**Lab 08**

**Laboratory Exercise**

**Part 1: Explore Minikube in Ubuntu**

Watch this video to understand what is Kubernetes.

<https://www.youtube.com/watch?v=l_lWfipUimk>

Watch this video to understand what is Minikube.

<https://www.youtube.com/watch?v=E2pP1MOfo3g>

**LAB EXERCISE**

This LAB exercise guide you in using Minikube. Minikube is an open-source tool that was developed to enable developers and system administrators to run a single cluster of Kubernetes on their local machine.

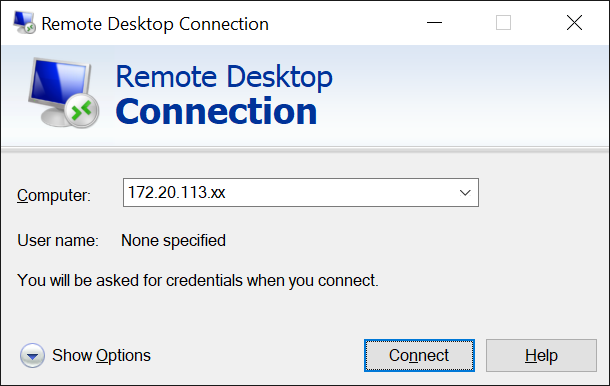
Minikube starts a single node Kubernetes cluster locally with small resource utilization.

**Time to Complete**

Approximately 30 Minutes

From your machine logged-in to RP VPN, run Remote Desktop Connection to connect to the ubuntu Linux Virtual Machine (VM). Please login based on your assigned VM as shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S/N** | **Name** | **VM** | **IP Address** | User Name | Password |
| 1 | ABDUL SALIM BIN ABDUL RASHITH | LABC03 - 172.20.115.50 | 172.20.115.50 | dockeradm | docker!2 |
| 2 | CASPER LEOW YU HAN (LIAO YU HANG) | LABC03 - 172.20.115.51 | 172.20.115.51 | dockeradm | docker!2 |
| 3 | CHAN JUN ZHI, GLENN | LABC03 - 172.20.115.52 | 172.20.115.52 | dockeradm | docker!2 |
| 4 | CHIA WAI TAT | LABC03 - 172.20.115.53 | 172.20.115.53 | dockeradm | docker!2 |
| 5 | HOI WAI TECK | LABC03 - 172.20.115.54 | 172.20.115.54 | dockeradm | docker!2 |
| 6 | KOH JIN CAI DAEMIAN | LABC03 - 172.20.115.55 | 172.20.115.55 | dockeradm | docker!2 |
| 7 | KYAW KYAW OO | LABC03 - 172.20.115.56 | 172.20.115.56 | dockeradm | docker!2 |
| 8 | LUM YOKE FAI | LABC03 - 172.20.115.57 | 172.20.115.57 | dockeradm | docker!2 |
| 9 | MUHAMMAD FADHLI BIN MOHAMED NOOR | LABC03 - 172.20.115.58 | 172.20.115.58 | dockeradm | docker!2 |
| 10 | MUHAMMAD HILMEE BIN MD ALI | LABC03 - 172.20.115.59 | 172.20.115.59 | dockeradm | docker!2 |
| 11 | NG SAY WEE | LABC03 - 172.20.115.60 | 172.20.115.60 | dockeradm | docker!2 |
| 12 | NGUI WEILY | LABC03 - 172.20.115.61 | 172.20.115.61 | dockeradm | docker!2 |
| 13 | NU'MAN HARITH BIN NORRAIMI | LABC03 - 172.20.115.62 | 172.20.115.62 | dockeradm | docker!2 |
| 14 | RULY JANUAR FACHMI | LABC03 - 172.20.115.76 | 172.20.115.76 | dockeradm | docker!2 |
| 15 | SEAH SHIH WEI GEROME | LABC03 - 172.20.115.64 | 172.20.115.64 | dockeradm | docker!2 |
| 16 | SEAN CHENG ZHI WEI | LABC03 - 172.20.115.65 | 172.20.115.65 | dockeradm | docker!2 |
| 17 | SEY KOK SIONG | LABC03 - 172.20.115.66 | 172.20.115.66 | dockeradm | docker!2 |
| 18 | TAN JOON YEE DOUGLAS | LABC03 - 172.20.115.67 | 172.20.115.67 | dockeradm | docker!2 |
| 19 | WU WAI TENG VANESSA | LABC03 - 172.20.115.68 | 172.20.115.68 | dockeradm | docker!2 |
| 20 | YAP KOON SING | LABC03 - 172.20.115.69 | 172.20.115.69 | dockeradm | docker!2 |
| 21 | YE CHENG LIM | LABC03 - 172.20.115.70 | 172.20.115.70 | dockeradm | docker!2 |
| 22 | SHAIFUL BIN ABDUL KARIM | LABC03 - 172.20.115.71 | 172.20.115.71 | dockeradm | docker!2 |
| 23 | CHAI RU YI | LABC03 - 172.20.115.72 | 172.20.115.72 | dockeradm | docker!2 |
| 24 | JWAY HWEE LING JULIE | LABC03 - 172.20.115.73 | 172.20.115.73 | dockeradm | docker!2 |
| 25 | SAMANTHA TEO XING YEE | LABC03 - 172.20.115.74 | 172.20.115.74 | dockeradm | docker!2 |
| 26 | ZIL AZZA HILMIAH BINTE RADUAN | LABC03 - 172.20.115.75 | 172.20.115.75 | dockeradm | docker!2 |



