SRPCE

NAME:- Akash R

ROLL.NO:- 2

REG.NO:- 422021104002

SEMESTER:-5

DEPARTMENT:-CSE

SUBJECT:- cloud Application development

SESSION:- 2021-2025

Chapter 2: Setting up a Working Environment

```
$ docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER        ERRORS
default   -        virtualbox   Running   tcp://192.168.99.100:2376           v18.04.0-ce
$
```

```
$ docker-machine create --driver virtualbox default
Running pre-create checks...
Creating machine...
(default) Copying /Users/gabriel/.docker/machine/cache/boot2docker.iso to /Users/gabriel/.docker/machine/
machines/default/boot2docker.iso...
(default) Creating VirtualBox VM...
(default) Creating SSH key...
(default) Starting the VM...
(default) Check network to re-create if needed...
(default) Waiting for an IP...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with boot2docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docke
r-machine env default
$
```

```
docker@default:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
ca4f61b1923c: Pull complete
Digest: sha256:97ce6fa4b6cdc0790cda65fe7290b74cfebd9fa0c9b8c38e979330d547d22ce1
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://cloud.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/

docker@default:~$
```

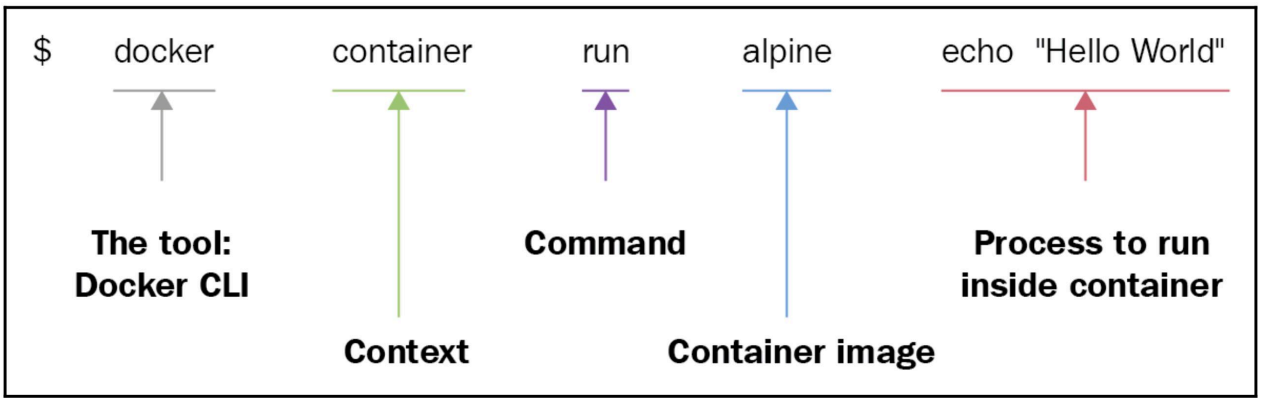
```
$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
ca4f61b1923c: Pull complete
Digest: sha256:97ce6fa4b6cdc0790cda65fe7290b74cfebd9fa0c9b8c38e979330d547d22ce1
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://cloud.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/

$
```

```
docker@default:~$ docker version
Client:
 Version:           18.06.1-ce
 API version:       1.38
 Go version:        go1.10.3
 Git commit:        e68fc7a
 Built:             Tue Aug 21 17:20:43 2018
 OS/Arch:           linux/amd64
 Experimental:      false

Server:
 Engine:
  Version:          18.06.1-ce
  API version:      1.38 (minimum version 1.12)
  Go version:       go1.10.3
  Git commit:       e68fc7a
  Built:            Tue Aug 21 17:28:38 2018
  OS/Arch:          linux/amd64
  Experimental:     false
docker@default:~$
```

```
Starting local Kubernetes v1.9.0 cluster...
Starting VM...
Downloading Minikube ISO
 142.22 MB / 142.22 MB [============================================] 100.00% 0s
Getting VM IP address...
Moving files into cluster...
Downloading localkube binary
 162.41 MB / 162.41 MB [============================================] 100.00% 0s
 0 B / 65 B [----------------------------------------------------------]   0.00%
 65 B / 65 B [==========================================================] 100.00% 0sSetting up certs...
Connecting to cluster...
Setting up kubeconfig...
Starting cluster components...
Kubectl is now configured to use the cluster.
Loading cached images from config file.
$
```

```
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"9", GitVersion:"v1.9.0", GitCommit:"925c127ec6b946659ad0fd596fa959be43f0cc05"
, GitTreeState:"clean", BuildDate:"2017-12-15T21:07:38Z", GoVersion:"go1.9.2", Compiler:"gc", Platform:"darwin/amd64"}
Server Version: version.Info{Major:"", Minor:"", GitVersion:"v1.9.0", GitCommit:"925c127ec6b946659ad0fd596fa959be43f0cc05",
GitTreeState:"clean", BuildDate:"2018-01-26T19:04:38Z", GoVersion:"go1.9.1", Compiler:"gc", Platform:"linux/amd64"}
$
```

