



B.M.S. COLLEGE OF ENGINEERING

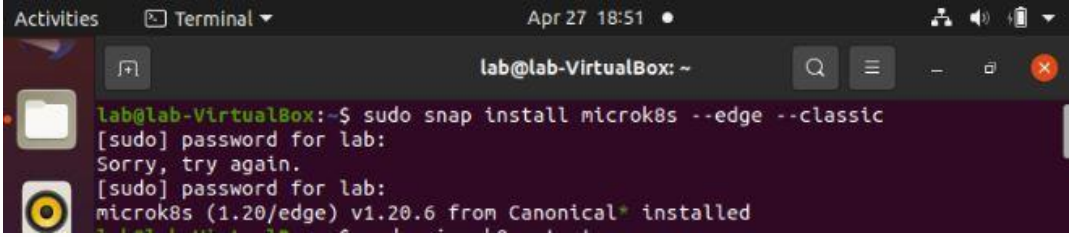
Autonomous Institute, Affiliated to VTU


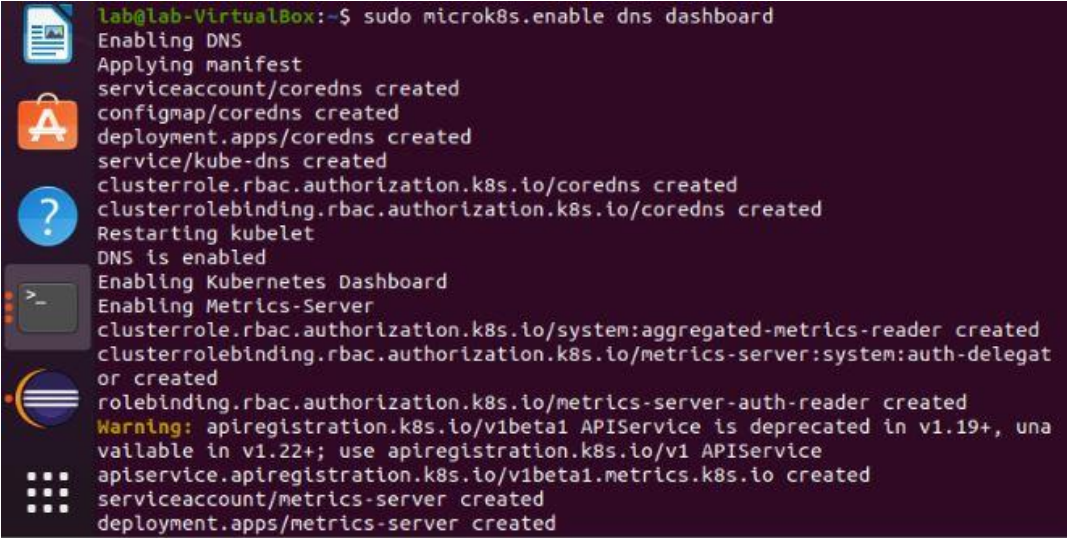
Estd. 1946

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	09-03-2021	18-03-2021	
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]	<p>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.</p> <p>Microk8s is a non-elastic, rails-based single-node Kubernetes tool that is focused primarily on offline development, prototyping, and testing.</p>		

Features of Microk8s [1,2]	<p>Microk8s includes the following features:</p> <ul style="list-style-type: none"> ● Istio ● GPGPU bindings ● Daily builds ● Local storage ● Local registry ● Updates ● Dashboard ● Metrics ● Upgrades ● IngressDNS ● Conformance ● 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: <ul style="list-style-type: none"> ○ 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 microk8s [1]	<p>Step 1) To install microk8s</p> <pre>sudo snap install microk8s --edge --classic</pre> 

