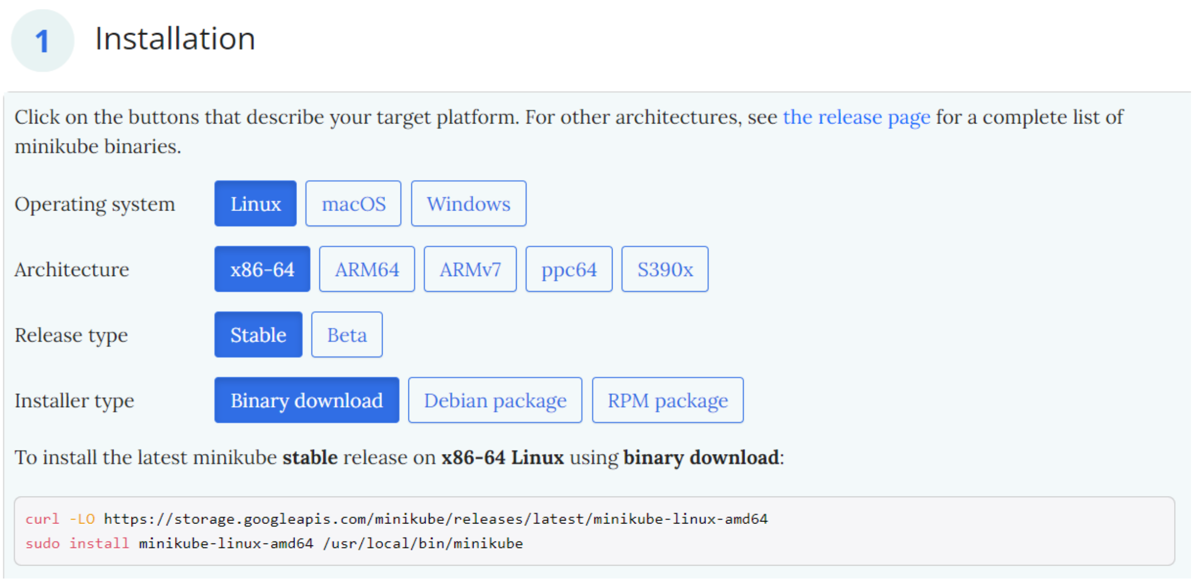
Containerizing an application with docker

Installing and Testing Docker:

- To install minikube, follow the instructions for your operating system at https://minikube.sigs.k8s.io/



1 Installation

Click on the buttons that describe your target platform. For other architectures, see the release page for a complete list of minikube binaries.

Operating system: [✓ Linux] macOS Windows

Architecture: [✓ x86-64] ARM64 ARMv7 ppc64 S390x

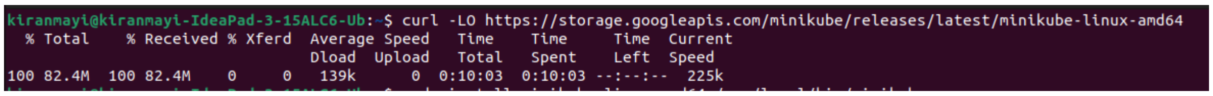
Release type: [✓ Stable] Beta

Installer type: [✓ Binary download] Debian package RPM package

To install the latest minikube **stable** release on **x86-64 Linux** using **binary download**:

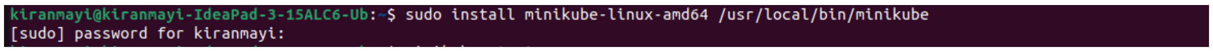
```
curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

Select like this in the link and copy the given link in to your terminal and run .



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 82.4M  100 82.4M    0     0   139k      0  0:10:03  0:10:03 --:--:--  225k
```

