

Multi-Tier Cloud Deployment Statement of Work

Prime Directive Technologies

Patrick R. Richardson
Prime Directive Technologies
100 Cloud Way
Ten Buck Two, OK 90210

Table of Contents

Project Overview.....	3
Project Goals & Objectives.....	3
Network Infrastructure Requirement.....	4
Assumptions & Risks.....	4
Network Configuration & Procedural Checklist.....	4
Network Topology Diagram with IP Addresses.....	5
Appendices Deliverables.....	6
Creating an EC2 Instance & Installation of a LAMP Stack.....	6
Creating an EC2 instance.....	12
Connecting to your EC2 instance via SSH.....	16
SSH into EC2 instance.....	18
Installation of MariaDB.....	19
Configuration of MariaDB.....	19
Accessing MariaDB.....	21
Creating The Database.....	22
Configuration of the SeedDMS Application Server.....	24
Connecting to the EC2 Instance.....	24
Apache installation.....	25
Installation of PHP.....	27
Installation and Configuration of SeedDMS.....	28
SeedDMS Installation Tool.....	31
Launching 2 additional EC2 instances from the AMI.....	37
SeedDMS Browser Configuration.....	44
Configuration of the Document Root file.....	47

Adding the Installation Tool.....	47
Accessing the Configuration File.....	48
Editing the Configuration file.....	49
Restarting Apache.....	50
HAProxy Server.....	53
Creating the EC2 Instance.....	53
SSH Into Instance.....	57
Installation of HAProxy.....	58
Verification of Correct Installation.....	58
Creation of Self-Signed SSL Certification.....	59
Enable and start HAProxy.....	60
Creating a Single-Sign SSL.....	61
Step 83.....	61
Configuring HAProxy to use the SSL Certificate.....	63
Failover and Cookies Test.....	66
Testing the Cookies Persistence.....	67
Failover Test.....	68
References.....	70

Scope of Work

Project Title

Multi-Tier Cloud Deployment

Project Overview

The project aims to design a multi-tier cloud-hosted management system using Amazon Web Services (AWS). This will consist of five AWS Amazon Linux EC2 instances. It will include a HAProxy server as the load balancer to distribute incoming traffic to the three independent application servers. The application servers will consist of 3 AWS EC2 instances running Amazon Linux and will include SeedDMS, Apache, and PHP for document management. All three application servers will gather and store information on an AWS EC2 instance with Amazon Linux, with MariaDB for the database server.

Milestone completion dates:

Milestone I – September 7, 2026

Milestone II – September 14, 2026

Milestone III – September 21, 2026

Milestone IV – September 27, 2026

Project Goals & Objectives

The goal is to implement a cloud-based document management system that provides session persistence to the application layer that is for a centralized documentation and management.

- Speed - The network must provide enough bandwidth to meet demand during periods of peak usage.
- Reliability - The network must contain redundancy to ensure that the failure of one application server will not bring the entire network down.

- Scalability - The network must scale without rebuilding the entire network. Changes and updates must be easy to implement.
- Security – With AWS handling the physical side of security, SeedDMS allows for system user security.

Network Infrastructure Requirement

Your choice of cloud provider and an internet connection.

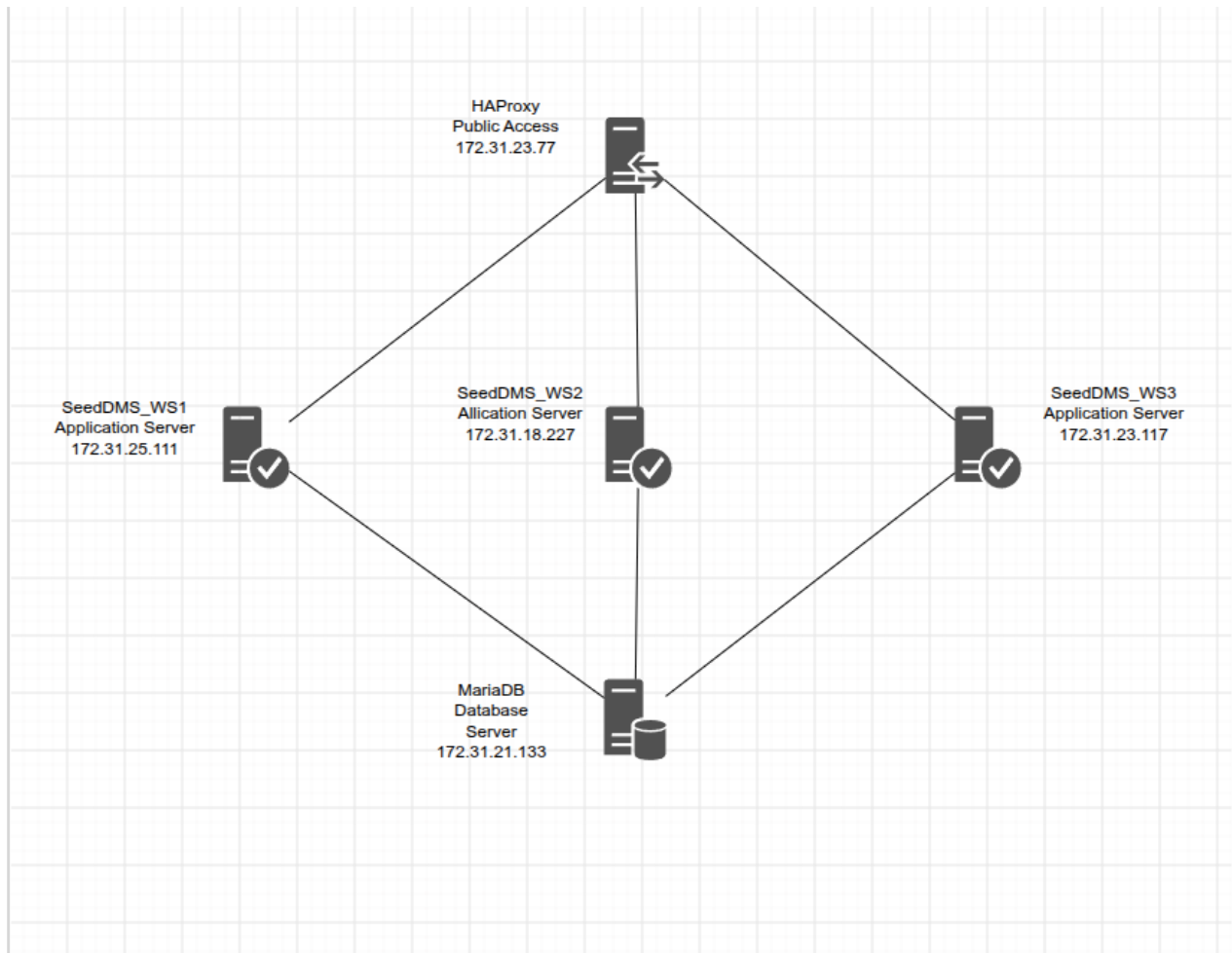
Assumptions & Risks

- AWS will provide a constant and available service.
- An internet connection is to be provided by the client
- The client will provide an internal IT team to deal with user-related issues.
- The client's internal IT team will handle all maintenance and security needs.

Network Configuration & Procedural Checklist

1. Create 5 AWS EC2 instance servers
2. Install and configure MariaDB database software on one server
3. Install and configure SeedDMS and Apache on 3 servers
4. Install HAProxy on the last server
5. Install SSL on the HAProxy server
6. Create a self-signed SSL certificate on the HAProxy server
7. Reconfigure the HAProxy configuration file to allow for cookies
8. Test all three servers for session persistence and fail over.

Network Topology Diagram with IP Addresses

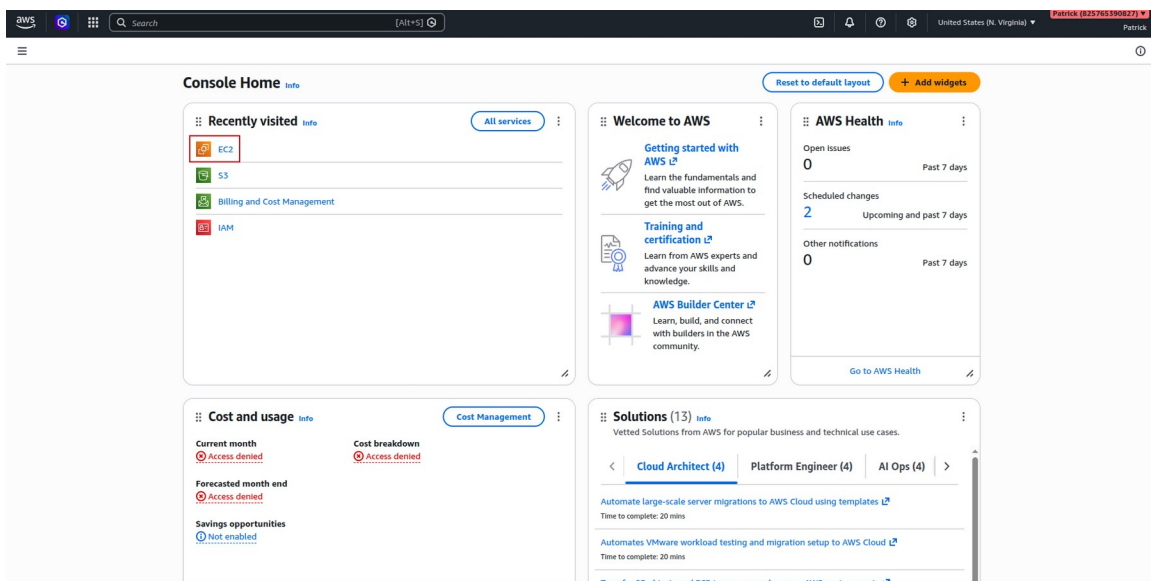


Appendices | Deliverables

Creating an EC2 Instance & Installation of a LAMP Stack

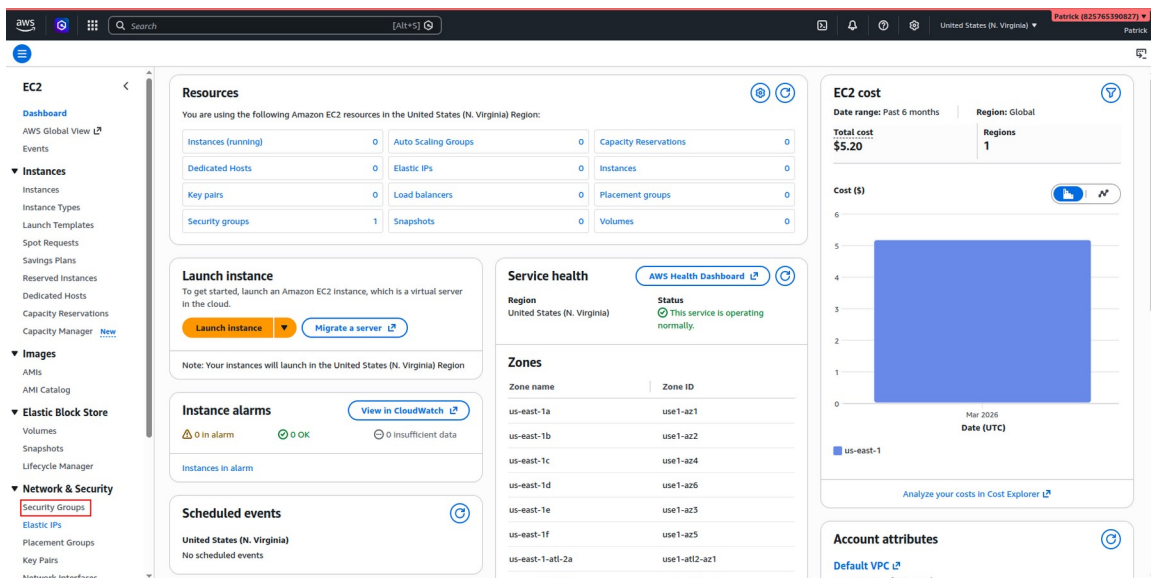
Step 1

Go to AWS.com and sign in. When you open it, it will look similar to the image to the bellow. Select the **EC2** service outlined in red in the image to the right.



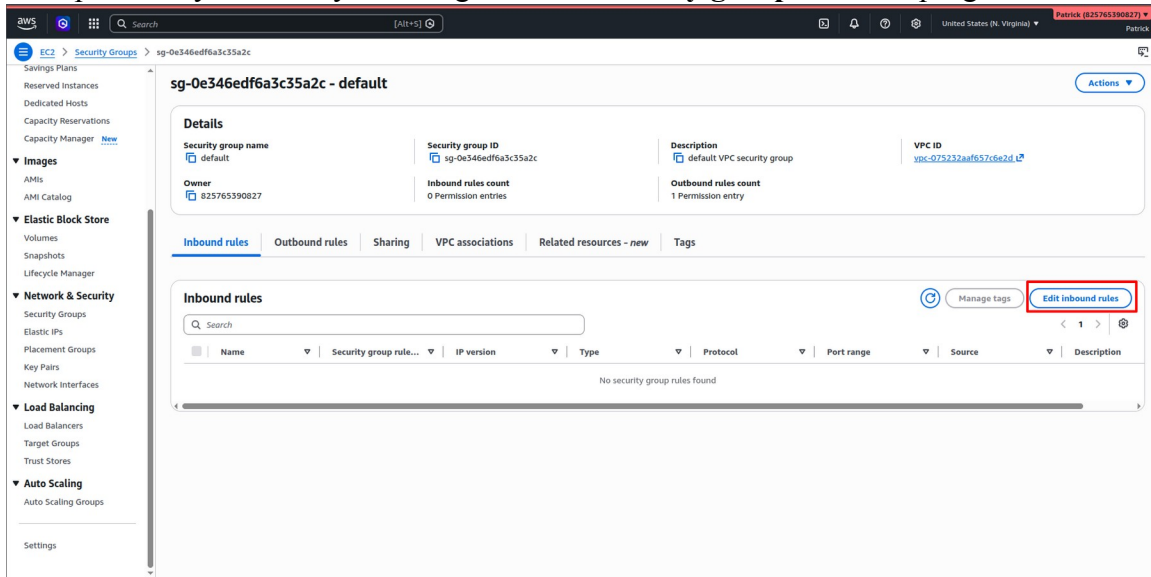
Step 2

Next we are going to create a security group for the instance. In this screen you will follow the list on the left down to where it says “**Security Groups**” and click on it.



Step 3

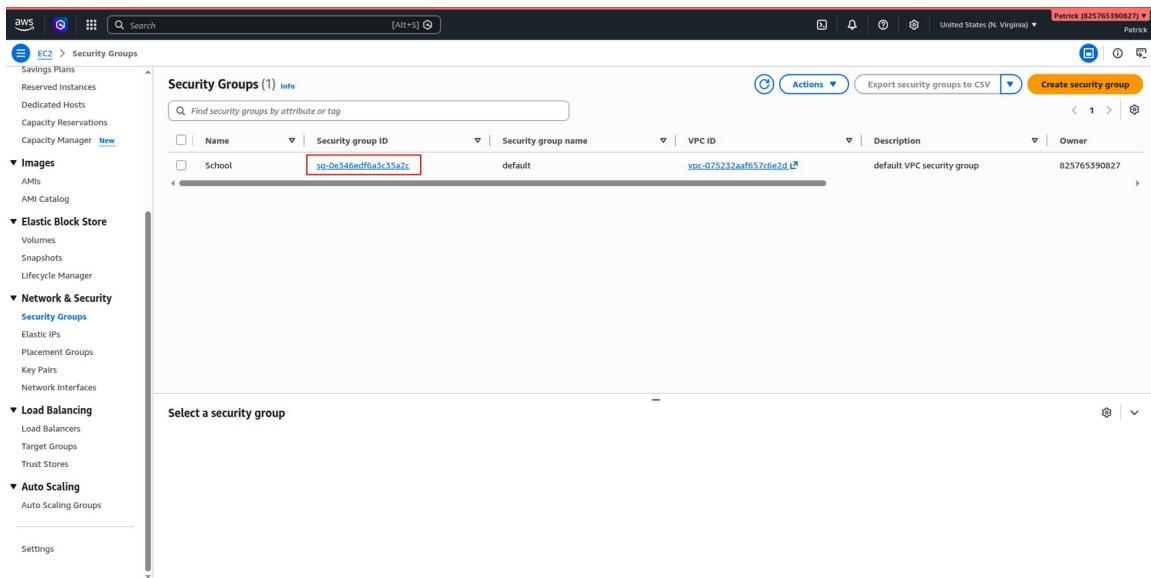
At this screen you will see your default security group in the middle. You can create a new one specifically for this by selecting **“Create security group”** in the top right corner.



Note: Best practice is to not use the default security group. Create a new one for the project at hand. This helps protect each project.

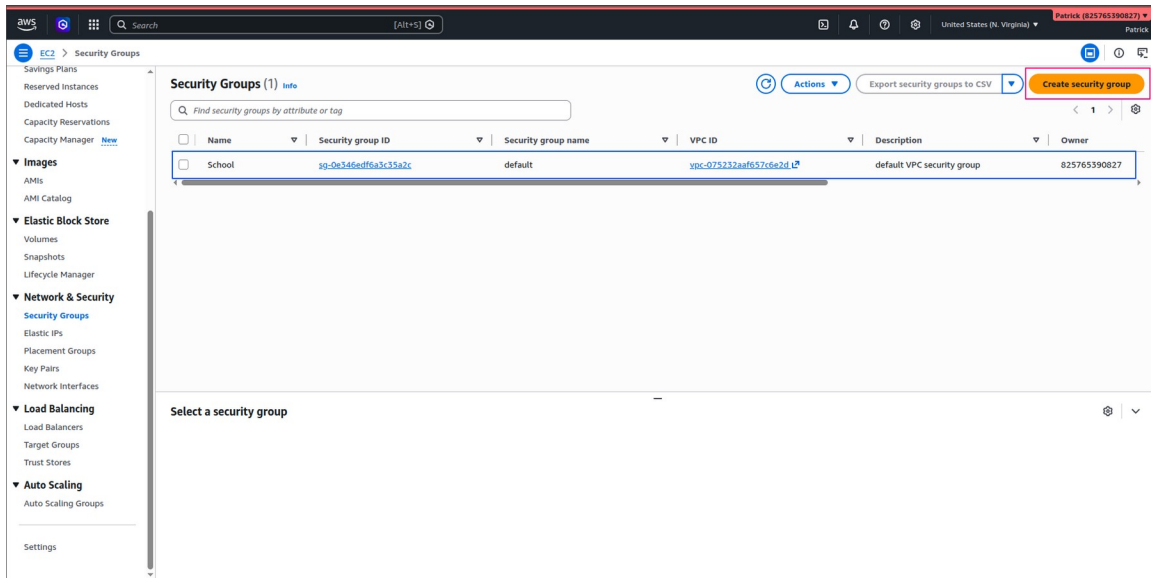
Step 4

Next we are going to add security rules. To start click on the security group **ID name**.



