B.M.S. COLLEGE OF ENGINEERING

Autonomous Institute, Affiliated to VTU

Estd. 1946

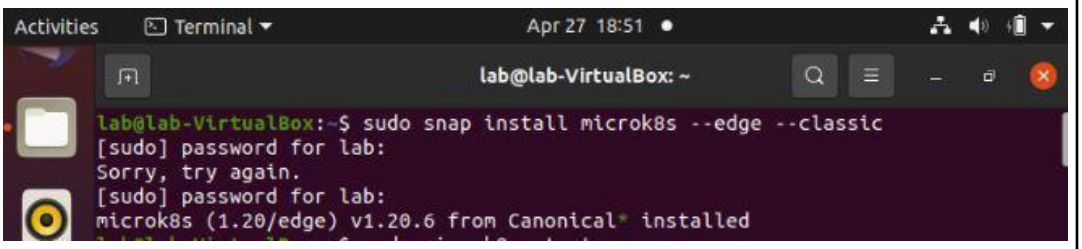
DEPARTMENT OF CSE

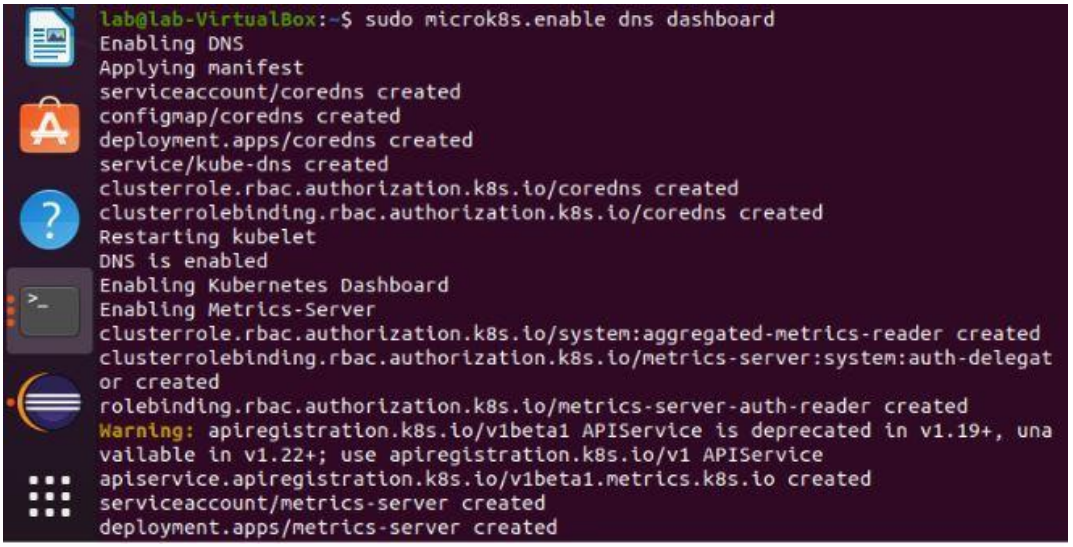
CTY Project Work In collaboration with HPE

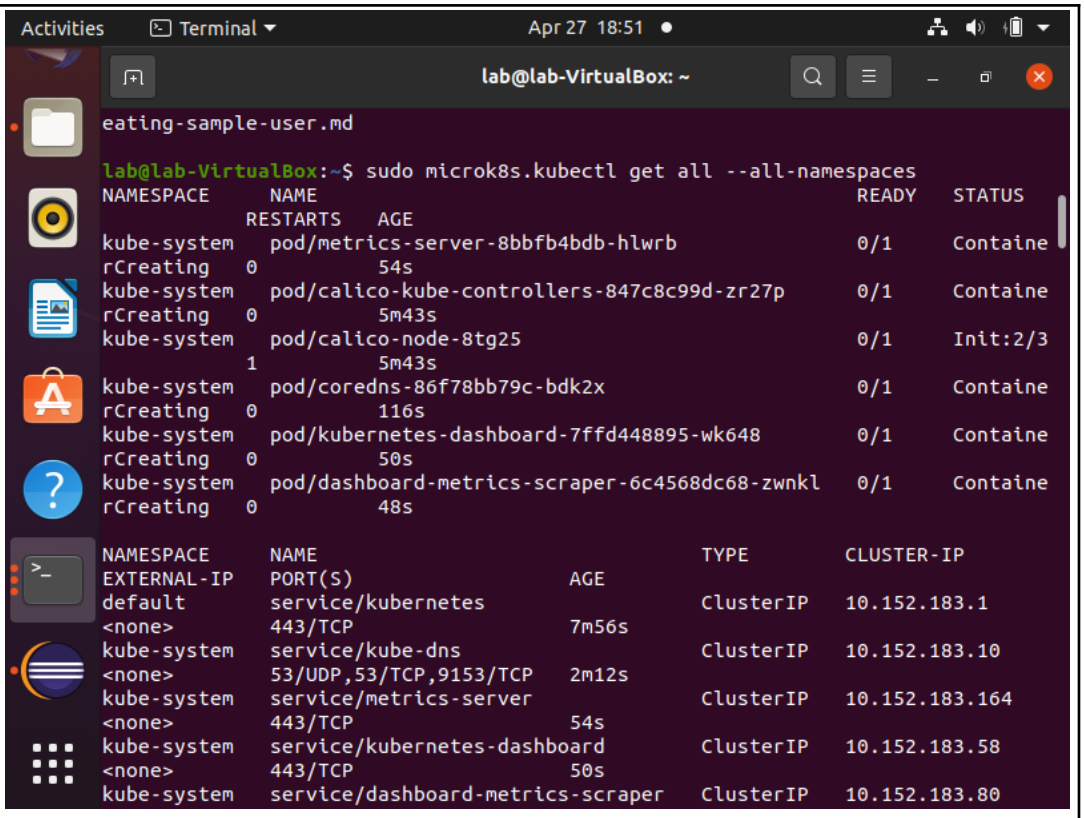
Project Title	Open source monitoring and observability stack on Kubernetes		
Student Team	Name: Pooja Srinivasan USN:1BM18CS069 SEM:UG-6th SEM	Name: Anusree Manoj K USN:1BM18CS017 SEM:UG-6th SEM	
	Name: Shikha N USN:1BM18CS149 SEM:UG-6th SEM	Name: Niha USN:1BM18CS060 SEM:UG-6th SEM	
Faculty Mentor	Dr. Nandhini V Associate Professor	**HPE Mentors**	Divakar Padiyar Sonu Sudhakaran
Review for the Period	18-04-2021	27-04-2021	
Task Given	Setting up and deploying a kubernetes cluster		
Difficulties Faced	None		
Libraries Used	None		
Github Link for the code:	None		