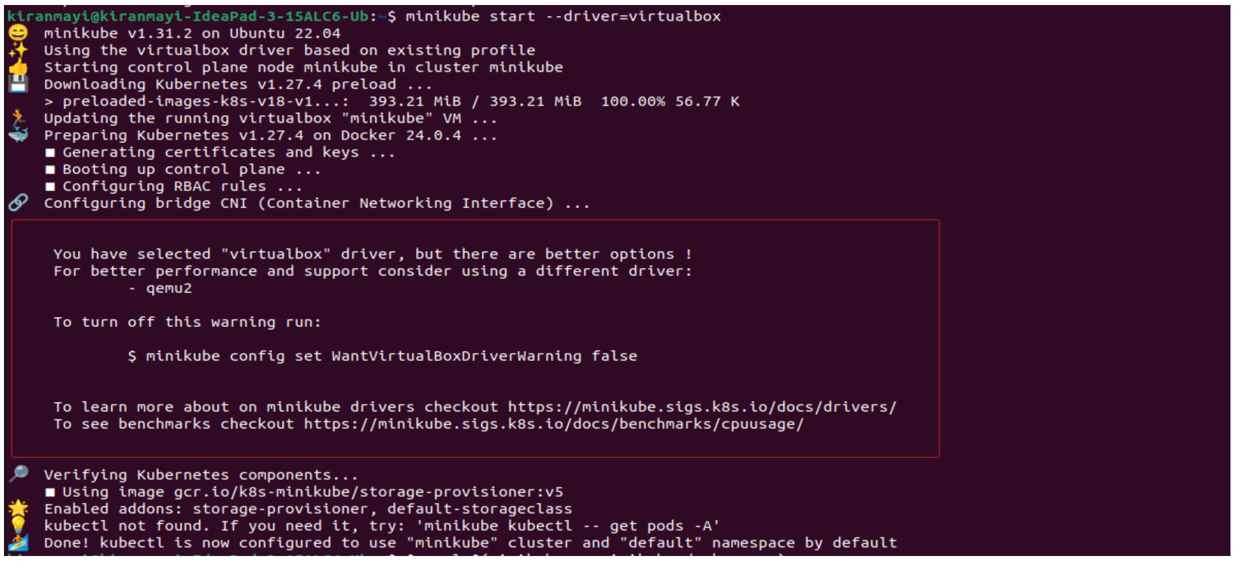
Next install the minikube by using sudo



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
[sudo] password for kiranmayi:
```

To start minikube using the resource defaults and VirtualBox as the VM manager, enter the following in a terminal:

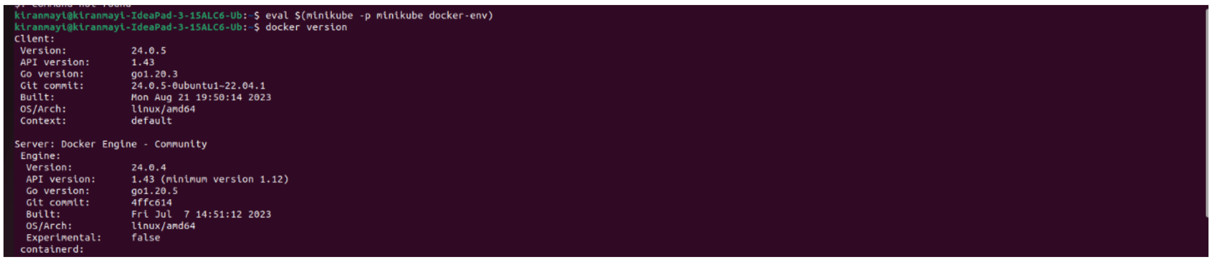
minikube start --driver=virtualbox



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~$ minikube start --driver=virtualbox
    minikube v1.31.2 on Ubuntu 22.04
    Using the virtualbox driver based on existing profile
    Starting control plane node minikube in cluster minikube
    Downloading Kubernetes v1.27.4 preload ...
    > preloaded-images-k8s-v18-v1...:  393.21 MiB / 393.21 MiB  100.00% 56.77 K
    Updating the running virtualbox "minikube" VM ...
    Preparing Kubernetes v1.27.4 on Docker 24.0.4 ...
    ■ Generating certificates and keys ...
    ■ Booting up control plane ...
    ■ Configuring RBAC rules ...
    Configuring bridge CNI (Container Networking Interface) ...

      You have selected "virtualbox" driver, but there are better options !
      For better performance and support consider using a different driver:
              - qemu2

      To turn off this warning run:

              $ minikube config set WantVirtualBoxDriverWarning false


      To learn more about on minikube drivers checkout https://minikube.sigs.k8s.io/docs/drivers/
      To see benchmarks checkout https://minikube.sigs.k8s.io/docs/benchmarks/cpuusage/

    Verifying Kubernetes components...
    ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
    Enabled addons: storage-provisioner, default-storageclass
    kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
    Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

In a terminal, enter the following to set your Docker environment variables:

eval $(minikube -p minikube docker-env)



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~$ eval $(minikube -p minikube docker-env)
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~$ docker version
Client:
 Version:           24.0.5
 API version:       1.43
 Go version:        go1.20.3
 Git commit:        24.0.5-0ubuntu1~22.04.1
 Built:             Mon Aug 21 19:50:14 2023
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:          24.0.4
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.20.5
  Git commit:       4ffc614
  Built:            Fri Jul  7 14:51:12 2023
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
```

You can test the installation of docker by typing docker version in your terminal.

Containerizing a Sample Application:

To containerize an application, you’ll need the source code or binary you want to run in the container plus the Dockerfile to build the container image. The sample application source code and Dockerfile are in the companion repository for this book at https://github.com/bradleyd/devops_for_the _desperate/ in the telnet-server/ folder.

Navigate to the telnet-server/ directory and open the Dockerfile, which should look like this:

```
# Build stage
FROM golang:alpine AS build-env
ADD . /
RUN cd / && go build -o telnet-server

# Final stage
FROM alpine:latest AS final
WORKDIR /app
ENV TELNET_PORT 2323
ENV METRIC_PORT 9000
COPY -from=build-env /telnet-server /app/
ENTRYPOINT ["./telnet-server"]
```

Building the Container Image:

Before building the container image you need test for the permissions.

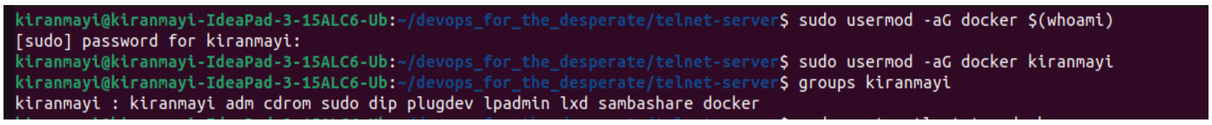
- **Add User to Docker Group:** You can add your user to the `docker` group, which allows it to run Docker commands without using `sudo`. Enter the following command

  sudo usermod -aG docker $(whoami)
- To allow the user `kiranmayi` to use Docker without using `sudo`, you should add the `kiranmayi` user to the `docker` group. Here's how you can do that:

  sudo usermod -aG docker kiranmayi
- Once the user `kiranmayi` is added to the `docker` group, they should be able to run Docker commands without using `sudo`. Ensure you're in the `docker` group by running:

