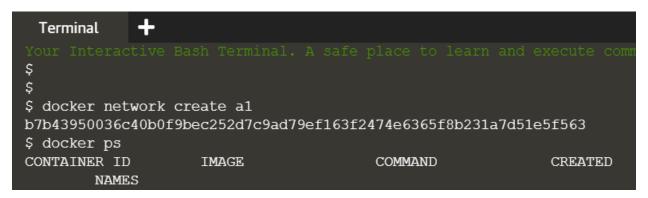
Name: Animesh Jain

Roll Number: R171218022

SAP ID: 500069453

Check if any container is running

Run command - \$ docker ps



Creating a container

Run command - \$ docker run -d -name=b1 -net=a1 redis



To check IP address

Run command - \$ docker inspect a180

(Note: a180 is unique initials of container ID you want ip)

```
Terminal
$ docker inspect a180
        "Id": "a18066204754382e13710dc1e06a6a597d0d3b0f3f1d0dd3ae995ec49aa130b5",
        "Created": "2021-02-05T06:13:18.577214385Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "redis-server"
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 1628,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2021-02-05T06:13:19.017604316Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
"Image": "sha256:4e8db158f18dc71307f95260e532df39a9b604b51d4e697468e82845c50cfe28",
        "ResolvConfPath": "/var/lib/docker/containers/a18066204754382e13710dc1e06a6a597d0d3b0f3f1d0dd3ae9
a130b5/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/a18066204754382e13710dc1e06a6a597d0d3b0f3f1d0dd3ae995
30b5/hostname",
        "HostsPath": "/var/lib/docker/containers/a18066204754382e13710dc1e06a6a597d0d3b0f3f1d0dd3ae995ec4
5/hosts",
"LogPath": "/var/lib/docker/containers/a18066204754382e13710dc1e06a6a597d0d3b0f3f1d0dd3ae995ec49a-
        "Name": "/b1",
"RestartCount": 0,
        "Driver": "overlay",
```

```
"a18066204754"
],
   "NetworkID": "b7b43950036c40b0f9bec252d7c9ad79ef1
   "EndpointID": "0bda81d19a819b43ae54b1d2552f9ee983
   "Gateway": "172.19.0.1",
   "IPAddress": "172.19.0.2",
   "IPPrefixLen": 16,
   "IPv6Gateway": "",
   "GlobalIPv6Address": "",
   "GlobalIPv6PrefixLen": 0,
   "MacAddress": "02:42:ac:13:00:02",
   "DriverOpts": null
}
```

```
Terminal
$ docker run -d --name=b2 --net=a1 redis
f51909d84c7d23264225be8bb8293c58914abef5c81da3f730500832b0e7130a
$ docker ps
CONTAINER ID
                                      COMMAND
                   IMAGE
                                                                CREATED
                                                                                    STATUS
            NAMES
f51909d84c7d
                 redis
                                       "docker-entrypoint.s..." 6 seconds ago
                                                                                   Up 4 secon
            b2
a18066204754
                                       "docker-entrypoint.s..." 5 minutes ago
                 redis
                                                                                    Up 5 minut
            b1
$ docker inspect f5
       "Id": "f51909d84c7d23264225be8bb8293c58914abef5c81da3f730500832b0e7130a",
        "Created": "2021-02-05T06:18:43.192236106Z",
        "Path": "docker-entrypoint.sh",
        "Args": [
            "redis-server"
       "State": {
           "Status": "running",
           "Running": true,
           "Paused": false,
           "Restarting": false,
           "OOMKilled": false,
           "Dead": false,
            "Pid": 1923,
           "ExitCode": 0,
           "Error": "",
           "StartedAt": "2021-02-05T06:18:43.692506156Z",
           "FinishedAt": "0001-01-01T00:00:00Z"
```

```
"f51909d84c7d"
],
"NetworkID": "b7b43950036c40b0f9bec252d7c9ad79ef163f2474e
"EndpointID": "e946ac1b0df4a58863585a1c65493cea63a6d6b1b9
"Gateway": "172.19.0.1",
"IPAddress": "172.19.0.3",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"MacAddress": "02:42:ac:13:00:03",
"DriverOpts": null
```

```
$ docker network create backend-network
cccd0d1b9c8e53932b2958d4744d862ecb39e90c470cf6b8980854f43c3ec783
$ docker run -d --name=redis --net=backend-network redis
b65148c352c140a03f13c1abc4ab0f779a667eec7ce43429490f26e0fe338890
$ docker run --net=backend-network alpine env
docker run --net=backend-network alpine cat /etc/hosts
docker run --net=backend-network alpine cat /etc/resolv.conf
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/sbin:/bin
HOSTNAME=0da2dfd579b3
HOME=/root
$ docker run --net=backend-network alpine cat /etc/hosts
docker run --net=backend-network alpine ping -c1 redis
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.20.0.3 77aca28a03bc
$ docker run --net=backend-network alpine cat /etc/resolv.conf
nameserver 127.0.0.11
options ndots:0
$ docker run --net=backend-network alpine ping -c1 redis
PING redis (172.20.0.2): 56 data bytes
64 bytes from 172.20.0.2: seq=0 ttl=64 time=0.131 ms
--- redis ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.131/0.131/0.131 ms
$
```

```
$ docker network create nw1
70e131726e7c2c29a86439b51a9a3dedbba56116b9411e7e8b0ae8701068cf4d
$ docker network ls
NETWORK ID
                  NAME
                                      DRIVER
b7b43950036c
                                      bridge
                                                          local
                 backend-network bridge
cccd0d1b9c8e
                                                         local
                 bridge
f35aae151499
                                    bridge
0c288d2ed25a
                 host
                                     host
                                                         local
4d3221fe852b
                 none
                                                          local
                                      null
70e131726e7c
                  nw1
                                      bridge
$ docker run -d --name=c1 --net==nw1 alpine
623eb7f294e06a1f1a49f0e1c10b60e1b02ac17f00d82d42e1050c722bebee80
docker: Error response from daemon: network =nw1 not found.
$ docker run -d --name=c1 --net=nw1 alpine
docker: Error response from daemon: Conflict. The container name "/c1" is already in use by containe
4e06a1f1a49f0e1c10b60e1b02ac17f00d82d42e1050c722bebee80". You have to remove (or rename) that contain
le to reuse that name.
See 'docker run --help'.
$ docker run -d --name=c2 --net=nw1 alpine
3fd9374cbcd6f32ee5169c0bbc74492593234b5431e6525eb32939248fdcc19c
$ docker ps
CONTAINER ID
                                     COMMAND
                                                               CREATED
                                                                                  STATUS
                   TMAGE
           NAMES
b65148c352c1
                   redis
                                      "docker-entrypoint.s..." 6 minutes ago
                                                                                  Up 6 minutes
            redis
09a58dd178a1
                                      "docker-entrypoint.s..." 10 minutes ago
                                                                                  Up 10 minutes
                   redis
            b3
f51909d84c7d
                                      "docker-entrypoint.s..." 13 minutes ago
                                                                                  Up 12 minutes
                   redis
           b2
a18066204754
                   redis
                                      "docker-entrypoint.s..." 18 minutes ago
                                                                                  Up 18 minutes
           b1
$ docker ps -a
```