Code: Deployment of microk8s and minikube

What is a Microk8 [1,2]	MicroK8s is a powerful, lightweight, reliable production-ready Kubernetes distribution. It is an enterprise-grade Kubernetes distribution that has a small disk and memory footprint while offering carefully selected add-ons out-the-box, such as Istio, Knative, Grafana, Cilium and more. Microk8s is a non-elastic, rails-based single-node Kubernetes tool that is focused primarily on offline development, prototyping, and testing.

Features of Microk8 [1,2]	Microk8s includes the following features: ● Istio ● GPGPU bindings ● Daily builds ● Local storage ● Local registry ● Updates ● Dashboard ● Metrics ● Upgrades ● IngressDNS ● Conforman ● Small: Developers want the smallest K8s for laptop and workstation development. MicroK8s provides a standalone K8s compatible with Azure AKS, Amazon EKS, Google GKE when you run it on Ubuntu. ● Simple: Minimize administration and operations with a single-package install that has no moving parts for simplicity and certainty. All dependencies and batteries included. ● Secure: Updates are available for all security issues and can be applied immediately or scheduled to suit your maintenance cycle. ● Current: MicroK8s tracks upstream and releases beta, RC and final bits the same day as upstream K8s. You can track the latest K8s or stick to any release version from 1.10 onwards. ● Comprehensive: MicroK8s includes a curated collection of manifests for common K8s capabilities and services: ○ Service Mesh: Istio, Linkerd ○ Serverless: Knative ○ Monitoring: Fluentd, Prometheus, Grafana, Metrics ○ Ingress, DNS, Dashboard, Clustering ○ Automatic updates to the latest Kubernetes version ○ GPGPU bindings for AI/ML ○ Cilium, Helm and Kubeflow!
Installation of microk8 [1]	Step 1) To install microk8s sudo snap install microk8s --edge --classic 

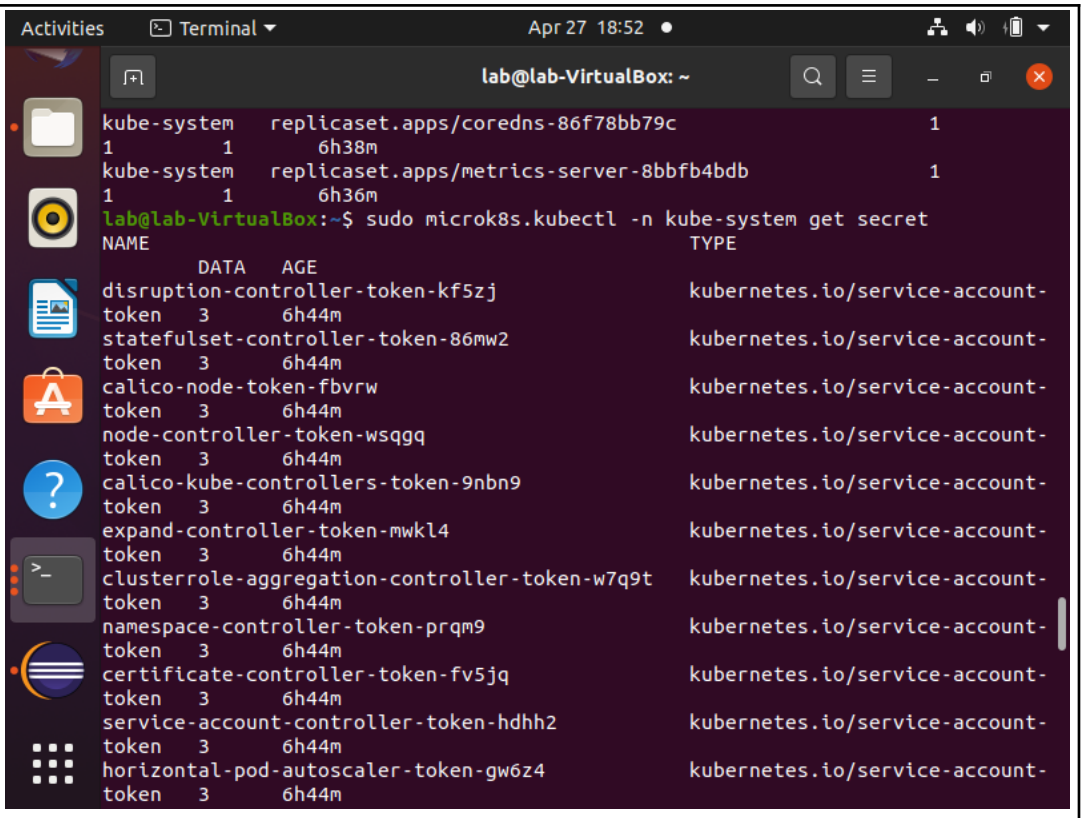
	Step 2) To start the microk8: sudo microk8s.start  The command should report that the service has started and pod scheduling has been enabled.
Microk8 dashboard [1]	The Dashboard is a web-based Dashboard that allows you to interact and manage Kubernetes Step 3) To install the dashboard we use the following command sudo microk8s.enable dns dashboard  Here we have enabled dns service but we can enable other services like **storage** (allows you to make use of storage on the host),**ingress**(create an Ingress controller),**gpu**(enable the nvidia-docker runtime and nvidia-device-plugin-daemonset),**istio**(enable the core Istio services),**registry** (deploy a private Docker registry) Step 4) We access the dashboard we need the address for that we us the following command: sudo microk8s.kubectl get all --all-namespaces



