

Experiment 04

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes Application.

Theory:

Kubectl:

The Kubernetes command-line tool, kubectl, allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs.

Deploying applications using kubectl:

Once you have a running Kubernetes cluster, you can deploy your containerized applications on top of it. To do so, you create a Kubernetes Deployment configuration. The Deployment instructs Kubernetes how to create and update instances of your application. Once you've created a Deployment, the Kubernetes control plane schedules the application instances included in that Deployment to run on individual Nodes in the cluster.

Once the application instances are created, a Kubernetes Deployment Controller continuously monitors those instances. If the Node hosting an instance goes down or is deleted, the Deployment controller replaces the instance with an instance on another Node in the cluster. This provides a self-healing mechanism to address machine failure or maintenance.

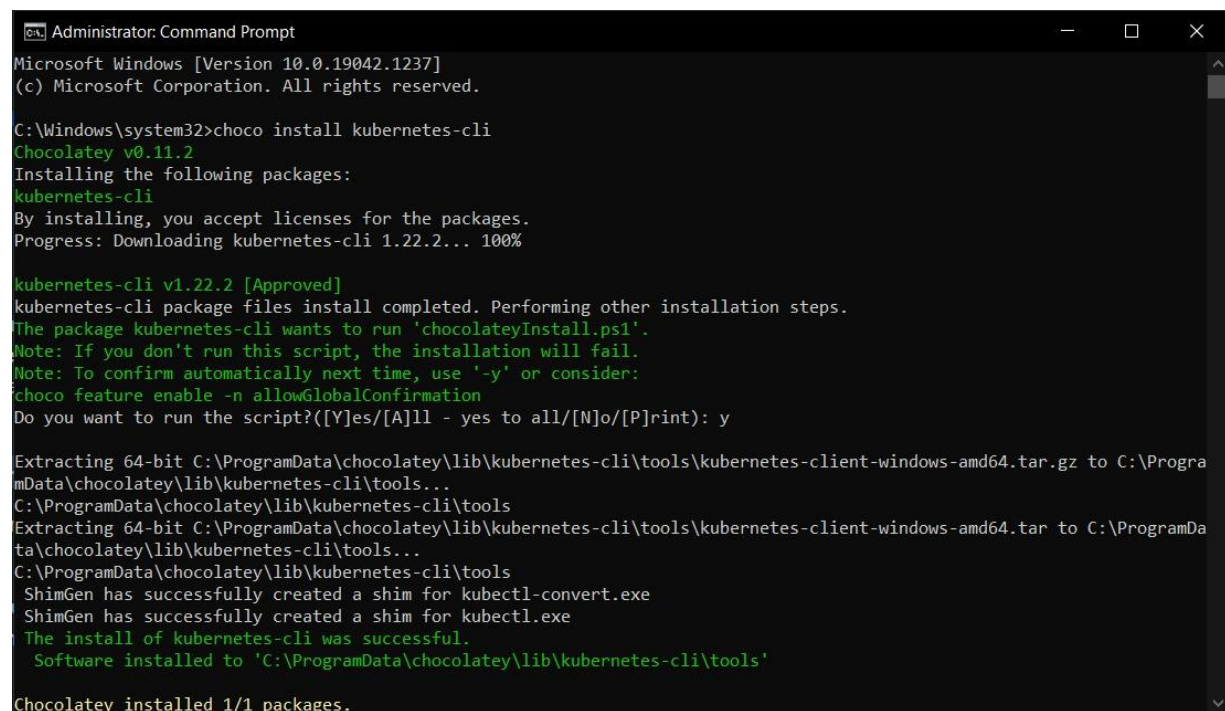
In a pre-orchestration world, installation scripts would often be used to start applications, but they did not allow recovery from machine failure. By both creating your application instances and keeping them running across Nodes, Kubernetes Deployments provide a fundamentally different approach to application management.

You can create and manage a Deployment by using the Kubernetes command line interface, Kubectl. Kubectl uses the Kubernetes API to interact with the cluster. In this module, you'll learn the most common Kubectl commands needed to create Deployments that run your applications on a Kubernetes cluster.

When you create a Deployment, you'll need to specify the container image for your application and the number of replicas that you want to run.