Chapter 3: Working with Containers




```
$ docker container ls -l
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS      NAMES
6ce5e46da7ce        alpine              "/bin/sh -c 'while :…"   41 seconds ago      Up 16 seconds                  quotes
$
```

```
$ docker container ls
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
31d719b2f439        nginx:alpine        "nginx -g 'daemon of…"   35 seconds ago      Up 30 seconds       80/tcp              cranky_curie
27b96de70b58        alpine:latest       "ping 127.0.0.1"         23 hours ago        Up 23 hours                             c2
35b8dd512acb        alpine:latest       "/bin/sh"                23 hours ago        Up 23 hours                             c1
$
```

```
/ # ps
PID   USER     TIME   COMMAND
    1 root       0:00 /bin/sh -c while :; do wget -qO- https://talaikis.com/api
   85 root       0:00 /bin/sh
  110 root       0:00 sleep 5
  111 root       0:00 ps
```

```
$ docker container exec quotes ps
PID   USER     TIME   COMMAND
    1 root       0:00 /bin/sh -c while :; do wget -qO- https://talaikis.com/api
  520 root       0:00 sleep 5
  521 root       0:00 ps
$
```

Chapter 5: Data Volumes and System Management

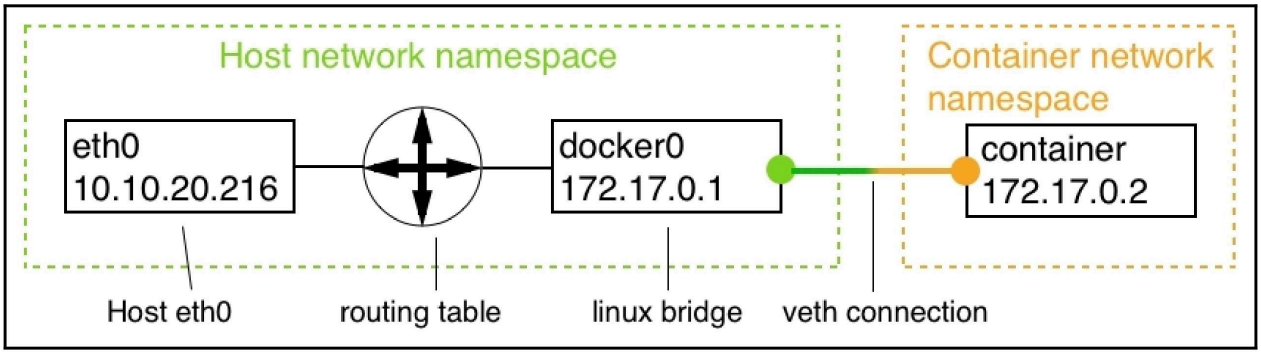
```
$ docker version
Client:
 Version:       18.04.0-ce
 API version:   1.37
 Go version:    go1.9.4
 Git commit:    3d479c0
 Built: Tue Apr 10 18:13:16 2018
 OS/Arch:       darwin/amd64
 Experimental:  true
 Orchestrator:  swarm

Server:
 Engine:
  Version:      18.04.0-ce
  API version:  1.37 (minimum version 1.12)
  Go version:   go1.9.4
  Git commit:   3d479c0
  Built:        Tue Apr 10 18:23:05 2018
  OS/Arch:      linux/amd64
  Experimental: true
$
```