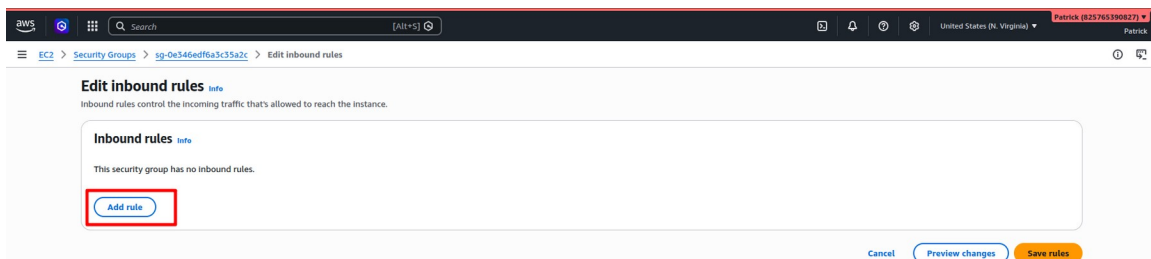
Step 5

Here we are going to add security rules. Click on the **“Edit inbound rules”** on the far right.



Step 6

At this screen you will click **“Add rule”**.



Step 7

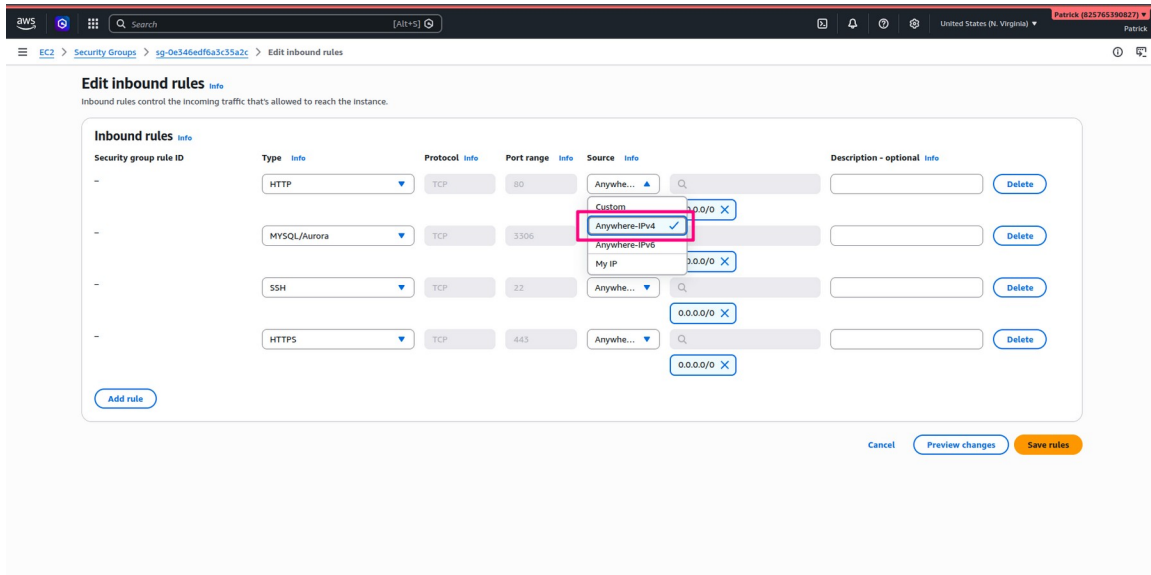
Here we will be adding multiple rules. The steps are the same for each rule. You are going to add 3 rules. HTTP, MYSQL/Aurora and SSH. By doing the following for each one.

The screenshot shows the 'Edit inbound rules' interface in the AWS IAM console. The 'Type' dropdown menu is open, showing 'Custom TCP' selected. The 'Protocol' is set to 'TCP', 'Port range' is '0', and 'Source' is 'Custom'. The 'Add rule' button is at the bottom left.

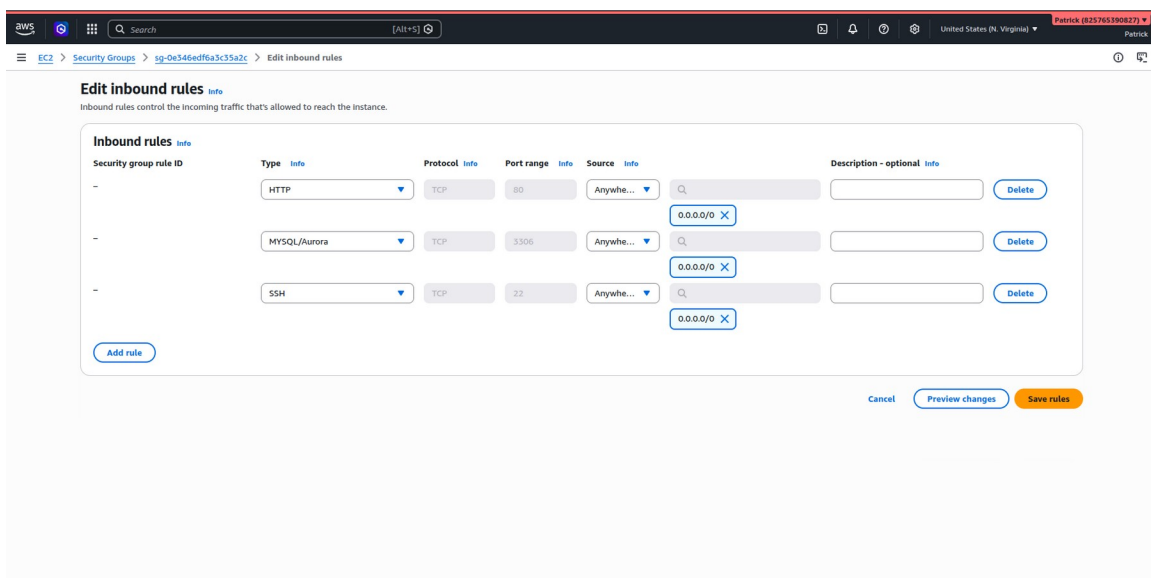
- a) Select the Type by clicking the “**Custom TCP**” and scroll down until you see **HTTP** and click on it.

The screenshot shows the 'Edit inbound rules' interface with the 'Type' dropdown menu open. The list of protocols is visible, and 'HTTP' is selected at the bottom of the list, indicated by a pink box and a checkmark. The 'Protocol' is set to 'TCP', 'Port range' is '80', and 'Source' is 'Anywhere...'. The 'Delete' button is at the bottom right.

- b) Next you will select “**Custom**” under the Source column. And scroll down to where it says “**Anywhere-IPv4**”.

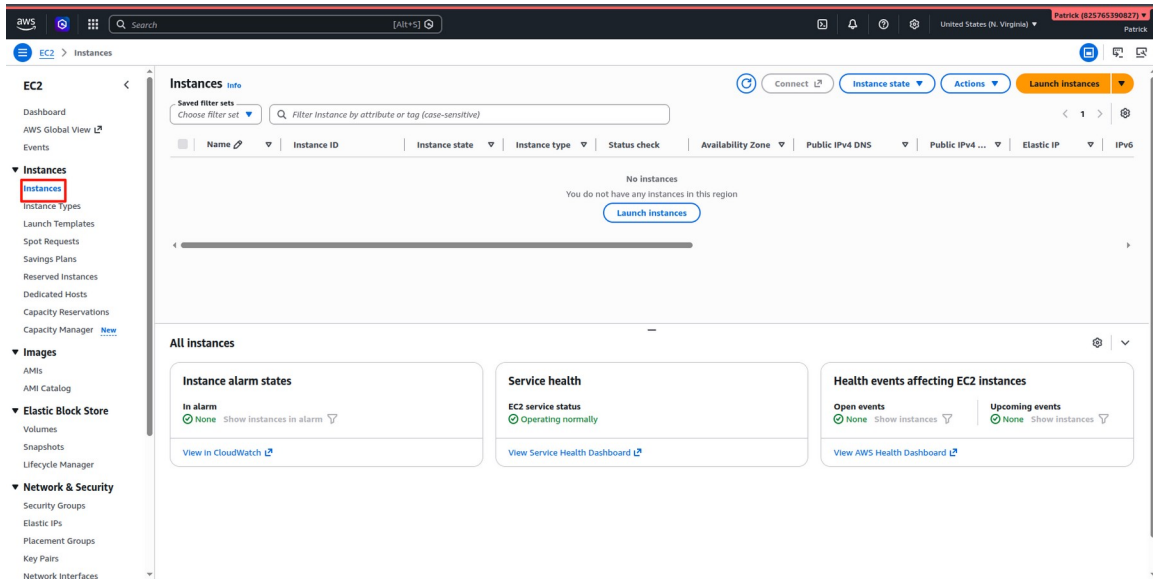


- c) Note: You will follow the same steps for the other 2 rules. MYSQL/Aurora and SSH. At the end it should look like this. Image to the right. Then click the **Save rules** at the bottom right.



Step 8

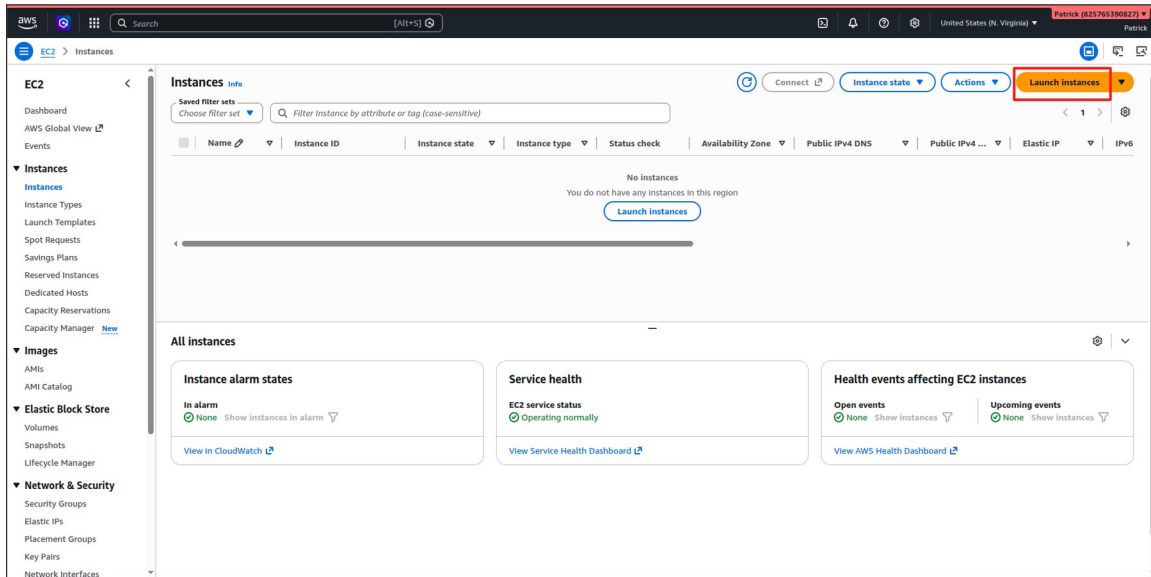
Getting to the EC2 page. Click on the Instance on the left.



Creating an EC2 instance

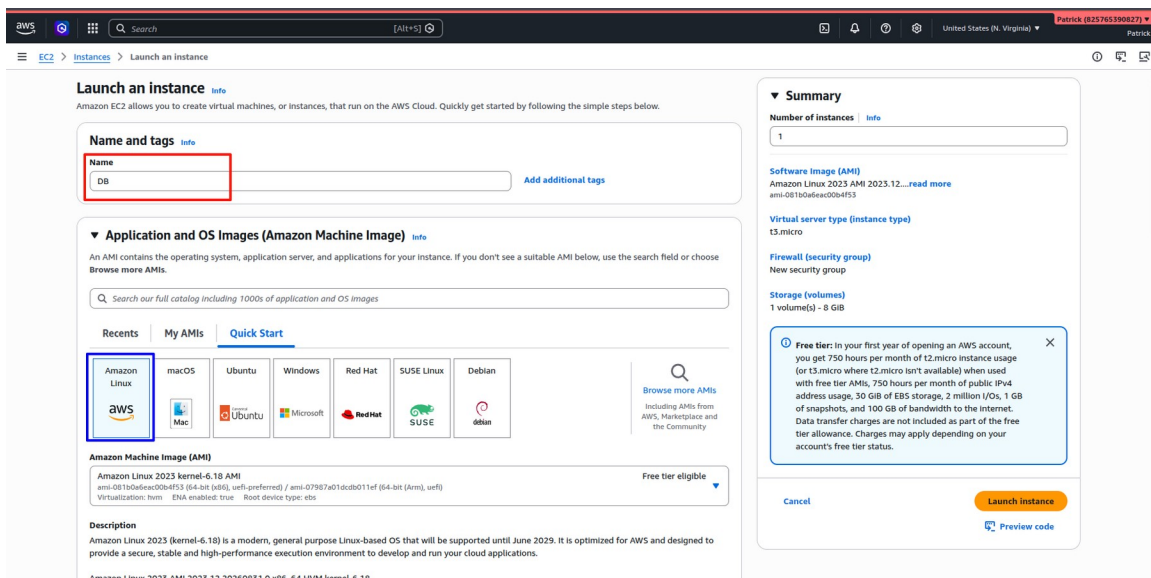
Step 9

To create an EC2 instance click “**Launch instance**” in the top right.



Step 10

Give the server a name. Helpful if you pick one that is relevant to its purpose.



Step 11

To make 2 or more EC2 instances that have the same configuration. On the right, right under “**Summary**” you can select how many to create.

The screenshot shows the AWS Management Console 'Launch an instance' page. The 'Summary' section on the right has a 'Number of instances' field set to 1. The 'Application and OS Images' section on the left shows 'Amazon Linux 2023 kernel-6.10 AMI' selected. A 'Free tier' notification is visible.

Step 12

Create a key pair. Click on “**Create new key pair**”. This authenticates the SSH login.

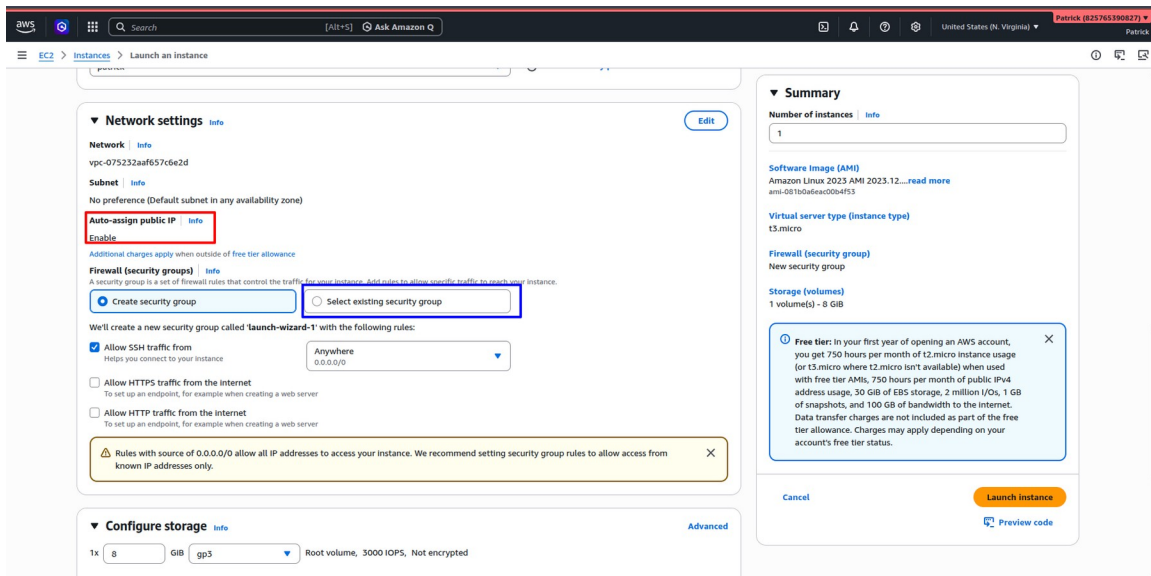
- Note: This step will pop up a download page for the private key “.pem”. Write down or remember where you put it, we will need it later.

The screenshot shows the AWS Management Console 'Launch an instance' page. The 'Key pair (login)' section shows 'Create new key pair' button highlighted. The 'Network settings' section shows 'Create security group' button highlighted. The 'Summary' section on the right shows 'Number of instances' set to 1.

Step 13

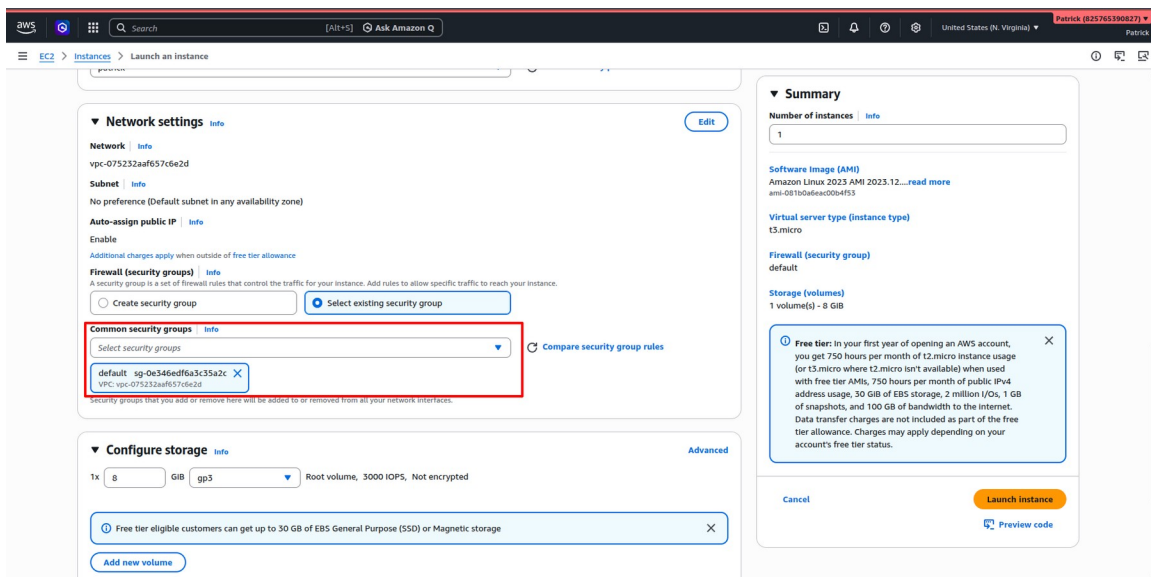
For the networking you want to make sure the “**Auto-assign public IP**” is enabled.

- Note: The IP address is auto assigned when launched. A new IP is created if the instance is stopped and then re-started.



