

Assignment 1: EC2 and Instance

- Logging into AWS (with Account ID)
- Launch a Virtual Machine (VM) with EC2
- Launch an Instance
- Installing Packages, user and root user
- Installing tree
- Using tree

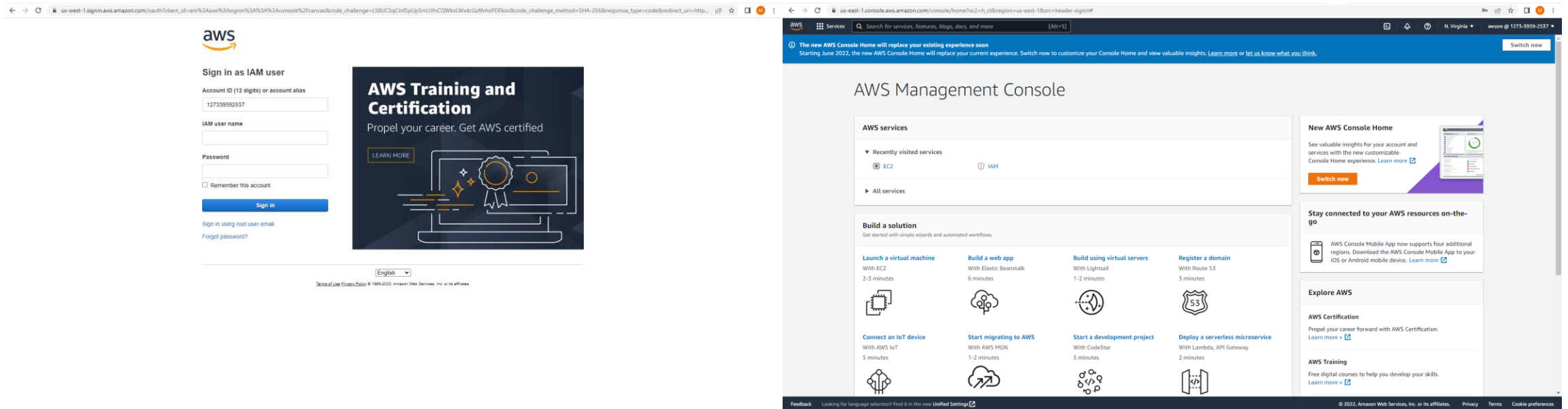
Logging into AWS (with Account ID)

Log in link: <https://console.aws.amazon.com/console/home>

Important link and account ID to remember: <https://127339592537.signin.aws.amazon.com/console>

Account ID: 127339592537 (same as code number in the above link)

IAM Username: awssre



Launch a Virtual Machine (VM) with EC2

Search for “EC2” at the top of services search bar. Should arrive at this page.

Next, create EC2 instance by clicking on **Launch Instance**

The screenshot shows the AWS Management Console for the EC2 service in the US East (N. Virginia) region. The interface includes a left-hand navigation menu, a central content area with several panels, and a right-hand sidebar with account and explore sections.

Left Navigation Menu:

- New EC2 Experience
- EC2 Dashboard
- EC2 Global View
- Events
- Tags
- Limits
- Instances
 - Instances (selected)
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Scheduled Instances
 - Capacity Reservations
- Images
 - AMIs
 - AMI Catalog
- Elastic Block Store
 - Volumes
 - Snapshots
 - Lifecycle Manager
- Network & Security
 - Security Groups
 - Elastic IPs

Central Content Area:

- Resources:** A summary of EC2 resources in the region. A table shows: Instances (running): 0, Dedicated Hosts: 0, Elastic IPs: 0, Instances: 1, Key pairs: 1, Load balancers: 0, Placement groups: 0, Security groups: 3, Snapshots: 0, Volumes: 1.
- Launch instance:** A section with a "Launch instance" button and a "Migrate a server" link. A note states: "Your instances will launch in the US East (N. Virginia) Region."
- Scheduled events:** A section showing "US East (N. Virginia)" with "No scheduled events".
- Migrate a server:** A section at the bottom.
- Service health:** A section showing the region "US East (N. Virginia)" with a status of "This service is operating normally".
- Zones:** A table listing availability zones and their IDs.

Zone name	Zone ID
us-east-1a	use1-az4
us-east-1b	use1-az6
us-east-1c	use1-az1
us-east-1d	use1-az2

Right Sidebar:

- Account attributes:** Includes links for "Supported platforms" (VPC), "Default VPC" (vpc-08c4253edb0f4bc29), "Settings" (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments).
- Explore AWS:** Contains promotional cards for "Enable Best Price-Performance with AWS Graviton2", "Save up to 90% on EC2 with Spot Instances", and "Get Up to 40% Better Price Performance".
- Additional information:** A section at the bottom of the sidebar.

Footer: © 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Launch an Instance: Settings

Arriving at this page, key in desired name. Name should be meaningful.

- **Application and OS Images (Amazon Machine Image):** Amazon Linux > Latest Kernel version (Free tier eligible)
- **Instance type:** t2.micro (1GB memory, Free tier eligible)
- **Key pair login:** Refer to next slide
- **Rest of the settings:** Default
- Click “Launch instance”
- Make sure it is successful
- Click on “View all instances”

The screenshot shows the AWS Management Console interface for launching an EC2 instance. The browser address bar indicates the URL: `us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstances:`. The console header shows the AWS logo, 'Services' menu, a search bar, and the user's account information (N. Virginia, awsre @ 1273-3959-2537).