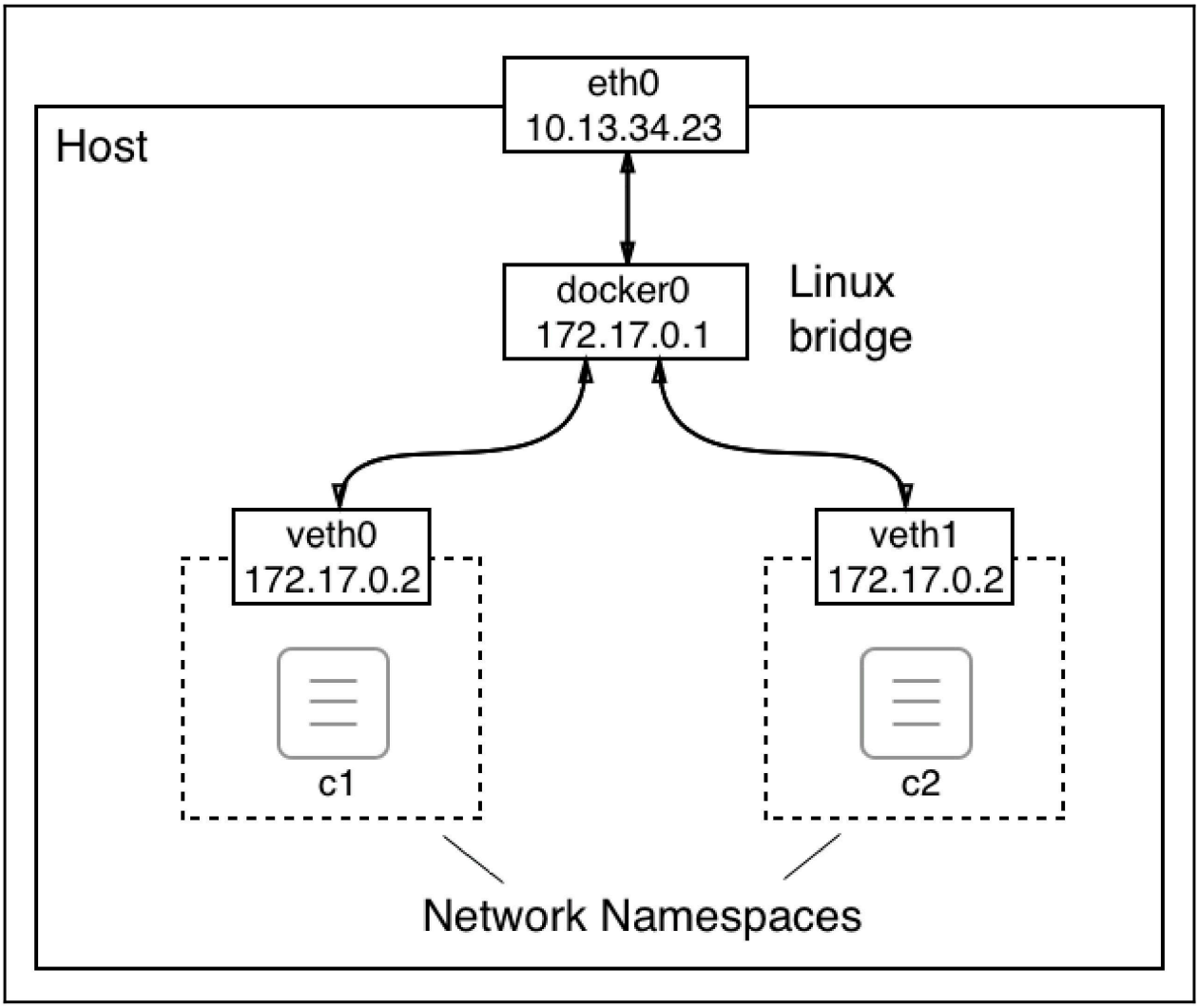
```
$ docker system info
Containers: 1
 Running: 0
 Paused: 0
 Stopped: 1
Images: 70
Server Version: 18.04.0-ce
Storage Driver: overlay2
 Backing Filesystem: extfs
 Supports d_type: true
 Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host ipvlan macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 773c489c9c1b21a6d78b5c538cd395416ec50f88
runc version: 4fc53a81fb7c994640722ac585fa9ca548971871
init version: 949e6fa
Security Options:
 seccomp
  Profile: default
Kernel Version: 4.9.87-linuxkit-aufs
Operating System: Docker for Mac
OSType: linux
Architecture: x86_64
CPUs: 4
Total Memory: 1.952GiB
Name: linuxkit-025000000001
ID: WV5X:CY7N:LHIP:SWJ2:T55W:P5QM:MEYU:MM3V:55OH:RALF:5ZDN:QH7Y
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): true
 File Descriptors: 22
 Goroutines: 42
 System Time: 2018-04-21T12:08:17.962868Z
 EventsListeners: 2
HTTP Proxy: gateway.docker.internal:3128
HTTPS Proxy: gateway.docker.internal:3129
Registry: https://index.docker.io/v1/
Labels:
Experimental: true
Insecure Registries:
 127.0.0.0/8
Live Restore Enabled: false
```

```
$ docker system df -v
Images space usage:

REPOSITORY                          TAG                 IMAGE ID            CREATED ago         SIZE                SHARED SIZE         UNIQUE SiZE         CONTAINERS
fundamentalsofdocker/ch14-web       1.0                 fe6612f845be        12 days ago ago     72.05MB             72.05MB             1.834kB             0
fundamentalsofdocker/web            2.0                 944523644dd7        12 days ago ago     72.05MB             72.05MB             1.827kB             0
fundamentalsofdocker/ch13-web       2.0                 b47675c8e53c        2 weeks ago ago     72.05MB             72.05MB             1.836kB             0
builder2                            latest              074a85b21f3a        3 weeks ago ago     1.514GB             1.456GB             58.7MB              0
builder                             latest              9864221c5187        3 weeks ago ago     1.464GB             1.456GB             8.051MB             0
ruby                                alpine              b620ae34414c        3 weeks ago ago     55.52MB             4.148MB             51.38MB             1
microsoft/azure-cli                 latest              a52f6e53da4c        4 weeks ago ago     400.4MB             0B                  400.4MB             0
nginx                               alpine              91ce6206f9d8        4 weeks ago ago     18MB                4.148MB             13.86MB             1