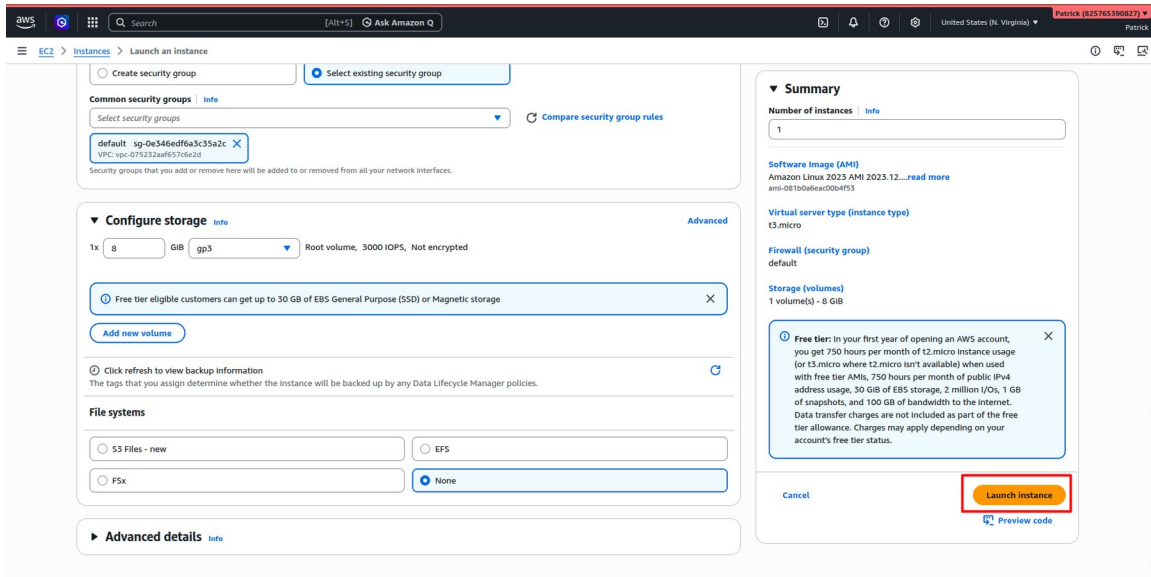
Next you want to make sure that you select “**existing security group**”.

- In the drop down menu, select the security group that you made earlier.



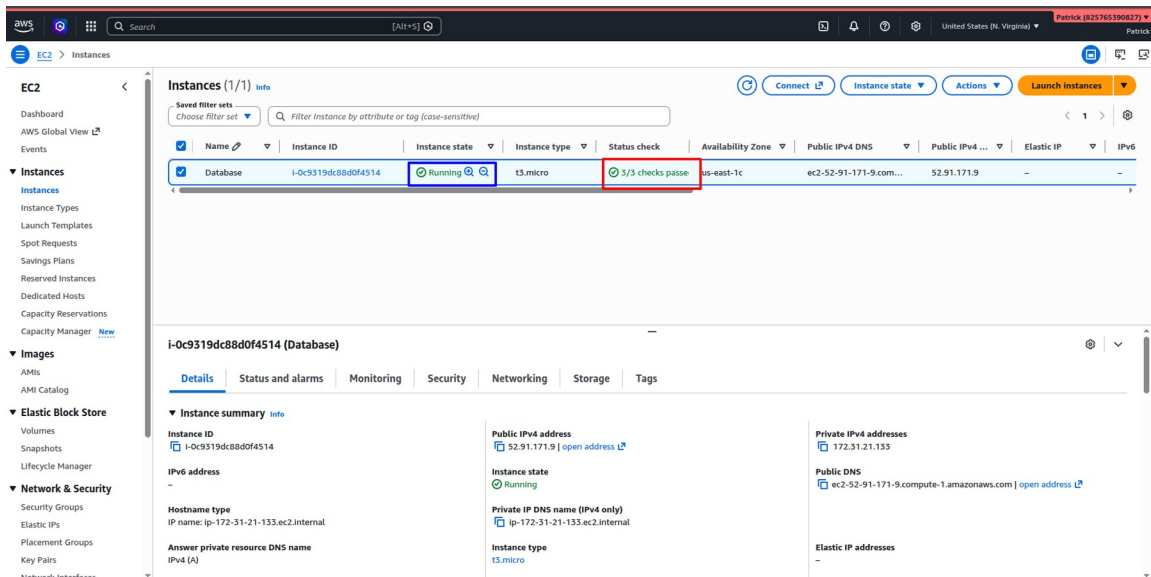
Step 14

Leave everything else as the default settings and click on “**Launch instance**” in the bottom right corner.



Step 15

Then you will need to make sure you get 3/3 checks passed. This shows that the instances are checked and running properly.



Note: If you did not create more than one EC2 instance, then you will need to follow the previous steps for each EC2 instance that you need.

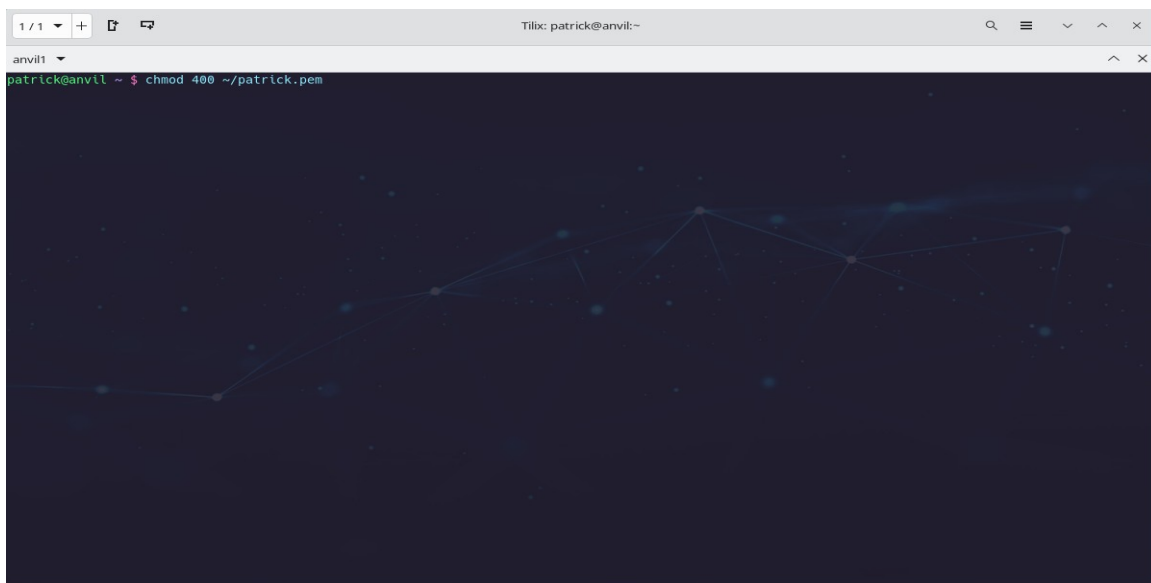
Connecting to your EC2 instance via SSH.

Note: Make sure you are in the directory where you saved your .pem key. To navigate to that directory, enter `cd` followed by the folder's name.

Step 16

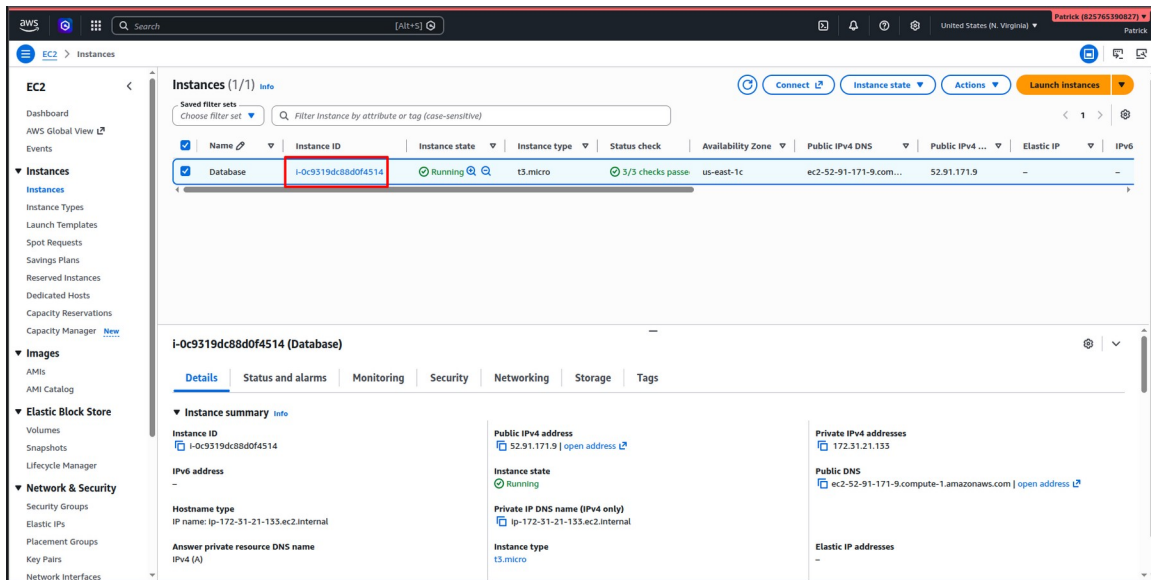
First we are going to secure the .pem file. To do this run “`chmod 400 ~/your file.pem`.”

- **Note:** This only makes the file where user can only read and everyone else and groups have zero access.

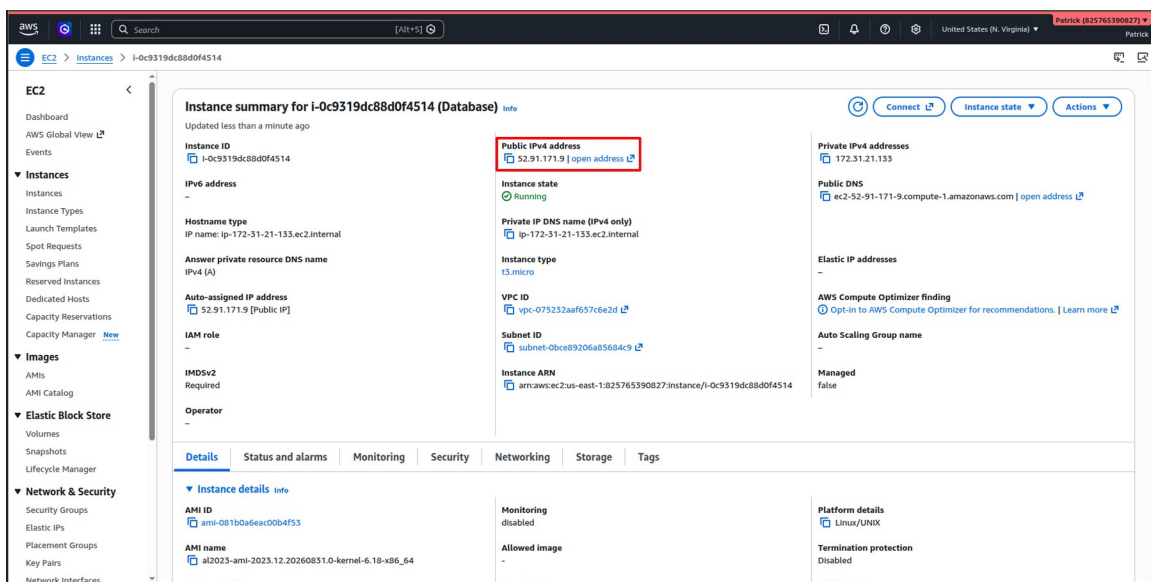


Step 17

Now we need to know what the EC2 instance IP address is. To do this, you need to go back to AWS and click on your EC2 instance.



A) Then in the middle of the screen you should see
“Public IPv4 address” the number under it is the Public IP. Copy or write that
number down.

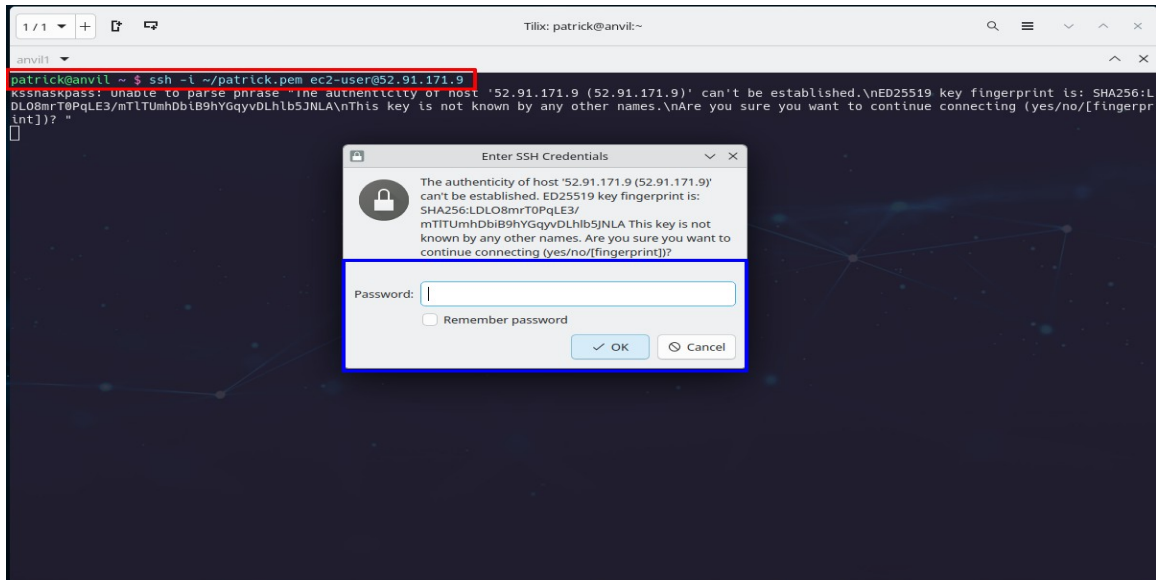


SSH into EC2 instance

Step 18

Now run the command “**ssh -i ~/file.pem ec2-user@public IP**”.

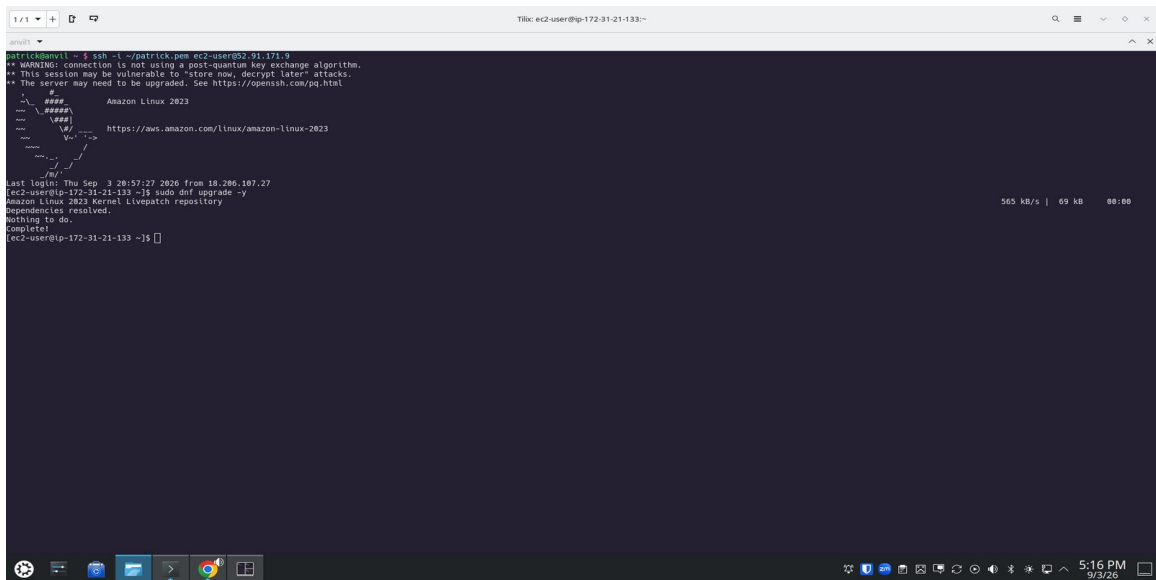
- The prompt is asking for approval, **not** your password. Type yes.



Step 19

Next we need to update the system. Run command “**sudo dnf upgrade -y**”

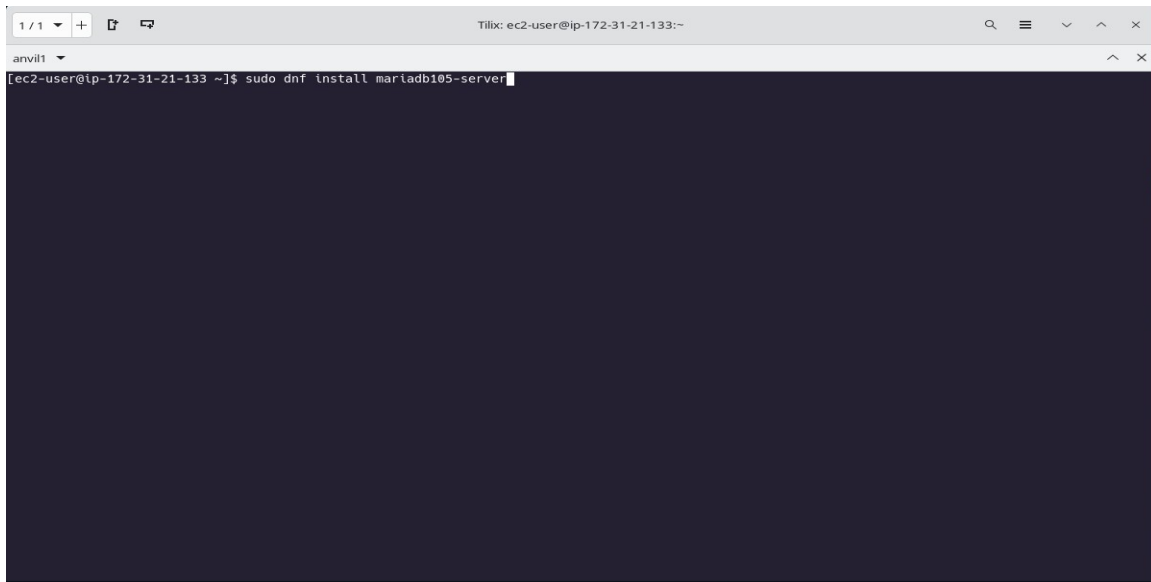
After the upgrade is done your screen should look like the picture to the left.



Installation of MariaDB

Step 20

To install MariaDB use the following command “**sudo dnf install mariadb105-server**”.

