The Commands Used In The Kubernetes Course !

"For Lesson 2"

Video 01. "Minikube & Kubectl"

- ***Minikube start***

This command is used to start a new cluster of the kubernetes

- ***Minikube Status***

This command is used to check the status of the minikube

- ***Kubectl cluster-info***

This command is used to check the information of the cluster either our cluster has made or not ?

Video 02. "Nodes"

- ***Kubectl get nodes***

This command is used to check the nodes form our kubernetes architecture, means that how many master nodes and worker nodes are available in our kubernetes architecture?

- ***Kubectl describe node minikube***

This command is used to check the description or details of the nodes. If we will write the command only till the node this will share details of all node that will confuse us, but if we will write the name of the specific node this will share the specified node's info

Video 03. "Alias"

- ***alias kgn="kubectl get nodes"***

This command is used to making an alias for long-tail commands because kubernetes uses the long-tail commands.

Video 04. "Pods"

This Video's Content Is Theory Based.

Video 05. "Why Pods"

This Video's Content Is Theory Based.

This Page Has 01 To 05 Video's Content.

Video 06. "About Pods Isolation"

This Video's Content Is Theory Based.

Video 07. "Multi Tier Apps Into Single Pod"

This Video's Content Is Theory Based.

Video 08. "Grouping The Pods"

This Video's Content Is Theory Based.

Video 09. "YAML File"

This Video's Content Is Theory Based.

Video 10. "Creating A Pod"

The YAML File

```
kind: Pod
apiVersion: v1
metadata:
  name: myfirstpod
spec:
  containers:
  - name: container1
    image: rizwansheikh7071/hello-world
    ports:
    - containerPort: 80
```

Note That : *During Creating the YAML file, always take care of the spaces the one space less or one more will create error when the file will be read by the* ***API SERVER*** *of the kubernetes.*

EXPLANATION OF THE YMAL FILE

The Kind key

Kind: Pod

This key tells the kubernetes that which type of resource you have to made, ***always keep in mind*** *that when you write the name of the resource so always write the first letter capital of the word like* ***Pod***

2nd Part is apiVersion

apiVersion: v1

We used v1 in the apiVersion in the resource of Pod
NOTE That: *Both the key and the value of the key have one space in betwee.*

3rd Part is metadata

metadata:
name: myfirstpod

In the metadata there are many things come in the metadata like ***name, namespaces, labels,annotations etc.*** *In this part the metadata is main heading and the name is the sub heading and so name is the field/key and the value is myfirstpod. T****he Imp Part Of The MetaData Is The Name Of The Pod Should Must Exist.***

4th Part is spec

In this part we tell that how much containers will be in this pod and many more important description.

- ***Kubectl create -f myfirstpod.yaml***

Through this command we give the instructions that create my first pod with the giving configuration file. ***The -f flag*** *is giving indication that we are giving a file of configuration for making resource.*

Video 11. "Pods Listing And Insights"

- ***kubeclt get pods/pod/po***

This command is used to check the list of the pods and get notice that ***the word get*** *is used to take information about any kind of resource.*

- ***Kubectl get pods/pod/po myfirstpod -o yaml***

The flag -o is used to seeing the out put in specified format like yaml or json.

- ***Kubectl describe pod myfirstpod***

This command is used to check the details and insights about our pod and many more information about the pod.

- ***kubectl explain pods***

DESCRIPTION:
Pod is a collection of containers that can run on a host. This resource is created by clients and scheduled onto hosts.

- ***kubectl explain pod.spec***

RESOURCE: spec <Object>
DESCRIPTION:
Specification of the desired behavior of the pod...

podSpec is a description of a pod.

Video 12. "Port Forwarding"

- ***kubectl port-forward myfirstpod 1500:80***

This command is used to forward the internal port to the external means that assigning the port for the external user more means that when a user will be hit our machine's or the server's IP address and then the kubernetes will be divert on the application's exposed port like 80 port is used for the nginx port.

Video 13. "Creating Pods From Command Line"

- ***kubectl run mythirdpod --image=aamirpinger/helloworld --port=80 –restart=Never***

This command is used to create the pod without creating the YAML file before.

Video 14. "Grouping Of Resources"

This Video's Content Is Theory Based.

