# Working with minikube

## Install minikube

Simply execute minikube-installer.exe, follow the dialog prompts, ensure to install minikube in C:\chaosengineering\Kubernetes\Minikube.

NOTE: This exe is available in this directory. Else you can download latest version from [here](https://minikube.sigs.k8s.io/docs/start/)

## Check minikube installation

Open command window

Type “minikube” ad hit enter

Output should be as shown below.

|  |
| --- |
| minikube provisions and manages local Kubernetes clusters optimized for development workflows.  **Basic Commands:**  start Starts a local Kubernetes cluster  status Gets the status of a local Kubernetes cluster  stop Stops a running local Kubernetes cluster  delete Deletes a local Kubernetes cluster  dashboard Access the Kubernetes dashboard running within the minikube cluster  pause pause Kubernetes  unpause unpause Kubernetes  **Images Commands:**  docker-env Provides instructions to point your terminal's docker-cli to the Docker Engine inside minikube. (Useful for building docker images directly inside minikube)  podman-env Configure environment to use minikube's Podman service  cache Manage cache for images  image Manage images  **Configuration and Management Commands:**  addons Enable or disable a minikube addon  config Modify persistent configuration values  profile Get or list the current profiles (clusters)  update-context Update kubeconfig in case of an IP or port change  **Networking and Connectivity Commands:**  service Returns a URL to connect to a service  tunnel Connect to LoadBalancer services  **Advanced Commands:**  mount Mounts the specified directory into minikube  ssh Log into the minikube environment (for debugging)  kubectl Run a kubectl binary matching the cluster version  node Add, remove, or list additional nodes  cp Copy the specified file into minikube  **Troubleshooting Commands:**  ssh-key Retrieve the ssh identity key path of the specified node  ssh-host Retrieve the ssh host key of the specified node  ip Retrieves the IP address of the specified node  logs Returns logs to debug a local Kubernetes cluster  update-check Print current and latest version number  version Print the version of minikube  options Show a list of global command-line options (applies to all commands).  **Other Commands:**  completion Generate command completion for a shell  license Outputs the licenses of dependencies to a directory  Use "minikube <command> --help" for more information about a given command. |

Now minikube is installed on the machine now we should start minikube and create images and containers in docker. To do that.

## Start minikube

Open command window

Type “minikube start”

The output should be as shown below.

|  |
| --- |
| \* minikube v1.30.1 on Microsoft Windows 10 Pro 10.0.19045.3208 Build 19045.3208  \* minikube 1.31.1 is available! Download it: [from here](https://github.com/kubernetes/minikube/releases/tag/v1.31.1)  \* To disable this notice, run: 'minikube config set WantUpdateNotification false'  \* Automatically selected the docker driver. Other choices: virtualbox, ssh  \* Using Docker Desktop driver with root privileges  \* Starting control plane node minikube in cluster minikube  \* Pulling base image ...  \* Creating docker container (CPUs=2, Memory=4000MB) ...  \* Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...  - Generating certificates and keys ...  - Booting up control plane ...  - Configuring RBAC rules ...  \* Configuring bridge CNI (Container Networking Interface) ...  \* Verifying Kubernetes components...  - Using image gcr.io/k8s-minikube/storage-provisioner:v5  \* Enabled addons: storage-provisioner, default-storageclass  \* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default |

* In the output “Automatically selected the docker driver. Other choices: virtualbox, ssh”, means minikube by default works with docker driver, you must have docker desktop installed to work with ninikube, there are other alternatives also provided in the documentation. [For more information visit this page](https://minikube.sigs.k8s.io/docs/drivers/).
* Also note that, default addons needed are also installed and configured.

Now minikube is started and in docker container you should get one image and one container t view the containers and images, lets learn few docker commands and minikube commands.