The main content area is titled 'Launch an instance' with a sub-header 'Launch an instance Info'. Below this, there's a section for 'Name and tags' with a text input field containing 'matthewlio_ec2' and a link to 'Add additional tags'. The 'Application and OS Images (Amazon Machine Image)' section is expanded, showing a search bar and a 'Quick Start' tab. Under 'Quick Start', there are buttons for 'Amazon Linux', 'Ubuntu', 'Windows', 'Red Hat', and 'SUSE Linux'. The 'Amazon Linux' button is selected, showing details for 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type' with a 'Free tier eligible' badge. A 'Free tier' notification box is visible on the right, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet'. The 'Summary' panel on the right shows 'Number of instances' set to 1, 'Software Image (AMI)' as 'Amazon Linux 2 Kernel 5.10 AMI...', 'Virtual server type (instance type)' as 't2.micro', 'Firewall (security group)' as 'New security group', and 'Storage (volumes)' as '1 volume(s) - 8 GiB'. At the bottom right, there are 'Cancel' and 'Launch instance' buttons.

Launch an Instance: New key pair

- Key pair name: something meaningful
- For now, use .pem

Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Key pair name

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ RSA
RSA encrypted private and public key pair

☐ ED25519
ED25519 encrypted private and public key pair (Not supported for Windows instances)

