DOCKER INSTALLATION

DevOps Certification Training

support@intellipaat.com
+91-7022374614
US: 1-800-216-8930(Toll Free)

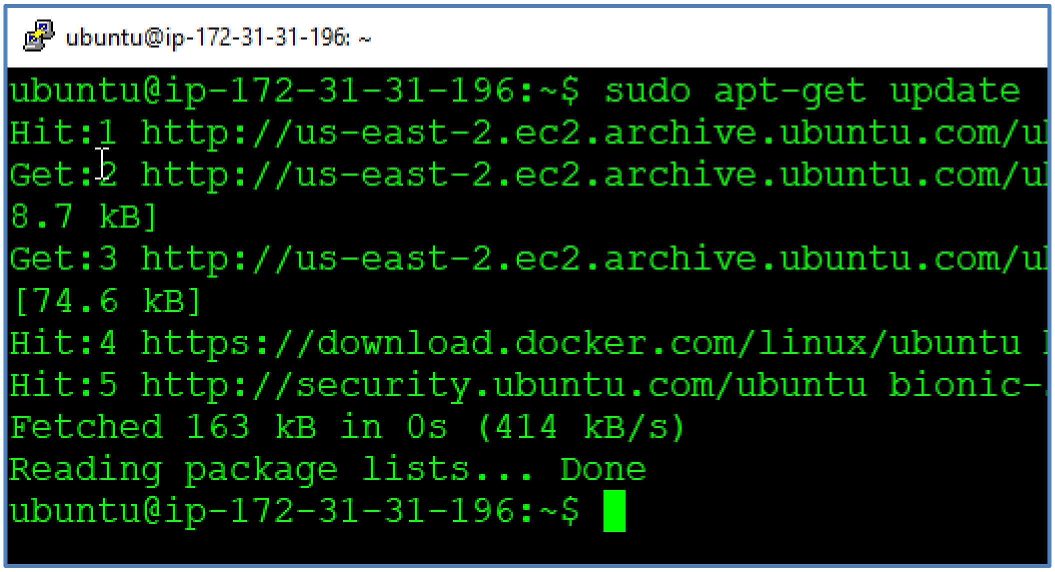
DOCKER INSTALLATION ON UBUNTU

Note: The terminal with green color commands represents terminal and the one with white color commands represents slave terminal.

Docker Installation

Step 1: Update your machine using the below command

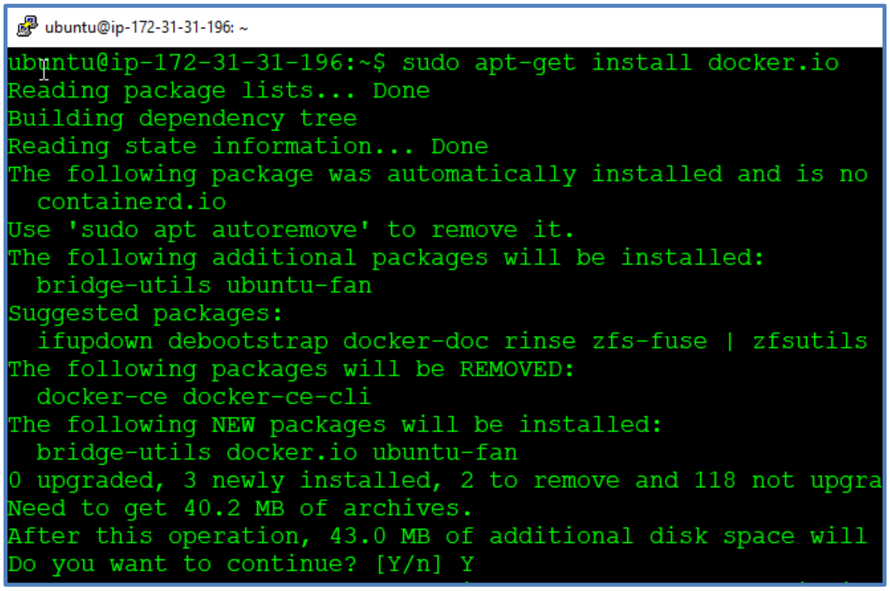
```
$ sudo apt-get update
```



```
ubuntu@ip-172-31-31-196: ~
ubuntu@ip-172-31-31-196:~$ sudo apt-get update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/u
Get:2 http://us-east-2.ec2.archive.ubuntu.com/u
8.7 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/u
[74.6 kB]
Hit:4 https://download.docker.com/linux/ubuntu
Hit:5 http://security.ubuntu.com/ubuntu bionic-
Fetched 163 kB in 0s (414 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-31-196:~$
```