## Docker commands

### Get List of docker command

Open command window

Type “docker --help” and hit enter

The output should something like below.

|  |
| --- |
| Usage: docker [OPTIONS] COMMAND  A self-sufficient runtime for containers  **Common Commands:**  run Create and run a new container from an image  exec Execute a command in a running container  ps List containers  build Build an image from a Dockerfile  pull Download an image from a registry  push Upload an image to a registry  images List images  login Log in to a registry  logout Log out from a registry  search Search Docker Hub for images  version Show the Docker version information  info Display system-wide information  **Management Commands:**  builder Manage builds  buildx\* Docker Buildx (Docker Inc., v0.11.0)  compose\* Docker Compose (Docker Inc., v2.19.0)  container Manage containers  context Manage contexts  dev\* Docker Dev Environments (Docker Inc., v0.1.0)  extension\* Manages Docker extensions (Docker Inc., v0.2.20)  image Manage images  init\* Creates Docker-related starter files for your project (Docker Inc., v0.1.0-beta.6)  manifest Manage Docker image manifests and manifest lists  network Manage networks  plugin Manage plugins  sbom\* View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc., 0.6.0)  scan\* Docker Scan (Docker Inc., v0.26.0)  scout\* Command line tool for Docker Scout (Docker Inc., 0.16.1)  system Manage Docker  trust Manage trust on Docker images  volume Manage volumes  **Swarm Commands:**  swarm Manage Swarm  **Commands:**  attach Attach local standard input, output, and error streams to a running container  commit Create a new image from a container's changes  cp Copy files/folders between a container and the local filesystem  create Create a new container  diff Inspect changes to files or directories on a container's filesystem  events Get real time events from the server  export Export a container's filesystem as a tar archive  history Show the history of an image  import Import the contents from a tarball to create a filesystem image  inspect Return low-level information on Docker objects  kill Kill one or more running containers  load Load an image from a tar archive or STDIN  logs Fetch the logs of a container  pause Pause all processes within one or more containers  port List port mappings or a specific mapping for the container  rename Rename a container  restart Restart one or more containers  rm Remove one or more containers  rmi Remove one or more images  save Save one or more images to a tar archive (streamed to STDOUT by default)  start Start one or more stopped containers  stats Display a live stream of container(s) resource usage statistics  stop Stop one or more running containers  tag Create a tag TARGET\_IMAGE that refers to SOURCE\_IMAGE  top Display the running processes of a container  unpause Unpause all processes within one or more containers  update Update configuration of one or more containers  wait Block until one or more containers stop, then print their exit codes  **Global Options:**  --config string Location of client config files (default  "C:\\Users\\E001150\\.docker")  -c, --context string Name of the context to use to connect to the  daemon (overrides DOCKER\_HOST env var and  default context set with "docker context use")  -D, --debug Enable debug mode  -H, --host list Daemon socket to connect to  -l, --log-level string Set the logging level ("debug", "info",  "warn", "error", "fatal") (default "info")  --tls Use TLS; implied by --tlsverify  --tlscacert string Trust certs signed only by this CA (default  "C:\\Users\\E001150\\.docker\\ca.pem")  --tlscert string Path to TLS certificate file (default  "C:\\Users\\E001150\\.docker\\cert.pem")  --tlskey string Path to TLS key file (default  "C:\\Users\\E001150\\.docker\\key.pem")  --tlsverify Use TLS and verify the remote  -v, --version Print version information and quit  Run 'docker COMMAND --help' for more information on a command.  For more help on how to use Docker, head to https://docs.docker.com/go/guides/ |

This command displays list all command options available for docker command line utility.

### Get List of Containers

Open command window

Type “docker ps” and hit enter

The output should something like below.

|  |
| --- |
| CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  9c742713ae53 gcr.io/k8s-minikube/kicbase:v0.0.39 "/usr/local/bin/entr…" 13 minutes ago Up 13 minutes 127.0.0.1:51625->22/tcp, 127.0.0.1:51626->2376/tcp, 127.0.0.1:51623->5000/tcp, 127.0.0.1:51624->8443/tcp, 127.0.0.1:51622->32443/tcp minikube |

This command displays the containers running at that moment in time,

### Get List of Images

Open command window

Type “docker image ls” and hit enter

The output should something like below.

|  |
| --- |
| REPOSITORY TAG IMAGE ID CREATED SIZE  gcr.io/k8s-minikube/kicbase v0.0.39 67a4b1138d2d 3 months ago 1.05GB |

This command displays the images available at that moment in time,

## minikube commands

### Using Kubectl commands

Open command window

Type “kubectl get pods -A”

The output should be as shown below.