Replace xx with the IP address of the VM that you have been assigned.

Exploring MiniKube on a Virtual Machine

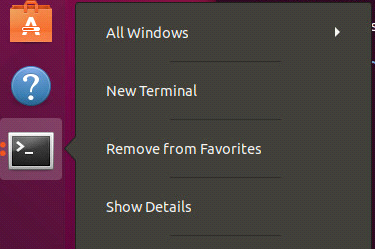
In the segment, we will be exploring the following:

a. How to launch Minikube.

b. Run a single cluster of Kubernetes.

1. Login to the VM using username "dockeradm" and password “docker!2”.

2. In the desktop space, “right mouse click” and select “Open Terminal”.

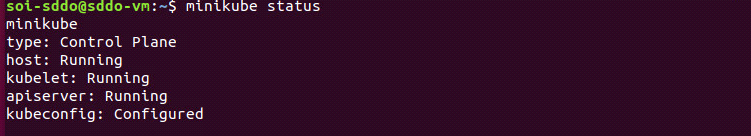


**Launch Kubernetes Cluster locally - Minikube start**  
3. From the command prompt on windows execute "minikube start”

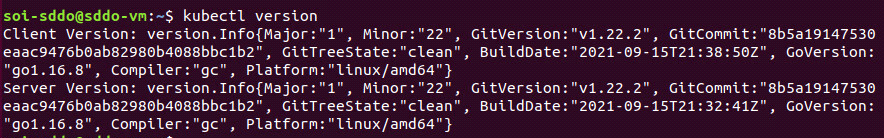


**Minikube commands**

3.1. Check the status of minikube using command “minikube status”



3.2. From the above command prompt execute "kubectl version"

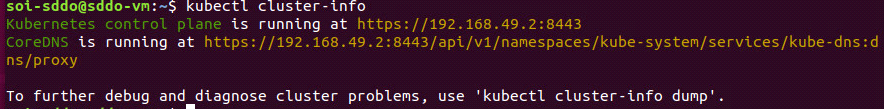


OK, kubectl is configured and we can see both the version of the client and as well as the server.

The client version is the kubectl version; the server version is the Kubernetes version installed on the master. You can also see details about the build.

Output will show both client and server versions

3.3. Use the kubectl CLI to get the cluster information: “kubectl cluster-info”



**Cluster IP Address**

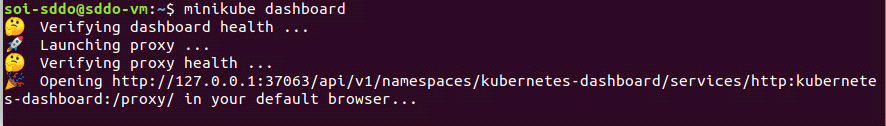
4. Get the IP address of the cluster via the ip command “minikube ip”