  groups kiranmayi

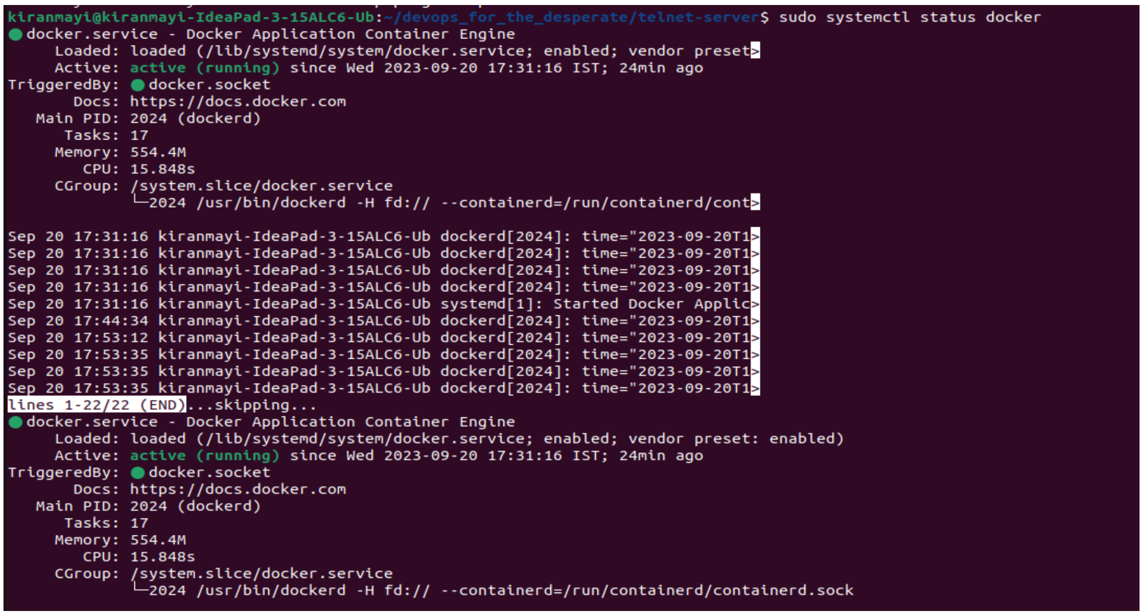
After running this command, you'll need to log out and log back in (or restart your system) for the changes to take effect.



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ sudo usermod -aG docker $(whoami)
[sudo] password for kiranmayi:
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ sudo usermod -aG docker kiranmayi
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ groups kiranmayi
kiranmayi : kiranmayi adm cdrom sudo dip plugdev lpadmin lxd sambashare docker
```

To check whether docker is running or not use the following command:

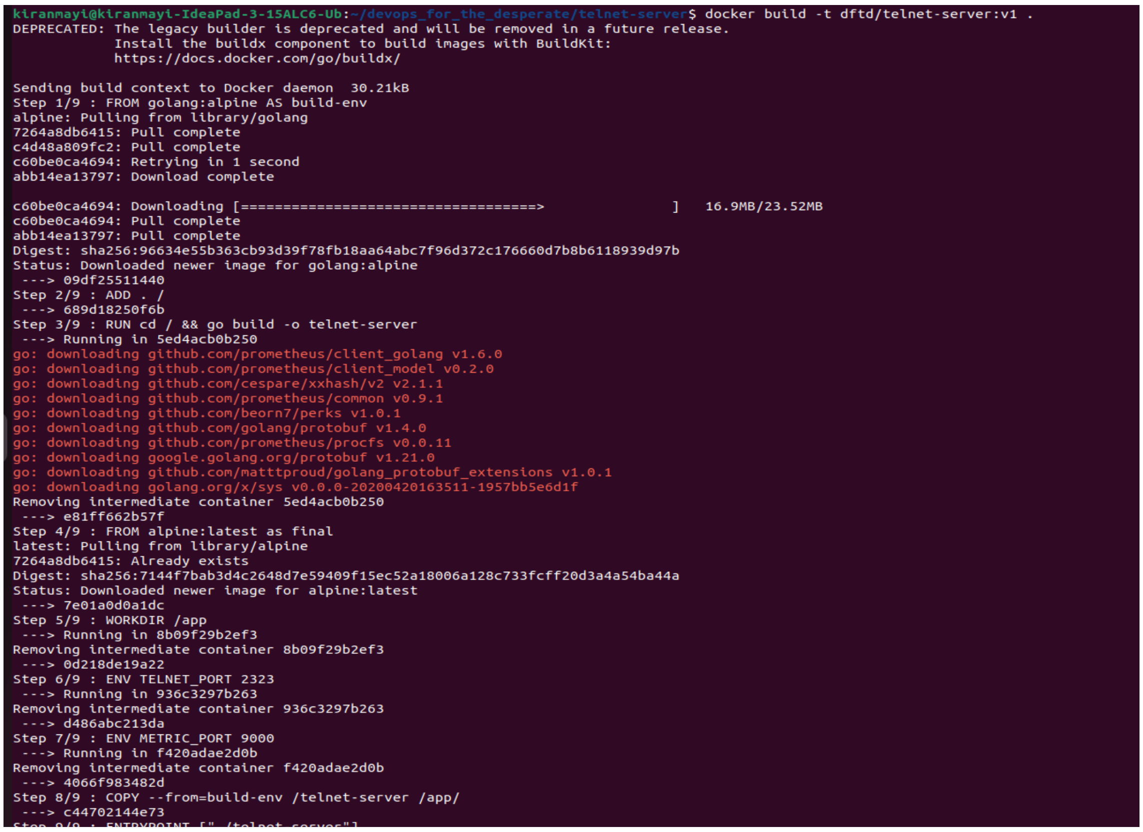
sudo systemctl status docker



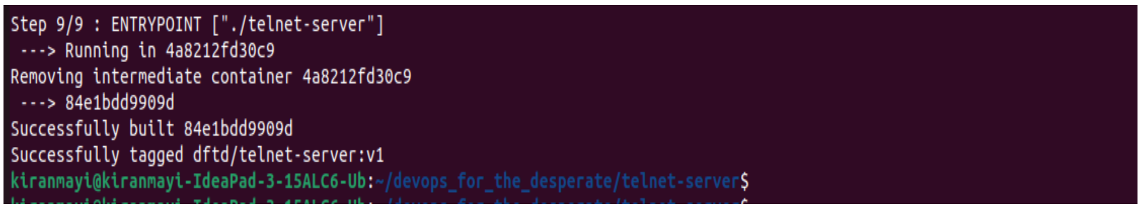
```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset>
     Active: active (running) since Wed 2023-09-20 17:31:16 IST; 24min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2024 (dockerd)
      Tasks: 17
     Memory: 554.4M
        CPU: 15.848s
     CGroup: /system.slice/docker.service
             └─2024 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