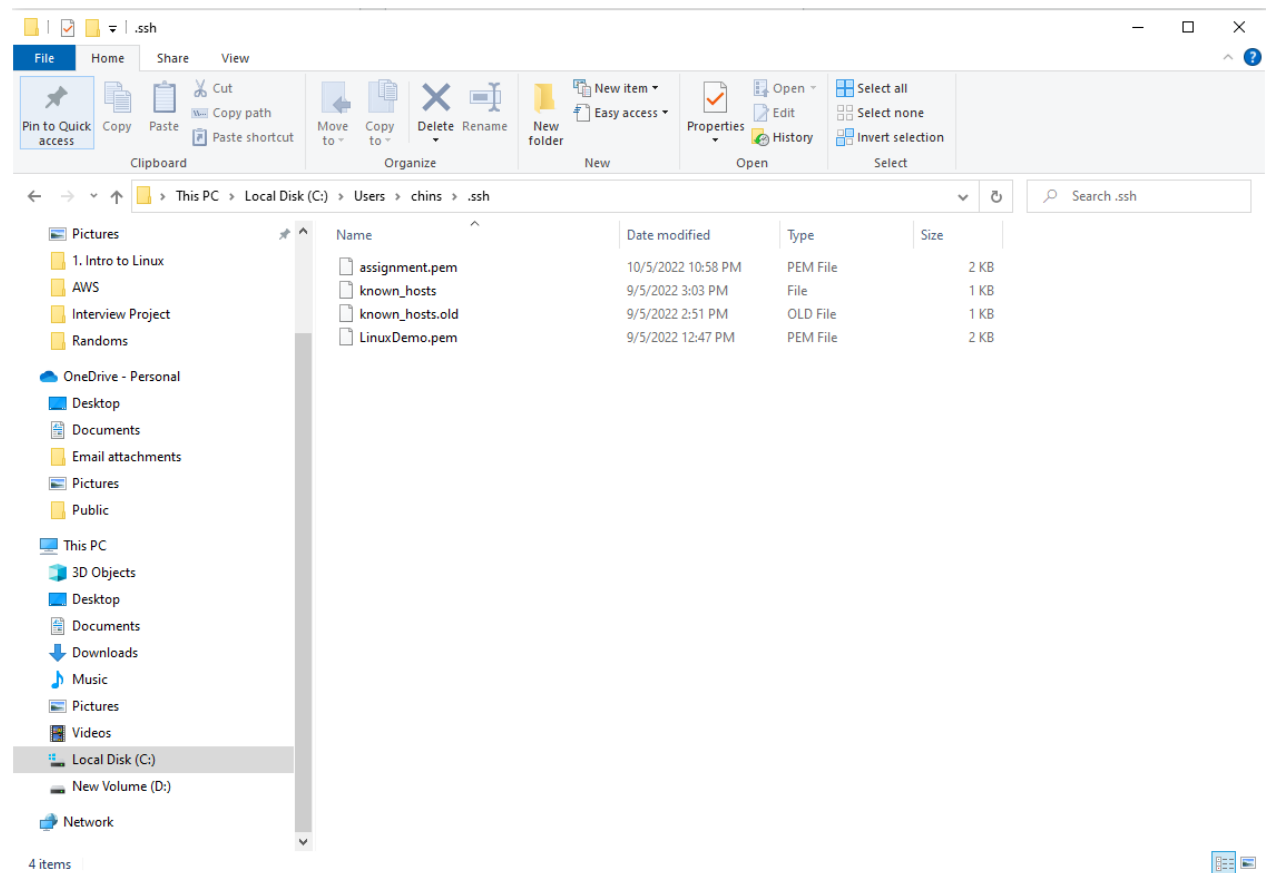
Private key file format

☒ .pem
For use with OpenSSH

☐ .ppk
For use with PuTTY

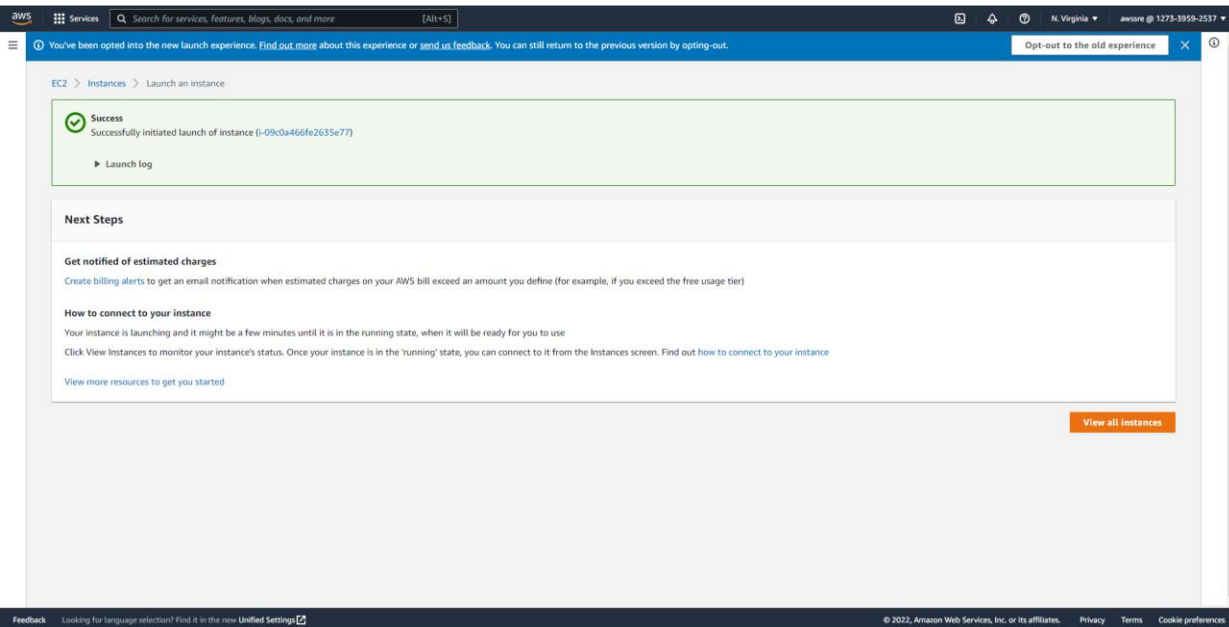
Cancel Create key pair

- After clicking “Create key pair”, computer will download key file
- Move key file to .ssh folder shown below (or your working folder)

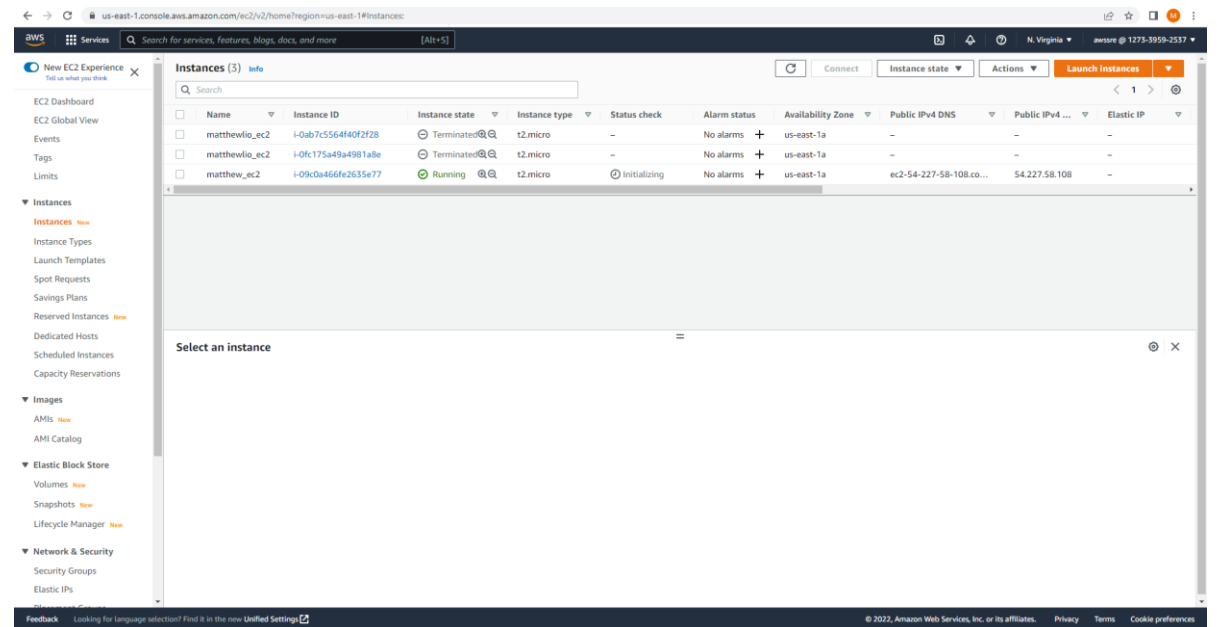


Launch an Instance: Successful

- Make sure it is successful
- Click on “View all instances”



- Instance state: Make sure it is running
- Status check: Wait while it is “Initializing”