Accessing the dashboard

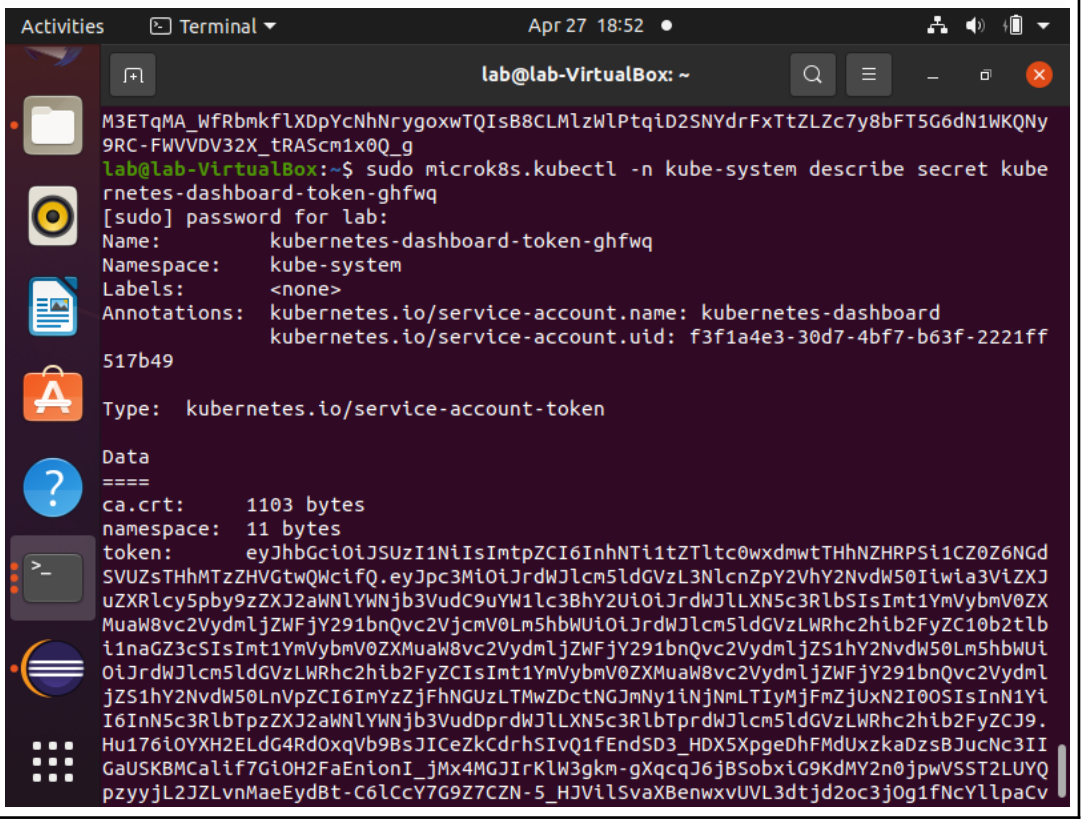
Step 5) We use the address obtained to access the dashboard.We will require a token to access the dashboard,for the token we use the following command:

sudo microk8s.kubectl -n kube-system get secret

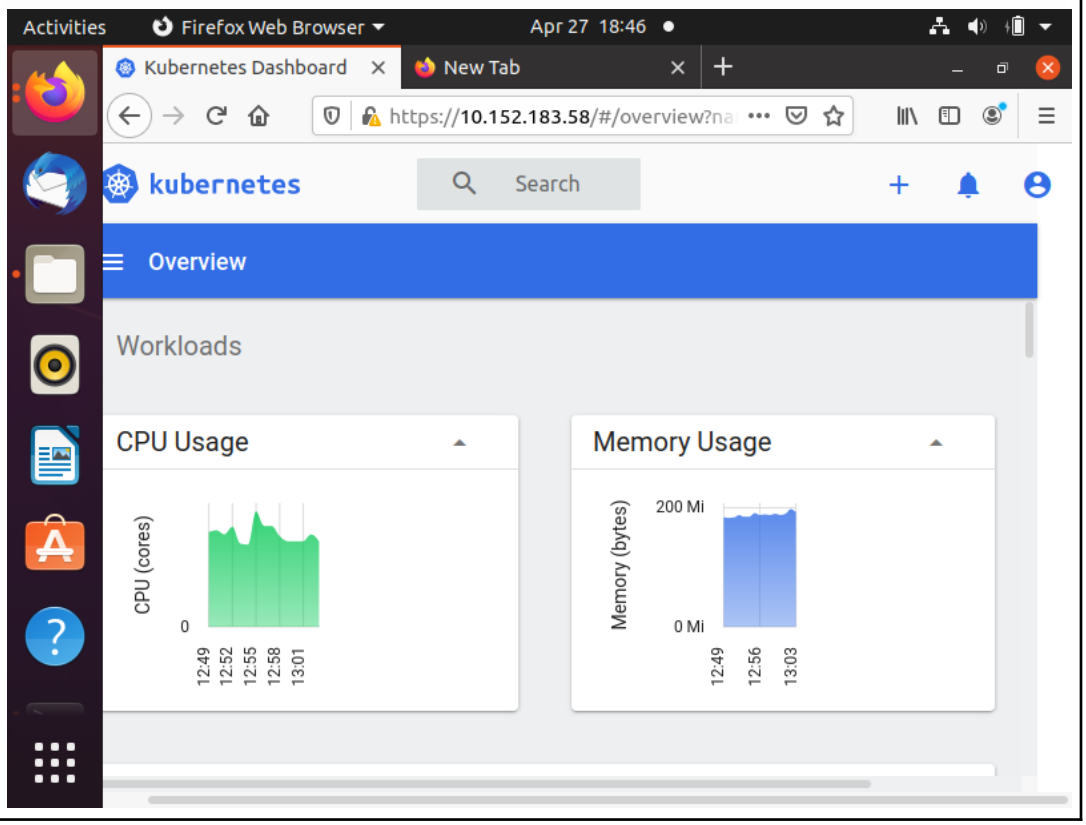


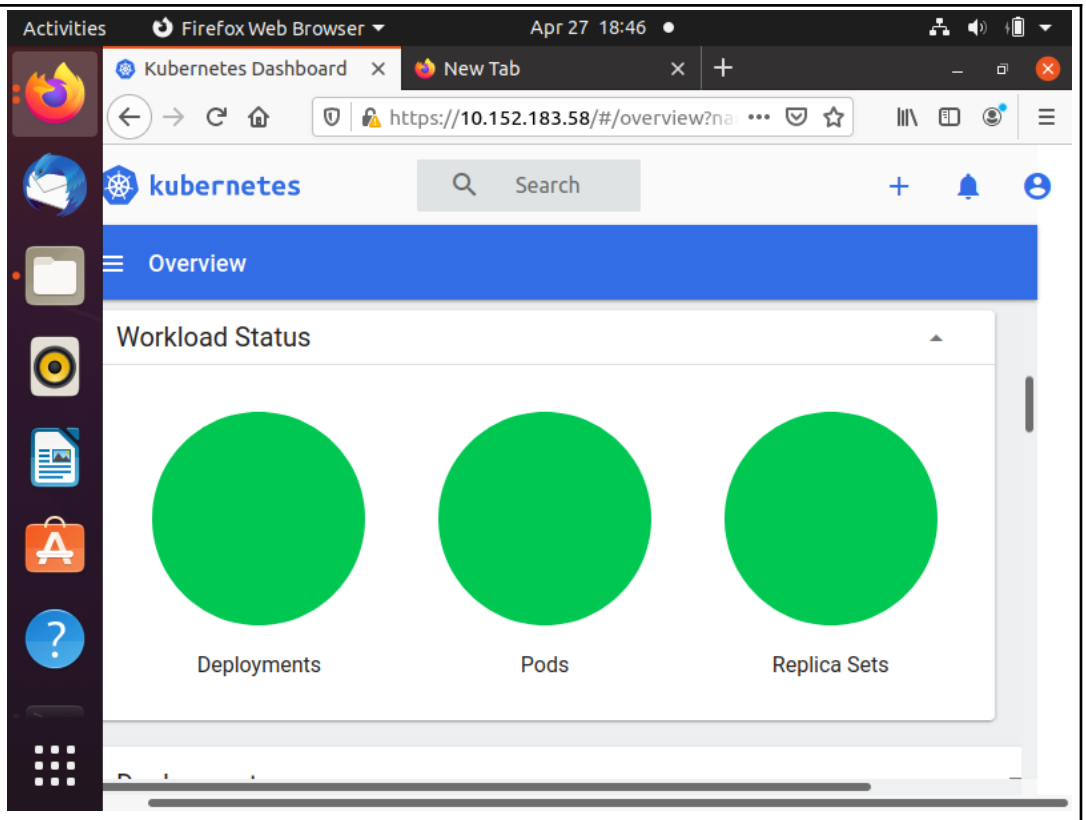
To retrieve the secret token for the service we perform the following command :