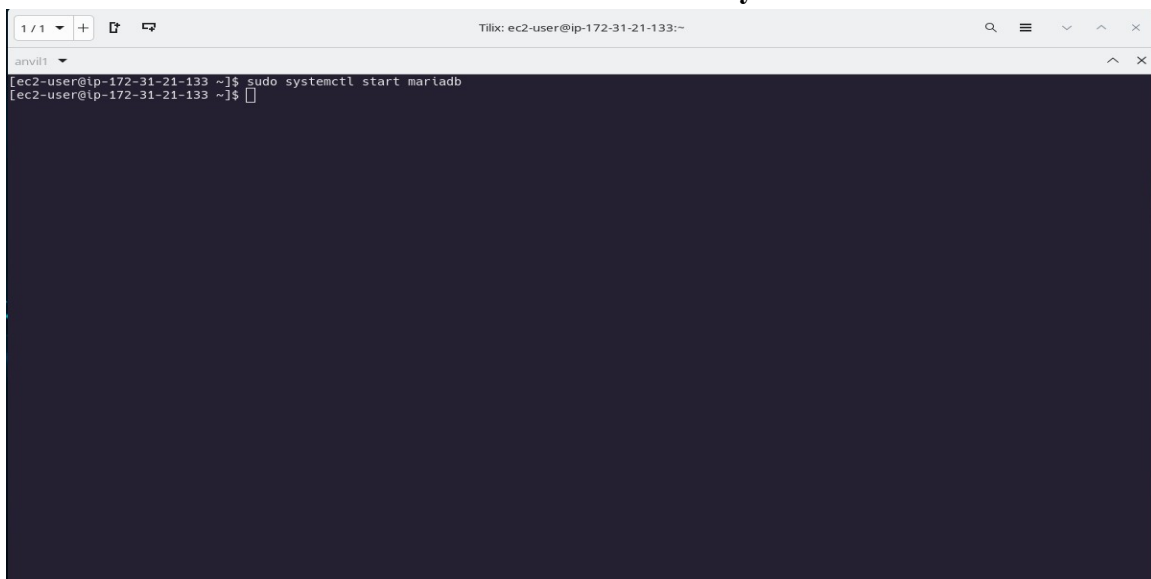
A terminal window titled 'Tilix: ec2-user@ip-172-31-21-133:~' with a dark background. The prompt is '[ec2-user@ip-172-31-21-133 ~]\$' and the command 'sudo dnf install mariadb105-server' is being typed. The window has standard Linux terminal window controls at the top and a search icon on the right.

```
1 / 1 + [Tilix: ec2-user@ip-172-31-21-133:~]
anvil1
[ec2-user@ip-172-31-21-133 ~]$ sudo dnf install mariadb105-server
```

Configuration of MariaDB

Step 21

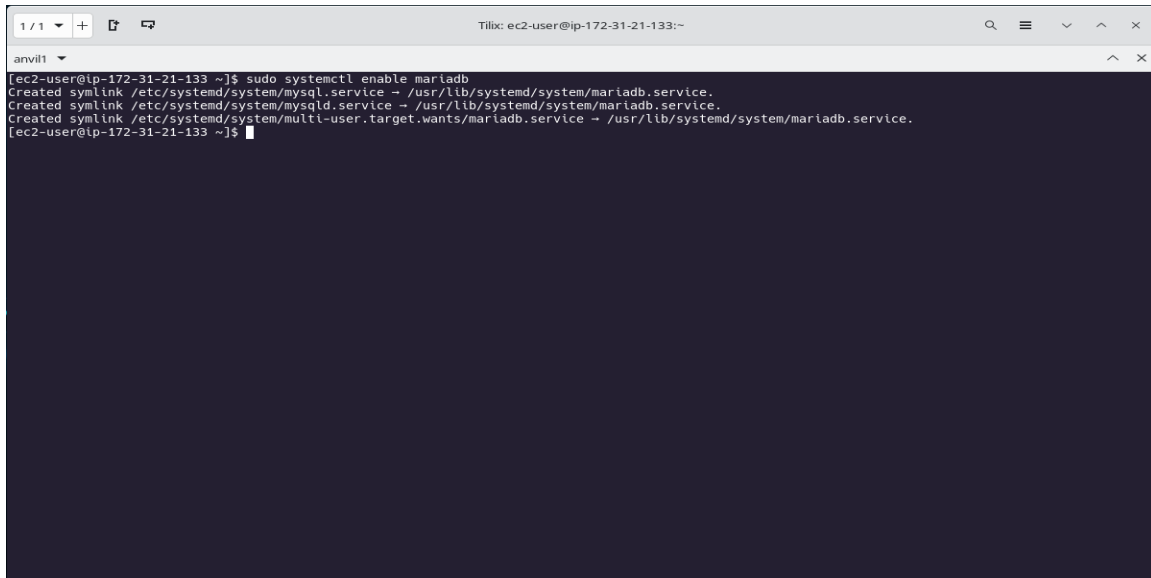
Now that it is installed we need to start it. “**sudo systemctl start mariadb**”

A terminal window titled 'Tilix: ec2-user@ip-172-31-21-133:~' with a dark background. The prompt is '[ec2-user@ip-172-31-21-133 ~]\$' and the command 'sudo systemctl start mariadb' is being typed. The window has standard Linux terminal window controls at the top and a search icon on the right.

```
1 / 1 + [Tilix: ec2-user@ip-172-31-21-133:~]
anvil1
[ec2-user@ip-172-31-21-133 ~]$ sudo systemctl start mariadb
[ec2-user@ip-172-31-21-133 ~]$
```

Step 22

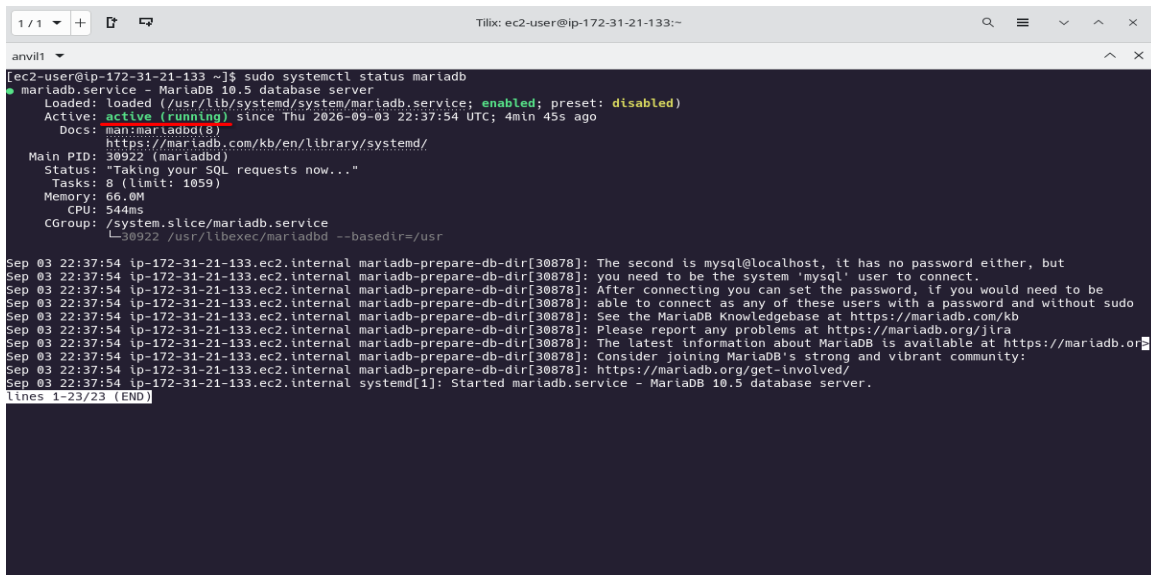
Next we enable mariadb. “**sudo systemctl enable mariadb**”



```
1 / 1 + [ ] [ ]
Tilix: ec2-user@ip-172-31-21-133:~
anvil1
[ec2-user@ip-172-31-21-133 ~]$ sudo systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
[ec2-user@ip-172-31-21-133 ~]$
```

Step 23

Now we need to make sure that mariadb is up and running. To do that we will use the command.
“**sudo systemctl status mariadb**”



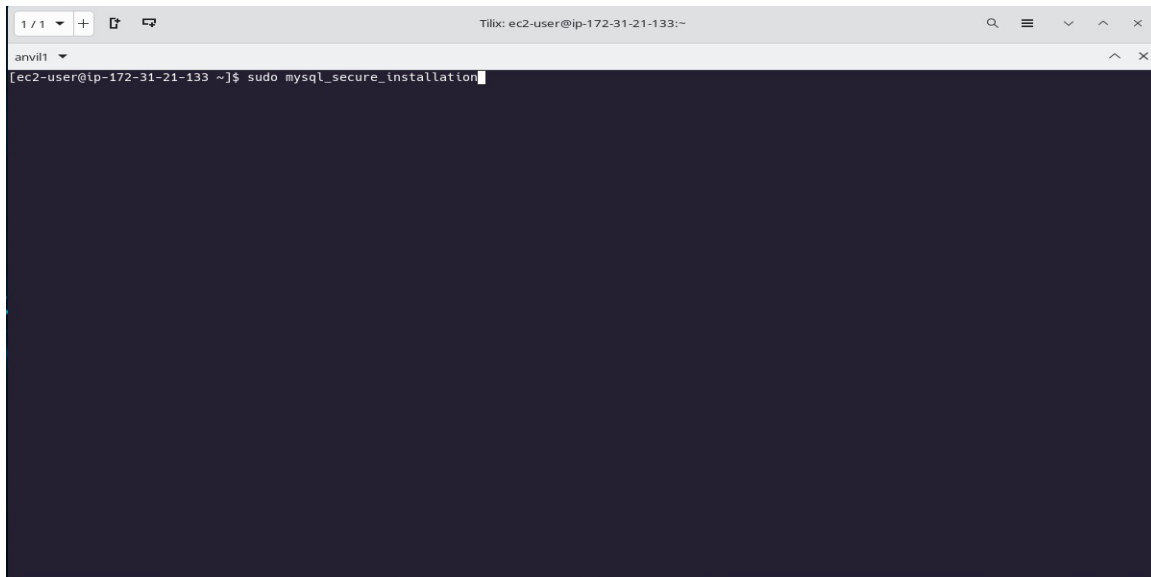
```
1 / 1 + [ ] [ ]
Tilix: ec2-user@ip-172-31-21-133:~
anvil1
[ec2-user@ip-172-31-21-133 ~]$ sudo systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
   Active: active (running) since Thu 2026-09-03 22:37:54 UTC; 4min 45s ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 30922 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 8 (limit: 1059)
    Memory: 66.0M
       CPU: 544ms
   CGroup: /system.slice/mariadb.service
           └─30922 /usr/libexec/mariabdd --basedir=/usr

Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: The second is mysql@localhost, it has no password either, but
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: you need to be the system 'mysql' user to connect.
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: After connecting you can set the password, if you would need to be
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: able to connect as any of these users with a password and without sudo
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: See the MariaDB Knowledgebase at https://mariadb.com/kb
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: Please report any problems at https://mariadb.org/jira
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: The latest information about MariaDB is available at https://mariadb.org
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: Consider joining MariaDB's strong and vibrant community:
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal mariadb-prepare-db-dir[30878]: https://mariadb.org/get-involved/
Sep 03 22:37:54 ip-172-31-21-133.ec2.internal systemd[1]: Started mariadb.service - MariaDB 10.5 database server.

lines 1-23/23 (END)
```

Step 24

Now to secure your MariaDB install. “**sudo mysql_secure_installation**”



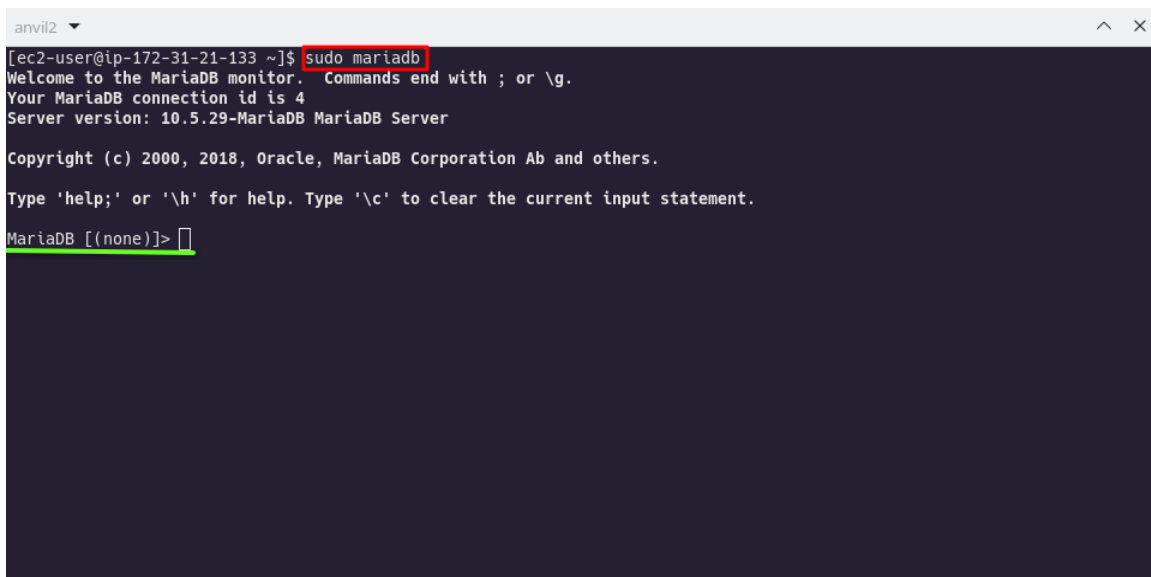
```
1 / 1 + [Tilix: ec2-user@ip-172-31-21-133:~]
anvil1
[ec2-user@ip-172-31-21-133 ~]$ sudo mysql_secure_installation
```

Accessing MariaDB

Step 25

Now to access Mariadb server need to run command “**sudo mariadb**” When you see:

“**MariaDB [(none)]>**” you know that you are inside the mariadb.



```
anvil2
[ec2-user@ip-172-31-21-133 ~]$ sudo mariadb
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.5.29-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Creating The Database

Step 26

Now we are going to create the database for SeedDMS. For this run the command: **“CREATE DATABASE seeddms; Verify that it is there with the command: “SHOW databases”**

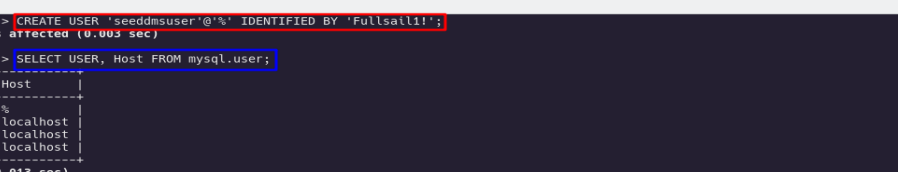
You can see in the image that seeddms is at the bottom of the list.

```
Tilix: ec2-user@ip-172-31-21-133:~  
anvil1 ▾  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> clear  
MariaDB [(none)]> clear  
MariaDB [(none)]>  
MariaDB [(none)]> CREATE DATABASE seeddms;  
Query OK, 1 row affected (0.000 sec)  
  
MariaDB [(none)]> SHOW database;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'database' at line 1  
MariaDB [(none)]> SHOW databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| seeddms |  
+-----+  
4 rows in set (0.002 sec)  
  
MariaDB [(none)]> █
```

Step 27

Next, we need to do for the database is to create the user.

```
“CREATE USER ‘newuser’@’%’ IDENTIFIED BY ‘user password’;”
```



The screenshot shows a terminal window with a dark background. At the top, a window title bar reads "Tilix: ec2-user@ip-172-31-21-133:~". Below the title bar, the terminal shows the following sequence of commands and output:

```
anvil1
MariaDB [(none)]> CREATE USER 'seddmsuser'@'%' IDENTIFIED BY 'Fullsatll!';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> SELECT USER, Host FROM mysql.user;
+-----+-----+
| User           | Host           |
+-----+-----+
| seddmsuser     | %              |
| mariadb.sys    | localhost      |
| mysql          | localhost      |
| root           | localhost      |
+-----+-----+
4 rows in set (0.013 sec)

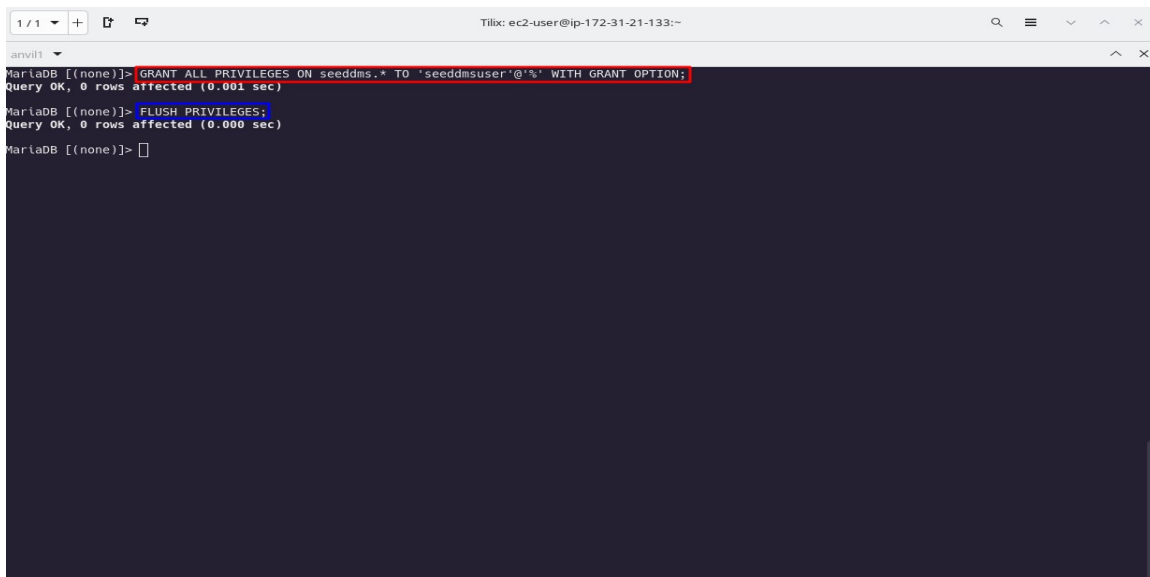
MariaDB [(none)]>
```

Step 28

Last thing we need to do is set privileges and save. Set privileges:

“GRANT ALL PRIVILEGES ON database_name.* TO 'databaseuser'@'%' WITH GRANT OPTION;

Save: **“FLUSH PRIVILEGES”**



```
1 / 1 + [icons] Tilix: ec2-user@ip-172-31-21-133:~
anvil1
MariaDB [(none)]> GRANT ALL PRIVILEGES ON seeddms.* TO 'seeddmsuser'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.001 sec)
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
MariaDB [(none)]> 
```

Configuration of the SeedDMS Application Server

Connecting to the EC2 Instance

Step 29

Now just like last time we have to ssh into the EC2 instance. Now run the command

“ssh -i ~/file.pem ec2-user@public IP”.

This prompt is asking for approval, **not** your password. Type yes.

[illegible]

Apache installation

Step 30

First part of a LAMP stack is installing Apache: “**sudo dnf install httpd.x86_64 -y**”

```
1/1 + T anvil
Tillic: ec2-user@ip-172-31-25-111:~

[ec2-user@ip-172-31-25-111 ~]$ sudo dnf install httpd.x86_64 -y
Last metadata expiration check: 0:03:08 ago on Sun Sep  6 15:08:44 2026.
Dependencies resolved.