|  |
| --- |
| NAMESPACE NAME READY STATUS RESTARTS AGE  kube-system coredns-787d4945fb-g7hp9 1/1 Running 1 (16m ago) 21m  kube-system etcd-minikube 1/1 Running 2 (16m ago) 21m  kube-system kube-apiserver-minikube 1/1 Running 2 (15m ago) 21m  kube-system kube-controller-manager-minikube 1/1 Running 3 (15m ago) 21m  kube-system kube-proxy-pggx8 1/1 Running 2 (16m ago) 21m  kube-system kube-scheduler-minikube 0/1 CrashLoopBackOff 8 (6s ago) 21m  kube-system storage-provisioner 1/1 Running 2 (16m ago) 21m |

This command listsout all the Pods available and their status along with Age and number of restarts

### Using minikube Kubectl commands

Open command window

Type “minikube kubectl -- get pods -A”

The output should be as shown below.

|  |
| --- |
| NAMESPACE NAME READY STATUS RESTARTS AGE  kube-system coredns-787d4945fb-g7hp9 1/1 Running 1 (16m ago) 21m  kube-system etcd-minikube 1/1 Running 2 (16m ago) 21m  kube-system kube-apiserver-minikube 1/1 Running 2 (15m ago) 21m  kube-system kube-controller-manager-minikube 1/1 Running 3 (15m ago) 21m  kube-system kube-proxy-pggx8 1/1 Running 2 (16m ago) 21m  kube-system kube-scheduler-minikube 0/1 CrashLoopBackOff 8 (6s ago) 21m  kube-system storage-provisioner 1/1 Running 2 (16m ago) 21m |

This command lists out all the Pods available and their status along with Age and number of restarts

### Enabling minikube Dashboard

For additional insight into your cluster state, minikube bundles the Kubernetes Dashboard, allowing you to get easily acclimated to your new environment:

Open command window

Type “minikube dashboard”

The output should be as shown below.

|  |
| --- |
| \* Enabling dashboard ...  - Using image docker.io/kubernetesui/dashboard:v2.7.0  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8  \* Some dashboard features require the metrics-server addon. To enable all features please run:  **minikube addons enable metrics-server**  \* Verifying dashboard health ...  \* Launching proxy ...  \* Verifying proxy health ...  \* Opening http://127.0.0.1:57064/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser... |

This command lists out all the Pods available and their status along with Age and number of restarts.

### Enabling metrics server

As suggested above, as metrics service is not enabled , lets try to enable metrics server using the following command.

Open command window

Type “minikube addons enable metrics-server”

The output should be as shown below.

|  |
| --- |
| \* metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.  You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS  - Using image registry.k8s.io/metrics-server/metrics-server:v0.6.3  \* The 'metrics-server' addon is enabled |

Observer that the metrics server is enabled.

## Deploy an application using minikube

Lets assume our application has 3 main parts to it.

1. Service
2. Load balancer
3. Ingress

### Deploying Service application

Create a sample deployment and expose it on port 8080:

kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0

output:

|  |
| --- |
| deployment.apps/hello-minikube created |

kubectl expose deployment hello-minikube --type=NodePort --port=8080

output:

|  |
| --- |
| service/hello-minikube exposed |

To see if these services are deployed properly or not.

minikube service list

|  |
| --- |
| |----------------------|---------------------------|--------------|-----|  | NAMESPACE | NAME | TARGET PORT | URL |  |----------------------|---------------------------|--------------|-----|  | default | hello-minikube | 8080 | |  | default | kubernetes | No node port | |  | kube-system | kube-dns | No node port | |  | kube-system | metrics-server | No node port | |  | kubernetes-dashboard | dashboard-metrics-scraper | No node port | |  | kubernetes-dashboard | kubernetes-dashboard | No node port | |  |----------------------|---------------------------|--------------|-----| |

It may take a moment, but your deployment will soon show up when you run:

kubectl get services hello-minikube

|  |
| --- |
| NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  hello-minikube NodePort 10.105.177.88 <none> 8080:31878/TCP 4m4s |