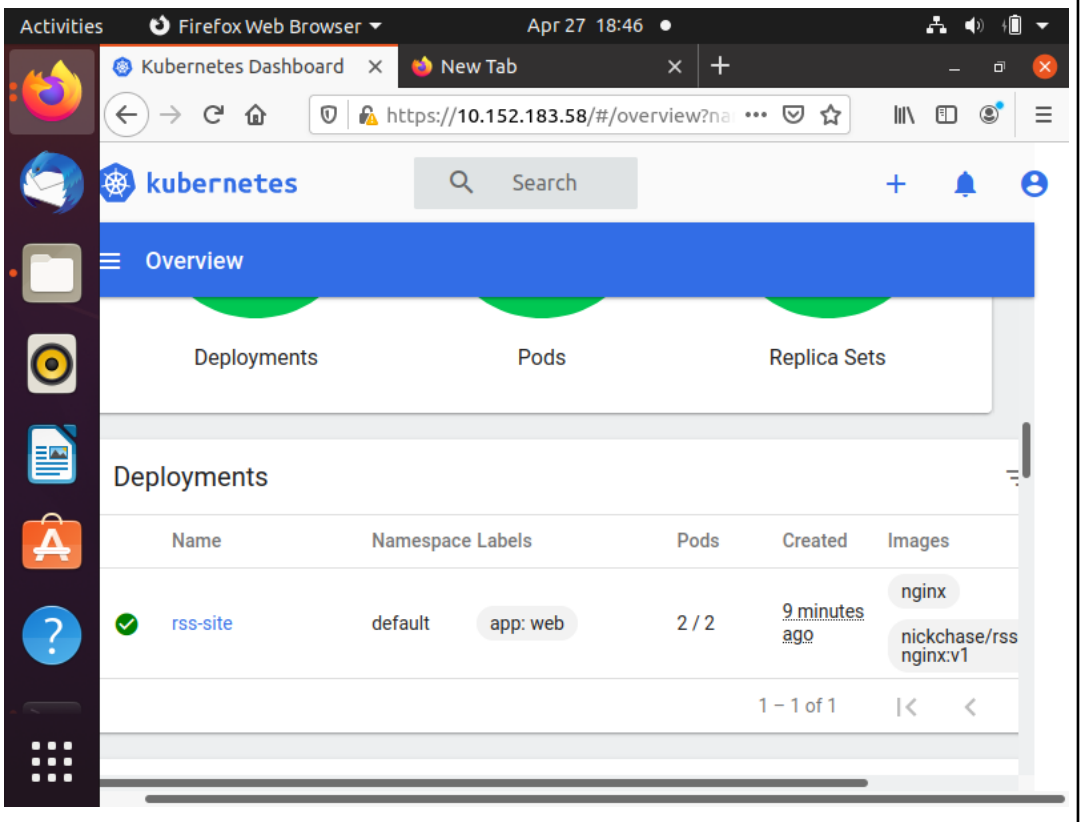
sudo microk8s.kubectl -n kube-system describe secret kubernetes-dashboard-token-ghfwq

