

# AWS SERVER SETUP WITH DOCKER CONTAINER

MUNTABIR HASAN CHOUDHURY | MCHOU001@ODU.EDU

CS 431/531 – WEB SERVER DESIGN

OLD DOMINION UNIVERSITY

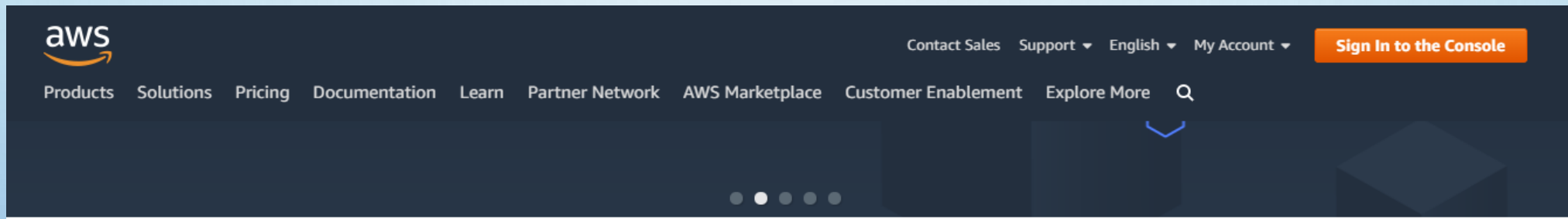
DEPARTMENT OF COMPUTER SCIENCE

# ACCOUNT SET UP

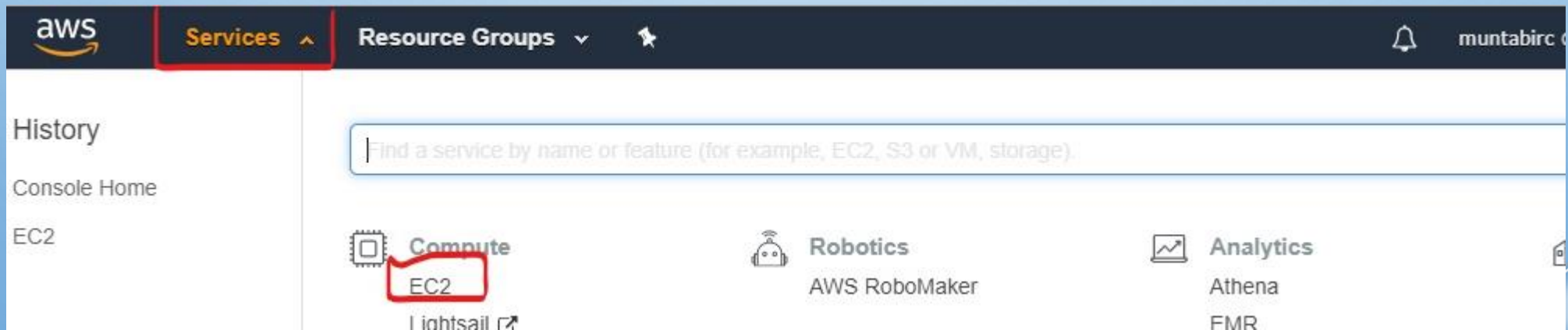
- OPEN A BROWSER AND GO TO AWS ([HTTPS://AWS.AMAZON.COM/](https://aws.amazon.com/))
- OPEN A FREE TIER ACCOUNT FOR 12 MONTHS
- AWS WILL ASK YOU TO PUT YOUR CREDIT CARD INFORMATION BUT DO NOT WORRY SINCE IT IS A FREE TIER ACCOUNT
- BEFORE 12 MONTHS, YOU CAN CANCEL YOUR SUBSCRIPTION