Step 2: Now use this command to install docker on your system

```
$ sudo apt-get install docker.io
```

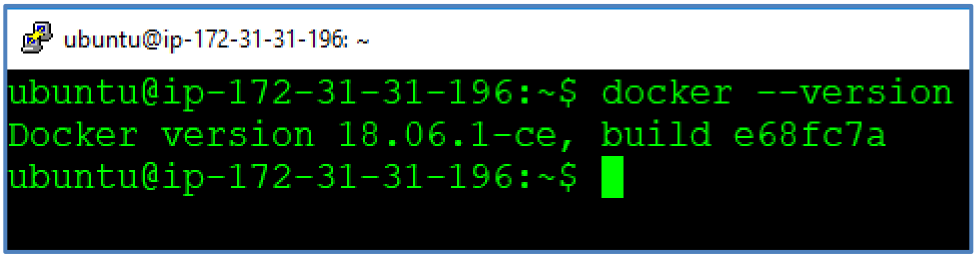


ubuntu@ip-172-31-31-196: ~

```
ubuntu@ip-172-31-31-196:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no
  containerd.io
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  bridge-utils ubuntu-fan
Suggested packages:
  ifupdown debootstrap docker-doc rinse zfs-fuse | zfsutils
The following packages will be REMOVED:
  docker-ce docker-ce-cli
The following NEW packages will be installed:
  bridge-utils docker.io ubuntu-fan
0 upgraded, 3 newly installed, 2 to remove and 118 not upgra
Need to get 40.2 MB of archives.
After this operation, 43.0 MB of additional disk space will
Do you want to continue? [Y/n] Y
```

Step 3: In order to check if docker installed properly in your system or not, use the command below:

```
$ docker --version
```



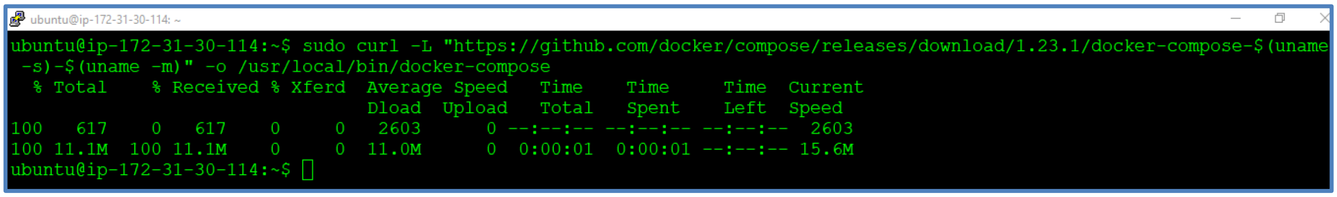
ubuntu@ip-172-31-31-196: ~

```
ubuntu@ip-172-31-31-196:~$ docker --version
Docker version 18.06.1-ce, build e68fc7a
ubuntu@ip-172-31-31-196:~$
```

Docker compose:

Step 1: Run this command to download the latest version of Docker Compos.

```
$ sudo curl -L "https://github.com/docker/compose/releases/download/1.23.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

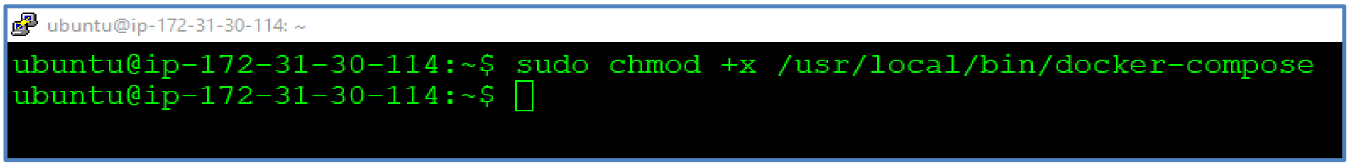


ubuntu@ip-172-31-30-114: ~