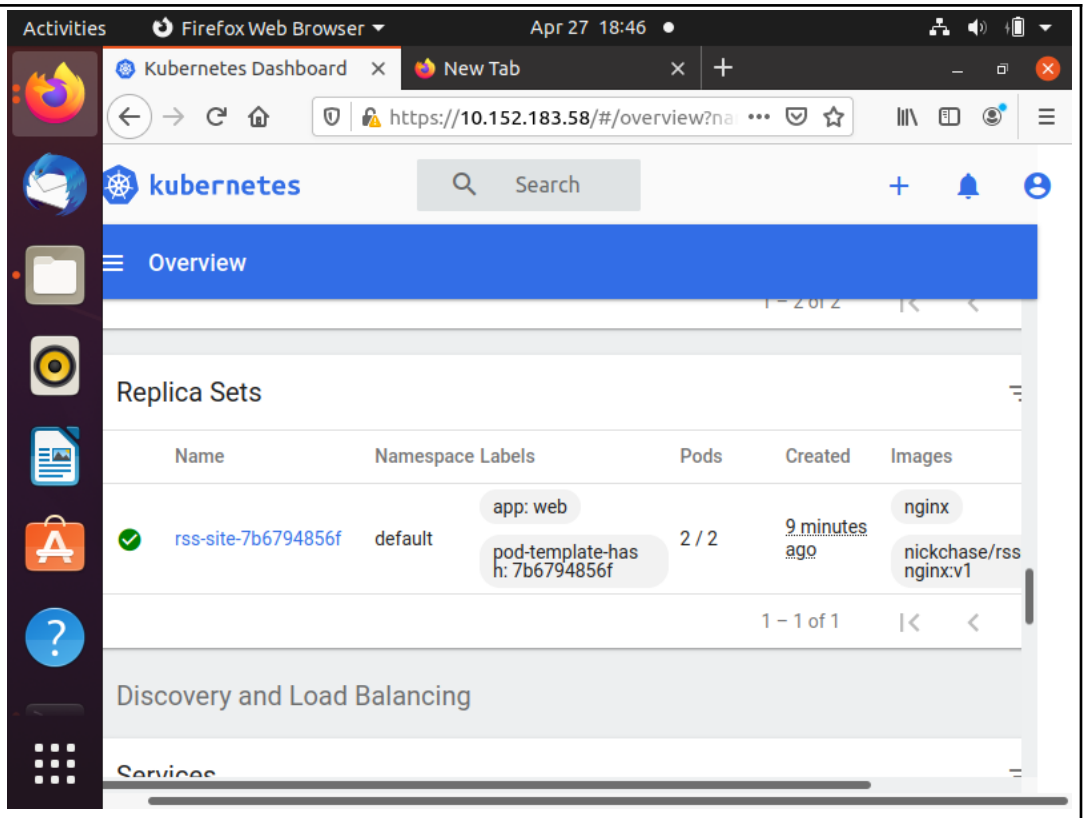


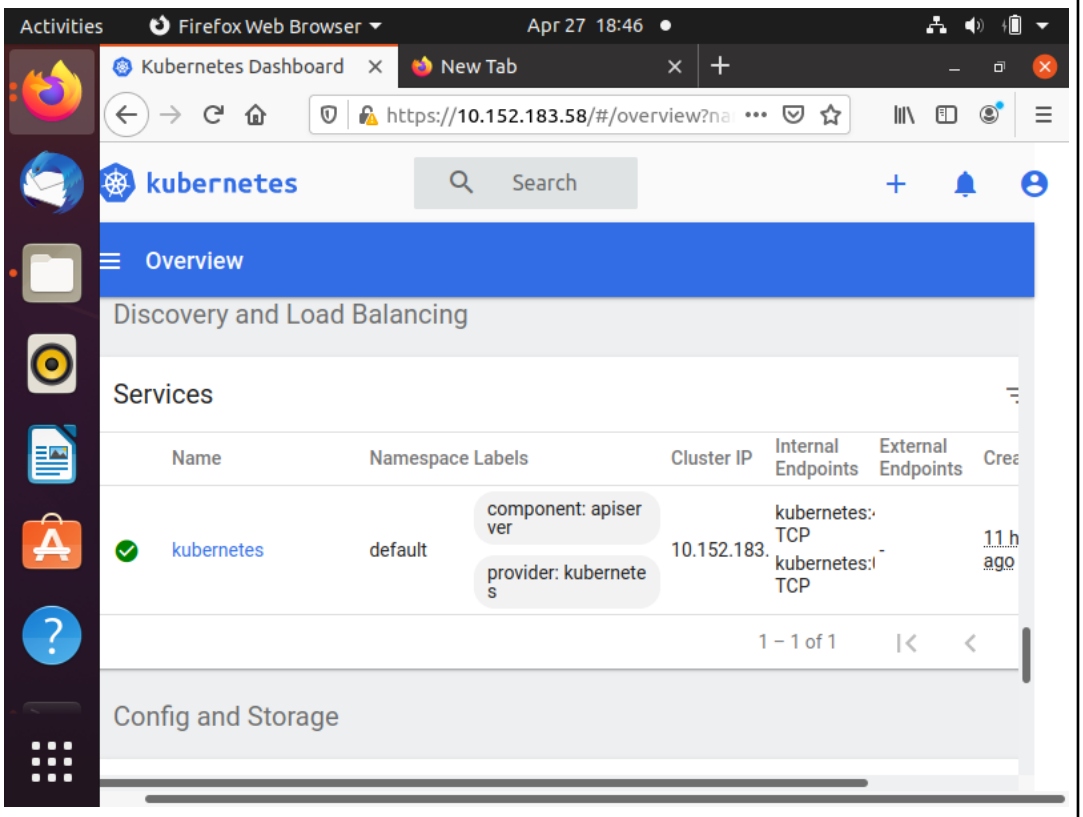
Pod creation [3]	Pod.yaml
	
	`---`
	`apiVersion: v1`
	`kind: Pod`
	`metadata:`
	`  name: rss-site`
	`  labels:`
	`  app: web`
	`spec:`
	`  containers:`
	`  - name: front-end`
	`  image: nginx`
	`  ports:`
	`  - containerPort: 80`
	`  - name: rss-reader`
	`  image: nickchase/rss-php-nginx:v1`
	`  ports:`
	`  - containerPort: 88`