	<p>Step 2) To start the microk8s:</p> <pre>sudo microk8s.start</pre>  <p>The command should report that the service has started and pod scheduling has been enabled.</p>
<p>Microk8s dashboard [1]</p>	<p>The Dashboard is a web-based Dashboard that allows you to interact and manage Kubernetes</p> <p>Step 3) To install the dashboard we use the following command</p> <pre>sudo microk8s.enable dns dashboard</pre>  <p>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)</p> <p>Step 4) We access the dashboard we need the address for that we us the following command:</p> <pre>sudo microk8s.kubectl get all --all-namespaces</pre>

```
Activities Terminal Apr 27 18:51 lab@lab-VirtualBox: ~
eating-sample-user.md
lab@lab-VirtualBox:~$ sudo microk8s.kubectl get all --all-namespaces
NAMESPACE      NAME                                     READY   STATUS
RESTARTS      AGE
kube-system    pod/metrics-server-8bbfb4bdb-hlwrb    0/1     Containe
rCreating     0      54s
kube-system    pod/calico-kube-controllers-847c8c99d-zr27p  0/1     Containe
rCreating     0      5m43s
kube-system    pod/calico-node-8tg25                  0/1     Init:2/3
1             5m43s
kube-system    pod/coredns-86f78bb79c-bdk2x          0/1     Containe
rCreating     0      116s
kube-system    pod/kubernetes-dashboard-7ffd448895-wk648  0/1     Containe
rCreating     0      50s
kube-system    pod/dashboard-metrics-scraper-6c4568dc68-zwnk1  0/1     Containe
rCreating     0      48s

NAMESPACE      NAME                                     TYPE      CLUSTER-IP
EXTERNAL-IP    PORT(S)                                AGE
default        service/kubernetes                     ClusterIP  10.152.183.1
<none>         443/TCP                                7m56s
kube-system    service/kube-dns                       ClusterIP  10.152.183.10
<none>         53/UDP,53/TCP,9153/TCP                 2m12s
kube-system    service/metrics-server                 ClusterIP  10.152.183.164
<none>         443/TCP                                54s
kube-system    service/kubernetes-dashboard           ClusterIP  10.152.183.58
<none>         443/TCP                                50s
kube-system    service/dashboard-metrics-scraper      ClusterIP  10.152.183.80
```

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:

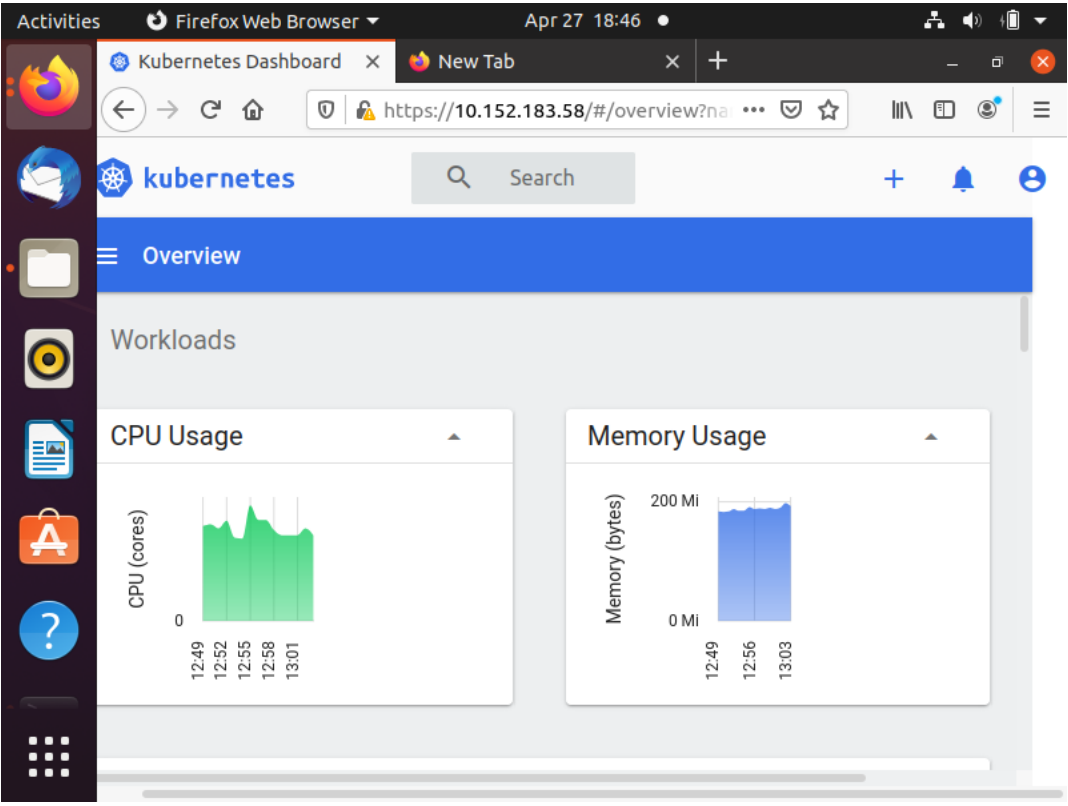
```
sudo microk8s.kubectl -n kube-system get secret
```


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 x 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 x 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 x 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