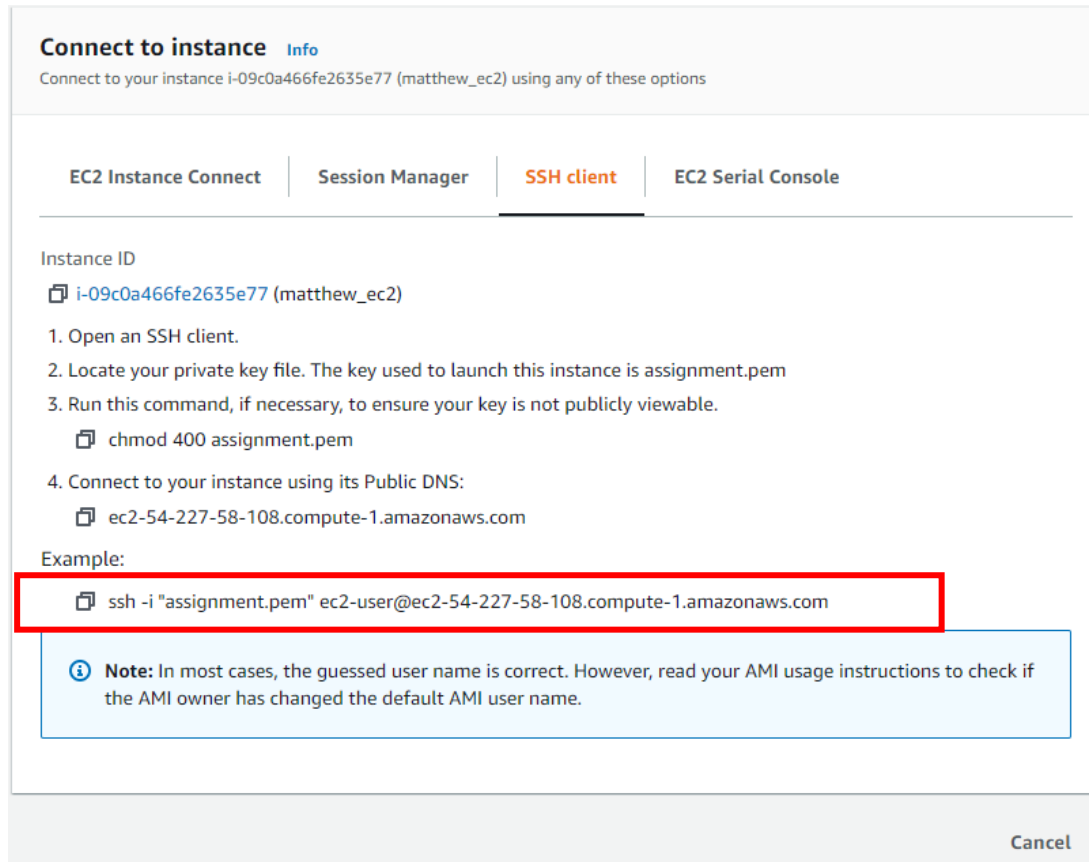
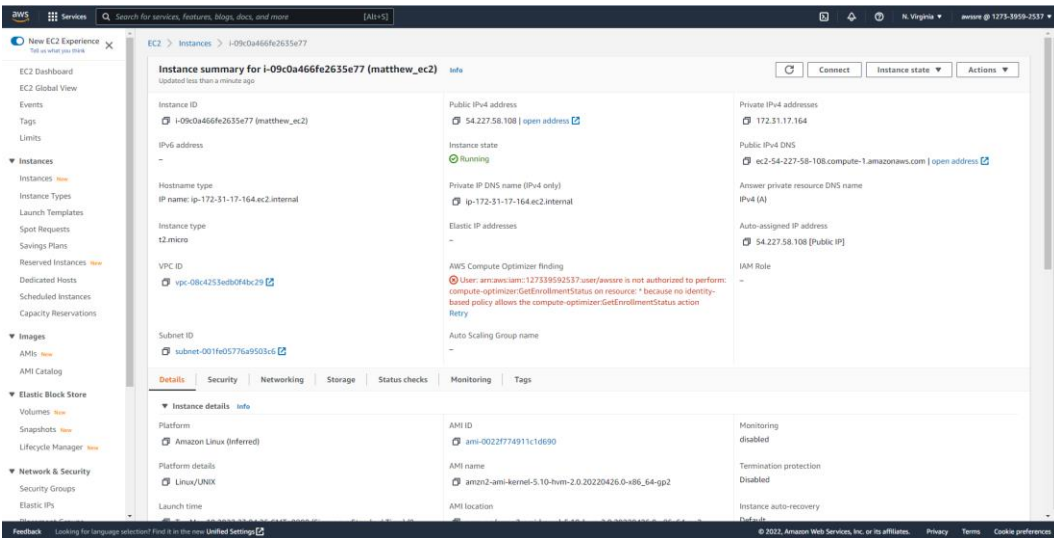
- After initializing, it should look like this



- Click on your instance

Launch an Instance: Connect

- Click on “Connect”
- Copy the “Example” string in the red box as shown



- Continuing connection on the next slide
- Open Git Bash terminal

Launch an Instance: Connect

- Using “cd” command, move to where you stored the key file that was downloaded (.ssh folder)
- Paste the “SSH Client” string that you have copied earlier into the terminal

```
chins@DESKTOP-BJ2N998 MINGW64 ~/.ssh
$ ssh -i "assignment.pem" ec2-user@ec2-54-227-58-108.compute-1.amazonaws.com
The authenticity of host 'ec2-54-227-58-108.compute-1.amazonaws.com (54.227.58.108)' can't be established.
ED25519 key fingerprint is SHA256:NvapzW7nqnFWV++6jTE8/Ji6Zjmk+Ep//6SyYdcrIyo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? |
```

- Type “yes” to continue connecting
- If unsuccessful: type **chmod 400 xxx.pem**
- Below shows how it should look like if it is successful. Note the red box when it is successful.

```
chins@DESKTOP-BJ2N998 MINGW64 ~/.ssh
$ ssh -i "assignment.pem" ec2-user@ec2-54-227-58-108.compute-1.amazonaws.com

 _ | _ | _ )
 _ | ( /   /   Amazon Linux 2 AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 2 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-17-164 ~]$ |
```