# EC2 INSTANCE

- SIGN INTO AWS MANAGEMENT CONSOLE

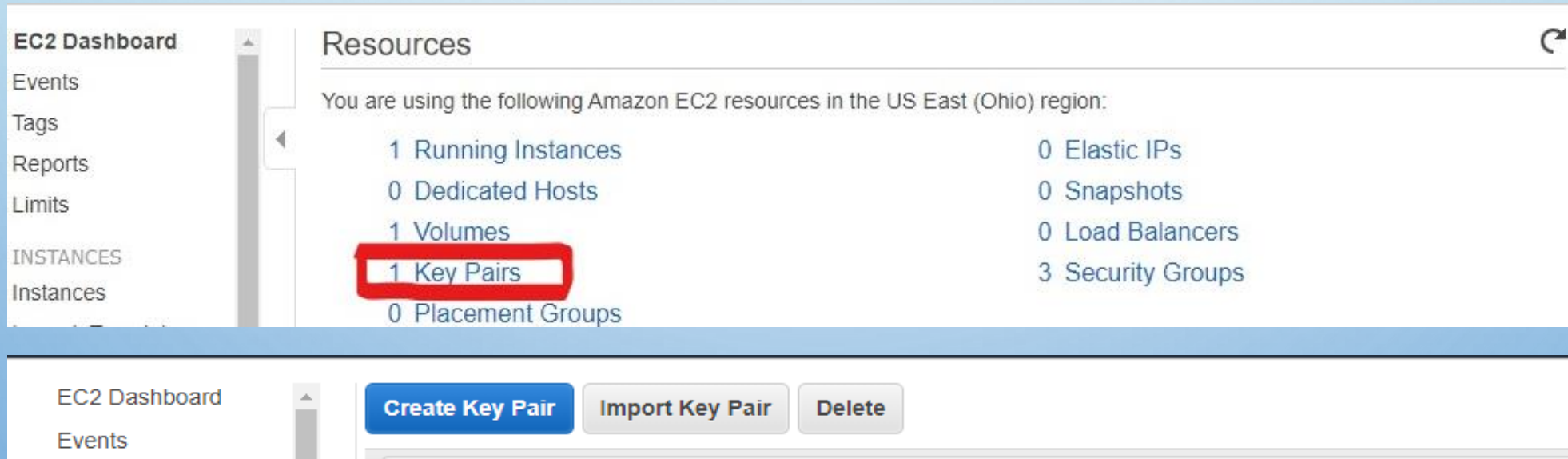


- CLICK ON SERVICES AND SELECT 'EC2'



# EC2 INSTANCE DASHBOARD

- CLICK ON KEY PAIRS AND CREATE A NEW KEY PAIR. IT WILL SAVED BE AS <YOUR\_SELECTED\_NAME>.PEM FILE IN YOUR COMPUTER



The screenshot displays the Amazon EC2 Dashboard. On the left, a sidebar lists navigation options: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, and Instances. The main content area, titled 'Resources', shows a summary of EC2 resources in the US East (Ohio) region. The resources listed are:

Resource Type	Count
Running Instances	1
Dedicated Hosts	0
Volumes	1
<b>Key Pairs</b>	<b>1</b>
Placement Groups	0
Elastic IPs	0
Snapshots	0
Load Balancers	0
Security Groups	3

Below the dashboard, a row of buttons is visible: 'Create Key Pair' (highlighted in blue), 'Import Key Pair', and 'Delete'.

# KEY PAIRS CONVERSION

- IF YOU ARE A MAC USER, YOU WILL NOT NEED TO CONVERT THE .PEM FILE (KEY PAIRS)
- IF YOU ARE A WINDOWS USER, YOU WILL NEED TO CONVERT THE .PEM FILE TO .PPK FILE USING 'PUTTY GEN'
- CLICK ON THE LINK BELOW FOR A COMPLETE GUIDE ON HOW TO CONVERT THE .PEM FILE TO .PPK FILE AND PERFORM SSH
- [HTTPS://STACKOVERFLOW.COM/QUESTIONS/3190667/CONVERT-PEM-TO-PPK-FILE-FORMAT](https://stackoverflow.com/questions/3190667/convert-pem-to-ppk-file-format)



# CREATE AN INSTANCE

- CLICK ON LAUNCH INSTANCE AND SELECT THE FREE TIER ON THE LEFT NAVIGATION PANE

The screenshot displays the AWS Management Console interface for creating an EC2 instance. On the left, the navigation pane shows 'INSTANCES' expanded, with 'Free tier only' selected under 'Community AMIs'. The main content area features the 'Create Instance' tab, which includes a 'Launch Instance' button and a description of Amazon EC2 instances. Below this, a list of AMIs is shown, including 'Amazon Linux 2 AMI (HVM), SSD Volume Type' and 'Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type'.