Sep 20 17:31:16 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
Sep 20 17:31:16 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
Sep 20 17:31:16 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
Sep 20 17:31:16 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
Sep 20 17:31:16 kiranmayi-IdeaPad-3-15ALC6-Ub systemd[1]: Started Docker Applic>
Sep 20 17:44:34 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
Sep 20 17:53:12 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
Sep 20 17:53:35 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
Sep 20 17:53:35 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
Sep 20 17:53:35 kiranmayi-IdeaPad-3-15ALC6-Ub dockerd[2024]: time="2023-09-20T1>
lines 1-22/22 (END)...skipping...
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
     Active: active (running) since Wed 2023-09-20 17:31:16 IST; 24min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2024 (dockerd)
      Tasks: 17
     Memory: 554.4M
        CPU: 15.848s
     CGroup: /system.slice/docker.service
             └─2024 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

Navigate to the telnet-server/ directory and enter the following to pass Docker the image name and Dockerfile location:

docker build -t dftd/telnet-server:v1



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker build -t dftd/telnet-server:v1 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  30.21kB
Step 1/9 : FROM golang:alpine AS build-env
alpine: Pulling from library/golang
7264a8db6415: Pull complete
c4d48a809fc2: Pull complete
c60be0ca4694: Retrying in 1 second
abb14ea13797: Download complete

c60be0ca4694: Downloading [====================================>               ]   16.9MB/23.52MB
c60be0ca4694: Pull complete
abb14ea13797: Pull complete
Digest: sha256:96634e55b363cb93d39f78fb18aa64abc7f96d372c176660d7b8b6118939d97b
Status: Downloaded newer image for golang:alpine
 ---> 09df25511440
Step 2/9 : ADD . /
 ---> 689d18250f6b
Step 3/9 : RUN cd / && go build -o telnet-server
 ---> Running in 5ed4acb0b250
go: downloading github.com/prometheus/client_golang v1.6.0
go: downloading github.com/prometheus/client_model v0.2.0
go: downloading github.com/cespare/xxhash/v2 v2.1.1
go: downloading github.com/prometheus/common v0.9.1
go: downloading github.com/beorn7/perks v1.0.1
go: downloading github.com/golang/protobuf v1.4.0
go: downloading github.com/prometheus/procfs v0.0.11
go: downloading google.golang.org/protobuf v1.21.0
go: downloading github.com/matttproud/golang_protobuf_extensions v1.0.1
go: downloading golang.org/x/sys v0.0.0-20200420163511-1957bb5e6d1f
Removing intermediate container 5ed4acb0b250
 ---> e81ff662b57f
Step 4/9 : FROM alpine:latest as final
latest: Pulling from library/alpine
7264a8db6415: Already exists
Digest: sha256:7144f7bab3d4c2648d7e59409f15ec52a18006a128c733fcff20d3a4a54ba44a
Status: Downloaded newer image for alpine:latest
 ---> 7e01a0d0a1dc
Step 5/9 : WORKDIR /app
 ---> Running in 8b09f29b2ef3
Removing intermediate container 8b09f29b2ef3
 ---> 0d218de19a22
Step 6/9 : ENV TELNET_PORT 2323
 ---> Running in 936c3297b263
Removing intermediate container 936c3297b263
 ---> d486abc213da
Step 7/9 : ENV METRIC_PORT 9000
 ---> Running in f420adae2d0b
Removing intermediate container f420adae2d0b
 ---> 4066f983482d
Step 8/9 : COPY --from=build-env /telnet-server /app/
 ---> c44702144e73
```

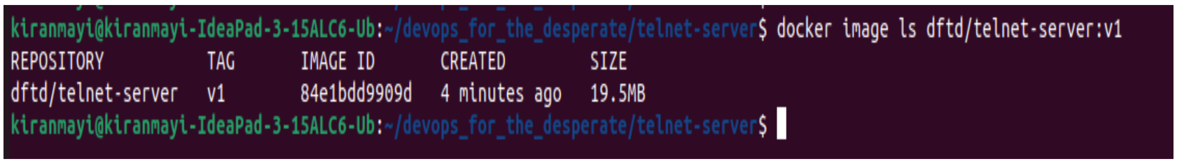


```
Step 9/9 : ENTRYPOINT ["./telnet-server"]
 ---> Running in 4a8212fd30c9
Removing intermediate container 4a8212fd30c9
 ---> 84e1bdd9909d
Successfully built 84e1bdd9909d
Successfully tagged dftd/telnet-server:v1
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

Verifying the Docker Image:

Next, verify that the Docker registry inside minikube is storing the telnetserver image. In a terminal, enter the following to list the Docker telnet-server image:

docker image ls dftd/telnet-server:v1

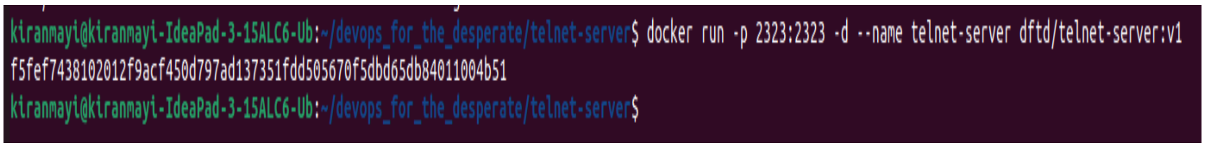


```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker image ls dftd/telnet-server:v1
REPOSITORY          TAG       IMAGE ID       CREATED         SIZE
dftd/telnet-server  v1        84e1bdd9909d   4 minutes ago   19.5MB
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

Running the Container :

The next step is to create and run the telnet-server container from the image you just built. Do this by entering the following:

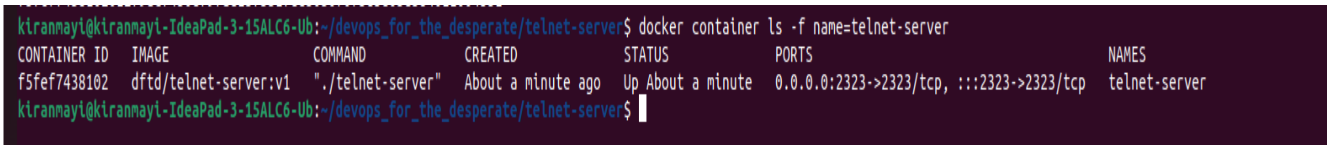
docker run -p 2323:2323 -d --name telnet-server dftd/telnet-server:v1



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker run -p 2323:2323 -d --name telnet-server dftd/telnet-server:v1
f5fef7438102012f9acf450d797ad137351fdd505670f5dbd65db84011004b51
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

Enter the following to verify that the container is actually running:

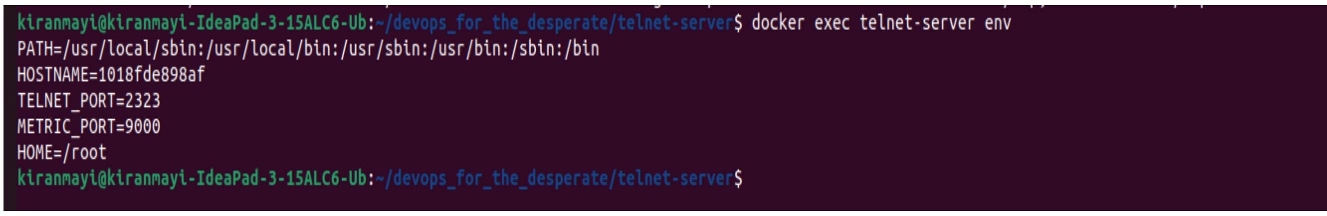
docker container ls -f name=telnet-server



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker container ls -f name=telnet-server
CONTAINER ID   IMAGE                   COMMAND             CREATED            STATUS            PORTS                                         NAMES
f5fef7438102   dftd/telnet-server:v1   "./telnet-server"   About a minute ago   Up About a minute   0.0.0.0:2323->2323/tcp, :::2323->2323/tcp   telnet-server
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

Other Docker Client Commands:

- exec:

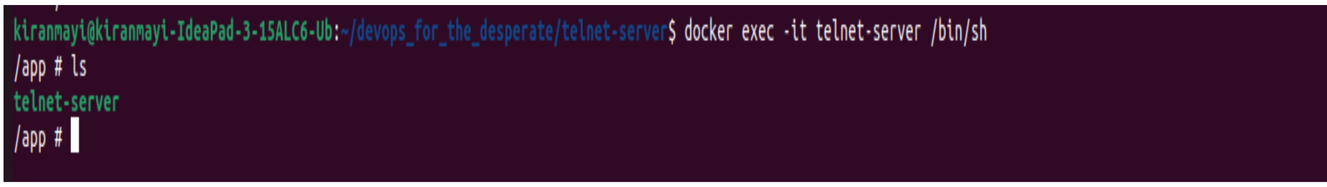


```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker exec telnet-server env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=1018fde898af
TELNET_PORT=2323
METRIC_PORT=9000
HOME=/root
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

Enter the following in a terminal to get a shell inside the container:

docker exec -it telnet-server /bin/sh

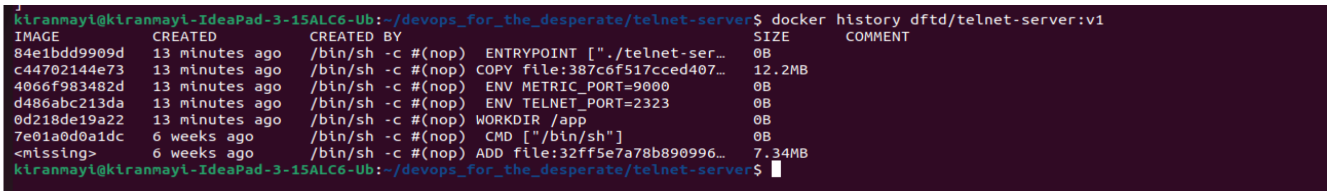
The ls command is issued to show you're inside the container you built. Input the exit command and press ENTER to leave the container and return to the local terminal.