Services

Activities Firefox Web Browser Apr 27 18:46

Kubernetes Dashboard x New Tab

https://10.152.183.58/#/overview?na

kubernetes Search

Overview

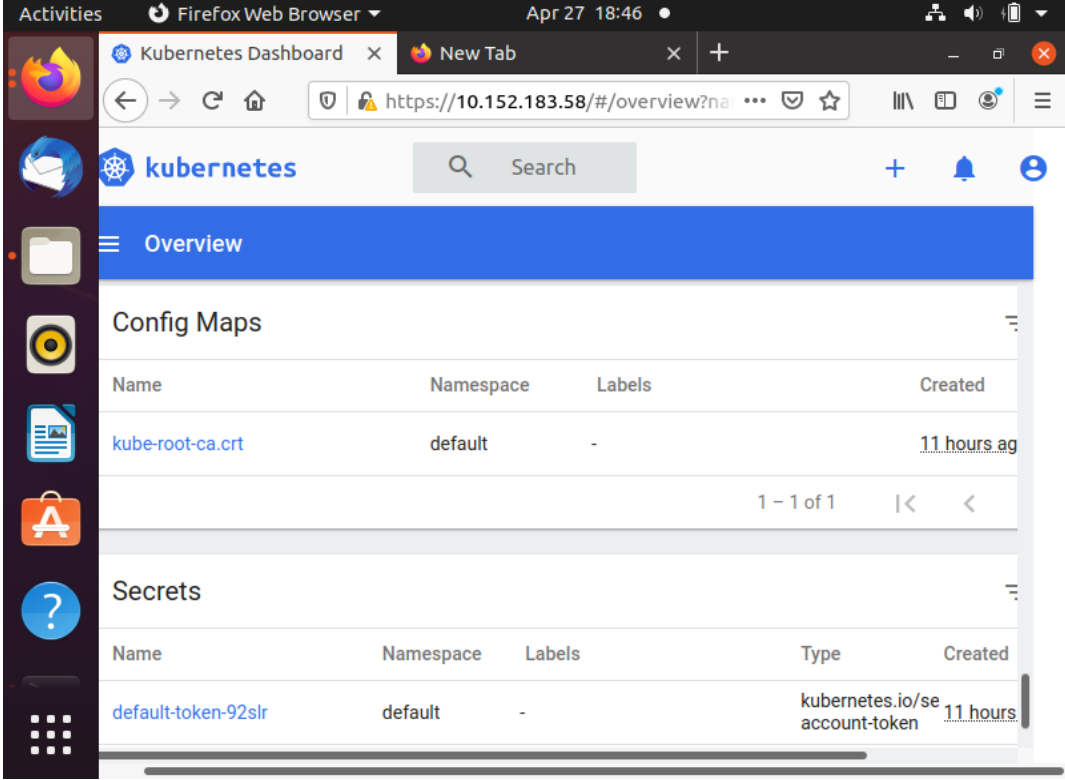
Discovery and Load Balancing

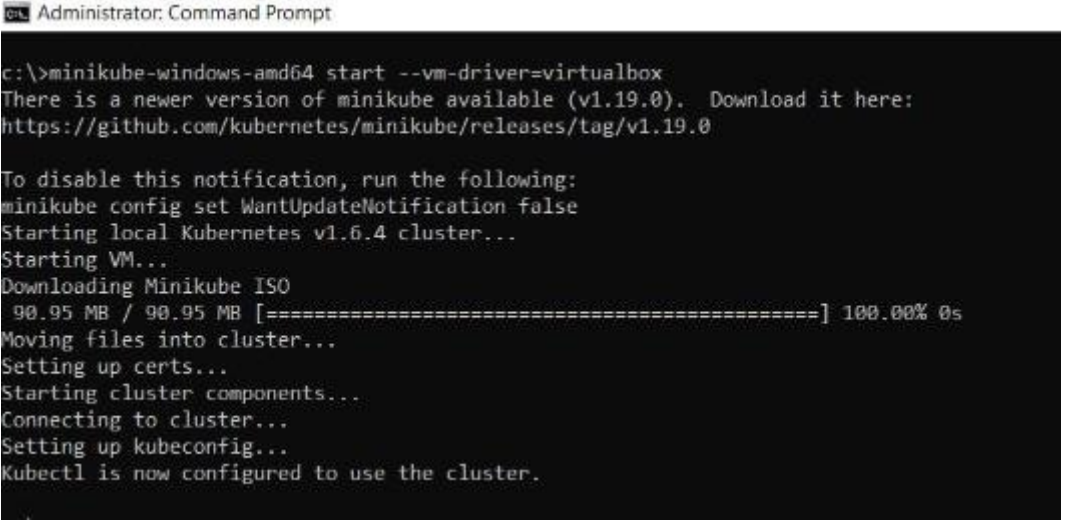
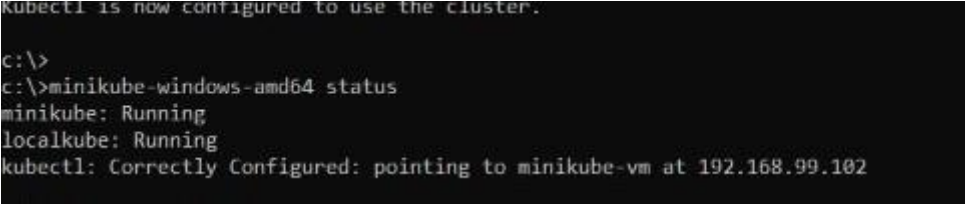
Services

