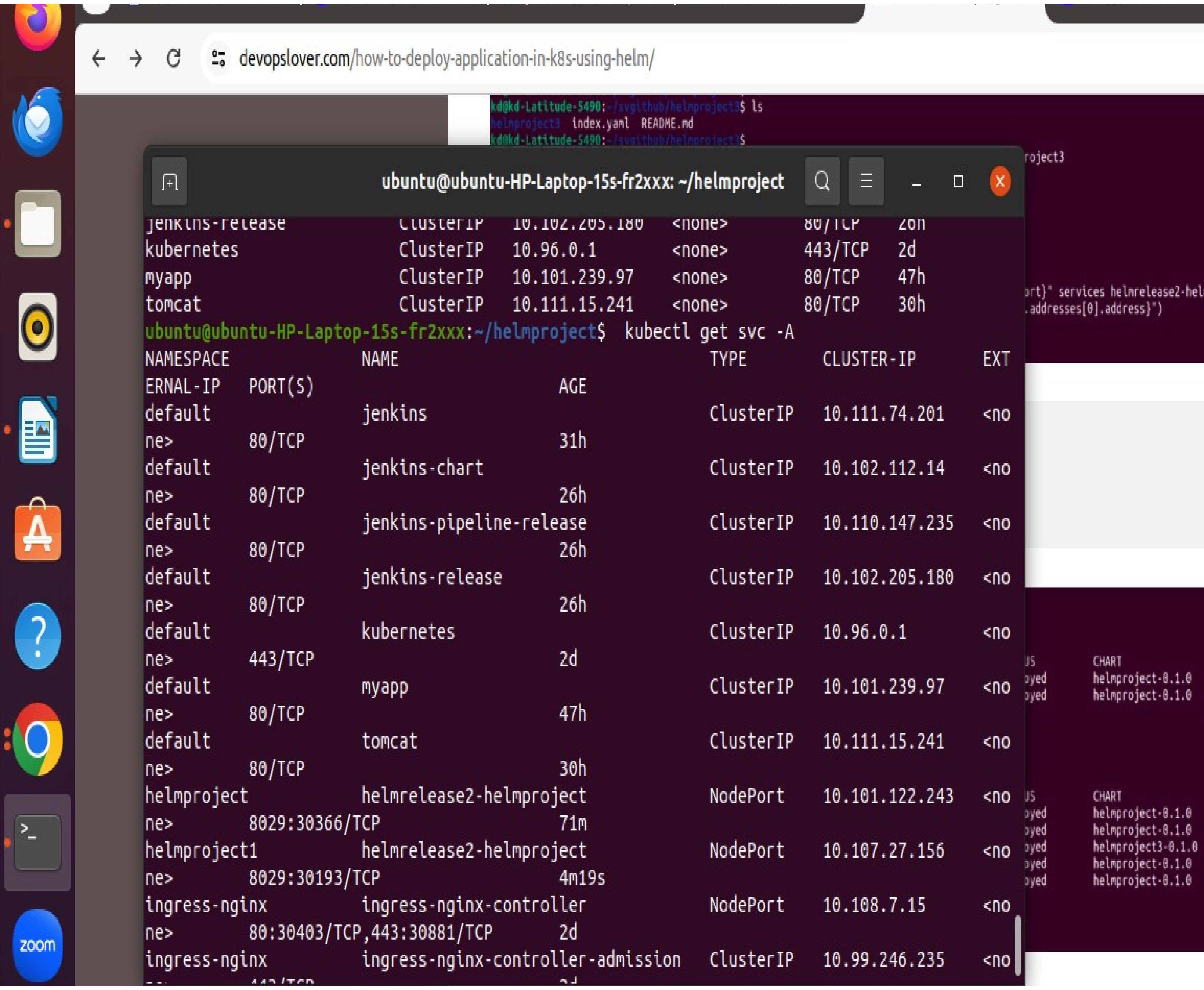
```
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
  export NODE_PORT=$(kubectl get --namespace helmproject3 -o jsonpath="{.spec.ports[0].nodePort}" services helmrelease2-helmproject3)
  export NODE_IP=$(kubectl get nodes --namespace helmproject3 -o jsonpath="{.items[0].status.addresses[0].address}")
  echo http://$NODE_IP:$NODE_PORT
kd@kd-Latitude-5490:~/svgithub/helmproject3$
```