=====
Package                                Architecture      Version            Repository          Size
=====
Installing:
httpd                                  x86_64            2.4.68-1.amzn2023.0.1  amazonlinux        46 k
Installing dependencies:
apr                                    x86_64            1.7.5-1.amzn2023.0.4   amazonlinux        129 k
apr-util                              x86_64            1.6.5-1.amzn2023.0.1   amazonlinux        15 k
apr-util-ldap                         x86_64            1.6.5-1.amzn2023.0.1   amazonlinux        19 k
generic-logos-httpd                  noarch            18.0.0-12.amzn2023.0.3  amazonlinux        1.4 M
httpd-core                           x86_64            2.4.68-1.amzn2023.0.1   amazonlinux        12 k
httpd-filesystem                     x86_64            2.4.68-1.amzn2023.0.1   amazonlinux        80 k
httpd-tools                          x86_64            2.4.68-1.amzn2023.0.1   amazonlinux        315 k
libbrotli                             x86_64            1.0.9-4.amzn2023.0.2    amazonlinux        33 k
mailcap                              noarch            2.1.49-3.amzn2023.0.3   amazonlinux
Installing weak dependencies:
apr-util-openssl                     x86_64            1.6.5-1.amzn2023.0.1   amazonlinux        17 k
mod_http2                           x86_64            2.0.42-1.amzn2023.0.1   amazonlinux        167 k
mod_lua                             x86_64            2.4.68-1.amzn2023.0.1   amazonlinux        59 k
=====

Transaction Summary
-----
Install 13 Packages

Total download size: 2.4 M
Installed size: 7.0 M
Downloading Packages:
(1/13): apr-1.7.5-1.amzn2023.0.4.x86_64.rpm                3.5 MB/s | 129 kB  00:00
(2/13): apr-util-1.6.5-1.amzn2023.0.1.x86_64.rpm           384 kB/s | 15 kB  00:00
(3/13): apr-util-1.6.5-1.amzn2023.0.1.x86_64.rpm           2.3 MB/s | 100 kB  00:00
(4/13): apr-util-openssl-1.6.5-1.amzn2023.0.1.x86_64.rpm   660 kB/s | 17 kB  00:00
(5/13): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm 737 kB/s | 19 kB  00:00
(6/13): httpd-2.4.68-1.amzn2023.0.1.x86_64.rpm             1.8 MB/s | 46 kB  00:00
(7/13): httpd-filesystem-2.4.68-1.amzn2023.0.1.noarch.rpm   589 kB/s | 12 kB  00:00
(8/13): httpd-tools-2.4.68-1.amzn2023.0.1.x86_64.rpm        3.0 MB/s | 80 kB  00:00
(9/13): httpd-core-2.4.68-1.amzn2023.0.1.x86_64.rpm         32 MB/s | 1.4 MB  00:00
(10/13): libbrotli-1.0.9-4.amzn2023.0.2.x86_64.rpm          9.4 MB/s | 315 kB  00:00
(11/13): mailcap-2.1.49-3.amzn2023.0.3.noarch.rpm           1.0 MB/s | 33 kB  00:00
(12/13): mod_http2-2.0.42-1.amzn2023.0.1.x86_64.rpm         5.2 MB/s | 167 kB  00:00
(13/13): mod_lua-2.4.68-1.amzn2023.0.1.x86_64.rpm           2.0 MB/s | 59 kB  00:00
```

Step 31

Now we have to start up Apache, use this command. “**sudo systemctl start httpd.service**”

After that you will want to check and make sure it is up and running, you can do the with the following command “**sudo systemctl status httpd.service**”

```
1/1 + T anvil
Tillic: ec2-user@ip-172-31-25-111:~

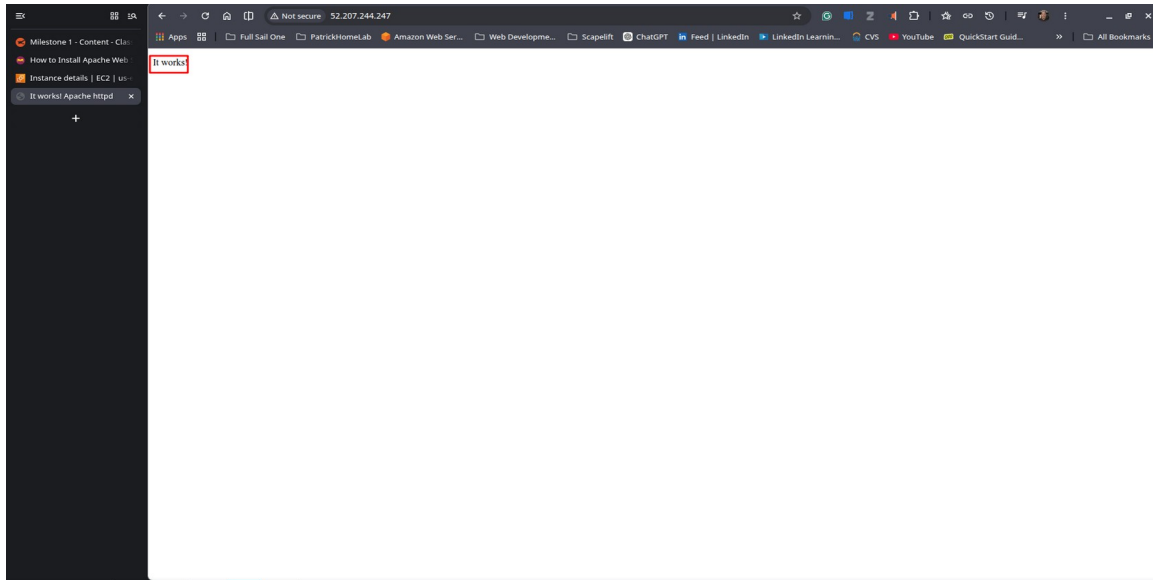
[ec2-user@ip-172-31-25-111 ~]$ sudo systemctl start httpd.service
[ec2-user@ip-172-31-25-111 ~]$ sudo systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Sun 2026-09-06 15:15:24 UTC; 16s ago
     Docs: man:httpd.service(8)
   Main PID: 27520 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
     Tasks: 177 (limit: 1059)
    Memory: 13.7M
       CPU: 73ms
   CGroup: /system.slice/httpd.service
           └─27520 /usr/sbin/httpd -DFOREGROUND
             └─27521 /usr/sbin/httpd -DFOREGROUND
               └─27522 /usr/sbin/httpd -DFOREGROUND
                 └─27573 /usr/sbin/httpd -DFOREGROUND
                   └─27589 /usr/sbin/httpd -DFOREGROUND

Sep 06 15:15:24 ip-172-31-25-111.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Sep 06 15:15:24 ip-172-31-25-111.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Sep 06 15:15:24 ip-172-31-25-111.ec2.internal httpd[27520]: Server configured, listening on: port 80
[ec2-user@ip-172-31-25-111 ~]$ sudo systemctl enable httpd.service
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[ec2-user@ip-172-31-25-111 ~]$
```

Step 32

Check in the browser to confirm that it can connect. To do so open your browser and enter:
`http://IP`

- now for the IP you will want to enter the public IP of your EC2 instance.

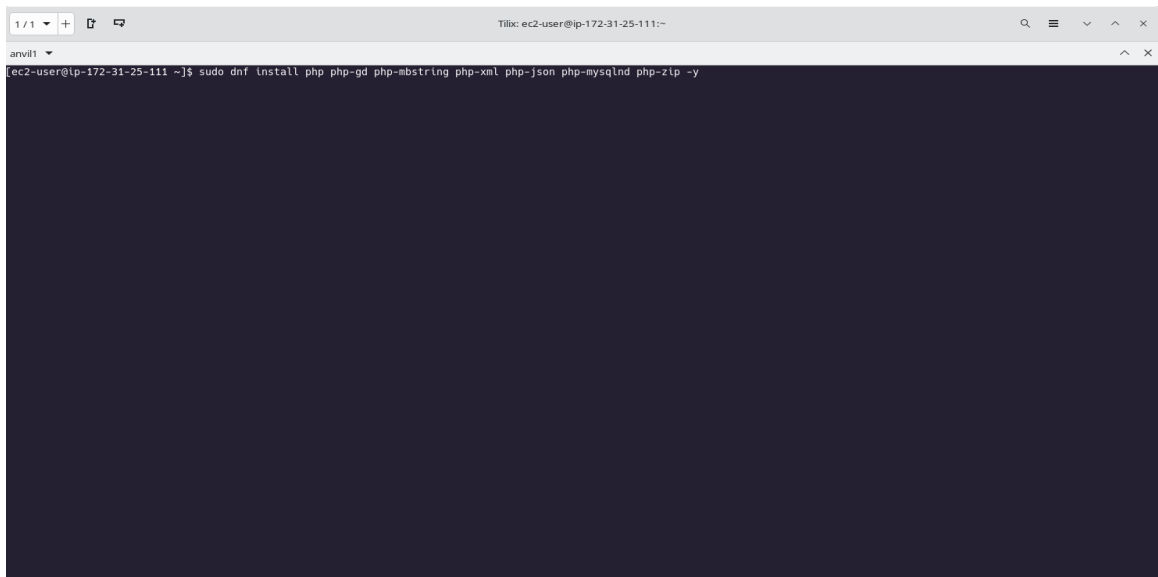


Installation of PHP

Step 33

Now we are going to install PHP. This command is a long one so make sure it typed out right, otherwise you will get an error.

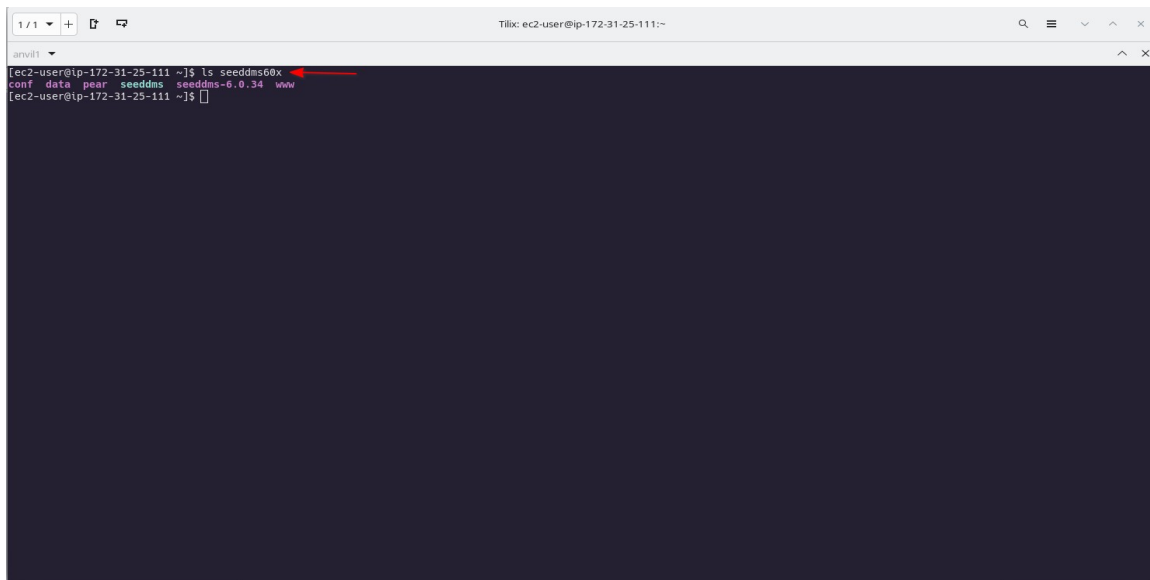
“sudo dnf install php php-gd php-mbstring php-xml php-json php-mysqlnd php-zip -y”

A terminal window with a dark background and light text. The window title bar shows 'Tilix: ec2-user@ip-172-31-25-111:~'. The terminal content shows the command 'sudo dnf install php php-gd php-mbstring php-xml php-json php-mysqlnd php-zip -y' being entered at the prompt. The prompt is '[ec2-user@ip-172-31-25-111 ~]\$'. The command is followed by a space and a hyphen, indicating it is being typed. The terminal window has standard Linux window controls (minimize, maximize, close) and a search icon in the top right corner. The terminal text is as follows:

```
ahvil1  
[ec2-user@ip-172-31-25-111 ~]$ sudo dnf install php php-gd php-mbstring php-xml php-json php-mysqlnd php-zip -y
```


Step 36

After extracting seeddms archive need to verify that the are all there Run: “**ls seeddms60x**”

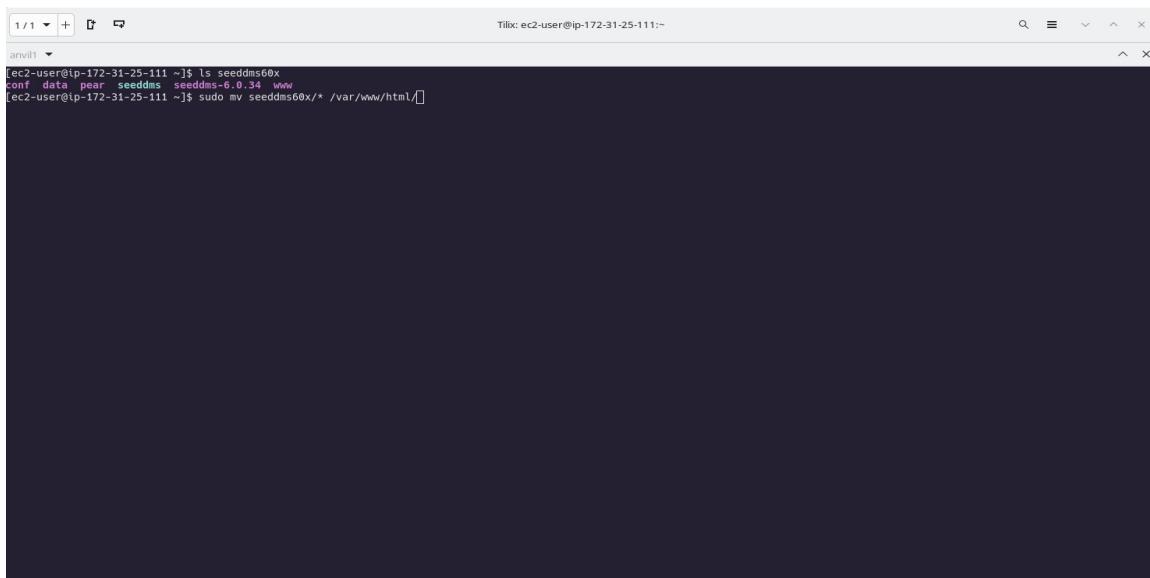


```
1/1 + [Tilix: ec2-user@ip-172-31-25-111:~]
[ec2-user@ip-172-31-25-111 ~]$ ls seeddms60x
conf  data  pear  seeddms  seeddms-6.0.34  www
[ec2-user@ip-172-31-25-111 ~]$
```

Step 37

Now that we have the files we need to move them to Apache root directory by running:

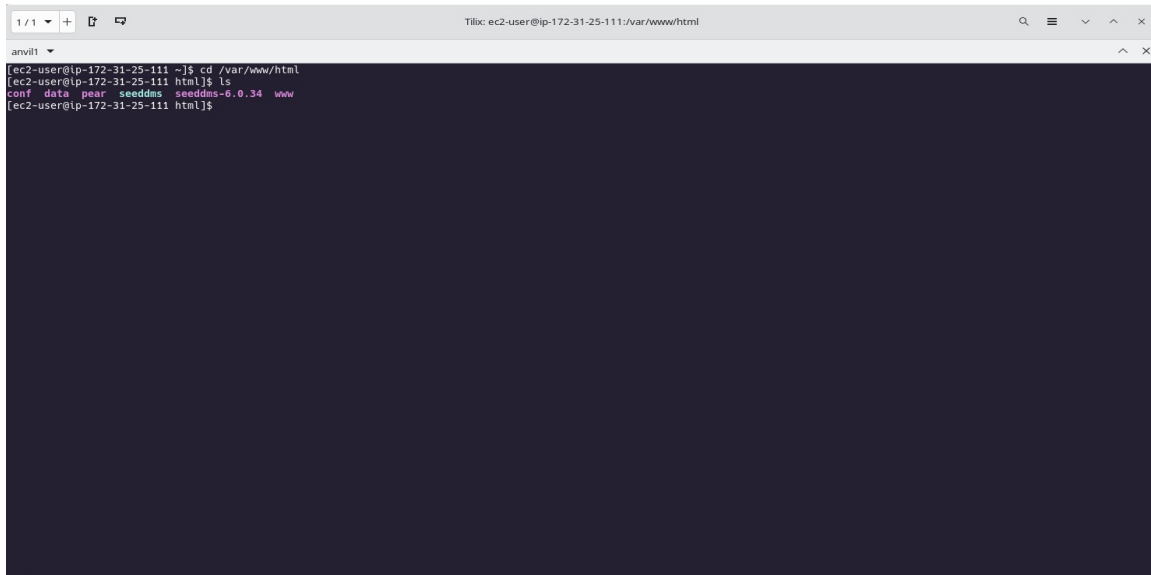
“sudo mv seeddms60x/* /var/www/html/”



```
1/1 + [Tilix: ec2-user@ip-172-31-25-111:~]
[ec2-user@ip-172-31-25-111 ~]$ ls seeddms60x
conf  data  pear  seeddms  seeddms-6.0.34  www
[ec2-user@ip-172-31-25-111 ~]$ sudo mv seeddms60x/* /var/www/html/
```

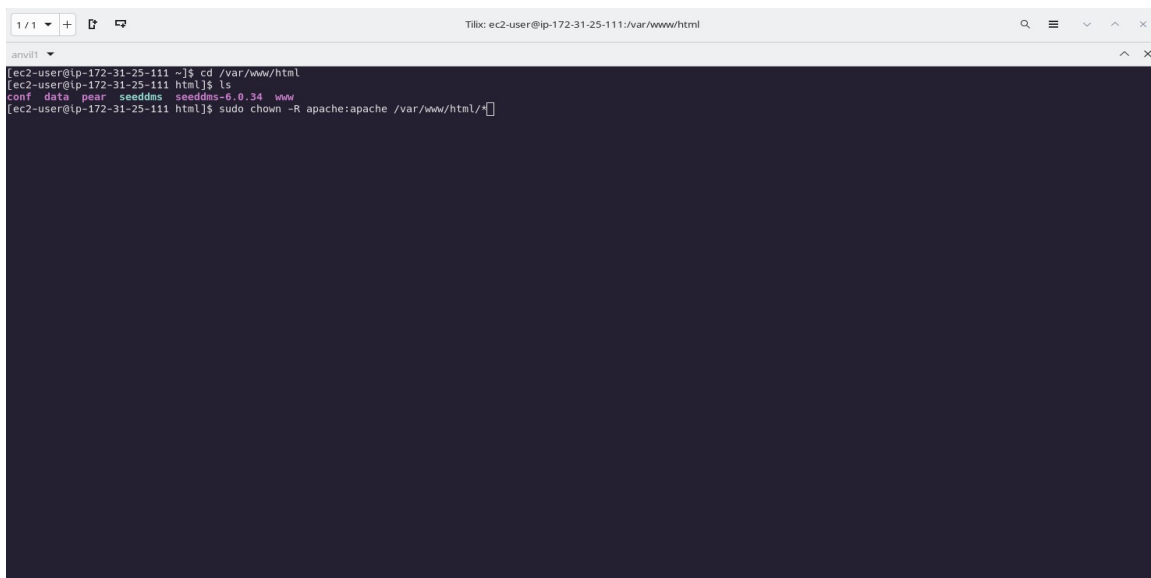
Step 38

Next we need to move to that directory to verify the files are there. Do that with the command: **“cd /var/www/html”** After your in the html directory run:”ls” to see the files have moved.

A terminal window titled 'Tilix ec2-user@ip-172-31-25-111:/var/www/html' showing a sequence of commands and their output. The user navigates to the /var/www/html directory and lists its contents, showing files like conf, data, pear, seeddms, and seeddms-6.0.34. The terminal has a dark background with light-colored text.