```
ubuntu@ip-172-31-30-114:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.23.1/docker-compose-$(uname
 -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   617    0   617    0     0   2603      0 --:--:-- --:--:-- --:--:--  2603
100 11.1M  100 11.1M    0     0  11.0M      0  0:00:01  0:00:01 --:--:-- 15.6M
ubuntu@ip-172-31-30-114:~$ 
```

Step 2: Apply executable permissions to the binary.

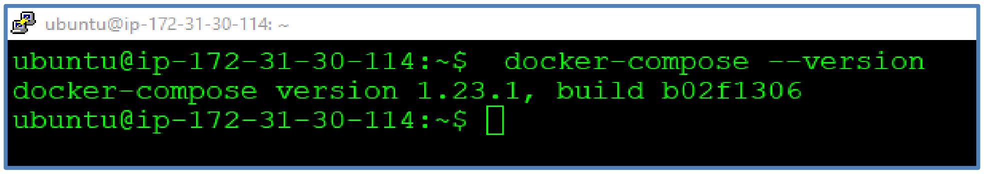
```
$ sudo chmod +x /usr/local/bin/docker-compose
```



```
ubuntu@ip-172-31-30-114:~$ sudo chmod +x /usr/local/bin/docker-compose
ubuntu@ip-172-31-30-114:~$ 
```

Step 3: Run the following command to test the installation.

```
$ docker-compose --version
```

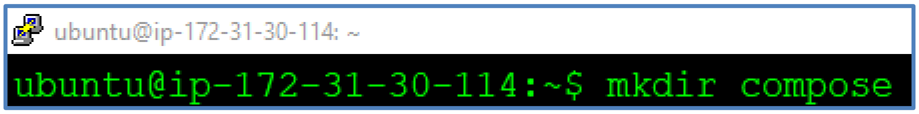


```
ubuntu@ip-172-31-30-114:~$  docker-compose --version
docker-compose version 1.23.1, build b02f1306
ubuntu@ip-172-31-30-114:~$ 
```

Before moving ahead to the installation of docker swarm let us create a compose directory. Follow the steps.

Step 1: Create the directory.

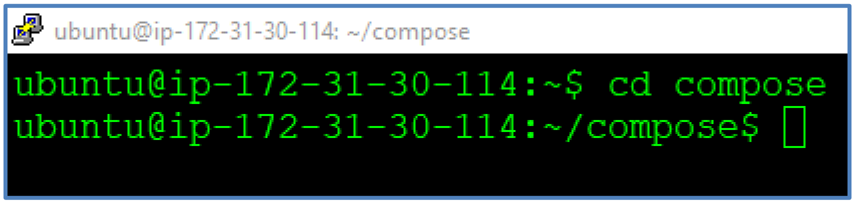
```
$ mkdir compose
```



```
ubuntu@ip-172-31-30-114:~$ mkdir compose
```

Step 2: Get inside the directory by using the following command.

```
$ cd compose
```



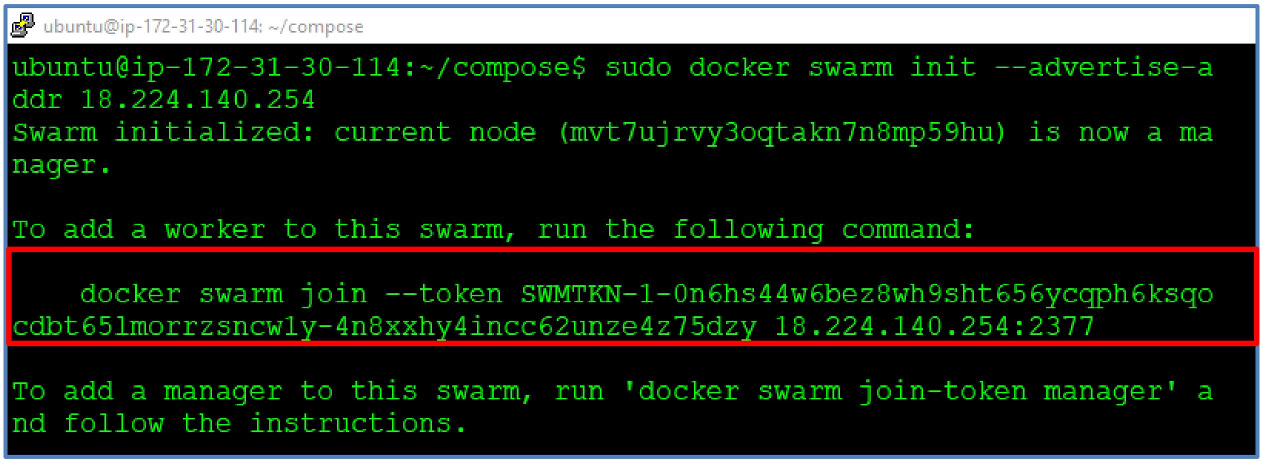
```
ubuntu@ip-172-31-30-114:~$ cd compose
ubuntu@ip-172-31-30-114:~/compose$ 
```

Initialization of Docker Swarm:

Since we have already installed docker in our system, along with that docker swarm is already installed. We just need to initialize the docker swarm

Step 1: Use the following command to create a new swarm.