$ kubectl get pod

$ kubectl get pod -A

$ kubectl get svc

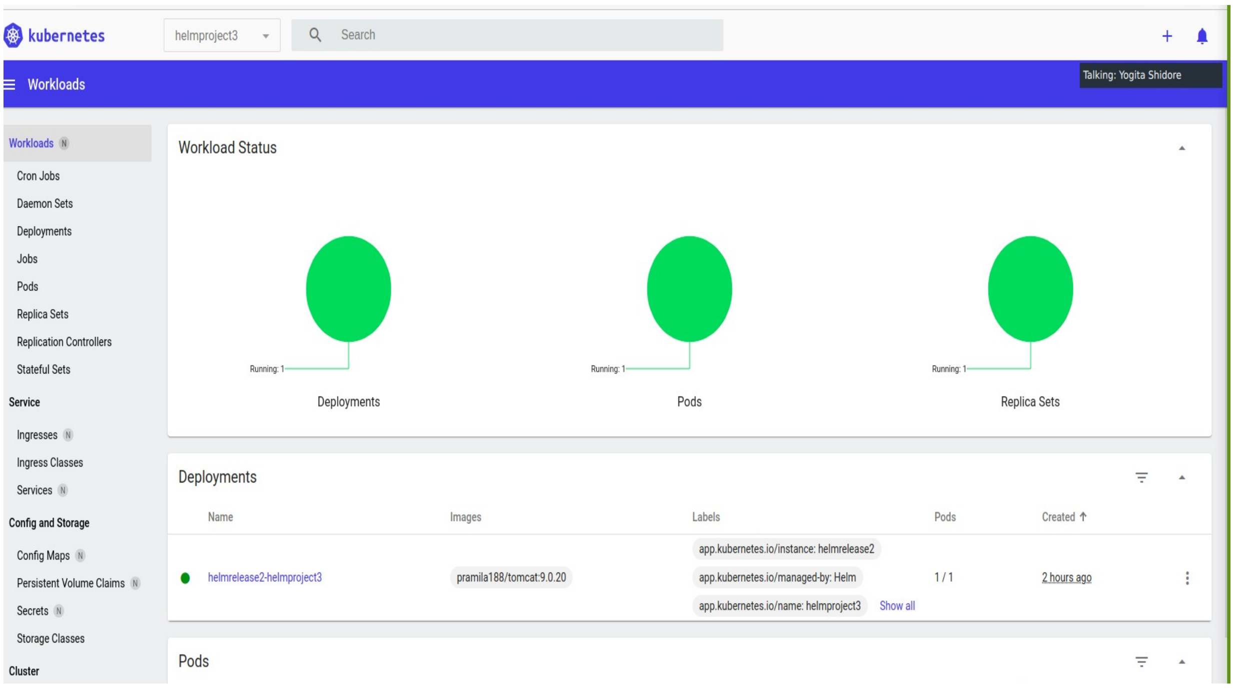
$ kubectl get svc-A



```
kd@kd-Latitude-5490:~/svgithub/helmproject3$ ls
helmproject3  index.yaml  README.md
kd@kd-Latitude-5490:~/svgithub/helmproject3$
```

```
jenkins-release            ClusterIP   10.102.205.180   <none>       80/TCP    26h
kubernetes                 ClusterIP   10.96.0.1        <none>       443/TCP   2d
myapp                      ClusterIP   10.101.239.97    <none>       80/TCP    47h
tomcat                     ClusterIP   10.111.15.241    <none>       80/TCP    30h
ubuntu@ubuntu-HP-Laptop-15s-fr2xxx:~/helmproject$  kubectl get svc -A
NAMESPACE            NAME                                  TYPE        CLUSTER-IP       EXT
ERNAL-IP   PORT(S)                      AGE
default              jenkins                               ClusterIP   10.111.74.201    <no
ne>        80/TCP                       31h
default              jenkins-chart                         ClusterIP   10.102.112.14    <no
ne>        80/TCP                       26h
default              jenkins-pipeline-release              ClusterIP   10.110.147.235   <no
ne>        80/TCP                       26h
default              jenkins-release                       ClusterIP   10.102.205.180   <no
ne>        80/TCP                       26h
default              kubernetes                            ClusterIP   10.96.0.1        <no
ne>        443/TCP                      2d
default              myapp                                 ClusterIP   10.101.239.97    <no
ne>        80/TCP                       47h
default              tomcat                                ClusterIP   10.111.15.241    <no
ne>        80/TCP                       30h
helmproject          helmrelease2-helmproject              NodePort    10.101.122.243   <no
ne>        8029:30366/TCP               71m
helmproject1         helmrelease2-helmproject              NodePort    10.107.27.156    <no
ne>        8029:30193/TCP               4m19s
ingress-nginx        ingress-nginx-controller              NodePort    10.108.7.15      <no
ne>        80:30403/TCP,443:30881/TCP   2d
ingress-nginx        ingress-nginx-controller-admission    ClusterIP   10.99.246.235    <no
ne>        443/TCP                      2d
```

$ minikube dashboard



Workload Status

Deployments	Pods	Replica Sets
Running: 1	Running: 1	Running: 1

Deployments

	Name	Images	Labels	Pods	Created ↑
●	helmrelease2-helmproject3	pramila188/tomcat:9.0.20	app.kubernetes.io/instance: helmrelease2 app.kubernetes.io/managed-by: Helm app.kubernetes.io/name: helmproject3 Show all	1 / 1	2 hours ago