Video 15. "Labels"

This Video's Content Is Theory & Little On Practical Based.

```
allii@ap-linux:~$ cat myfirstpodwithlabes.yaml
kind: Pod
apiVersion: v1
metadata:
  name: myfirstpodwithlabels
  labels:
    type: backend
    env: production
spec:
  containers:
  - image: aamirpinger/helloworld
    name: container1
    ports:
   - containerPort: 80
```

- ***kubectl create -f myfirstpodwithlabes.yaml***

This command is used to create for the pod with running the container1.

- ***kubectl run anotherpodwithlabel --image aamirpinger/helloworld --port=80 --restart=Never –labels=type=frontend,env=development***

This command is used to create pod with running the container1 with the of ***aamirpinger/helloworld*** *and assigning the* ***port 80 for nginx*** *application and restart=Never is used to assigning the specific task to making the resource of pod if we will not do this so the kubernetres will make another pod that will not required and also* ***assigning the labels*** *to this pod.*

Video 16. "Pods Listing With Labels"

- ***kubectl get po –show-labels***

This command is used to get the information about the labels. This command will show the results all pod having or not the labels.

```
NAME                    READY   STATUS    RESTARTS   AGE   LABELS
anotherpodwithlabel     1/1     Running   0          27m   env=development,type=frontend
myfirstpod              1/1     Running   4          22h   <none>
myfirstpodwithlabels    1/1     Running   0          33m   env=production,type=backend
mysecondpod             0/1     Error     0          18h   run=mysecondpod
mythirdpod              0/1     Error     0          18h   run=mythirdpod
```

This will shows us if the we created with command line or the giving the ***YAML file.***

- ***kubectl get po -L env,type,run***

This command is used to check the results of labels in the columns.

Video 17. "Labelling Pod At Runtime"

- ***kubectl get po –show-labels***

Firstly use this command to check the list of labels and then use below command

- ***kubectl label pod myfirstpod app=helloworld type=frontend***

By using this command we will be able to assign the labels to the running pods if the pod has no any label before.

- ***kubectl label po anotherpodwithlabel env=production --overwrite***

This command is used to change the label at the running time of the pod.

- ***kubectl label pod myfirstpod app-***

This command is used to remove the label from the running time of the pod.

Video 18. "Label Selector"

- ***kubectl get pods -l type=frontend***

This command is used to get the label that which has only the value of frontend but without showing the labels in the results

- ***kubectl get pods -l type=frontend –show-labels***

While this command is used to get the list of labels in the results.

- ***kubectl get pods -l type!=frontend –show-labels***

This command is used to get results of labels that should be consists on the others labels except of frontend mtllab keh frontend k ilawa jo bhee han wo show kr do or yaad rhy keh frontend wala label show niii krna.

- ***kubectl get pod -l type!=frontend,env=production –show-labels***

*This command is used to implement to conditions in the single command like **"type"** should not be frontend but the **"env"** should must be production.*

- ***kubectl get pods -l env –show-labels***

This command is used to get the results by providing the key means that we have no problem whatever the key has the value.

- ***kubectl get pods -l '!env' –show-labels***

*This command is used to take result that consists on other than the env key. Means that the result should be except of **env** value.*

- ***kubectl get pods -l 'type in (frontend,backend)' –show-labels***

This command is used to check the list of the frontend and backend against the type key.

- ***kubectl get pods -l 'type notin (frontend,backend)' –show-labels***

This command will show the result of the other keys except than the type.

Video 19. "Pod Scheduling With Node Selector"

This video is all about to scheduling our pod on the matched nodes..

- ***kubectl label node minikube thetypeofharddisk=ssd***

This command is used to assigning the label with our required name. After this our pod will be must be deploy on the that node which will be match with our pod name.

Video 20. "Annotation"

This Video's Content Is Theory Based.

Video 21. "Describing Pod Insights"

- ***kubectl describe pod my1stpod***

This command is used to take the details of the pod or any other resource of the kubernetes. This command describes the pod in detailed form.

Video 22. "Overlapping Labels"

This Video's Content Is Theory Based.

Video 23. "NameSpace"

This Video's Content Is Theory Based.

This Page Has 19 To 23 Video's Content!