```
$ sudo docker swarm init –advertise-addr  <master IP>
```



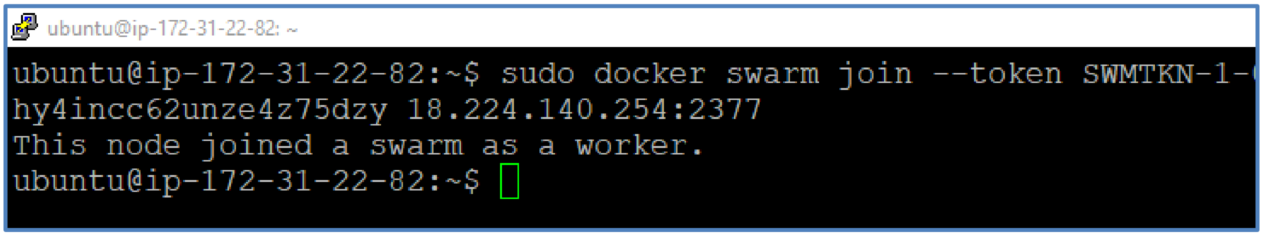
ubuntu@ip-172-31-30-114: ~/compose

```
ubuntu@ip-172-31-30-114:~/compose$ sudo docker swarm init --advertise-a
ddr 18.224.140.254
Swarm initialized: current node (mvt7ujrvy3oqtakn7n8mp59hu) is now a ma
nager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-0n6hs44w6bez8wh9sht656ycqph6ksqo
cdbt65lmorrzsncwly-4n8xxhy4incc62unze4z75dzy 18.224.140.254:2377

To add a manager to this swarm, run 'docker swarm join-token manager' a
nd follow the instructions.
```

Copy the token (marked in red) to clipboard.

Step 2: Now we will start a new session as worker, and we will join the swarm that we just created. Paste the copied token shown below.

```
$ sudo <token>
```

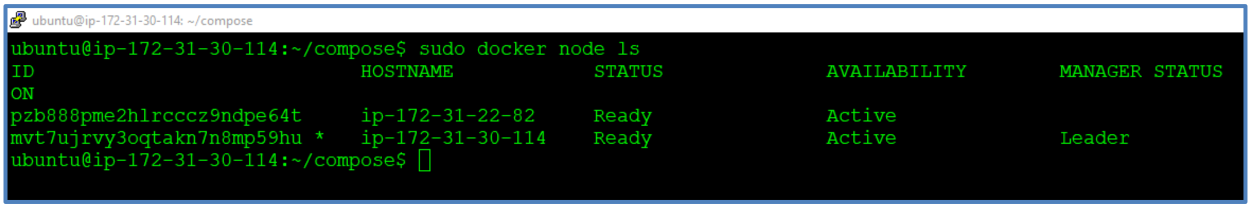


ubuntu@ip-172-31-22-82: ~

```
ubuntu@ip-172-31-22-82:~$ sudo docker swarm join --token SWMTKN-1-
hy4incc62unze4z75dzy 18.224.140.254:2377
This node joined a swarm as a worker.
ubuntu@ip-172-31-22-82:~$
```

Step 3: Now check we will check the node list as the manager.