**Left Navigation Pane:**

- Tags
- Reports
- Limits
- INSTANCES**
  - Instances
  - Launch Templates

**Create Instance Tab:**

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

**Launch Instance**

**Migrate a Machine**

Use CloudEndure Migration to simplify, expect automate large-scale migrations from physical cloud-based infrastructure to AWS.

[Get started with CloudEndure Migration](#)

**Community AMIs:**

- ☒ Free tier only ⓘ

**Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-00c03f7f7f2ec15c3 (64-bit x86) / ami-014d175d64de0a174 (Arm)**

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

**Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-0c64dd618a49ae8**

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby

# LAUNCH INSTANCE

- CHOOSE YOUR FAVORITE MACHINE IMAGE BASED ON THE OS (64 BIT, 32BIT) YOUR MACHINE IS RUNNING.
- PREFERABLY SELECT UBUNTU 18.04 64BIT IMAGE.



Amazon Linux  
Free tier eligible

**Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type** - ami-0c64dd618a49aeee8

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Select

64-bit (x86)



Red Hat  
Free tier eligible

**Red Hat Enterprise Linux 8 (HVM), SSD Volume Type** - ami-0520e698dd500b1d1 (64-bit x86) / ami-0099847d600887c9f (64-bit Arm)

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Select

☒ 64-bit (x86)

☐ 64-bit (Arm)



SUSE Linux  
Free tier eligible

**SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type** - ami-0e0bae59dc35fe89a (64-bit x86) / ami-0b49a8f443e46ff20 (64-bit Arm)

SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Select

☒ 64-bit (x86)

☐ 64-bit (Arm)



Free tier eligible

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-05c1fa8df71875112 (64-bit x86) / ami-0606a0d9f566249d3 (64-bit Arm)

Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Select

☒ 64-bit (x86)

☐ 64-bit (Arm)

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

## Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.small	2	2	EBS only	Yes	Up to 5 Gigabit	Yes
<input type="checkbox"/>	General purpose	t3a.medium	2	4	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

## LAUNCH INSTANCE CONT..

- CLICK ON REVIEW AND LAUNCH



# CONNECT YOUR INSTANCE

- ONCE YOU LAUNCHED YOUR INSTANCE,
- FOR WINDOWS USER –
  - CONNECT THE INSTANCE USING 'PUTTY'
  - A COMPLETE GUIDE ([HTTPS://STACKOVERFLOW.COM/QUESTIONS/3190667/CONVERT-PEM-TO-PPK-FILE-FORMAT](https://stackoverflow.com/questions/3190667/convert-pem-to-ppk-file-format)) CAN BE FOUND ON HOW TO CONNECT
  - HOWEVER, WHEN CONNECTING, USE THE "PUBLIC DNS / PUBLIC IP" FROM AWS AS A 'HOST NAME' ON PUTTY (DEMONSTRATED IN NEXT SLIDE)
  - KEEP THE PORT AS DEFAULT
  - CLICK ON OPEN
  - LOGIN AS "UBUNTU"

# CONNECT YOUR INSTANCE CONT..

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name
	i-03c7c2c4e7b066b24	t2.micro	us-east-2c	running	2/2 checks ...	None	ec2-18-222-188-28.us-east-2.compute.amazonaws.com	18.222.188.28	-	docker

Instance: i-03c7c2c4e7b066b24 Public DNS: ec2-18-222-188-28.us-east-2.compute.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID: i-03c7c2c4e7b066b24  
Instance state: running  
Instance type: t2.micro  
Elastic IPs: -

Public DNS (IPv4): ec2-18-222-188-28.us-east-2.compute.amazonaws.com  
IPv4 Public IP: 18.222.188.28  
Private DNS: ip-172-31-39-60.us-east-2.compute.internal

```
ubuntu@ip-172-31-39-60: ~  
login as: ubuntu  
Authenticating with public key "imported-openssh-key"  
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1048-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage
```

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port  
ec2-18-222-188-28.us-east-2.compute.ar 22

Connection type:  
☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

Load, save or delete a stored session

Saved Sessions  
docker

Default Settings  
1LinuxServer  
CS domain  
Hubble  
docker  
linuxconnect

Load Save Delete

Close window on exit:  
☐ Always ☐ Never ☒ Only on clean exit

About Help Open Cancel

## Connect To Your Instance

I would like to connect with

- ☒ A standalone SSH client ⓘ
- ☐ EC2 Instance Connect (browser-based SSH connection) ⓘ
- ☐ A Java SSH Client directly from my browser (Java required) ⓘ