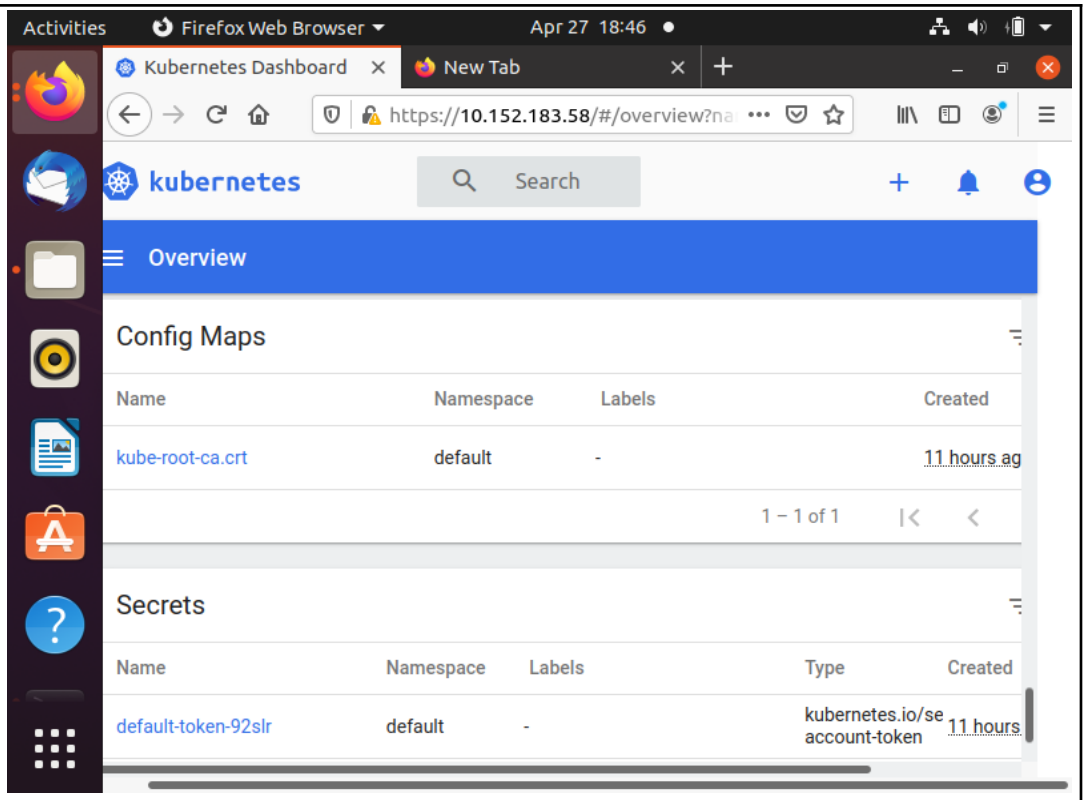
Screenshots of pod creation and deployments	


Activities
Firefox Web Browser
Apr 27 18:46
Kubernetes Dashboard
New Tab
https://10.152.183.58/#/overview?na
kubernetes
Search
Overview
Workload Status
Deployments
Pods
Replica Sets


Activities
Firefox Web Browser
Apr 27 18:46
Kubernetes Dashboard
New Tab
https://10.152.183.58/#/overview?na
kubernetes
Search
Overview
Deployments
Pods
Replica Sets
Deployments
Name
Namespace
Labels
Pods
Created
Images
rss-site
default
app: web
2 / 2
9 minutes ago
nginx
nickchase/rss nginx:v1
1 – 1 of 1


Activities
Firefox Web Browser
Apr 27 18:46
Kubernetes Dashboard
New Tab
https://10.152.183.58/#/overview?na
kubernetes
Search
Overview
Replica Sets
Name
Namespace Labels
Pods
Created
Images
rss-site-7b6794856f
default
app: web
pod-template-hash: 7b6794856f
2 / 2
9 minutes ago
nginx
nickchase/rss-nginx:v1
1 – 1 of 1
Discovery and Load Balancing


Activities
Firefox Web Browser
Apr 27 18:46
Kubernetes Dashboard
New Tab
https://10.152.183.58/#/overview?na
kubernetes
Search
Overview
Discovery and Load Balancing
Services
Name
Namespace Labels
Cluster IP
Internal Endpoints
External Endpoints
kubernetes
default
component: apiserver
provider: kubernetes
10.152.183.
kubernetes: TCP
kubernetes: TCP
11 h ago
1 – 1 of 1
Config and Storage

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>What is Minikube [4,5]</td><td>Minikube is a tool that lets you run Kubernetes locally. minikube runs a single-node Kubernetes cluster on your personal computer (including Windows, macOS and Linux PCs) so that you can try out Kubernetes, or for daily development work.</td></tr>
<tr><td>Installation of Minikube [5]</td><td>Note:-The installation is done on windows

Step 1) Intially the package Chocolatery was installed for installing kubectl.The command is as follows:

- @"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALUSERSPROFILE%\chocolatey\bin"

Step 2) Then to install kubectl we use the following command.

- choco install kubernetes-cli</td></tr>
</table>

```
G:\>choco
Chocolatey v0.10.15
Please run 'choco -?' or 'choco <command> -?' for help menu.

G:\>choco install kubernetes-cli
Chocolatey v0.10.15
Installing the following packages:
kubernetes-cli
By installing you accept licenses for the packages.
Progress: Downloading kubernetes-cli 1.21.0... 100%

kubernetes-cli v1.21.0 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): Y

Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
 ShimGen has successfully created a shim for kubectl-convert.exe
 ShimGen has successfully created a shim for kubectl.exe
 The install of kubernetes-cli was successful.
  Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-cli\tools'

Chocolatey installed 1/1 packages.
 See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
```

The binary for Minikube was downloaded, and installed.

After installation was complete the following command was executed to start Minikube.