```
$ sudo docker node ls
```

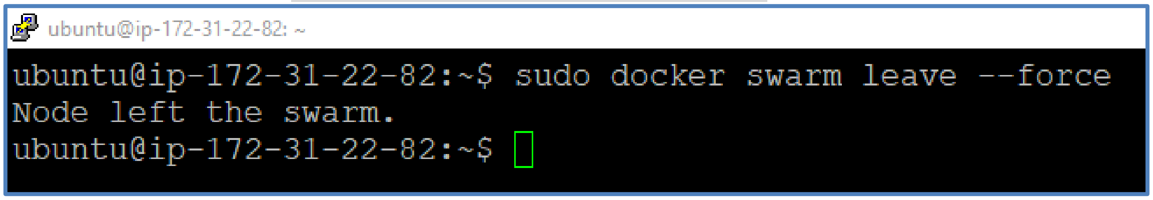


```
ubuntu@ip-172-31-30-114:~/compose$ sudo docker node ls
ID                              HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS
ON
pzb888pme2hlrcccz9ndpe64t     ip-172-31-22-82     Ready               Active
mvt7ujrvy3oqtakn7n8mp59hu *   ip-172-31-30-114    Ready               Active              Leader
ubuntu@ip-172-31-30-114:~/compose$
```

As you can see worker has joined and status of both nodes are ready.

Step 4: Follow the commands given below to leave the swarm.

```
$ sudo docker leave --force
```

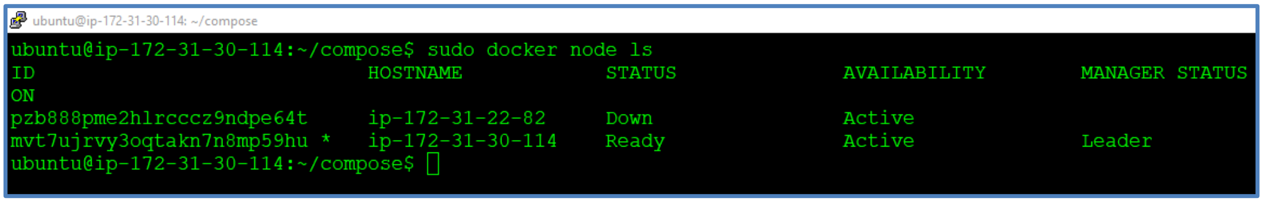


```
ubuntu@ip-172-31-22-82:~$ sudo docker swarm leave --force
Node left the swarm.
ubuntu@ip-172-31-22-82:~$
```

Now that the node left the swarm, let's check the node list as manager and check the status of the nodes.

Step 5: To check the node list as manager follow the command given below.

```
$ sudo docker node ls
```

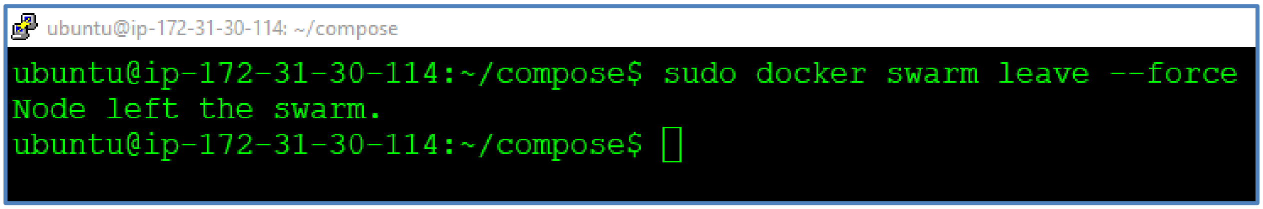


```
ubuntu@ip-172-31-30-114:~/compose$ sudo docker node ls
ID                              HOSTNAME            STATUS              AVAILABILITY        MANAGER STATUS
ON
pzb888pme2hlrcccz9ndpe64t     ip-172-31-22-82     Down                Active
mvt7ujrvy3oqtakn7n8mp59hu *   ip-172-31-30-114    Ready               Active              Leader
ubuntu@ip-172-31-30-114:~/compose$
```

As you can, the status of the node that left the swarm is no longer ready.

Step 6: To leave the swarm as manager follow the command given below.

```
$ sudo docker swarm leave --force
```



```
ubuntu@ip-172-31-30-114:~/compose$ sudo docker swarm leave --force
Node left the swarm.
ubuntu@ip-172-31-30-114:~/compose$ 
```