To access your instance:

1. Open an SSH client. (find out how to [connect using PuTTY](#))
2. Locate your private key file (docker.pem). The wizard automatically detects the key you used to launch the instance.
3. Your key must not be publicly viewable for SSH to work. Use this command if needed:

```
chmod 400 docker.pem
```

4. Connect to your instance using its Public DNS:

```
ec2-18-222-188-28.us-east-2.compute.amazonaws.com
```

Example:

```
ssh -i "docker.pem" ubuntu@ec2-18-222-188-28.us-east-2.compute.amazonaws.com
```

Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our [connection documentation](#).

Close

# CONNECT YOUR INSTANCE

- FOR MAC USER,
  - GO TO THE TERMINAL
  - CHANGE THE MODE OF THE .PEM FILE
  - THEN CONNECT TO THE INSTANCE BY FOLLOWING THE "CONNECT TO YOUR INSTANCE"

# COPY YOUR DOCKERFILE AND ECHO FILE

- ONCE YOU LOGGED INTO THE INSTANCE,
  - IF YOU ARE A WINDOWS USER, YOU CAN USE WINSCP TO TRANSFER YOUR FILE FROM LOCAL TO REMOTE
  - DOWNLOADABLE LINK FOR WINSCP: [HTTPS://WINSKP.NET/ENG/DOWNLOAD.PHP](https://winscp.net/eng/download.php)
  - IF YOU ARE A MAC USER, YOU CAN USE “SCP” COMMAND TO TRANSFER YOUR FILE FROM LOCAL TO REMOTE ([HTTPS://LINUXIZE.COM/POST/HOW-TO-USE-SCP-COMMAND-TO-SECURELY-TRANSFER-FILES/](https://linuxize.com/post/how-to-use-scp-command-to-securely-transfer-files/))
  - EXAMPLE: SCP FILE.TXT [REMOTE USERNAME@10.10.0.2:/REMOTE/DIRECTORY](#)
- UPON TRANSFERRING THE FILES FROM LOCAL TO REMOTE -
  - IN THE EC2 INSTANCE, SIMPLY CHANGE THE DIRECTORY TO THE FOLDER WHICH CONTAINS THE DOCKER FILE
  - BUILD THE DOCKER IMAGE
  - RUN YOUR ECHO SERVER



ubuntu@ip-172-31-39-60: ~/Assignment

```
ubuntu@ip-172-31-39-60:~/Assignment$ docker build -t docker-python-echoserver-app .  
Sending build context to Docker daemon 7.168kB  
Step 1/7 : FROM python:3.6  
--> 1c515a624542  
Step 2/7 : WORKDIR /app  
--> Using cache  
--> d6f0dd07e6e2  
Step 3/7 : COPY src/requirements.txt ./  
--> Using cache  
--> 06da4a9991aa  
Step 4/7 : RUN pip install -r requirements.txt  
--> Using cache  
--> eeb8alc932e2  
Step 5/7 : COPY src /app  
--> 4c8474023ela  
Step 6/7 : EXPOSE 80  
--> Running in 143d8b7e9956  
Removing intermediate container 143d8b7e9956  
--> 9e926c5d5a08  
Step 7/7 : CMD ["python", "echo-server.py"]  
--> Running in 2ff748b540b4  
Removing intermediate container 2ff748b540b4  
--> dc146d069f6c  
Successfully built dc146d069f6c  
Successfully tagged docker-python-echoserver-app:latest  
ubuntu@ip-172-31-39-60:~/Assignment$ docker run --rm -it -p 80:80 docker-python-echoserver-app  
Listening on 0.0.0.0:80 for HTTP connection  
█
```