Installing Packages, user and root user

- To install packages in Linux, type: yum install **package**
- **Package**: package name
- But you can only install packages as root user. Type: sudo -i
- Take note the differences below between normal user and root user (ec2-user and root, \$ and #)

```
[ec2-user@ip-172-31-17-164 ~]$ sudo -i  
[root@ip-172-31-17-164 ~]# |
```

Installing tree

- Yum install tree

```
[ec2-user@ip-172-31-17-164 ~]$ sudo -i
[root@ip-172-31-17-164 ~]# sudo -i
[root@ip-172-31-17-164 ~]# yum install tree
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
Resolving Dependencies
--> Running transaction check
---> Package tree.x86_64 0:1.6.0-10.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
tree x86_64 1.6.0-10.amzn2.0.1 amzn2-core 47 k
=====

Transaction Summary
=====
Install 1 Package

Total download size: 47 k
Installed size: 83 k
Is this ok [y/d/N]: y
Downloading packages:
tree-1.6.0-10.amzn2.0.1.x86_64.rpm | 47 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : tree-1.6.0-10.amzn2.0.1.x86_64 1/1
  Verifying : tree-1.6.0-10.amzn2.0.1.x86_64 1/1

Installed:
tree.x86_64 0:1.6.0-10.amzn2.0.1

Complete!
[root@ip-172-31-17-164 ~]# |
```

Using tree package

- Tree package from Google:
- It outputs the directory paths and files in each sub-category and a summary of a total number of sub-directories and files.
- Below, we print out the /opt directory: cd /opt, then tree
- Red box shows the commands used

```
[root@ip-172-31-17-164 ~]# cd /opt
[root@ip-172-31-17-164 opt]# tree

aws
├── apitools
│   ├── cfn-init -> ./cfn-init-2.0-10
│   └── cfn-init-2.0-10
│       ├── bin
│       │   ├── cfn-elect-cmd-leader
│       │   ├── cfn-get-metadata
│       │   ├── cfn-hup
│       │   ├── cfn-init
│       │   ├── cfn-send-cmd-event
│       │   ├── cfn-send-cmd-result
│       │   └── cfn-signal
│       ├── init
│       │   ├── redhat
│       │   │   └── cfn-hup
│       │   ├── systemd
│       │   │   └── cfn-hup.service
│       │   └── ubuntu
│       │       └── cfn-hup
│       └── share
│           └── doc
│               └── aws-cfn-bootstrap-2.0
│                   ├── CHANGELOG.txt
│                   ├── LICENSE.txt
│                   └── NOTICE.txt
└── bin
    ├── cfn-elect-cmd-leader -> ../apitools/cfn-init/bin/cfn-elect-cmd-leader
    ├── cfn-get-metadata -> ../apitools/cfn-init/bin/cfn-get-metadata
    ├── cfn-hup -> ../apitools/cfn-init/bin/cfn-hup
    ├── cfn-init -> ../apitools/cfn-init/bin/cfn-init
    ├── cfn-send-cmd-event -> ../apitools/cfn-init/bin/cfn-send-cmd-event
    ├── cfn-send-cmd-result -> ../apitools/cfn-init/bin/cfn-send-cmd-result
    ├── cfn-signal -> ../apitools/cfn-init/bin/cfn-signal
    ├── ec2-metadata -> /usr/bin/ec2-metadata
    ├── eic_curl_authorized_keys
    ├── eic_harvest_hostkeys
    ├── eic_parse_authorized_keys
    └── eic_run_authorized_keys

rh

14 directories, 25 files
[root@ip-172-31-17-164 opt]#
```

Assignment 2: Uninstalling packages

- Uninstalling tree package with erase

Uninstalling tree package with erase

- Uninstalling tree package: `yum erase tree -y`
- `-y`: to automatically “yes” the command

```
[root@ip-172-31-17-164 opt]# yum erase tree -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
---> Package tree.x86_64 0:1.6.0-10.amzn2.0.1 will be erased
--> Finished Dependency Resolution

amzn2-core/2/x86_64 | 3.7 kB 00:00:00

Dependencies Resolved

=====
Package                Arch          Version              Repository            Size
=====
Removing:
tree                   x86_64        1.6.0-10.amzn2.0.1   @amzn2-core           83 k
=====

Transaction Summary
=====
Remove 1 Package

Installed size: 83 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Erasing      : tree-1.6.0-10.amzn2.0.1.x86_64        1/1
  Verifying    : tree-1.6.0-10.amzn2.0.1.x86_64        1/1

Removed:
  tree.x86_64 0:1.6.0-10.amzn2.0.1

Complete!
[root@ip-172-31-17-164 opt]# |
```

- No more tree package to use

```
[root@ip-172-31-17-164 opt]# tree
-bash: /bin/tree: No such file or directory
[root@ip-172-31-17-164 opt]# |
```

Assignment 3: Creating and printing new files

- Making directories with mkdir command
- Creating new files with echo command
- Printing content in files with cat command
- Echo and cat together

Making directories with mkdir command

- Here, we create new folder called “demolinux” in the /opt directory, using the mkdir command
- With ls command, we can see that the folder has been created
- We then enter (using cd command) into “demolinux” folder

```
[root@ip-172-31-17-164 opt]# mkdir demolinux
[root@ip-172-31-17-164 opt]# ls
aws  demolinux  rh
[root@ip-172-31-17-164 opt]# cd demolinux
[root@ip-172-31-17-164 demolinux]# |
```