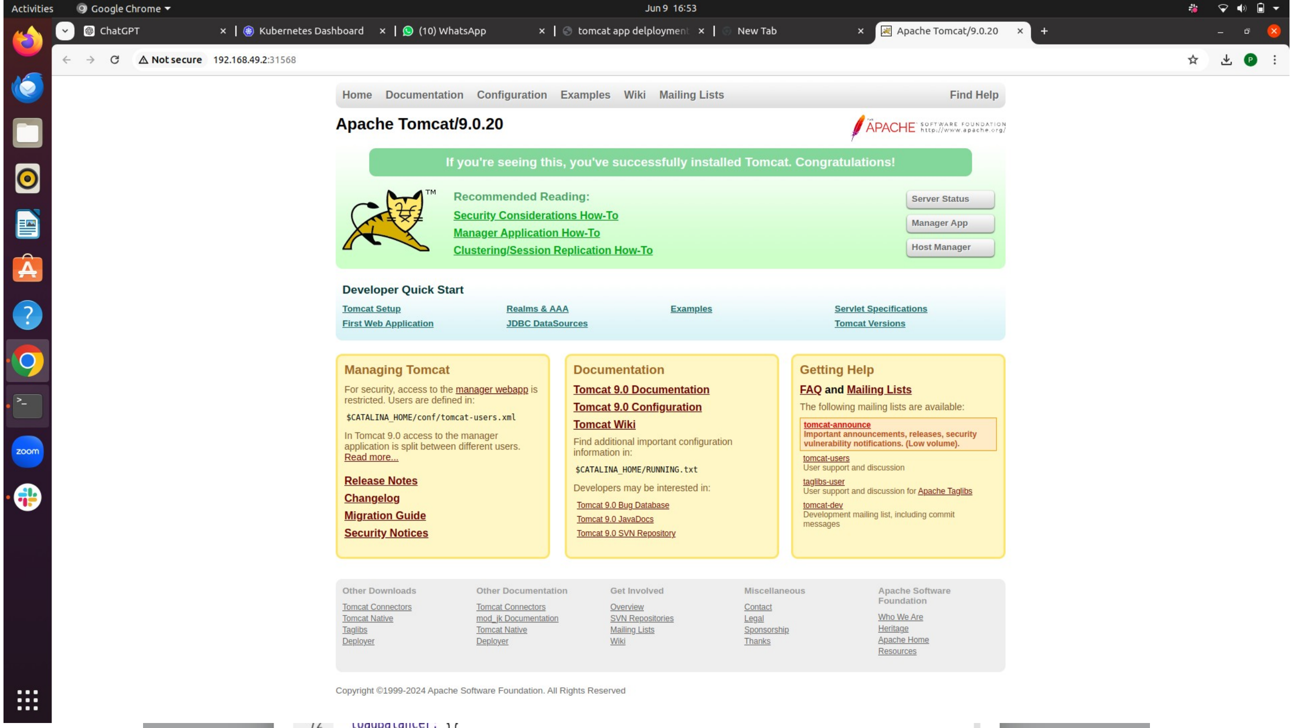
$ minikube service helmrelease2-helmproject3

$ minikube service helmrelease2-helmproject3 -n helmproject3 –url

automatically open Browser and show the output:

OR open browser and minikube ip:NodePort

NodePort is here:



Apache Tomcat/9.0.20

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
```
$CATALINA_HOME/conf/tomcat-users.xml
```
In Tomcat 9.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation
Tomcat 9.0 Documentation
Tomcat 9.0 Configuration
Tomcat Wiki
Find additional important configuration information in:
```
$CATALINA_HOME/RUNNING.txt
```
Developers may be interested in:
Tomcat 9.0 Bug Database
Tomcat 9.0 JavaDocs
Tomcat 9.0 SVN Repository

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs
tomcat-dev
Development mailing list, including commit messages

Other Downloads
Tomcat Connectors
Tomcat Native
Taglibs
Deployer

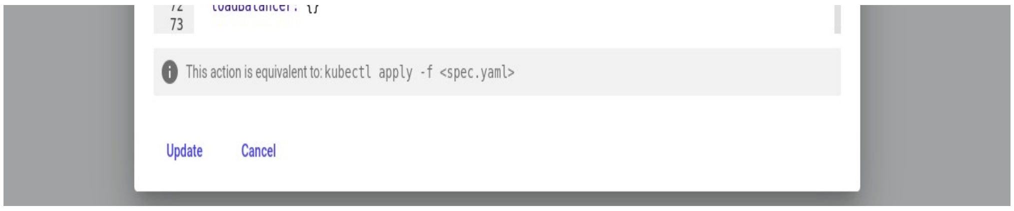
Other Documentation
Tomcat Connectors
mod_jk Documentation
Tomcat Native
Deployer

Get Involved
Overview
SVN Repositories
Mailing Lists
Wiki

Miscellaneous
Contact
Legal
Sponsorship
Thanks

Apache Software Foundation
Who We Are
Heritage
Apache Home
Resources

Copyright ©1999-2024 Apache Software Foundation. All Rights Reserved


```
  loadBalancer: {}
```
This action is equivalent to: kubectl apply -f <spec.yaml>

Output:

a

----------------------------------*******************-----------------------------