Ping network

\$ docker run -name=c3 -net=backend-network alpine ping redis

```
$ docker run --name=c3 --net=backend-network alpine ping redis
PING redis (172.20.0.2): 56 data bytes
64 bytes from 172.20.0.2: seq=0 ttl=64 time=0.152 ms
64 bytes from 172.20.0.2: seq=1 ttl=64 time=0.097 ms
64 bytes from 172.20.0.2: seq=2 ttl=64 time=0.086 ms
64 bytes from 172.20.0.2: seq=3 ttl=64 time=0.139 ms
64 bytes from 172.20.0.2: seq=4 ttl=64 time=0.090 ms
64 bytes from 172.20.0.2: seq=5 ttl=64 time=0.120 ms
64 bytes from 172.20.0.2: seq=6 ttl=64 time=0.097 ms
64 bytes from 172.20.0.2: seq=7 ttl=64 time=0.109 ms
64 bytes from 172.20.0.2: seq=8 ttl=64 time=0.128 ms
64 bytes from 172.20.0.2: seq=9 ttl=64 time=0.135 ms
64 bytes from 172.20.0.2: seq=10 ttl=64 time=0.153 ms
64 bytes from 172.20.0.2: seq=11 ttl=64 time=0.121 ms
64 bytes from 172.20.0.2: seq=12 ttl=64 time=0.102 ms
64 bytes from 172.20.0.2: seg=13 ttl=64 time=0.092 ms
```

```
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS

NAMES

860f5e151296 alpine "ping redis" About a minute ago Up About a minute
```