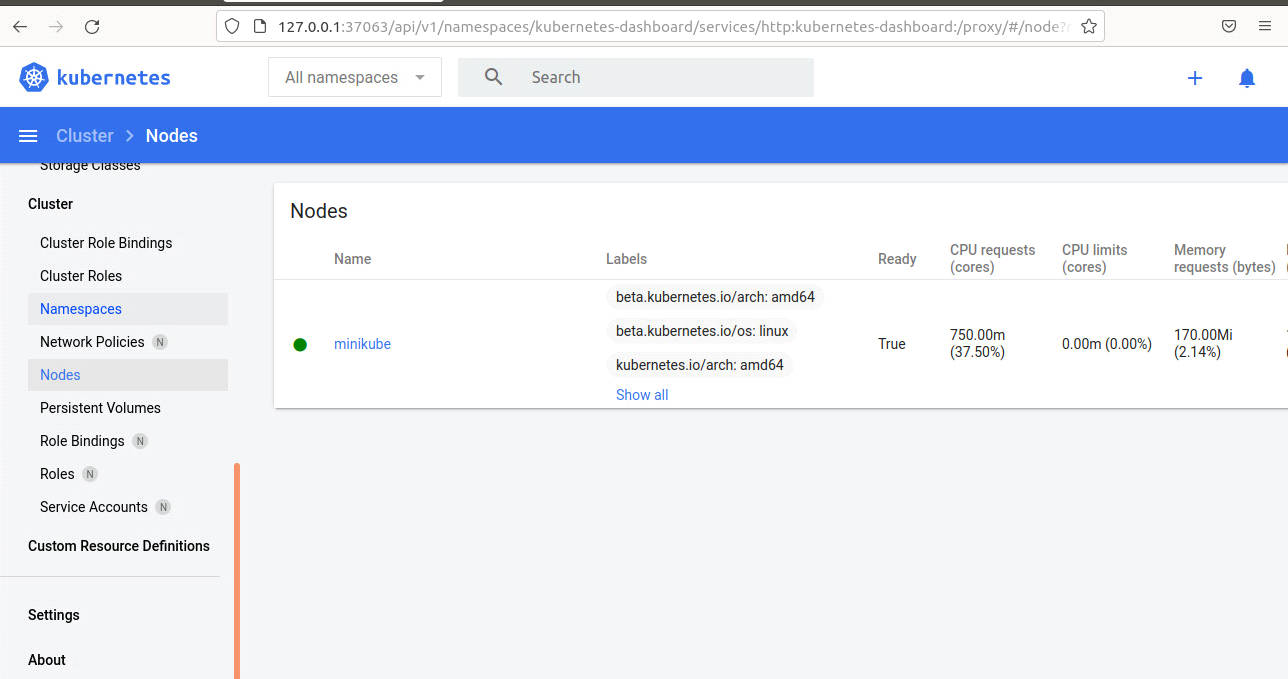
**Kubernetes Dashboard**

**Note:** If you installed minikube locally, run minikube start. Before you run minikube dashboard, you should open a new terminal, start minikube dashboard there, and then switch back to the main terminal.

4.1. Launch the Kubernetes Dashboard at any point via the dashboard command as shown below: “minikube dashboard”



4.2. Click on the Node link and you will see that information:



4.3. The above node information can also be obtained by using the kubectl CLI to get the list of nodes “kubectl get nodes” on a new terminal.

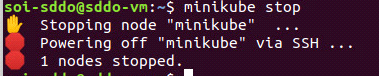


Now we have only one node, and we can see that its status is ready (it is ready to accept applications for deployment).

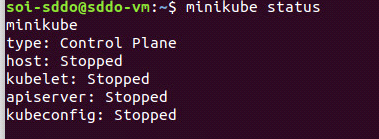
Kubernetes will choose where to deploy our application based on Node available resources.

**Stop the Kubernetes Cluster**

5.1. Run the below command to stop the Kubernetes cluster using command “minikube stop”



5.2. Check the status of cluster using minikube command “minikube status”



**Part 2: Kubectl Commands**

This LAB exercise is to practice the kubectl commands to work with Kubernetes deployment

Please refer to this kubectl cheat sheet for details of the various kubectl commands and options:

Ref: <https://phoenixnap.com/kb/kubectl-commands-cheat-sheet>



**Time to Complete**

Approximately 30 Minutes

**What You Need**

Kubernetes Cluster should be running. If not start the cluster using the command “minikube start”

**Kubectl Commands**

1. Open a terminal as dockeradm.

2. Check the Kubernetes cluster status using the “**minikube status**” command.

3. If the cluster is not running, start the cluster using command “**minikube start**”.

4. Show merged kube config settings using the command “**kubectl config view**”.

**Deploy first app**

We need to provide the deployment name and app image location (include the full repository url for images hosted outside Docker hub).