Name	Namespace Labels	Cluster IP	Internal Endpoints	External Endpoints	Created
✓ kubernetes	default component: apiserver provider: kubelets	10.152.183.	kubernetes: TCP kubernetes: TCP		11 h ago

1 - 1 of 1

Config and Storage

	
What is Minikube [4,5]	<p>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.</p>
Installation of Minikube [5]	<p>Note:-The installation is done on windows</p> <p>Step 1) Initially the package Chocolatey was installed for installing kubectl. The command is as follows:</p> <ul style="list-style-type: none"> - <code>@"%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%;%ALLUSERSPROFILE%\chocolatey\bin"</code> <p>Step 2) Then to install kubectl we use the following command.</p> <ul style="list-style-type: none"> - <code>choco install kubernetes-cli</code>

	<div data-bbox="501 197 1572 647"><pre>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 kubect1-convert.exe ShimGen has successfully created a shim for kubect1.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).</pre></div> <p>The binary for Minikube was downloaded, and installed.</p> <p>After installation was complete the following command was executed to start Minikube.</p> <ul style="list-style-type: none">- minikube start --vm-driver=virtualbox <div data-bbox="501 913 1572 1431"><pre>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... Kubect1 is now configured to use the cluster.</pre></div> <p>Then we can check the running status of the minikube using the following command.</p> <ul style="list-style-type: none">- minikube-windows-amd64 status <div data-bbox="501 1650 1471 1854"><pre>Kubect1 is now configured to use the cluster. c:\> c:\>minikube-windows-amd64 status minikube: Running localkub: Running kubect1: Correctly Configured: pointing to minikube-vm at 192.168.99.102</pre></div>
<p>Accessing Minikube</p> <p>[5]</p>	<p>There are 3 ways to access a running Minikube.</p>

	<h2>1. From Command Line using kubectl</h2>
--	---

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

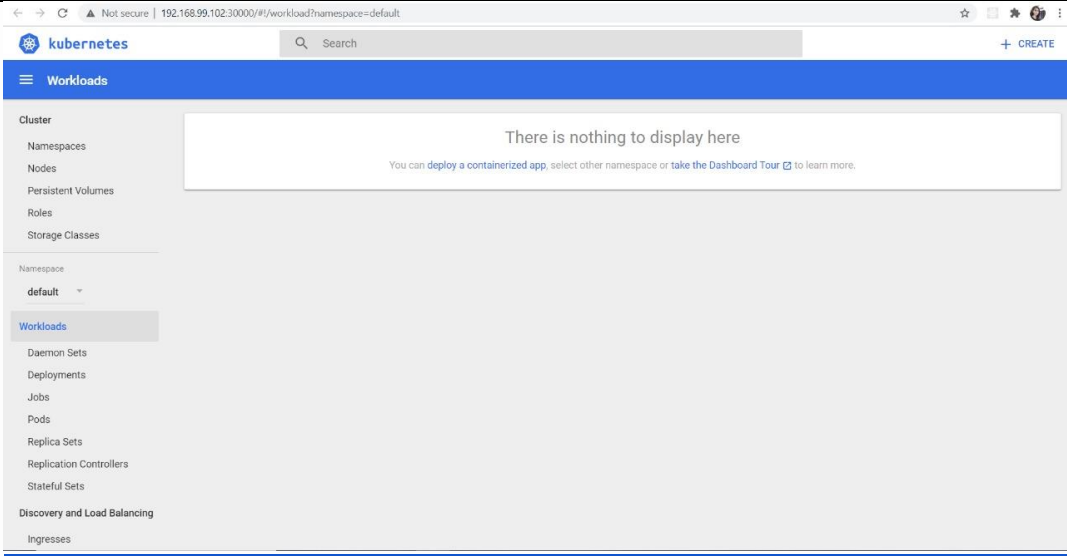
```
c:\>kubectl config view
apiVersion: v1
clusters:
- cluster:
  certificate-authority: C:\Users\Anusree\.minikube\ca.crt
  server: https://192.168.99.102:8443
  name: minikube
contexts:
- context:
  cluster: minikube
  user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: C:\Users\Anusree\.minikube\apiserver.crt
    client-key: C:\Users\Anusree\.minikube\apiserver.key
```

The following is used to get more information about the minikube

- \$ kubectl cluster-info

```
c:\>kubectl cluster-info
Kubernetes control plane is running at https://192.168.99.102:8443
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Kubernetes master is running at <https://192.168.99.100:8443>

	
References	<ol style="list-style-type: none">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