Creating new files with echo command

- With echo command, we create a new text file called abc.txt
- The content of the file is just a simple string "abc"

echo "abc" > abc.txt

Command echo

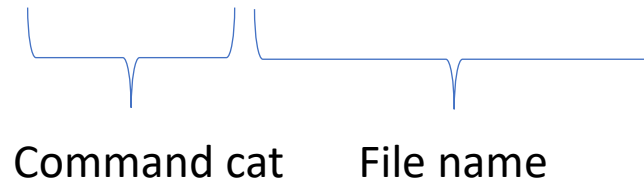
Content
string

File name

```
[root@ip-172-31-17-164 opt]# cd demolinux
[root@ip-172-31-17-164 demolinux]# ls
[root@ip-172-31-17-164 demolinux]# echo "abc" > abc.txt
[root@ip-172-31-17-164 demolinux]# ls
abc.txt
[root@ip-172-31-17-164 demolinux]# |
```


Printing content in files with cat command

cat abc.txt



Command cat File name

- Printing content of abc.txt file that we created earlier
- Content shown in red box

```
[root@ip-172-31-17-164 opt]# cd demolinux
[root@ip-172-31-17-164 demolinux]# ls
[root@ip-172-31-17-164 demolinux]# echo "abc" > abc.txt
[root@ip-172-31-17-164 demolinux]# ls
abc.txt
[root@ip-172-31-17-164 demolinux]# cat abc.txt
abc
[root@ip-172-31-17-164 demolinux]# |
```

Echo and cat together

- Using both echo and cat commands, create “abc1.txt” file and print content

```
[root@ip-172-31-17-164 demolinux]# ls
abc.txt
[root@ip-172-31-17-164 demolinux]# echo "abc1" > abc1.txt
[root@ip-172-31-17-164 demolinux]# ls
abc1.txt  abc.txt
[root@ip-172-31-17-164 demolinux]# cat abc1.txt
abc1
[root@ip-172-31-17-164 demolinux]# |
```

Assignment 4: Extra basic ls commands

- `ls -ltr` command
- `ls -ltra` command

Ls -ltr command

- WHEN IN DOUBT: use manual page command: `man ls`
- `-l` : List all files in the long listing format view
- `-t` : sort by modification time, newest first
- `-r`: reverse order while sorting

```
[root@ip-172-31-17-164 demolinux]# ls
abc1.txt  abc.txt
[root@ip-172-31-17-164 demolinux]# ls -ltr
total 8
-rw-r--r-- 1 root root 4 May 10 16:22 abc.txt
-rw-r--r-- 1 root root 5 May 10 16:42 abc1.txt
[root@ip-172-31-17-164 demolinux]# |
```

Ls -ltra command

- WHEN IN DOUBT: use manual page command: `man ls`
- `-l` : List all files in the long listing format view
- `-t` : sort by modification time, newest first
- `-r`: reverse order while sorting
- `-a`: all files, including parent directory and current directory ("`..`" is parent directory, "`.`" is current directory)

```
[root@ip-172-31-17-164 demolinux]# ls
abc1.txt  abc.txt
[root@ip-172-31-17-164 demolinux]# ls -ltr
total 8
-rw-r--r-- 1 root root 4 May 10 16:22 abc.txt
-rw-r--r-- 1 root root 5 May 10 16:42 abc1.txt
[root@ip-172-31-17-164 demolinux]# ls -ltra
total 8
drwxr-xr-x 5 root root 44 May 10 16:08 ..
-rw-r--r-- 1 root root  4 May 10 16:22 abc.txt
-rw-r--r-- 1 root root  5 May 10 16:42 abc1.txt
drwxr-xr-x 2 root root 37 May 10 16:42 .
[root@ip-172-31-17-164 demolinux]# |
```

Assignment 5: Display a HTTP Web Page

- Allow HTTP traffic (existing instance)
- Allow HTTP traffic (when creating new instance)
- Installing httpd package
- Creating HTTP file
- Starting httpd service (and using chkconfig)
- Public IP Address
- HTML Webpage

Allow HTTP traffic (existing instance)

- Go to instance page. View under Security tab
- Under Security groups, click available security group (in red box)

EC2 > Instances > i-09c0a466fe2635e77

Instance summary for i-09c0a466fe2635e77 (matthew_ec2) Info

Instance ID: i-09c0a466fe2635e77 (matthew_ec2)

Public IPv4 address: 54.227.58.108 | [open address](#)

Instance state: Running

Private IPv4 addresses: 172.31.17.164

Public IPv4 DNS: ec2-54-227-58-108.compute-1.amazonaws.com | [open address](#)

Private IP DNS name (IPv4 only): ip-172-31-17-164.ec2.internal

Elastic IP addresses: -

Auto-assigned IP address: 54.227.58.108 [Public IP]

IAM Role: -

AWS Compute Optimizer finding: ⚠ User: arn:aws:iam::127339592537:user/awsre is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action [Retry](#)

Auto Scaling Group name: -

Security details

IAM Role: -

Owner ID: 127339592537

Launch time: Tue May 10 2022 23:04:26 GMT+0800 (Singapore Standard Time)

Security groups (highlighted in red box)

sg-041eaefeda3ba34ff (launch-wizard-4)