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker exec -it telnet-server /bin/sh
/app # ls
telnet-server
/app #
```

- history:

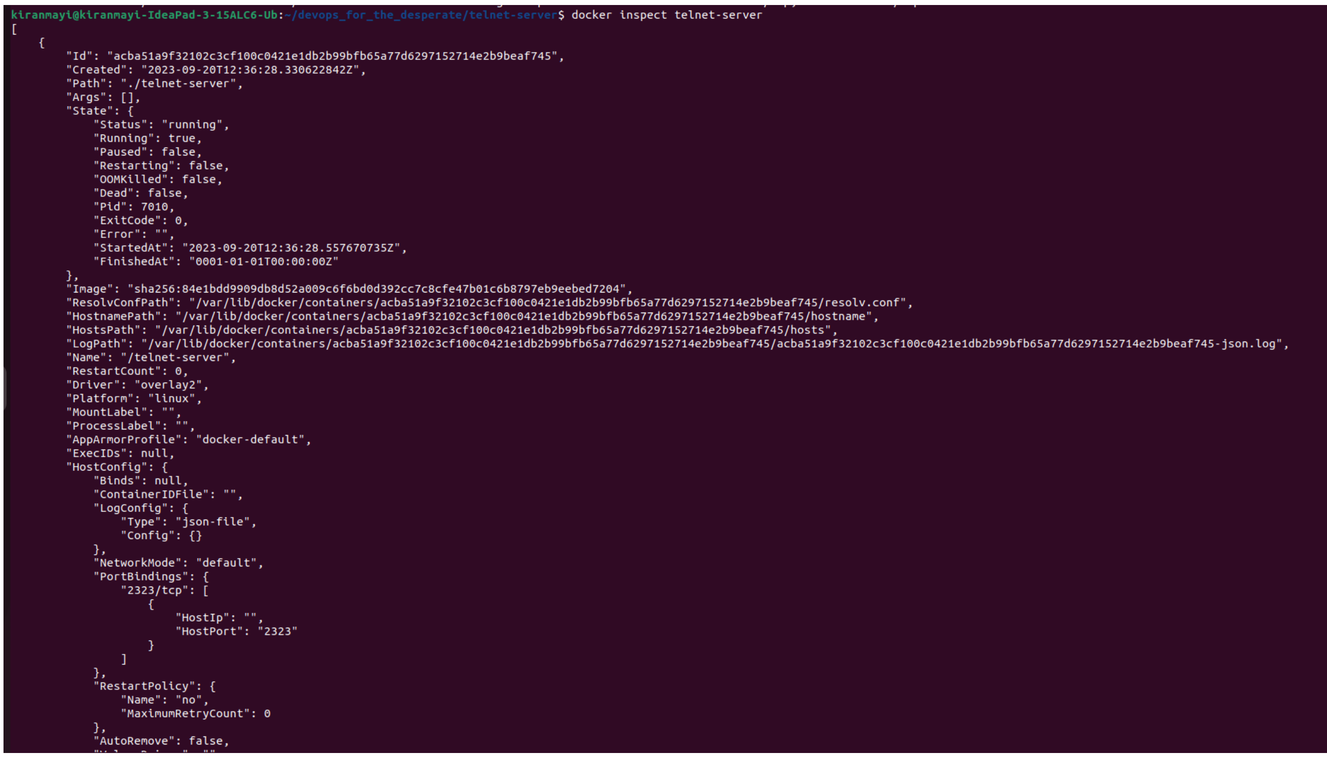
docker history dftd/telnet-server:v1



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker history dftd/telnet-server:v1
IMAGE          CREATED          CREATED BY                                      SIZE      COMMENT
84e1bdd9909d   13 minutes ago   /bin/sh -c #(nop)  ENTRYPOINT ["./telnet-ser…   0B
c44702144e73   13 minutes ago   /bin/sh -c #(nop) COPY file:387c6f517cced407…   12.2MB
4066f983482d   13 minutes ago   /bin/sh -c #(nop)  ENV METRIC_PORT=9000         0B
d486abc213da   13 minutes ago   /bin/sh -c #(nop)  ENV TELNET_PORT=2323         0B
0d218de19a22   13 minutes ago   /bin/sh -c #(nop) WORKDIR /app                  0B
7e01a0d0a1dc   6 weeks ago      /bin/sh -c #(nop)  CMD ["/bin/sh"]              0B
<missing>      6 weeks ago      /bin/sh -c #(nop) ADD file:32ff5e7a78b890996…   7.34MB
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

- inspect:

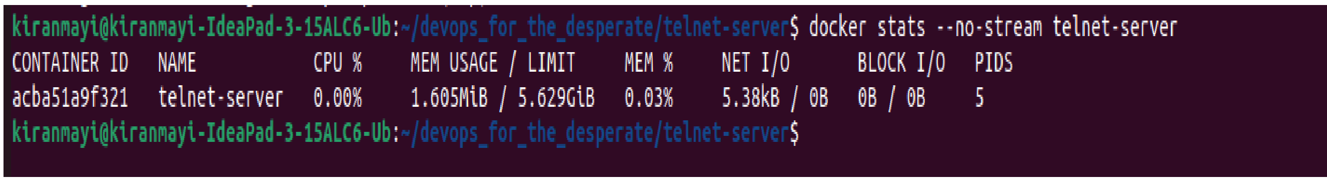
docker inspect telnet-server



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker inspect telnet-server
[
    {
        "Id": "acba51a9f32102c3cf100c0421e1db2b99bfb65a77d6297152714e2b9beaf745",
        "Created": "2023-09-20T12:36:28.330622842Z",
        "Path": "./telnet-server",
        "Args": [],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 7010,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2023-09-20T12:36:28.557670735Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:84e1bdd9909db8d52a009c6f6bd0d392cc7c8cfe47b01c6b8797eb9eebed7204",
        "ResolvConfPath": "/var/lib/docker/containers/acba51a9f32102c3cf100c0421e1db2b99bfb65a77d6297152714e2b9beaf745/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/acba51a9f32102c3cf100c0421e1db2b99bfb65a77d6297152714e2b9beaf745/hostname",
        "HostsPath": "/var/lib/docker/containers/acba51a9f32102c3cf100c0421e1db2b99bfb65a77d6297152714e2b9beaf745/hosts",
        "LogPath": "/var/lib/docker/containers/acba51a9f32102c3cf100c0421e1db2b99bfb65a77d6297152714e2b9beaf745/acba51a9f32102c3cf100c0421e1db2b99bfb65a77d6297152714e2b9beaf745-json.log",
        "Name": "/telnet-server",
        "RestartCount": 0,
        "Driver": "overlay2",
        "Platform": "linux",
        "MountLabel": "",
        "ProcessLabel": "",
        "AppArmorProfile": "docker-default",
        "ExecIDs": null,
        "HostConfig": {
            "Binds": null,
            "ContainerIDFile": "",
            "LogConfig": {
                "Type": "json-file",
                "Config": {}
            },
            "NetworkMode": "default",
            "PortBindings": {
                "2323/tcp": [
                    {
                        "HostIp": "",
                        "HostPort": "2323"
                    }
                ]
            },
            "RestartPolicy": {
                "Name": "no",
                "MaximumRetryCount": 0
            },
            "AutoRemove": false,
```

- stats:

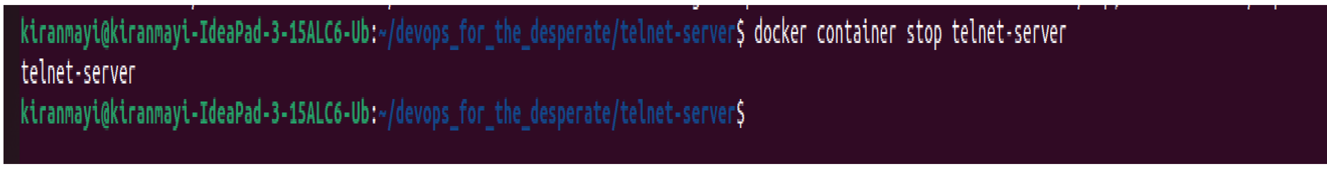
docker stats --no-stream telnet-server



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker stats --no-stream telnet-server
CONTAINER ID   NAME            CPU %     MEM USAGE / LIMIT     MEM %     NET I/O        BLOCK I/O   PIDS
acba51a9f321   telnet-server   0.00%     1.605MiB / 5.629GiB   0.03%     5.38kB / 0B    0B / 0B     5
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

- stop:

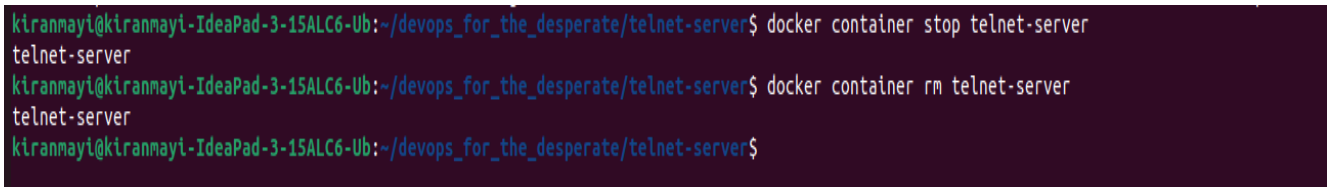
docker container stop telnet-server



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker container stop telnet-server
telnet-server
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

- rm:

docker container rm telnet-server



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker container stop telnet-server
telnet-server
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker container rm telnet-server
telnet-server
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

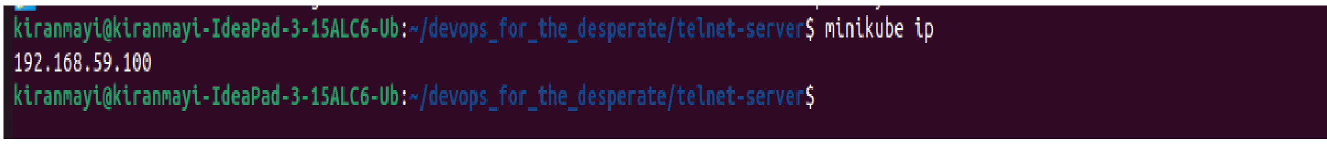
Testing the Container:

To find out whether the sample application you've containerized actually works, you'll connect to the telnet-server on port 2323 and run some basic commands. Then you'll view the container logs to verify that the applica�tion is working correctly.

Connecting to the Telnet-Server:

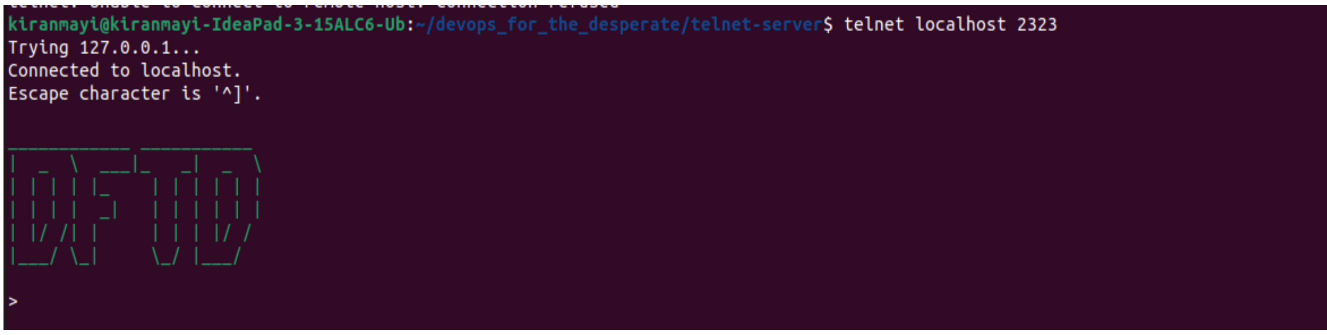
To connect to the server, pass telnet the hostname or IP address of the server plus the port to which you want to connect. Since the Docker server is running inside a VM (minikube), you'll need the IP address minikube exposes to your local host. Enter the following in a terminal to get the IP address:

minikube ip



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ minikube ip
192.168.59.100
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

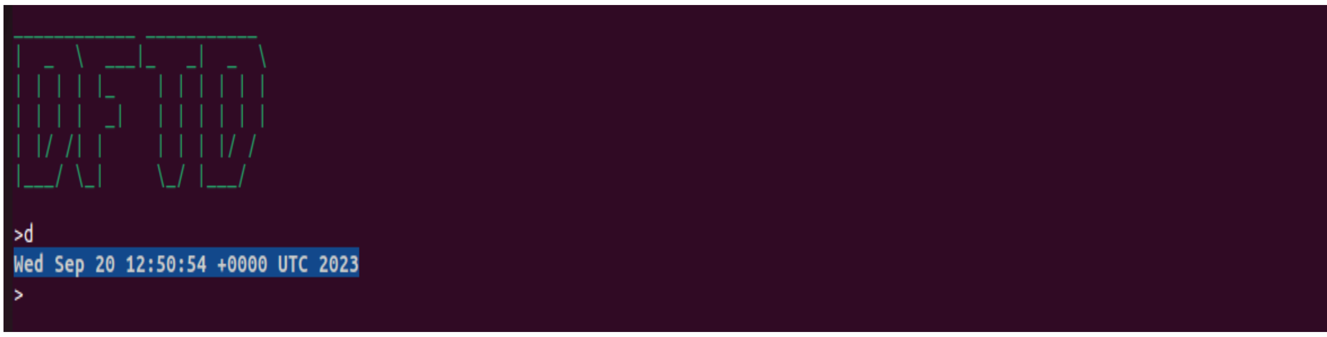
To connect to the telnet-server running inside the container, pass the IP address or localhost and port (2323) to the telnet command:



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ telnet localhost 2323
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

______________ ___________
|  _  \  ___|_   _|  _  \
| | | | |_    | | | | | |
| | | |  _|   | | | | | |
| |/ /| |     | | | |/ /
|___/ \_|     \_/ |___/

>
```

While still connected to the telnet-server, enter the following to print the current date and time:

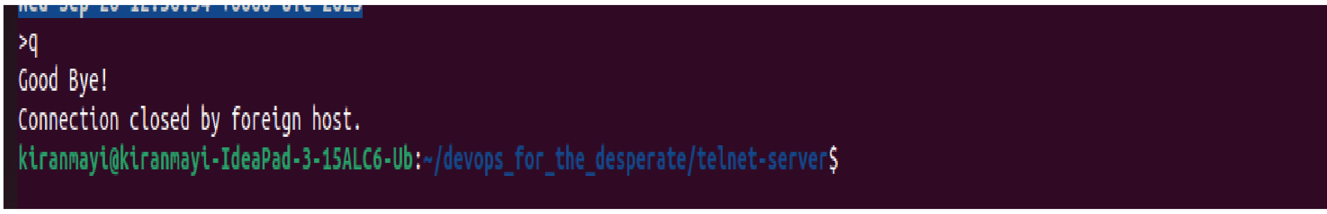
d



```
______________ ___________
|  _  \  ___|_   _|  _  \
| | | | |_    | | | | | |
| | | |  _|   | | | | | |
| |/ /| |     | | | |/ /
|___/ \_|     \_/ |___/

>d
Wed Sep 20 12:50:54 +0000 UTC 2023
>
```

Enter the following to quit the telnet-server session:

q

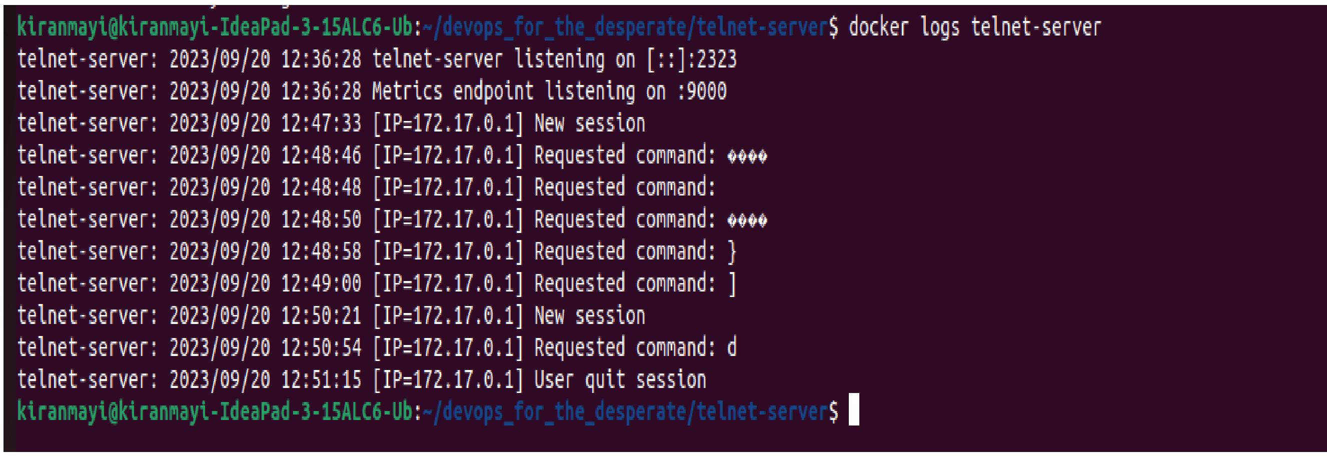


```
>q
Good Bye!
Connection closed by foreign host.
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```

Getting Logs from the Container:

Docker provides a simple way to retrieve logs from a running container. This is important for troubleshooting and forensics purpose. To see all the logs for the telnet-server, which is logging to STDOUT, enter the following in your terminal:

docker logs telnet-server



```
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$ docker logs telnet-server
telnet-server: 2023/09/20 12:36:28 telnet-server listening on [::]:2323
telnet-server: 2023/09/20 12:36:28 Metrics endpoint listening on :9000
telnet-server: 2023/09/20 12:47:33 [IP=172.17.0.1] New session
telnet-server: 2023/09/20 12:48:46 [IP=172.17.0.1] Requested command: ����
telnet-server: 2023/09/20 12:48:48 [IP=172.17.0.1] Requested command:
telnet-server: 2023/09/20 12:48:50 [IP=172.17.0.1] Requested command: ����
telnet-server: 2023/09/20 12:48:58 [IP=172.17.0.1] Requested command: }
telnet-server: 2023/09/20 12:49:00 [IP=172.17.0.1] Requested command: ]
telnet-server: 2023/09/20 12:50:21 [IP=172.17.0.1] New session
telnet-server: 2023/09/20 12:50:54 [IP=172.17.0.1] Requested command: d
telnet-server: 2023/09/20 12:51:15 [IP=172.17.0.1] User quit session
kiranmayi@kiranmayi-IdeaPad-3-15ALC6-Ub:~/devops_for_the_desperate/telnet-server$
```