- minikube start --vm-driver=virtualbox

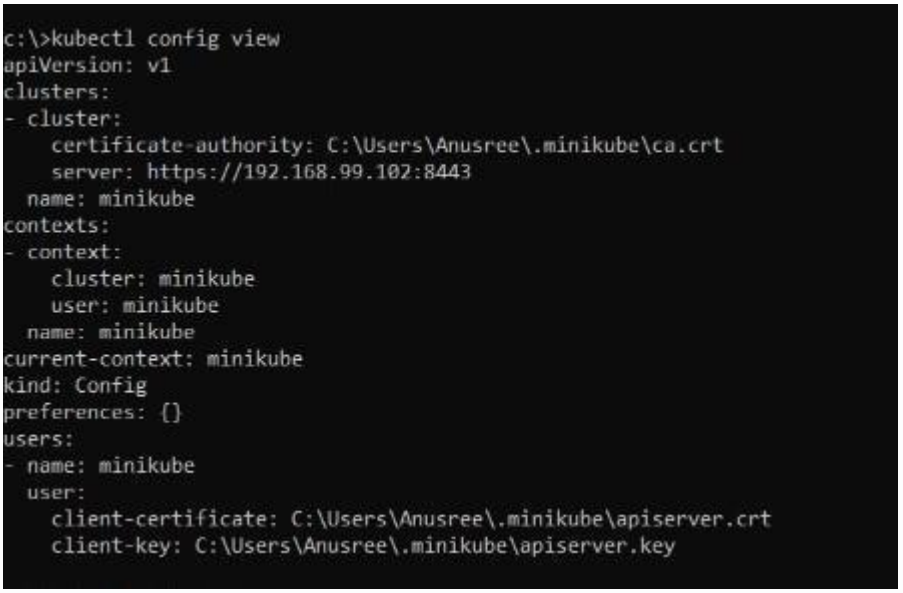
Administrator: Command Prompt

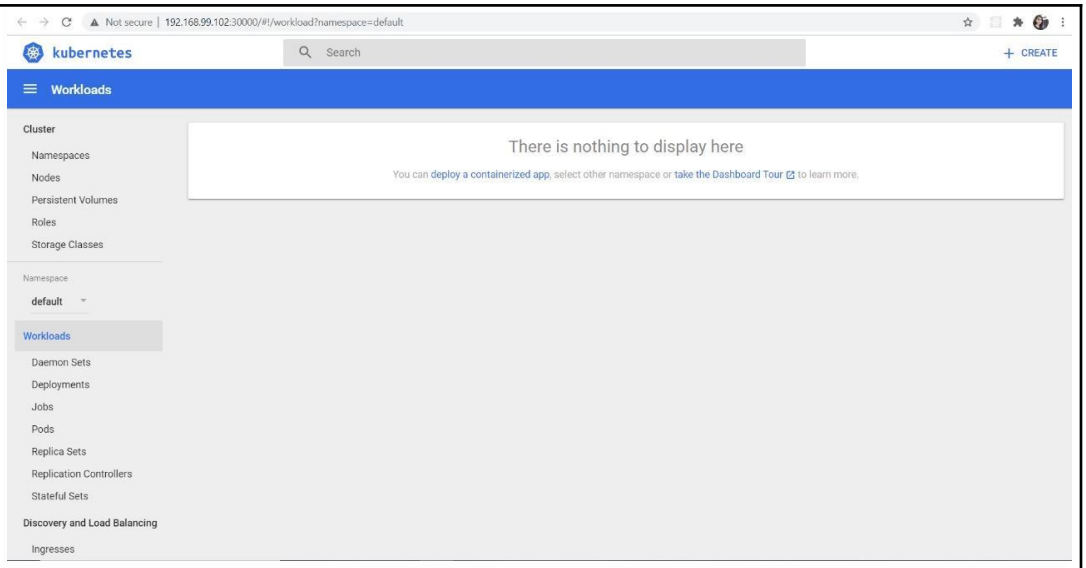
```
c:\>minikube-windows-amd64 start --vm-driver=virtualbox
There is a newer version of minikube available (v1.19.0).  Download it here:
https://github.com/kubernetes/minikube/releases/tag/v1.19.0

To disable this notification, run the following:
minikube config set WantUpdateNotification false
Starting local Kubernetes v1.6.4 cluster...
Starting VM...
Downloading Minikube ISO
 90.95 MB / 90.95 MB [============================================] 100.00% 0s
Moving files into cluster...
Setting up certs...
Starting cluster components...
Connecting to cluster...
Setting up kubeconfig...
Kubectl is now configured to use the cluster.
```

Then we can check the running status of the minikube using the following command.

- minikube-windows-amd64 status

```
Kubectl is now configured to use the cluster.

c:\>
c:\>minikube-windows-amd64 status
minikube: Running
localkube: Running
kubectl: Correctly Configured: pointing to minikube-vm at 192.168.99.102
```

Accessing Minikube [5]	There are 3 ways to access a running Minikube. 1. From Command Line using kubectl 2. From Graphical User Interface 3. APIs To connect to the Kubernetes cluster, kubectl needs the Master Node endpoint and the credentials to connect to it. In the process, Minikube creates a configuration file inside the .kube directory, which resides in the user's home directory. That configuration file has all the connection details. By default, the kubectl binary accesses this file to find the Master Node's connection endpoint, along with the credentials. To look at the connection details, we can run the following command. - kubectl config view  The following is used to get more information about the minikube - $ kubectl cluster-info  Kubernetes master is running at https://192.168.99.100:8443



References

1. https://thenewstack.io/deploy-a-single-node-kubernetes-instance-in-seconds-with-microk8s/
2. https://ubuntu.com/blog/introduction-to-microk8s-part-1-2#:~:text=MicroK8s%20is%20a%20powerful%2C%20lightweight,%2C%20Grafana%2C%20Cilium%20and%20more.
3. https://www.mirantis.com/blog/introduction-to-yaml-creating-a-kubernetes-deployment/
4. https://kubernetes.io/docs/tasks/tools/#:~:text=Quick%20Start%20Guide-,minikube,or%20for%20daily%20development%20work.
5. https://medium.com/@vidurada/getting-start-with-kubernetes-in-windows-10-with-minikube-and-kubectl-32320d75d910