Vidoe 24. "Creating NameSpaces"

- ***kubectl create namespace development***

This command is used to create the namespace ***(Development)***

- ***kubectl create ns production***

This command is used to create the namespace ***(Production)***

Video 25. "Pod Inside Namespace"

- ***kubectl create -f my1stpod.yaml***

This command will create the new pod and now this pod will be create in the namespace of Production not in the default namespace, and before using this namespace command I have created pods but those saved into the default namespace.

- ***kubectl get pod –namespace=production***

This command is used to check the status of the production namespace so this command will show the pod in this namespace.

```
allii@kubernetes:~$ nano my1stpod.yaml
allii@kubernetes:~$ kubectl create -f my1
error: the path "my1" does not exist
allii@kubernetes:~$ kubectl create -f my1stpod.yaml
pod/my1stpod created
allii@kubernetes:~$ kubectl get pod
NAME               READY   STATUS             RESTARTS   AGE
aboutannotations   0/1     ImagePullBackOff   0          10h
my1stpod           0/1     ImagePullBackOff   0          10h
pwannotation       0/1     ImagePullBackOff   0          10h
allii@kubernetes:~$ kubectl get pod --namespace=production
NAME       READY   STATUS             RESTARTS   AGE
my1stpod   0/1     ImagePullBackOff   0          75s
allii@kubernetes:~$ kubectl get pod --namespace=production --show-ns
Error: unknown flag: --show-ns
See 'kubectl get --help' for usage.
allii@kubernetes:~$ kubectl get pod --namespace=default
NAME               READY   STATUS             RESTARTS   AGE
aboutannotations   0/1     ImagePullBackOff   0          10h
my1stpod           0/1     ImagePullBackOff   0          10h
pwannotation       0/1     ImagePullBackOff   0          10h
allii@kubernetes:~$
```

- ***kubectl run nsexample --image=aamirpinger/helloworld --port=80 --restart=Never --namespace=development***

This command is used to create pod in the namespace of Development directly through the command line.

- ***kubectl get pod -n=production***
- ***kubectl get pod -n=development***

The flag ***-n, or complete word of --namespace*** *these kubernetes will be understand and they are work will be same.*

Video 26. "Listing Pod from All Namespaces"

- ***kubectl get pod --all-namespaces***

This command is used to get information of all pods either they are running in the specified namespace or not..
means that all pod on this machine or the cluster.

Video 27. "Deleting Resource"

- ***kubeclt delete pod myfirstpod***

This command will delete the myfirstpod form the default namespace.

- ***kubectl delete pod myfirst pod –namespce=production***

Now this command will delete the pod from the namespace of the ***Production***

- ***kubectl delete ns development***

If we want to delete the namespace directly so we can do through this command
But Remember that *This will delete all resources form inside of the namespace.*

We can delete the pods by implementing the conditions of the labels!

- ***kubectl get pods –show-labels***

Firstly show the labels and then delete the pod using labels

- ***kubeclt delete -l pod type=backend***

This means that those pods delete that which have the ***type=backend***
The ***-l*** *is being use for giving the criteria means labels.*

- ***kubeclt delete -l pod type!=backend***

This command will delete those pods that have the ***key "type"****value other than the backend.*

- ***kubeclt delete -l pod !type***

Means those pods delete that having not the key of the ***type.***

- ***kubeclt delete -l pod type***

By using this there no condition that the key of type has which value ?

- ***kubeclt delete -l pod 'type in (frontend,backend)'***

This command will delete those pods that have value of ***frontend and backend*** *in front of type key.*

- ***Kubeclt delete pods –all***

This will delete all pods that are in the default namespace on the machine.

The Extra Command About Linux

- ***--dry-run***

This command is used to the status of the of the operation running before if the error will come then means that debug before running.

- ***Declarative Mode***

This mode is used to take results about operation before running in this mode we create an template and this template is not the final result.

Alhamdolillah!

Finally The Lesson Two OF The Kubernetes Has Been Completed.

Firstly Thanks Of Allah Almighty
And Then Sir Aamir Pinger.

This Document Is Created By "Rizwan Sheikh"
Email: rizwansheikh7071@gmail.com
Docker ID: rizwansheikh7071
Due To Human Being The Mistake Can Be Done!
Thank You Happy Learning!