Inbound rules

- Click Edit inbound rules

EC2 > Security Groups > sg-041eaefeda3ba34ff - launch-wizard-4

sg-041eaefeda3ba34ff - launch-wizard-4 Actions

Details

Security group name: launch-wizard-4

Security group ID: sg-041eaefeda3ba34ff

Description: launch-wizard created 2022-05-10T14:56:00.211Z

VPC ID: vpc-08c4253edb0f4bc29

Owner: 127339592537

Inbound rules count: 1 Permission entry

Outbound rules count: 1 Permission entry

Inbound rules | Outbound rules | Tags

You can now check network connectivity with Reachability Analyzer [Run Reachability Analyzer](#)

Inbound rules (1/1) [Manage tags](#) [Edit inbound rules](#) (highlighted in red box)

Filter security group rules

<input checked="" type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input checked="" type="checkbox"/>	-	sg-01092590318266...	IPv4	SSH	TCP	22	0.0.0.0/0	-

Allow HTTP traffic (existing instance)

BEFORE



Editing inbound rules

- Add rule

The screenshot shows the AWS Management Console 'Edit inbound rules' page for security group sg-041eafeda3ba34ff. The page lists one existing rule for SSH on port 22. The 'Add rule' button at the bottom left is highlighted with a red rectangle. The 'Save rules' button at the bottom right is also visible.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sg-r-01092590318266558	SSH	TCP	22	Custom	

AFTER



- Type: HTTP
- 0.0.0.0/0
- Save rules

The screenshot shows the same AWS Management Console 'Edit inbound rules' page, but now with two rules. A new rule for HTTP on port 80 has been added. The 'Type' dropdown for the new rule is highlighted with a red rectangle, and the 'Source' dropdown is also highlighted. The 'Save rules' button at the bottom right is highlighted with a red rectangle.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sg-r-01092590318266558	SSH	TCP	22	Custom	
-	HTTP	TCP	80	Anywhere-I	

Allow HTTP traffic (when creating new instance)

Allowing HTTP traffic when creating new instance

- Check the appropriate box

▼ Network settings

Edit

Network

vpc-08c4253edb0f4bc29

Subnet

No preference (Default subnet in any availability zone)

Auto-assign public IP

Enable

Security groups (Firewall) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

We'll create a new security group called 'launch-wizard-5' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere
0.0.0.0/0

☐ Allow HTTPs traffic from the internet

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

×

Installing httpd package

- As root user, install httpd package
- `sudo yum install -y httpd`

```
[root@ip-172-31-17-164 demolinux]# sudo yum install -y httpd
```

- Installation in progress...

```
[root@ip-172-31-17-164 demolinux]# sudo yum install -y httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.7 kB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.53-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.53-1.amzn2 for package: httpd-2.4.53-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.53-1.amzn2 for package: httpd-2.4.53-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.53-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.53-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.53-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.53-1.amzn2.x86_64
--> Processing Dependency: libaprutil1.so.0(64bit) for package: httpd-2.4.53-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0(64bit) for package: httpd-2.4.53-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed
--> Package httpd-filesystem.noarch 0:2.4.53-1.amzn2 will be installed
--> Package httpd-tools.x86_64 0:2.4.53-1.amzn2 will be installed
--> Package mailcap.noarch 0:2.1.41-2.amzn2 will be installed
--> Package mod_http2.x86_64 0:1.15.19-1.amzn2.0.1 will be installed
--> Running transaction check
--> Package apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

```
Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
httpd x86_64 2.4.53-1.amzn2 amzn2-core 1.3 M
Installing for dependencies:
apr x86_64 1.7.0-9.amzn2 amzn2-core 122 k
apr-util x86_64 1.6.1-5.amzn2.0.2 amzn2-core 99 k
apr-util-bdb x86_64 1.6.1-5.amzn2.0.2 amzn2-core 19 k
generic-logos-httpd noarch 18.0.0-4.amzn2 amzn2-core 19 k
httpd-filesystem noarch 2.4.53-1.amzn2 amzn2-core 24 k
httpd-tools x86_64 2.4.53-1.amzn2 amzn2-core 88 k
mailcap noarch 2.1.41-2.amzn2 amzn2-core 31 k
mod_http2 x86_64 1.15.19-1.amzn2.0.1 amzn2-core 149 k
=====

Transaction Summary
Install 1 Package (+8 Dependent packages)

Total download size: 1.9 M
Installed size: 5.2 M
Downloading packages:
(1/9): apr-1.7.0-9.amzn2.x86_64.rpm | 122 kB 00:00:00
(2/9): apr-util-1.6.1-5.amzn2.0.2.x86_64.rpm | 99 kB 00:00:00
(3/9): apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64.rpm | 19 kB 00:00:00
(4/9): generic-logos-httpd-18.0.0-4.amzn2.noarch.rpm | 19 kB 00:00:00
(5/9): httpd-filesystem-2.4.53-1.amzn2.noarch.rpm | 24 kB 00:00:00
(6/9): httpd-2.4.53-1.amzn2.x86_64.rpm | 1.3 MB 00:00:00
(7/9): httpd-tools-2.4.53-1.amzn2.x86_64.rpm | 88 kB 00:00:00
(8/9): mailcap-2.1.41-2.amzn2.noarch.rpm | 31 kB 00:00:00
(9/9): mod_http2-1.15.19-1.amzn2.0.1.x86_64.rpm | 149 kB 00:00:00
-----
Total 8.3 MB/s | 1.9 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : apr-1.7.0-9.amzn2.x86_64 1/9
Installing : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 2/9
Installing : apr-util-1.6.1-5.amzn2.0.2.x86_64 3/9
Installing : httpd-tools-2.4.53-1.amzn2.x86_64 4/9
Installing : generic-logos-httpd-18.0.0-4.amzn2.noarch 5/9
Installing : mailcap-2.1.41-2.amzn2.noarch 6/9
Installing : httpd-filesystem-2.4.53-1.amzn2.noarch 7/9
Installing : mod_http2-1.15.19-1.amzn2.0.1.x86_64 8/9
Installing : httpd-2.4.53-1.amzn2.x86_64 9/9
Verifying : apr-util-1.6.1-5.amzn2.0.2.x86_64 1/9
Verifying : apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 2/9
Verifying : mod_http2-1.15.19-1.amzn2.0.1.x86_64 3/9
Verifying : httpd-filesystem-2.4.53-1.amzn2.noarch 4/9
Verifying : httpd-tools-2.4.53-1.amzn2.x86_64 5/9
Verifying : mailcap-2.1.41-2.amzn2.noarch 6/9
Verifying : generic-logos-httpd-18.0.0-4.amzn2.noarch 7/9
Verifying : httpd-2.4.53-1.amzn2.x86_64 8/9
Verifying : apr-1.7.0-9.amzn2.x86_64 9/9