```
1/1 + [ ] [ ]
Tilix ec2-user@ip-172-31-25-111:/var/www/html
anvil
[ec2-user@ip-172-31-25-111 ~]$ cd /var/www/html
[ec2-user@ip-172-31-25-111 html]$ ls
conf  data  pear  seeddms  seeddms-6.0.34  www
[ec2-user@ip-172-31-25-111 html]$
```

Step 39

A terminal window titled 'Tilix ec2-user@ip-172-31-25-111:/var/www/html' showing the same sequence of commands as Step 38, followed by the command 'sudo chown -R apache:apache /var/www/html/*' which is partially entered. The terminal has a dark background with light-colored text.

```
1/1 + [ ] [ ]
Tilix ec2-user@ip-172-31-25-111:/var/www/html
anvil
[ec2-user@ip-172-31-25-111 ~]$ cd /var/www/html
[ec2-user@ip-172-31-25-111 html]$ ls
conf  data  pear  seeddms  seeddms-6.0.34  www
[ec2-user@ip-172-31-25-111 html]$ sudo chown -R apache:apache /var/www/html/*
```

Now that we can see the files are there we need to give owner ship to Apache. Run: **“sudo chown -R apache:apache /var/www/html/*”**

SeedDMS Installation Tool

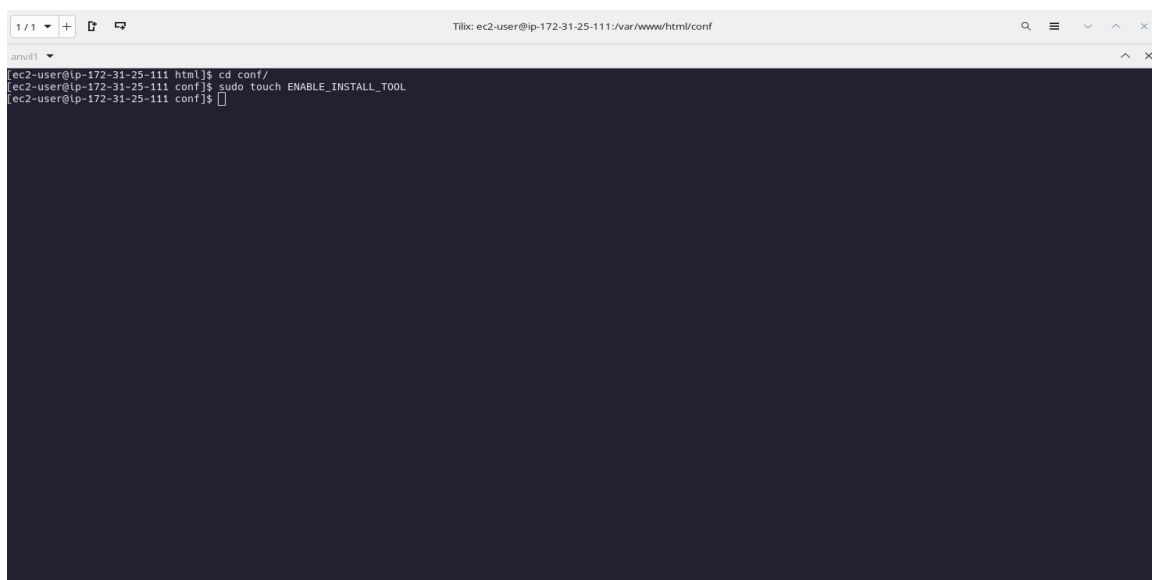
Step 40

Now in order to be able to access the web based installation tool we need to put the tool in the file. First step is to move to the configuration directory.

Run: “**cd conf/**”

note: if you are not in the html directory you will need to run: “**cd /var/www/html/conf/**”

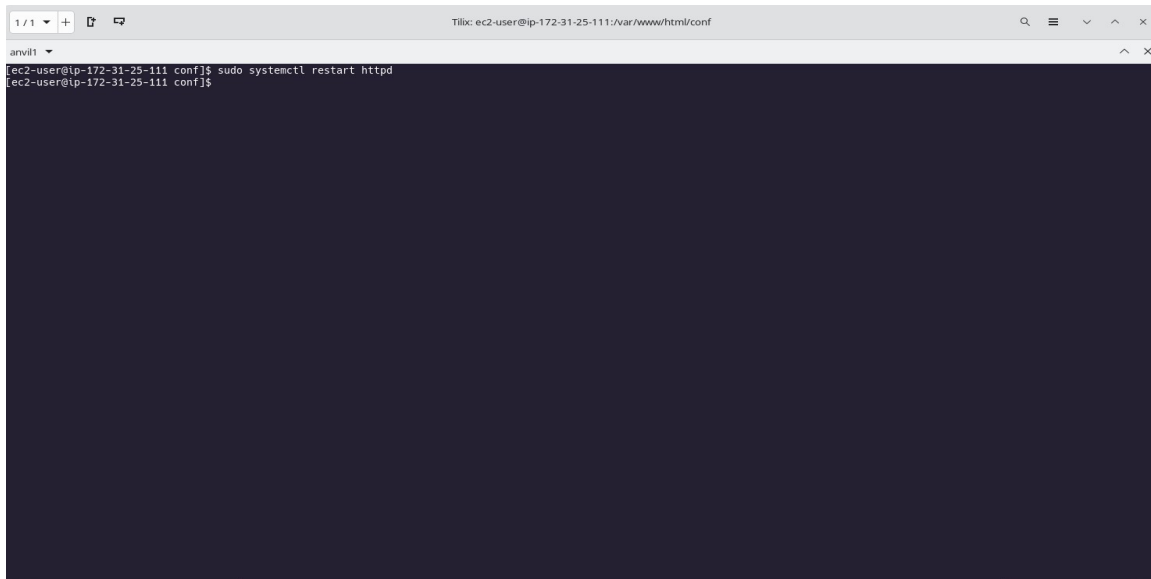
Now that we are in the configuration file we need to create the installation tool running the following command will create the file needed. “**sudo touch ENABLE_INSTALL_TOOL**”



```
anyll |
[ec2-user@ip-172-31-25-111 html]$ cd conf/
[ec2-user@ip-172-31-25-111 conf]$ sudo touch ENABLE_INSTALL_TOOL
[ec2-user@ip-172-31-25-111 conf]$
```

Step 41

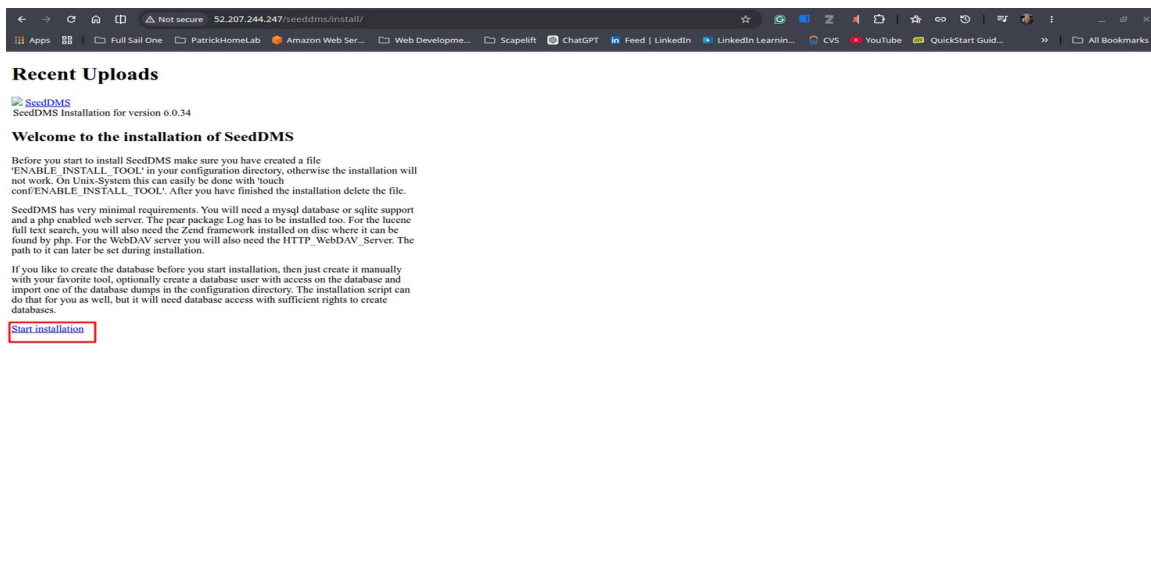
Now that we have the all the files and configuration done we need to restart Apache for it to all take affect. Run: “**sudo systemctl restart httpd**”