5.1 Start a single instance of nginx using the command “**kubectl create deployment nginx --image=nginx**”.

Great! You just deployed your first application by creating a deployment. This performed a few things for you:

* searched for a suitable node where an instance of the application could be run (we have only 1 available node)
* scheduled the application to run on that Node
* configured the cluster to reschedule the instance on a new Node when needed

6. Get the POD documents using the command “**kubectl explain pods**” for pod manifests.

7. View and find resources using below commands.

Pods that are running inside Kubernetes are running on a private, isolated network. By default they are visible from other pods and services within the same kubernetes cluster, but not outside that network. When we use kubectl, we're interacting through an API endpoint to communicate with our application.

|  |  |
| --- | --- |
| **Commands** | **Screenshot of resources** |
| # List all services in the namespaces **kubectl get services** |  |
| # List all pods in all namespaces **kubectl get pods --all-namespaces** |  |
| # List all pods in the namespace, with more details  **kubectl get pods -o wide** |  |
| #List a particular deployment  **kubectl get deployment nginx** |  |
| # Describe commands with verbose output  **kubectl describe pods nginx** |  |
| # Retrieve nginx logs  **kubectl logs <nginx pods>** |  |

8. Print the supported versions of API on the cluster using the command: **kubectl api-versions**

9. Displays the cluster Info using the command:

**kubectl cluster-info**

10. Display the current context of the cluster using the command :

**kubectl config current-context**

11. Create new Pods that send a message in stdout.  
  
11.1. Create a directory using the command: **mkdir pods**   
11.2. vi ~/pods/01\_pods.yaml

apiVersion: v1

kind: Pod

metadata:   
 name: busybox-logs   
spec:   
 containers:   
 - name: busybox   
 image: busybox

command: ["/bin/sh"]   
 args: ["-c", "echo \"$(date) - INFO - My first logs output on $(hostname)\""]   
# restartPolicy: Never

Create the resource based on the previous yaml file definition.

kubectl create -f ~/data/pods/01\_pods.yaml

Get the logs of the busybox-logs Pod created previously.

kubectl logs busybox-logs

**--End of Lab Exercise --**

Those documentations can help you to go further in this topic :

* Kubernetes official [HYPERLINK "https://kubernetes.io/docs/reference/kubectl/cheatsheet/"cheat sheet](https://kubernetes.io/docs/reference/kubectl/cheatsheet/) documentation
* [Kubectl official Reference](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-strong-getting-started-strong-) documentation
* Kubernetes official documentation [HYPERLINK "https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/"pod overview](https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/)
* Kubernetes official documentation on [HYPERLINK "https://kubernetes.io/docs/concepts/overview/object-management-kubectl/overview/"object management](https://kubernetes.io/docs/concepts/overview/object-management-kubectl/overview/)
* Kubernetes official documentation on [HYPERLINK "https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/"list pods](https://kubernetes.io/docs/tasks/access-application-cluster/list-all-running-container-images/)
* Kubernetes official documentation on [HYPERLINK "https://kubernetes.io/docs/tasks/debug-application-cluster/debug-application-introspection/#using-kubectl-describe-pod-to-fetch-details-about-pods"introspection and debugging](https://kubernetes.io/docs/tasks/debug-application-cluster/debug-application-introspection/#using-kubectl-describe-pod-to-fetch-details-about-pods)
* Kubernetes official documentation on [HYPERLINK "https://kubernetes.io/docs/tasks/debug-application-cluster/get-shell-running-container/"getting a shell to a running container](https://kubernetes.io/docs/tasks/debug-application-cluster/get-shell-running-container/)
* Kubernetes official documentation on [HYPERLINK "https://kubernetes.io/docs/concepts/cluster-administration/manage-deployment/"resources management](https://kubernetes.io/docs/concepts/cluster-administration/manage-deployment/)