The easiest way to access this service is to let minikube launch a web browser for you:

minikube service hello-minikube

|  |
| --- |
| |-----------|----------------|-------------|---------------------------|  | NAMESPACE | NAME | TARGET PORT | URL |  |-----------|----------------|-------------|---------------------------|  | default | hello-minikube | 8080 | http://192.168.49.2:31878 |  |-----------|----------------|-------------|---------------------------|  \* Starting tunnel for service hello-minikube.  |-----------|----------------|-------------|------------------------|  | NAMESPACE | NAME | TARGET PORT | URL |  |-----------|----------------|-------------|------------------------|  | default | hello-minikube | | http://127.0.0.1:52672 |  |-----------|----------------|-------------|------------------------|  \* Opening service default/hello-minikube in default browser...  ! Because you are using a Docker driver on windows, the terminal needs to be open to run it. |

NOTE: as long as command window is open, the service will be up and running, to close the service you must press Ctrl+C.

Browser automatically opens navigates to the above URL and you would see something similar to below.

A screenshot of a computer

Description automatically generated

Alternatively, use kubectl to forward the port:

kubectl port-forward service/hello-minikube 7080:8080

|  |
| --- |
| Forwarding from 127.0.0.1:7080 -> 8080  Forwarding from [::1]:7080 -> 8080 |

NOTE: as long as command window is open, the service will be up and running, to close the service you must press Ctrl+C.

Tada! Your application is now available at <http://localhost:7080/>.

A screenshot of a computer

Description automatically generated

You should be able to see the request metadata in the application output. Try changing the path of the request and observe the changes. Similarly, you can do a POST request and observe the body show up in the output.

## Manage your cluster

Pause Kubernetes without impacting deployed applications:

minikube pause

Unpause a paused instance:

minikube unpause

Halt the cluster:

minikube stop

Change the default memory limit (requires a restart):

minikube config set memory 9001

Browse the catalog of easily installed Kubernetes services:

minikube addons list

Create a second cluster running an older Kubernetes release:

minikube start -p aged --kubernetes-version=v1.16.1

Delete all of the minikube clusters:

minikube delete –all

**C:\Users\E001150\Downloads\minikube>minikube delete --all**

\* Deleting "minikube" in docker ...

\* Removing C:\Users\E001150\.minikube\machines\minikube ...

\* Removing C:\Users\E001150\.minikube\machines\minikube-m02 ...

\* Removed all traces of the "minikube" cluster.

\* Successfully deleted all profiles

**C:\Users\E001150\Downloads\minikube>minikube delete --all --purge**

\* Successfully deleted all profiles

\* Successfully purged minikube directory located at - [C:\Users\E001150\.minikube]

\* Kicbase images have not been deleted. To delete images run:

- docker rmi gcr.io/k8s-minikube/kicbase:v0.0.39

## Some useful minikube commands

minikube profile list

|----------|-----------|---------|--------------|------|---------|---------|-------|--------|

| Profile | VM Driver | Runtime | IP | Port | Version | Status | Nodes | Active |

|----------|-----------|---------|--------------|------|---------|---------|-------|--------|

| minikube | docker | docker | 192.168.49.2 | 8443 | v1.26.3 | Running | 1 | \* |

|----------|-----------|---------|--------------|------|---------|---------|-------|--------|

Note number of nodes is shown as 1 here.

**minikube status**

minikube

type: Control Plane

host: Running

kubelet: Running

apiserver: Running

kubeconfig: Configured

## Add a Node to existing cluster

minikube node add

|  |
| --- |
| \* Adding node m02 to cluster minikube  ! Cluster was created without any CNI, adding a node to it might cause broken networking.  \* Starting worker node minikube-m02 in cluster minikube  \* Pulling base image ...  \* Creating docker container (CPUs=2, Memory=2200MB) ...  \* Preparing Kubernetes v1.26.3 on Docker 23.0.2 ...  \* Verifying Kubernetes components...  \* Successfully added m02 to minikube! |

To make sure new node is added to existing cluster, type following command and observe the number of nodes is shown as 2.

C:\Users\E001150\Downloads\minikube>minikube profile list

|----------|-----------|---------|--------------|------|---------|---------|-------|--------|

| Profile | VM Driver | Runtime | IP | Port | Version | Status | **Nodes** | Active |

|----------|-----------|---------|--------------|------|---------|---------|-------|--------|

| minikube | docker | docker | 192.168.49.2 | 8443 | v1.26.3 | Running | **2** | \* |

|----------|-----------|---------|--------------|------|---------|---------|-------|--------|