perl                                5.26                3c2c4c3b2e15        5 weeks ago ago     879.2MB             879.2MB             0B                  0
fundamentalsofdocker/ch08-web       1.0                 922e085ed002        7 weeks ago ago     72.01MB             68.02MB             3.992MB             0
fundamentalsofdocker/ch08-db        1.0                 4953d5353c17        7 weeks ago ago     39.46MB             4.148MB             35.31MB             0
node                                9.6-alpine          a88ff852e3d4        8 weeks ago ago     68.02MB             68.02MB             0B                  0
alpine                              latest              3fd9065eaf02        3 months ago ago    4.148MB             4.148MB             0B                  0
confluentinc/cp-enterprise-kafka    4.0.0               07d41f8648f5        3 months ago ago    565.1MB             0B                  565.1MB             0
hello-world                         latest              f2a91732366c        5 months ago ago    1.848kB             0B                  1.848kB             0
hseeberger/scala-sbt                latest              da0e1be3bb79        9 months ago ago    925.4MB             0B                  925.4MB             0

Containers space usage:

CONTAINER ID        IMAGE               COMMAND             LOCAL VOLUMES       SIZE                CREATED ago         STATUS                      NAMES
afe0dcab9bc4        nginx:alpine        "ping 8.8.8.8"      0                   0B                  6 seconds ago ago   Up 4 seconds                xenodochial_easley
2a2d742604af        ruby:alpine         "/bin/sh"           0                   16.7MB              5 hours ago ago     Exited (0) 13 minutes ago   keen_lumiere

Local Volumes space usage:

VOLUME NAME         LINKS               SIZE
ch08_pets-data      0                   47.24MB

Build cache usage: 0B

$
```

```
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
928c8ce47bf2        bridge              bridge              local
bdb36adcf70c        host                host                local
af82006f2f2d        none                null                local
$
```

```
C:\Users\admin>docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "3b08c1c711ada84ae859c4bed48b5af1f45b68db89356ca5045dc7ee8672e946",
        "Created": "2018-04-09T09:47:29.9424652Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
```


Host network namespace
Container network namespace
eth0
10.10.20.216
docker0
172.17.0.1
container
172.17.0.2
Host eth0
routing table
linux bridge
veth connection


eth0
10.13.34.23
Host
docker0
172.17.0.1
Linux
bridge
veth0
172.17.0.2
veth1
172.17.0.2
c1
c2
Network Namespaces