Create a network

\$ docker network create backend-network

```
$ docker network create backend-network
1dbc0e12f98214e52a76ae34082cd6cfec750186935dc2ee7bc29e0896c9b788
$ docker ps
CONTAINER ID
                                               CREATED
                                                               STATUS
                                                                               PORTS
               IMAGE
                               COMMAND
     NAMES
$ docker run -d --name=redis --net=backend-network redis
a695e68f0304cc2a09206eacd5b860def7f1fbc481f0fd7d11a5bf7867f1a151
$ docker ps
                               COMMAND
                                                   CREATED
                                                                   STATUS
                                                                                   PORT
CONTAINER ID
               IMAGE
         NAMES
                               "docker-entrypoint.s..." 9 seconds ago
a695e68f0304
                                                                                   6379
               redis
                                                                   Up 7 seconds
$ docker run --net=backend-network alpine cat /etc/resolv.conf
nameserver 127.0.0.11
options ndots:0
$ docker run --net=backend-network alpine ping -c1 redis
PING redis (172.19.0.2): 56 data bytes
64 bytes from 172.19.0.2: seq=0 ttl=64 time=0.144 ms
--- redis ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.144/0.144/0.144 ms
```

Create a network

\$ docker network create frontend-network

```
$ docker network create frontend-network
9a5cb2c53263ea87ad3a48a7e11110e3fb7fbf203e98b294aea987400add6d19
$ docker ps
CONTAINER ID
                   IMAGE
                                      COMMAND
                                                                CREATED
                                                                                    STATU
            NAMES
a695e68f0304
                   redis
                                       "docker-entrypoint.s..." 6 minutes ago
                                                                                    Up 6
            redis
$ docker inspect redis
        "Id": "a695e68f0304cc2a09206eacd5b860def7f1fbc481f0fd7d11a5bf7867f1a151",
        "Created": "2021-02-05T06:41:08.934384463Z",
```

```
},
"NetworkMode": "backend-network",
"PortBindings": {},
"RestartPolicy": {
    "Name": "no",
    "MaximumRetryCount": 0
```

```
"Networks": {
    "backend-network": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": [
            "a695e68f0304"
        ],
        "NetworkID": "1dbc0e12f98214e52a76ae34082cd6cfec7501
        "EndpointID": "3c5c86fbc51672fb30ad80134edb0411650e6
        "Gateway": "172.19.0.1",
        "IPAddress": "172.19.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:13:00:02",
        "DriverOpts": null
    },
    "frontend-network": {
        "IPAMConfig": {},
        "Links": null,
        "Aliases": [
            "a695e68f0304"
        "NetworkID": "9a5cb2c53263ea87ad3a48a7e11110e3fb7fbf
        "EndpointID": "4d37cb45fd39d23b7d24da3280962608f8316
        "Gateway": "172.20.0.1",
        "IPAddress": "172.20.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0.
```

\$ docker run -d -p 3000:3000 -net=frontend-network katacoda/redis-nodedocker-example

```
$ docker run -d -p 3000:3000 --net=frontend-network katacoda/redis-node-docker-example
Unable to find image 'katacoda/redis-node-docker-example:latest' locally
latest: Pulling from katacoda/redis-node-docker-example
12b41071e6ce: Pull complete
a3ed95caeb02: Pull complete
49a025abf7e3: Pull complete
1fb1c0be01ab: Pull complete
ae8c1f781cde: Pull complete
db73207ad2ae: Pull complete
446b13034c13: Pull complete
Digest: sha256:1aae9759464f00953c8e078a0e0d0649622fef9dd5655b1491f9ee589ae904b4
Status: Downloaded newer image for katacoda/redis-node-docker-example:latest
79eb06d49f6487b71ebbc1cdef8a4ef5cc71920685e7be30cd4b9824df8493e0
$ curl docker:3000
This page was generated after talking to redis.
Application Build: 1
Total requests: 1
IP count:
  ::ffff:172.17.0.58: 1
```

To create a network

\$ docker network create frontend-network

```
$ docker network create frontend-network2
e58e9123a83f5f264e5c39bee58fd22a55c7c8366e76233254e8817695e77c35
$ docker network connect --alias db frontend-network2 redis
$ docker run --net=frontend-network2 alpine ping -c1 db
PING db (172.21.0.2): 56 data bytes
64 bytes from 172.21.0.2: seq=0 ttl=64 time=0.204 ms
--- db ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.204/0.204/0.204 ms
$ [
```

\$ docker network 1s			
NETWORK ID	NAME	DRIVER	SCOPE
1dbc0e12f982	backend-network	bridge	local
f70cdd723831	bridge	bridge	local
9a5cb2c53263	frontend-network	bridge	local
e58e9123a83f	frontend-network2	bridge	local
0c288d2ed25a	host	host	local
4d3221fe852b	none	null	local

To disconnect from server

\$ docker network disconnect frontend-network redis

(frontend-network is name of network)

```
$ docker network disconnect frontend-network redis
$
```