## RUN ECHO SERVER IN DOCKER

# PING AND TELNET YOUR IP

```
ubuntu@ip-172-31-39-60:~$ ping 3.19.143.89
PING 3.19.143.89 (3.19.143.89) 56(84) bytes of data.
64 bytes from 3.19.143.89: icmp_seq=1 ttl=63 time=0.282 ms
64 bytes from 3.19.143.89: icmp_seq=2 ttl=63 time=0.395 ms
64 bytes from 3.19.143.89: icmp_seq=3 ttl=63 time=0.379 ms
64 bytes from 3.19.143.89: icmp_seq=4 ttl=63 time=0.414 ms
64 bytes from 3.19.143.89: icmp_seq=5 ttl=63 time=0.553 ms
64 bytes from 3.19.143.89: icmp_seq=6 ttl=63 time=0.348 ms
64 bytes from 3.19.143.89: icmp_seq=7 ttl=63 time=0.339 ms
^C
--- 3.19.143.89 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 615lms
rtt min/avg/max/mdev = 0.282/0.387/0.553/0.079 ms
ubuntu@ip-172-31-39-60:~$ telnet 3.19.143.89 80
Trying 3.19.143.89...
Connected to 3.19.143.89.
Escape character is '^]'.
█
```

- PING <YOUR\_IP\_ADDRESS>
- TELNET <YOUR\_IP\_ADDRESS>  
<PORT>
- YOU SHOULD BE ABLE TO SEE  
A MESSAGE THAT IS:  
“CONNECTED  
<YOUR\_IP\_ADDRESS>”
- SO, THE ECHO-SERVER CAN BE  
ACCESSIBLE FROM ELSEWHERE

# ECHOED BACK FROM THE SERVER AFTER TELNET

```
ubuntu@ip-172-31-39-60: ~/Assignment
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1048-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Sep 18 18:44:31 UTC 2019

System load:  0.0           Processes:           99
Usage of /:   67.8% of 7.69GB Users logged in:       0
Memory usage: 24%          IP address for eth0: 172.31.39.60
Swap usage:   0%           IP address for docker0: 172.17.0.1

 * Congrats to the Kubernetes community on 1.16 beta 1! Now available
   in MicroK8s for evaluation and testing, with upgrades to RC and GA

   snap info microk8s

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

48 packages can be updated.
0 updates are security updates.

Last login: Wed Sep 18 17:53:40 2019 from 68.229.97.243
ubuntu@ip-172-31-39-60:~$ ls
Assignment      forMLN          opencv.zip
CS531-WebServerDesign  get-docker.sh   opencv_contrib
Dockerfile      opencv          opencv_contrib.zip
foo.bar         opencv-text-recognition
ubuntu@ip-172-31-39-60:~$ cd Assignment/
ubuntu@ip-172-31-39-60:~/Assignment$ docker run --rm -it -p 80:80 docker-python-echoserver-app
Listening on 0.0.0.0:80 for HTTP connection
Connected to: ('3.19.143.89', 55880)
```

```
ubuntu@ip-172-31-39-60: ~
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1048-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Sep 18 18:45:27 UTC 2019

System load:  0.05          Processes:           107
Usage of /:   67.8% of 7.69GB Users logged in:       1
Memory usage: 29%          IP address for eth0: 172.31.39.60
Swap usage:   0%           IP address for docker0: 172.17.0.1

 * Congrats to the Kubernetes community on 1.16 beta 1! Now available
   in MicroK8s for evaluation and testing, with upgrades to RC and GA

   snap info microk8s

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

48 packages can be updated.
0 updates are security updates.

Last login: Wed Sep 18 18:44:32 2019 from 68.229.97.243
ubuntu@ip-172-31-39-60:~$ ping 3.19.143.89
PING 3.19.143.89 (3.19.143.89) 56(84) bytes of data.
64 bytes from 3.19.143.89: icmp_seq=1 ttl=63 time=0.335 ms
64 bytes from 3.19.143.89: icmp_seq=2 ttl=63 time=0.326 ms
64 bytes from 3.19.143.89: icmp_seq=3 ttl=63 time=0.431 ms
64 bytes from 3.19.143.89: icmp_seq=4 ttl=63 time=0.347 ms
64 bytes from 3.19.143.89: icmp_seq=5 ttl=63 time=0.355 ms
64 bytes from 3.19.143.89: icmp_seq=6 ttl=63 time=0.459 ms
64 bytes from 3.19.143.89: icmp_seq=7 ttl=63 time=0.355 ms
64 bytes from 3.19.143.89: icmp_seq=8 ttl=63 time=0.310 ms
^C
--- 3.19.143.89 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7159ms
rtt min/avg/max/mdev = 0.310/0.364/0.459/0.054 ms
ubuntu@ip-172-31-39-60:~$ telnet 3.19.143.89 80
Trying 3.19.143.89...
Connected to 3.19.143.89.
Escape character is '^]'.

```