Installed:
httpd.x86_64 0:2.4.53-1.amzn2

Dependency Installed:
apr.x86_64 0:1.7.0-9.amzn2 apr-util.x86_64 0:1.6.1-5.amzn2.0.2
apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2 generic-logos-httpd.noarch 0:18.0.0-4.amzn2
httpd-filesystem.noarch 0:2.4.53-1.amzn2 httpd-tools.x86_64 0:2.4.53-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2 mod_http2.x86_64 0:1.15.19-1.amzn2.0.1

Complete!
[root@ip-172-31-17-164 demolinux]#
```

- Installation complete!

Creating HTTP file

Creating HTML file using echo command

- echo "<html><body><h1>Welcome! </br> Design & Developed by Matt</h1></body></html>" > /var/www/html/index.html
- HTML file: /var/www/html/index.html

```
[root@ip-172-31-31-168 ~]# echo "<html><body><h1>Welcome! </br> Design & Developed by Matt</h1></body></html>" > /var/www/html/index.html
[root@ip-172-31-31-168 ~]# cat /var/www/html/index.html
<html><body><h1>Welcome! </br> Design & Developed by Matt</h1></body></html>
[root@ip-172-31-31-168 ~]# |
```

Starting httpd service (and using chkconfig)

Necessary commands for starting httpd service

- `sudo service httpd start`
- `sudo chkconfig httpd on`

```
[root@ip-172-31-31-168 ~]# sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-31-168 ~]# sudo chkconfig httpd on
Note: Forwarding request to 'systemctl enable httpd.service'.
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-31-168 ~]# |
```

Public IP Address

To view our HTTP webpage, we first need to obtain our public IP address

- On AWS Console, go to our instance page
- Our public IP address is shown in the red box
- Copy this IP address
- Open a new tab, and paste the IP address in the address bar

The screenshot displays the AWS Management Console interface for an EC2 instance. The breadcrumb navigation at the top shows 'EC2 > Instances > i-0bc7260f620a52185'. The main header for the instance summary is 'Instance summary for i-0bc7260f620a52185 (matt-ec2)' with an 'Info' link and a 'Updated 3 minutes ago' timestamp. On the right side of the header, there are buttons for 'Refresh', 'Connect', 'Instance state' (with a dropdown arrow), and 'Actions' (with a dropdown arrow). The instance details are organized into three columns. The first column on the left lists attributes: Instance ID (i-0bc7260f620a52185 (matt-ec2)), IPv6 address (none), Hostname type (IP name: ip-172-31-31-168.ec2.internal), Instance type (t2.micro), VPC ID (vpc-08c4253edb0f4bc29), and Subnet ID (subnet-001fe05776a9503c6). The middle column shows: Public IPv4 address (3.87.80.99 | open address), Instance State (Running), Private IP DNS name (ip-172-31-31-168.ec2.internal), Elastic IP addresses (none), AWS Compute Optimizer finding (a warning message about permissions), and Auto Scaling Group name (none). The third column on the right shows: Private IPv4 addresses (172.31.31.168), Public IPv4 DNS (ec2-3-87-80-99.compute-1.amazonaws.com | open address), Answer private resource DNS name (IPv4 (A)), Auto-assigned IP address (3.87.80.99 [Public IP]), and IAM Role (none). A red rectangular box is drawn around the 'Public IPv4 address' section in the middle column, specifically highlighting the IP address 3.87.80.99 and the 'open address' link.

Attribute	Value
Instance ID	i-0bc7260f620a52185 (matt-ec2)
IPv6 address	None
Hostname type	IP name: ip-172-31-31-168.ec2.internal
Instance type	t2.micro
VPC ID	vpc-08c4253edb0f4bc29
Subnet ID	subnet-001fe05776a9503c6
Public IPv4 address	3.87.80.99 open address
Instance State	Running
Private IP DNS name (IPv4 only)	ip-172-31-31-168.ec2.internal
Elastic IP addresses	None
AWS Compute Optimizer finding	⚠ User: arn:aws:iam::127339592537:user/awssre is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action Retry
Auto Scaling Group name	None
Private IPv4 addresses	172.31.31.168
Public IPv4 DNS	ec2-3-87-80-99.compute-1.amazonaws.com open address
Answer private resource DNS name	IPv4 (A)
Auto-assigned IP address	3.87.80.99 [Public IP]
IAM Role	None

HTML Webpage



⚠ Not secure | 3.87.80.99

Welcome!

Design & Developed by Matt