Outputs:

Installing kubectl using chocolatey:



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>choco install kubernetes-cli
Chocolatey v0.11.2
Installing the following packages:
kubernetes-cli
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-cli 1.22.2... 100%

kubernetes-cli v1.22.2 [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.
```

```
Administrator: Command Prompt
C:\Windows\system32>kubectl version
Client Version: version.Info{Major:"1", Minor:"21", GitVersion:"v1.21.5", GitCommit:"aea7bbadd2fc0cd689de94a54e5b7b758869d691", GitTreeState:"clean", BuildDate:"2021-09-15T21:10:45Z", GoVersion:"go1.16.8", Compiler:"gc", Platform:"windows/amd64"}
Server Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.2", GitCommit:"8b5a19147530eaac9476b0ab82980b4088bbc1b2", GitTreeState:"clean", BuildDate:"2021-09-15T21:32:41Z", GoVersion:"go1.16.8", Compiler:"gc", Platform:"linux/amd64"}

C:\Windows\system32>kubectl get nodes
NAME          STATUS    ROLES          AGE   VERSION
minikube      Ready     control-plane,master  2m48s  v1.22.2

C:\Windows\system32>
```

Deploying first application using Kubernetes:

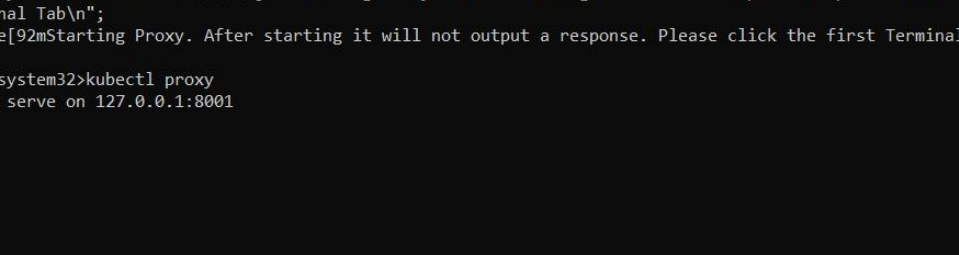
```
Administrator: Command Prompt
C:\Windows\system32>kubectl create deployment kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1
deployment.apps/kubernetes-bootcamp created

C:\Windows\system32>
C:\Windows\system32>kubectl get deplyments
error: the server doesn't have a resource type "deplyments"

C:\Windows\system32>kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp  0/1     1            0           24s

C:\Windows\system32>kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp  1/1     1            1           86s

C:\Windows\system32>
```



```
Administrator: Command Prompt - kubectl proxy
C:\Windows\system32>echo -e "\n\n\n[e[92mStarting Proxy. After starting it will not output a response. Please click the
first Terminal Tab\n";
-e "\n\n\n[e[92mStarting Proxy. After starting it will not output a response. Please click the first Terminal Tab\n";

C:\Windows\system32>kubectl proxy
Starting to serve on 127.0.0.1:8001
```

Checking the deployed application:

```

- paths: [
  "/.well-known/openid-configuration",
  "/api",
  "/api/v1",
  "/apis",
  "/apis/",
  "/apis/admissionregistration.k8s.io",
  "/apis/admissionregistration.k8s.io/v1",
  "/apis/apixtensions.k8s.io",
  "/apis/apixtensions.k8s.io/v1",
  "/apis/apiregistration.k8s.io",
  "/apis/apiregistration.k8s.io/v1",
  "/apis/apps",
  "/apis/apps/v1",
  "/apis/authentication.k8s.io",
  "/apis/authentication.k8s.io/v1",
  "/apis/authorization.k8s.io",
  "/apis/authorization.k8s.io/v1",
  "/apis/autoscaling",
  "/apis/autoscaling/v1",
  "/apis/autoscaling/v2beta1",
  "/apis/autoscaling/v2beta2",
  "/apis/batch",
  "/apis/batch/v1",
  "/apis/batch/v1beta1",
  "/apis/certificates.k8s.io",
  "/apis/certificates.k8s.io/v1",
  "/apis/coordination.k8s.io",
  "/apis/coordination.k8s.io/v1",
  "/apis/discovery.k8s.io",
  "/apis/discovery.k8s.io/v1",
  "/apis/discovery.k8s.io/v1beta1",
  "/apis/events.k8s.io",
  "/apis/events.k8s.io/v1",
  "/apis/events.k8s.io/v1beta1",
  "/apis/flowcontrol.apiserver.k8s.io",
  "/apis/flowcontrol.apiserver.k8s.io/v1beta1",
  "/apis/networking.k8s.io",
  "/apis/networking.k8s.io/v1",
  "/apis/node.k8s.io",
  "/apis/node.k8s.io/v1",

```

Conclusion: Kubectl was successfully installed and an application was successfully deployed using Kubernetes.