```
1/1 + [T] [C]
Tilix: ec2-user@ip-172-31-25-111:/var/www/html/conf
anvil1
[ec2-user@ip-172-31-25-111 conf]$ sudo systemctl restart httpd
[ec2-user@ip-172-31-25-111 conf]$
```

Step 42

Now that the server has restarted we will go to the browser to configure the website side. If you still have the browser up just refresh the page. If not that is fine. Just go to **http://public-IP/seaddms/install** in the browser. You want to get to the page the is to the right. And then click “**Start installation**”



After clicking “Start installation” you should be at this page.

SeedDMS Installation for version 6.0.34

Your Root directory has been modified to fit your installation path!

Server settings

Root directory: /var/www/html/www/

Http Root: /seeddms/

Content directory: /var/www/html/data/

Directory for full text index: /var/www/html/data/ucene/

Directory for partial uploads: /var/www/html/data/staging/

Cache directory: /var/www/html/data/cache/

Extra PHP include Path: /var/www/html/pear/

Database settings

Database Type: mysql

Server name: 172.31.25.111

Database: Database

Username: seeddmsuser

Password: *****

Create database tables: ☒ [Apply](#)

Step 43

Next need to go back to AWS and click on your Database. Here we need to copy the **Privet IP**

Instance summary for i-04513506e419c8dc2 (SeedDMS)

Updated about 1 hour ago

Instance ID: i-04513506e419c8dc2

IPv6 address: -

Hostname type: IP name: ip-172-31-25-111.ec2.internal

Answer private resource DNS name: IPv4 (A)

Auto-assigned IP address: 52.207.244.247 [Public IP]

IAM role: -

IMDSv2: Required

Operator: -

Public IPv4 address: 52.207.244.247 | [open address](#)

Instance state: Running

Private IP DNS name (IPv4 only): ip-172-31-25-111.ec2.internal

Instance type: t3.micro

VPC ID: vpc-075232aaf657c6e2d

Subnet ID: subnet-0bce89206a05684c9

Instance ARN: arn:aws:ec2:us-east-1:825765390827:instance/i-04513506e419c8dc2

Private IPv4 addresses: 172.31.25.111

Public DNS: ec2-52-207-244-247.compute-1.amazonaws.com | [open address](#)

Elastic IP addresses: -

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)

Auto Scaling Group name: -

Managed: false

Instance details

AMI ID: ami-081b0a6eac00b4f53

AMI name: al2023-ami-2023.12.20260831.0-kernel-6.18-x86_64

Monitoring: disabled

Allowed image: -

Platform details: Linux/UNIX

Termination protection: Disabled

Step 44

Back at the installation web page we need fill some of this out.

- **Extra PHP include Path:** needs to be: /var/www/html/pear/
- **Database Type:** needs to be: mysql
- **Server Name:** This is the private IP of your database that we just saved.
- **Database:** is the name of your database.
- Then username and password

Create database tables: Check mark on this **ONLY** for the first application server. Click apply

SeedDMS

SeedDMS Installation for version 6.0.34

Your Root directory has been modified to fit your installation path!

Server settings

Root directory: /var/www/html/www/

Http Root: /seeddms/

Content directory: /var/www/html/data/

Directory for full text index: /var/www/html/data/ucense/

Directory for partial uploads: /var/www/html/data/staging/

Cache directory: /var/www/html/data/cache/

Extra PHP include Path: /var/www/html/pear/

Database settings

Database Type: mysql

Server name: 172.31.25.111

Database: Database

Username: seeddmsuser

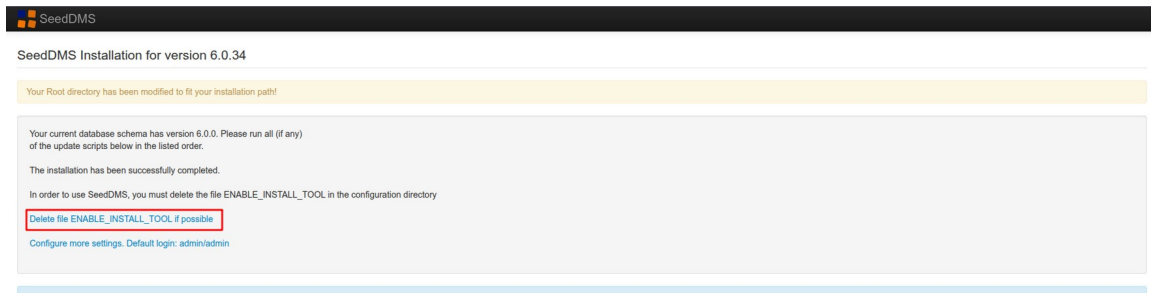
Password: *****

Create database tables: ☒

Apply

Step 45

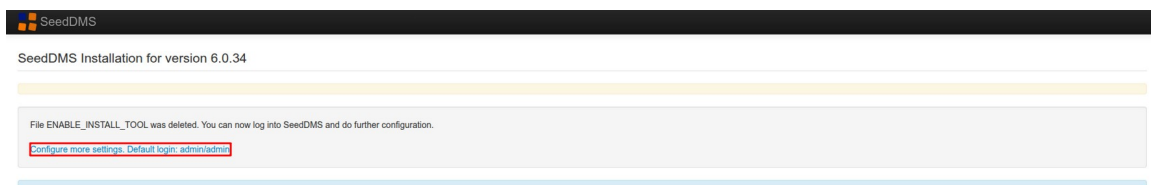
At this time you should be at this page to the right. Here we are going to delete the installation tool.



Step 46

Next we are going to go into the application and configure a little in there.

Note: The default login is admin/admin



Step 47

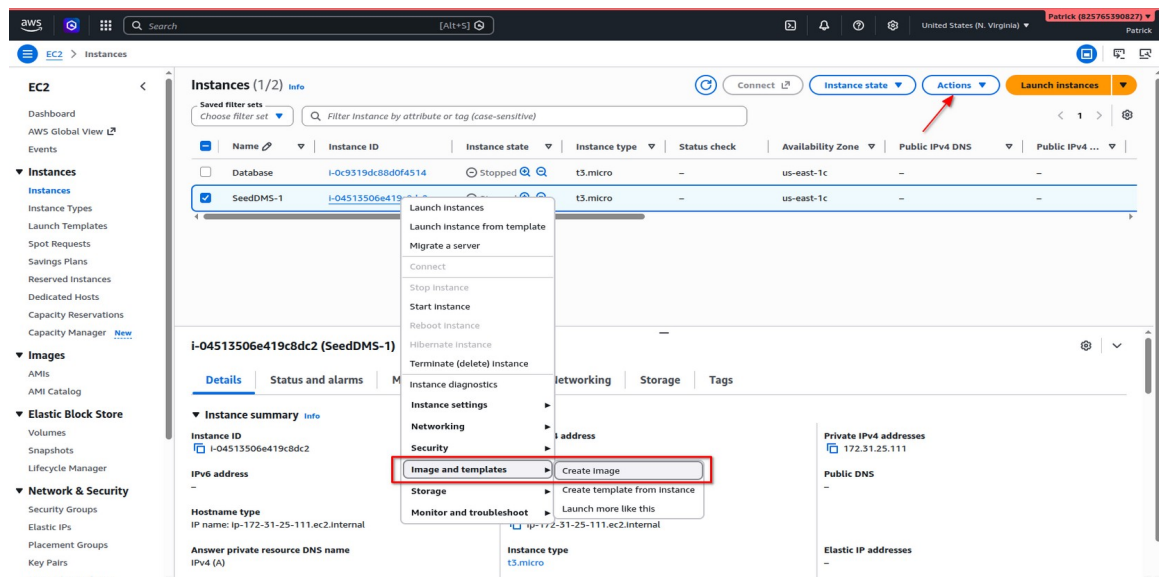
The only thing that we need to do is change the name of the site. When choosing a name it is good to pick one that is relevant to its job also if you know you are going to make multiple one, to put a number convention in the name.

The screenshot shows the administration interface of SeedDMS. At the top, there is a navigation bar with the site name 'Prime Directive Technologies_WS1', a search bar, and links to 'Calendar' and 'Admin tools'. Below this is a sub-navigation bar with 'Admin tools' and several expandable menus: 'Users/Groups management', 'Definitions', 'Backup/Logging', and 'Misc'. The main content area is titled 'Settings' and has four tabs: 'Site' (selected), 'System', 'Advanced', and 'Extensions'. Under the 'Site' tab, there is a section for 'Display settings' containing various configuration options. The 'Site Name' field is highlighted with a red box and contains the text 'Prime Directive Technologies_WS1'. Other visible settings include 'Foot Note' (SeedDMS free document management system - www.seeddms.org), 'Print Disclaimer' (checked), 'Available languages' (empty), 'Default language' (English (GB)), 'Date format' (empty), 'Date/Time format' (empty), 'Default theme' (bootstrap4), 'Override theme' (unchecked), 'One page mode' (checked), and several 'Width of preview images' fields (list: 40, menu list: 40, dropfolder list: 100, detail: 100). The 'Action when clicking on' field is partially visible at the bottom.

Launching 2 additional EC2 instances from the AMI

Step 48

The first thing is to select the EC2 instance the you want to use. Next you can either click **Actions** in the top right, or you can right click on the instance (both option will work the same way). In the drop down menu scroll down to **image and templates**, the in the next menu select **Create Image**.



Step 49

You will want to name the instance the same as the first one but change what number server it is.

The screenshot shows the 'Create image' page in the AWS Management Console. The page title is 'Create image' with a sub-header 'Info'. Below this, it states: 'An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.'

Image details

Instance ID
I-04513506e419c8dc2 (SeedDMS-1)

Image name
Enter image name
Maximum 127 characters. Can't be modified after creation.

Image description - optional
Image description
Maximum 255 characters

☒ **Reboot instance**
When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/xv...	Create new snapshot from v...	8	EBS General Purpose SSD - ...	3000		☒ Enable	☐ Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together
Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately
Tag the image and the snapshots with different tags.

Step 50

Next you can give it a description if you want, but it is not necessary. The last thing is to click **Create Image** at the bottom right.

The screenshot shows the 'Create image' page in the AWS Management Console. The page title is 'Create image' with a sub-header 'Info'. Below this, it states: 'An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.'

Image details

Instance ID
I-04513506e419c8dc2 (SeedDMS-1)

Image name
Enter image name
Maximum 127 characters. Can't be modified after creation.

Image description - optional
Image description
Maximum 255 characters

☒ **Reboot instance**
When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/xv...	Create new snapshot from v...	8	EBS General Purpose SSD - ...	3000		☒ Enable	☐ Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

☒ Tag image and snapshots together
Tag the image and the snapshots with the same tag.

☐ Tag image and snapshots separately
Tag the image and the snapshots with different tags.

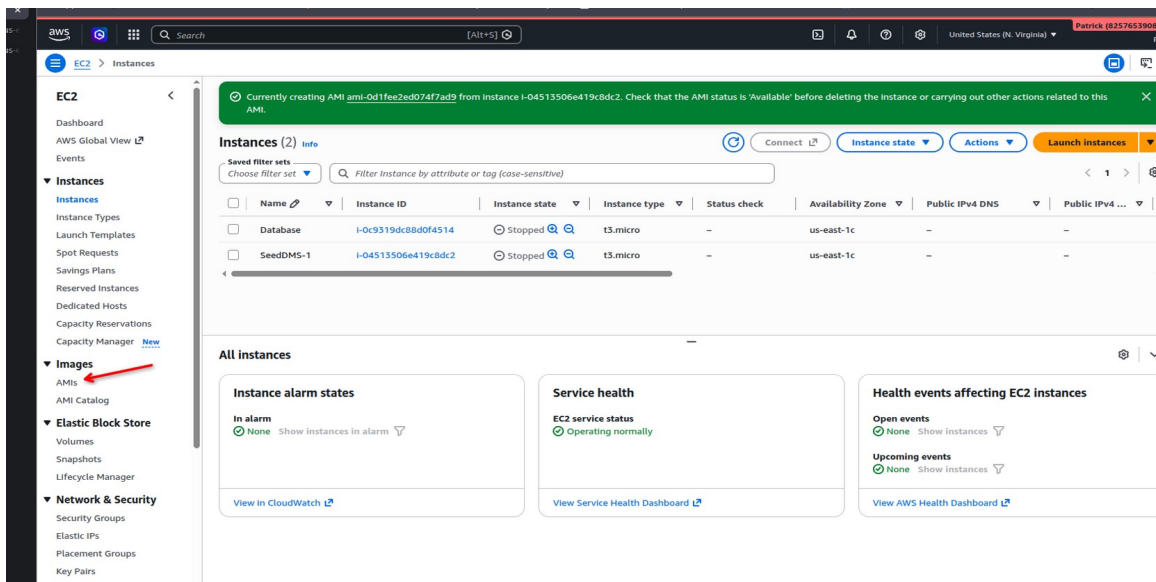
No tags associated with the resource.

[Add new tag](#)
You can add up to 50 more tags.

[Cancel](#) [Create image](#)

Step 51

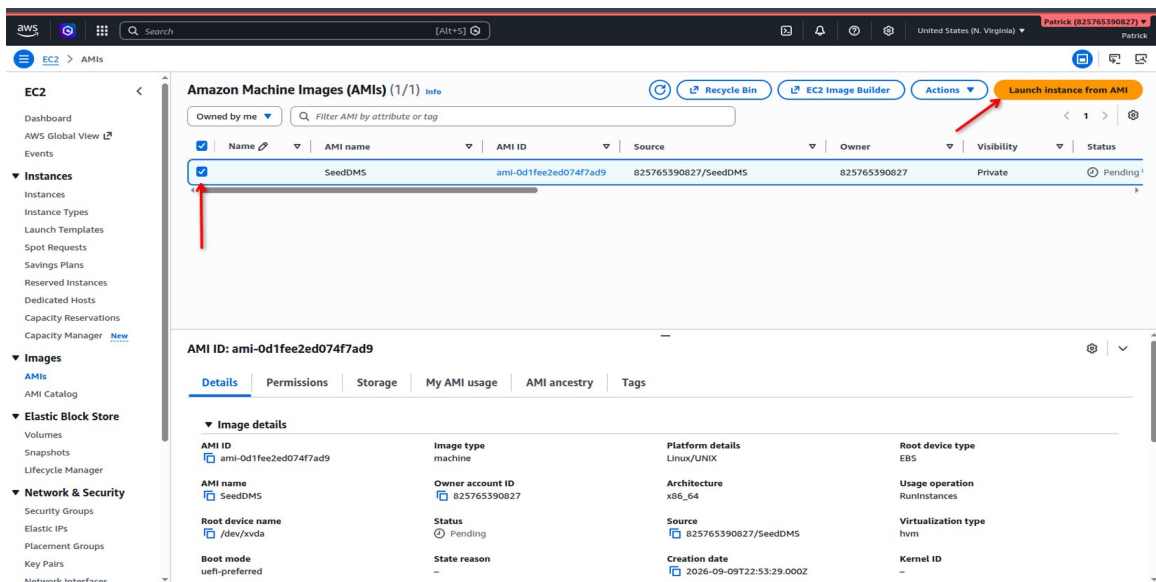
To see the AMI that you just created, need to go to AMI. In the menu on the left scroll down to where it say **AMI**.



Step 52

The page you will see the AMI that you created. Select the instance,. Then in the top right click:

Launch instance from AMI



Step 53

At this screen you will give it a similar name then we will adjust it later. Then on the far right under **Summary** select how many instances you want. For this project I'm going to select 2. Then at the bottom select **Launch instance**.

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

[AMI from catalog](#) Recents My AMIs Quick Start

Name
SeedDM5

Description
-

Image ID
ami-0d1fee2ed074f7ad9

Username ⓘ
root (Check with the AMI provider.)

Creation date
2026-09-09T22:53:29.000Z

Architecture
x86_64

Virtualization
hvm

Root device type
ebs

ENA Enabled
Yes

Boot mode
uefi-preferred

Summary

Number of instances | Info
1

Software Image (AMI)
SeedDM5
ami-0d1fee2ed074f7ad9

Virtual server type (instance type)
-

Firewall (security group)
New security group

Storage (volumes)
-

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the Internet. Data transfer charges are not included as part of the free tier allowance. Charges may apply depending on your account's free tier status.

[Cancel](#) [Launch instance](#) [Preview code](#)

Step 54

With the image already configured all we need to do is select your **key-pair**. Then make sure that you select: **Select existing security group** then click on your security group from the list. When done click **Launch instance** at the bottom right.

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required
 [Create new key pair](#)

Network settings Info [Edit](#)

Network | Info
vpc-075232aaf657c6e2d

Subnet | Info
No preference (Default subnet in any availability zone)

Auto-assign public IP | Info
Enable

Firewall (security groups) | 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.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

☒ Allow SSH traffic from

☐ 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.

Summary

Number of instances | Info
2

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)
SeedDM5
ami-0d1fee2ed074f7ad9

Virtual server type (instance type)
t3.micro

Firewall (security group)
New security group

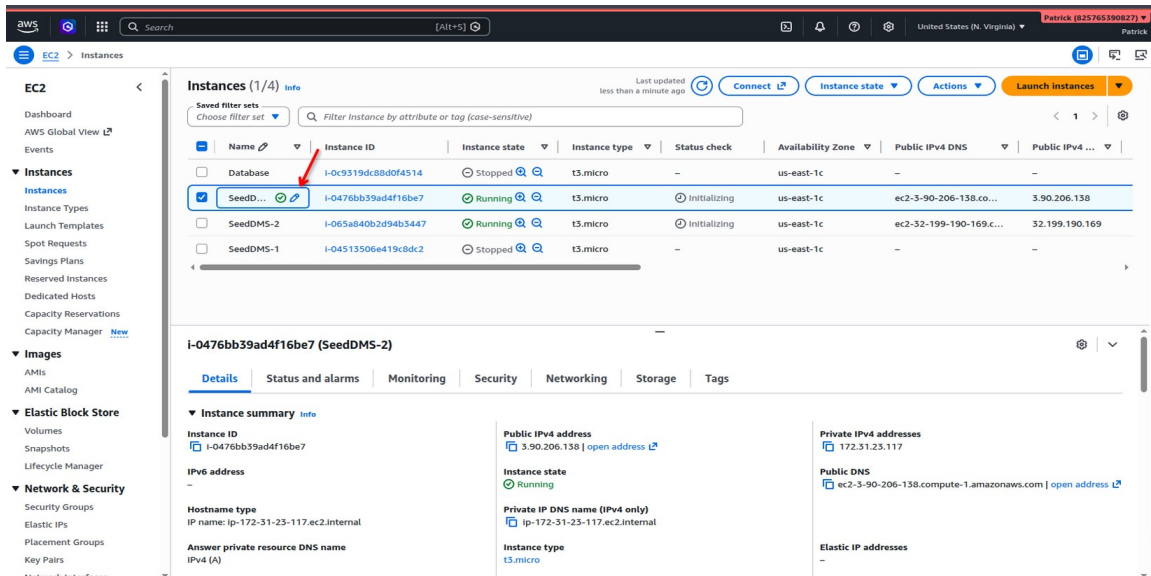
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the Internet. Data transfer charges are not included as part of the free tier allowance. Charges may apply depending on your account's free tier status.

[Cancel](#) [Launch instance](#) [Preview code](#)

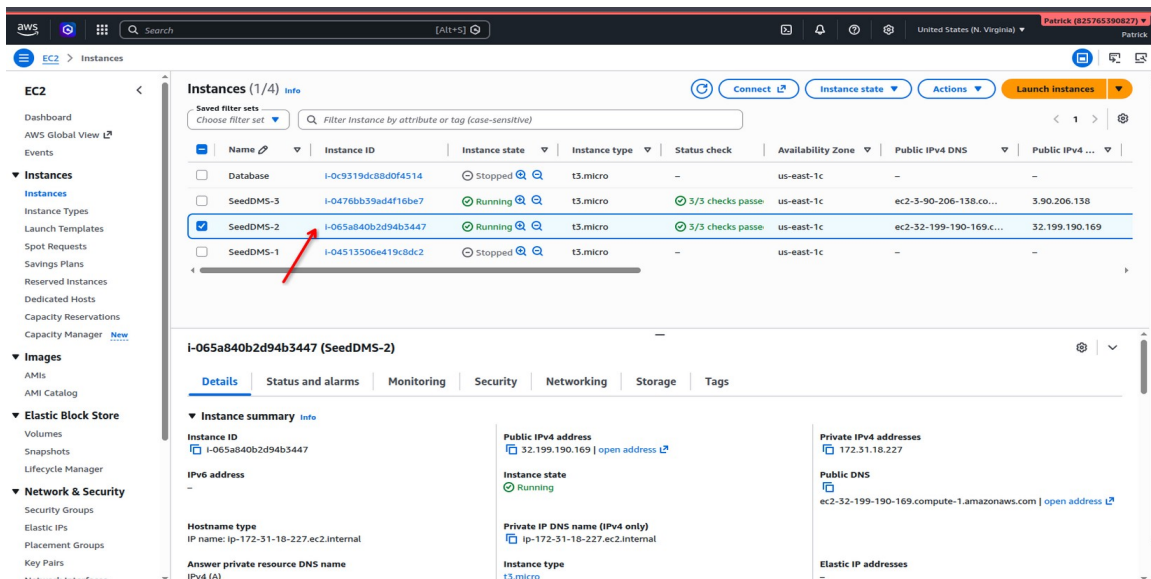
Step 55

Now back at your instances page we will rename the 2 instances to match the first one but with number to tell which one is which. Hover over the instance that you want to rename and click on the pencil icon. Select one of the 2 new instances that we just created we, will do the extra configuration.



Step 56

Select one of the new instances.



Step 57

Need to grab a copy of the public IP so we can SSH into it.

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with categories like EC2, Images, Elastic Block Store, and Network & Security. The main content area displays the 'Instance summary' for instance ID i-065a840b2d94b3447. Key information includes: Instance ID, Public IPv4 address (32.199.190.169), Instance state (Running), Hostname type, Answer private resource DNS name, Auto-assigned IP address, IAM role, IMDSv2 (Required), and Operator. A red arrow points to the Public IPv4 address. Below the summary are tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is active, showing AMI ID, Monitoring status (disabled), and Platform details (Linux/UNIX).

Step 58

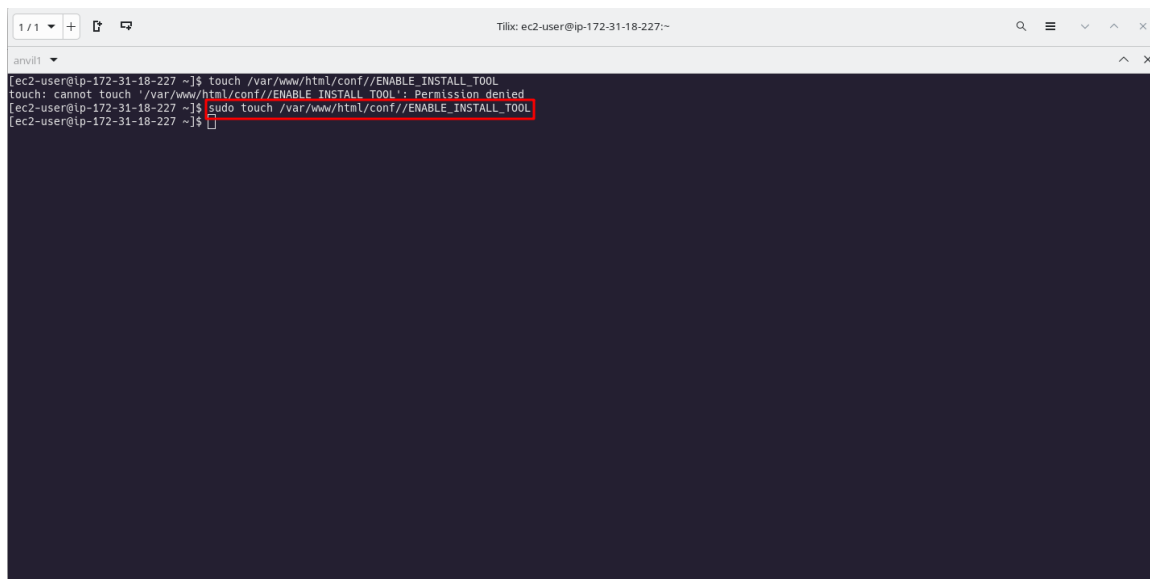
Just like the other times SSH into the instance using it's IP address. **ssh -i ~/file.pem ec2-user@IP**

The screenshot shows a terminal window titled 'Tilix: ec2-user@ip-172-31-18-227:~'. The user has executed the command `ssh -i ~/patrick.pem ec2-user@32.199.190.169`. The terminal output shows a warning about the connection not using a post-quantum key exchange algorithm, followed by the Amazon Linux 2023 logo and the URL <https://aws.amazon.com/linux/amazon-linux-2023>. The last login information is displayed: 'Last login: Sun Sep 6 15:02:51 2026 from 8.40.186.164'. The prompt is now `[ec2-user@ip-172-31-18-227 ~]$`.

Step 59

Create the installation tool use the command.

Sudo touch /var/www/html/conf/ENABLE_INSTALL_TOOL



```
1 / 1 + [ ] [ ]  
Tilix: ec2-user@ip-172-31-18-227:~  
-anvil1  
[ec2-user@ip-172-31-18-227 ~]$ touch /var/www/html/conf/ENABLE_INSTALL_TOOL  
touch: cannot touch '/var/www/html/conf/ENABLE_INSTALL_TOOL': Permission denied  
[ec2-user@ip-172-31-18-227 ~]$ sudo touch /var/www/html/conf/ENABLE_INSTALL_TOOL  
[ec2-user@ip-172-31-18-227 ~]$
```

SeedDMS Browser Configuration

Step 60

Next we are going to go to the browser so we can configure the settings for the SeedDMS application like we did on the first one. So go to: **http://IP address/seeddms/install**

Your browser should look similar to the image below.

SeedDMS Installation for version 6.0.34

Server settings

Root directory:

Http Root:

Content directory:

Directory for full text index:

Directory for partial uploads:

Cache directory:

Extra PHP include Path:

Database settings

Database Type:

Server name:

Database:

Username:

Password:

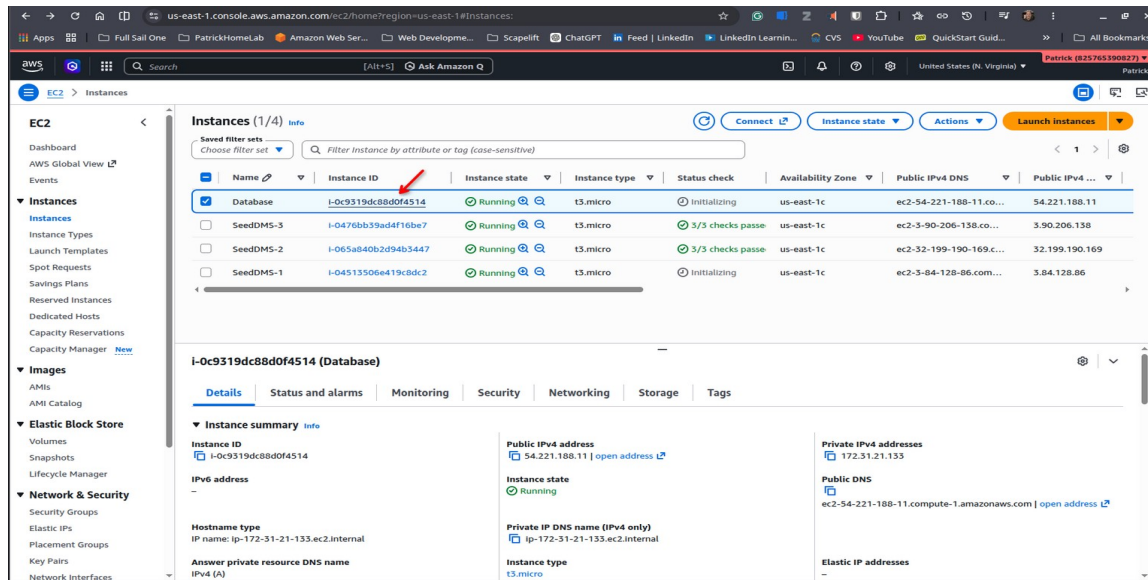
Create database tables: ☐

Red arrows point to the following fields:

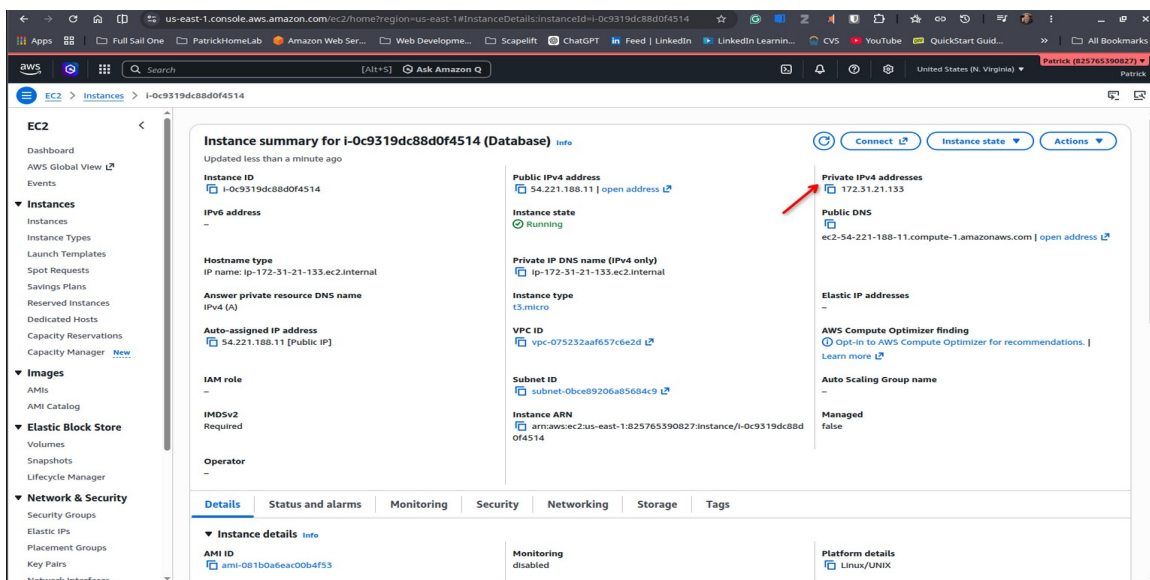
- Extra PHP include Path
- Database private IP (pointing to the Server name field)
- Database

Step 61

Now back in AWS we need to go to the database by clicking on the instance.



Here we are need to grab a copy of the **Private IP** address.



Step 62

Now come back to SeedDMS install site. Just like with the first one need to fill out the following sections.

SeedDMS Installation for version 6.0.34

Server settings

Root directory:

Http Root:

Content directory:

Directory for full text index:

Directory for partial uploads:

Cache directory:

Extra PHP include Path: ←

Database settings

Database Type: ← Database type

Server name: ← Database private IP

Database: ←

Username:

Password:

Create database tables: ☐

- Extra PHP include Path: needs to be: /var/www/html/pear/
- Database Type: needs to be: mysql
- Server Name: This is the private IP of your database that we just saved.
- Database: is the name of your database.
- Then of course your username and password.
 - Note: you are not creating a new username and password. You are using the ones that we already created.
- Create database tables: **Leave this unchecked**
- Then click apply

NOTE: Make sure you do this entire section again for the 3rd SeedDMS EC2 instance that we created.

Configuration of the Document Root file.

Note: This will be done on all three SeedDMS web servers. In order to do this you will need the public IP of all three SeedDMS servers.

Adding the Installation Tool

Step 63

After SSH into the first instance **ssh -i ~/file.pem ec2-user@IP**. You can run an update if you want to however, it is not required. **sudo dnf update -y** The need to create the install tool file:

```
sudo touch /var/www/html/conf/ENABLE_INSTALL_TOOL
```

```

1/1 | + - + | Titled: ec2-user@ip-172-31-23-117:~
patrick@milv ~$ ssh -l -o /etc/pam.d/sshd ec2-user@100.27.197.2
Warning: Permanently added '100.27.197.2' (ED25519) key fingerprint to the list of known hosts.
Warning: Permanently added '100.27.197.2' (ED25519) to the list of known hosts.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html

A newer release of "Amazon Linux" is available.
Version 2023.12.28260989:
Run "/usr/bin/dnf check-release-update" for full release and version update info

#####
# Amazon Linux 2023
#####
# https://aws.amazon.com/linux/amazon-linux-2023
#####

Last login: Thu Sep 10 00:46:16 2026 from 8.40.186.164
[ec2-user@ip-172-31-23-117 ~]$ sudo dnf update --releasever=2023.12.28260989
Last metadata expiration check: 0:13:46 ago on Sun Sep 13 00:35:26 2026.
=====
WARNING:
A newer release of "Amazon Linux" is available.

Available Versions:
Version 2023.12.28260989:
Run the following command to upgrade to 2023.12.28260989:
dnf update --releasever=2023.12.28260989

Release notes:
https://docs.aws.amazon.com/linux/al2023/release-notes/releNotes-2023.12.28260989.html
=====
Dependencies resolved.
Nothing to do.
Completed.
[ec2-user@ip-172-31-23-117 ~]$ sudo touch /var/www/html/conf/ENABLE_INSTALL_TOOL
bash: sudo: command not found
[ec2-user@ip-172-31-23-117 ~]$ sudo touch /var/www/html/conf/ENABLE_INSTALL_TOOL
[ec2-user@ip-172-31-23-117 ~]$

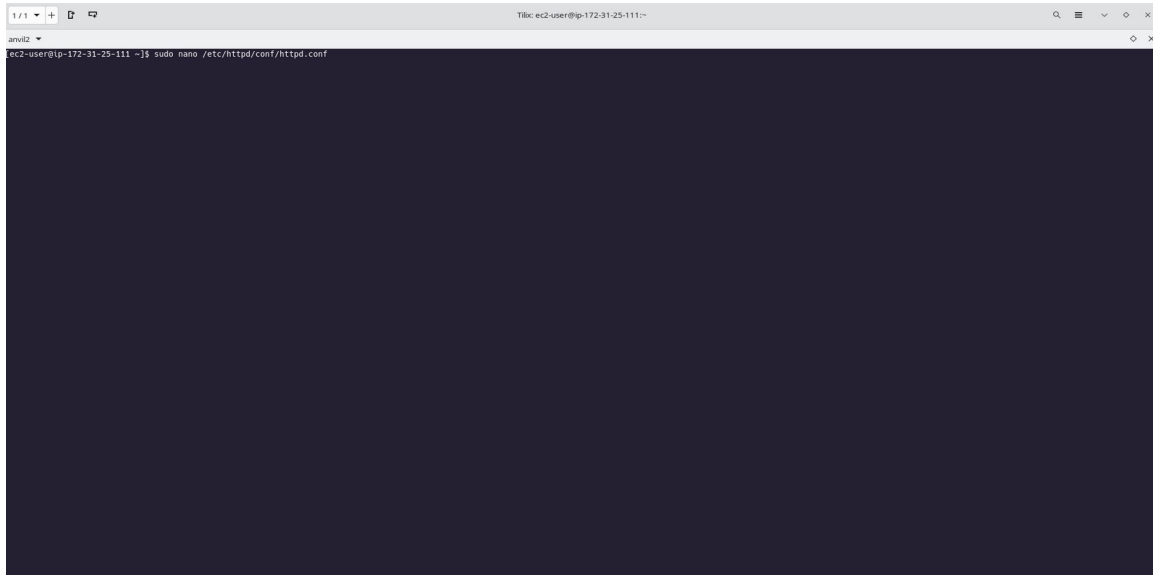
```

Accessing the Configuration File

Step 64

Next we are going to access Apache's configuration file.

`sudo nano /etc/httpd/conf/httpd.conf`



Editing the Configuration file

Step 65

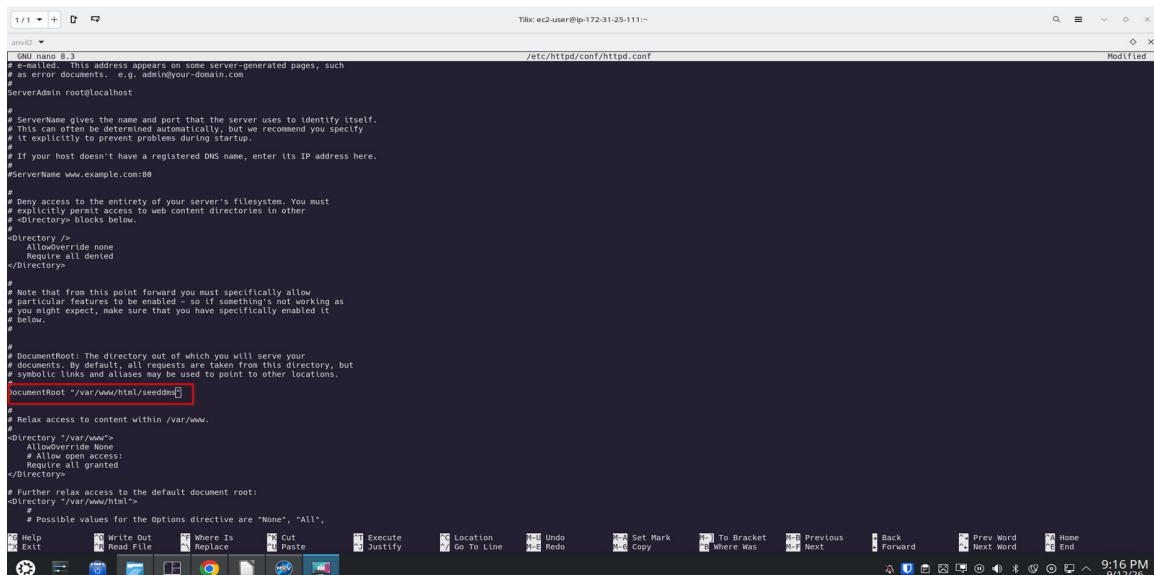
Warning: Edit the Apache configuration file carefully. Incorrect changes could brake the server and it will not load or load wrong.

Now in the config file, need to scroll down to where is says
“/var/www/html”.

DocumentRoot

Once you find it we are going to add seeddms at the end of it. It should look like the image below.

DocumentRoot “/var/www/html/seeddms”



```
GNU nano 8.3 /etc/httpd/conf/httpd.conf
# e-mailed. This address appears on some server-generated pages, such
# as error documents.  e.g. admin@your-domain.com
ServerAdmin root@localhost

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#ServerName www.example.com:80

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride none
    Require all denied
</Directory>

#
# Note that from this point forward you must specifically allow
# particular features to be enabled - so if something's not working as
# you might expect, make sure that you have specifically enabled it
# below.
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
DocumentRoot "/var/www/html/seeddms"

#
# Relax access to content within /var/www.
#
<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

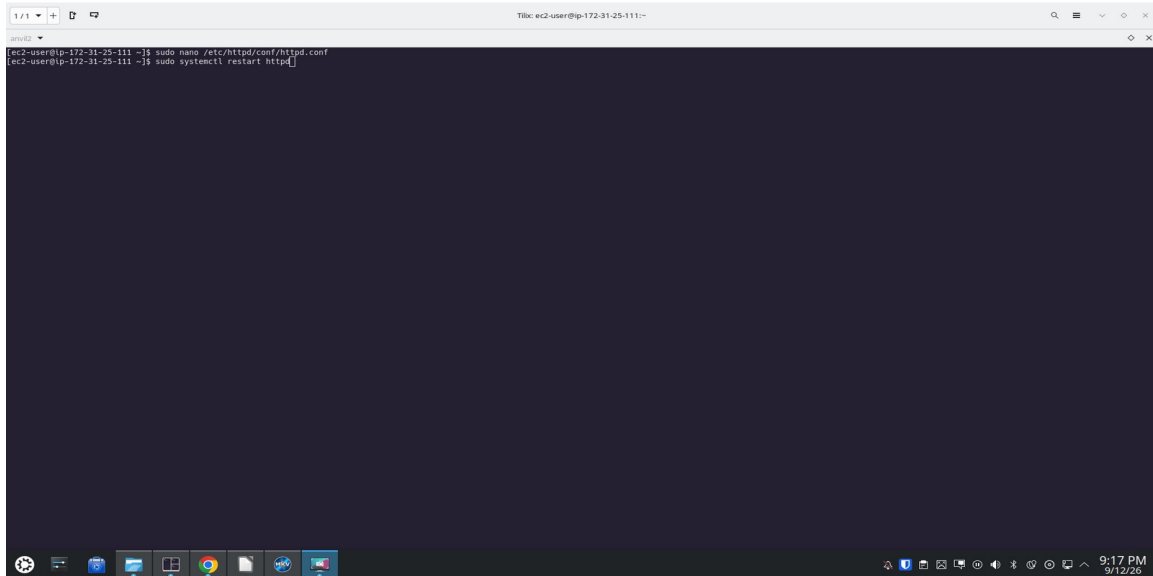
# Further relax access to the default document root:
<Directory "/var/www/html">
    #
    # Possible values for the Options directive are "None", "All",
    #
```

After you have made that change you will press **CTRL+O** then **Enter** (this will save the changes). The press **CTRL+X** to exit the config file.

Restarting Apache

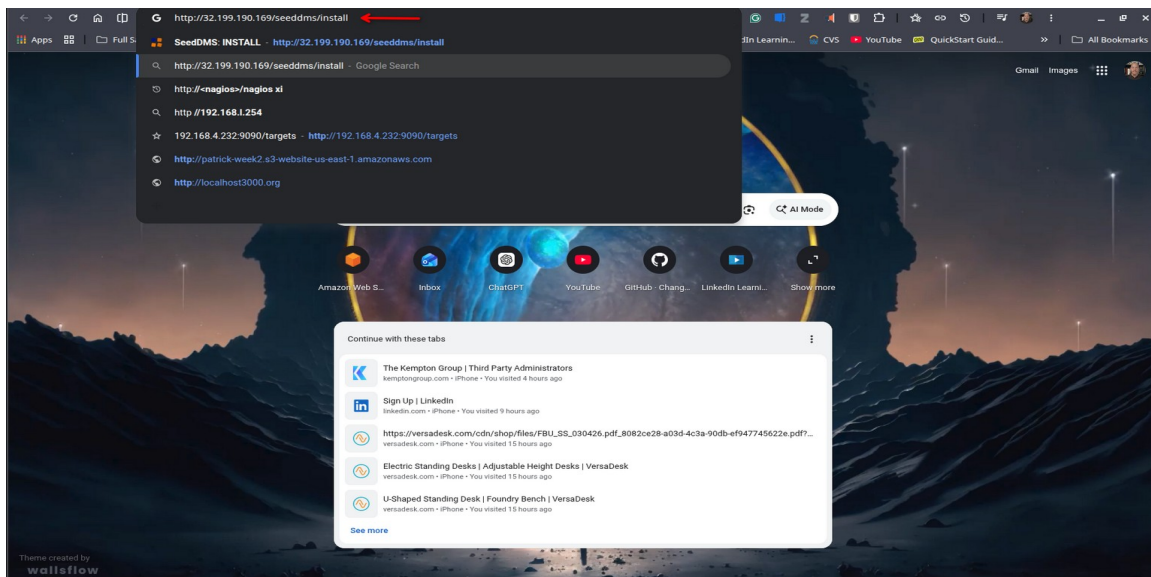
Step 66

Next we need to restart Apache for the changes to take affect. Run: **sudo systemctl restart httpd**



Step 67

Next we are going to go to the browser so we can configure the settings for the SeedDMS application like we did on the first one. So go to: **http://IP address/seeddms/install**



Your browser should look similar to the image Below

SeedDMS Installation for version 6.0.34

Server settings

Root directory:

Http Root:

Content directory:

Directory for full text index:

Directory for partial uploads:

Cache directory:

Extra PHP include Path:

Database settings

Database Type:

Server name:

Database:

Username:

Password:

Create database tables: ☐

Red arrows point to the following fields:

- Extra PHP include Path
- Database private IP
- Database

Step 68

Now you will go through the steps as you have previously without changing anything. When you come to the login page, go a head and log in.

SeedDMS

Sign in

User ID:

Password:

Language:

Theme:

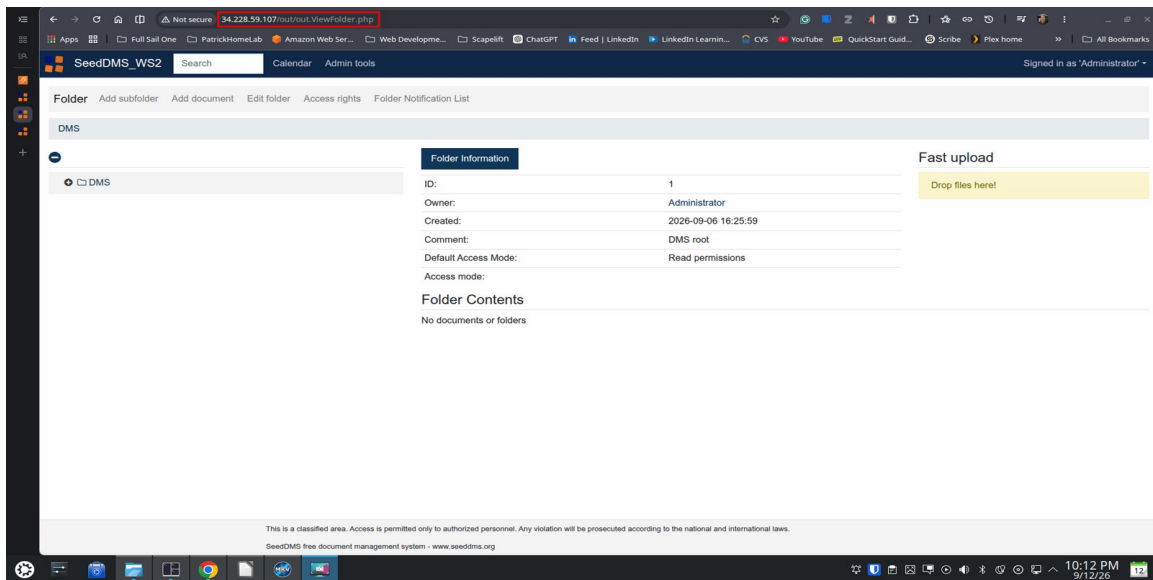
This is a classified area. Access is permitted only to authorized personnel. Any violation will be prosecuted according to the national and international laws.

SeedDMS free document management system - www.seeddms.org

Step 69

Now open a new browser and enter the IP address only for one of your SeedDMS servers.

At this point it should have taken you to the SeedDMS website.



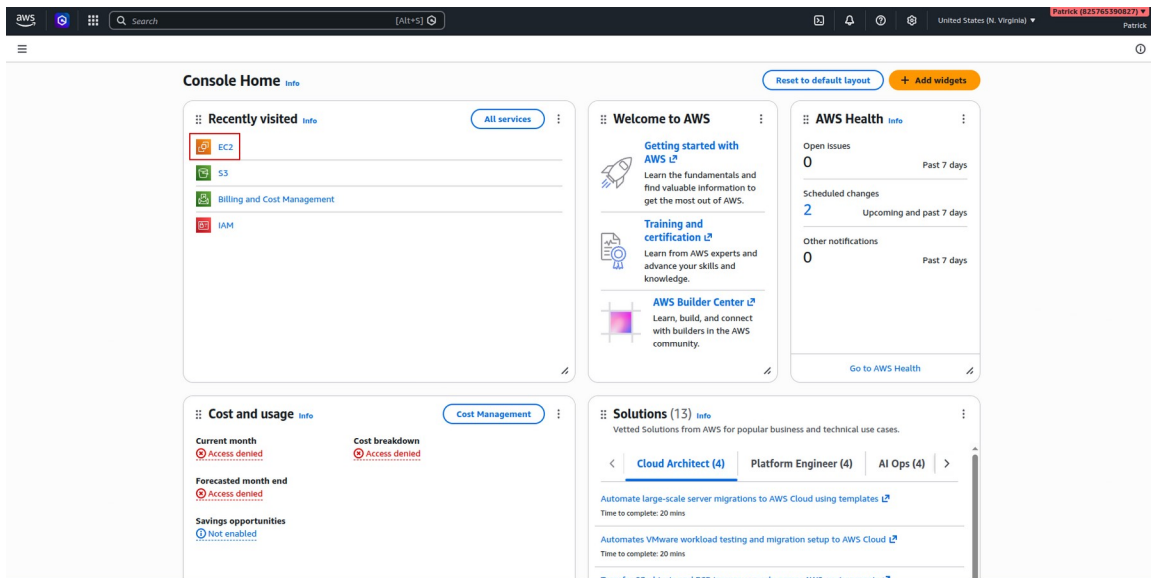
HAProxy Server

Creating the EC2 Instance

Step 70

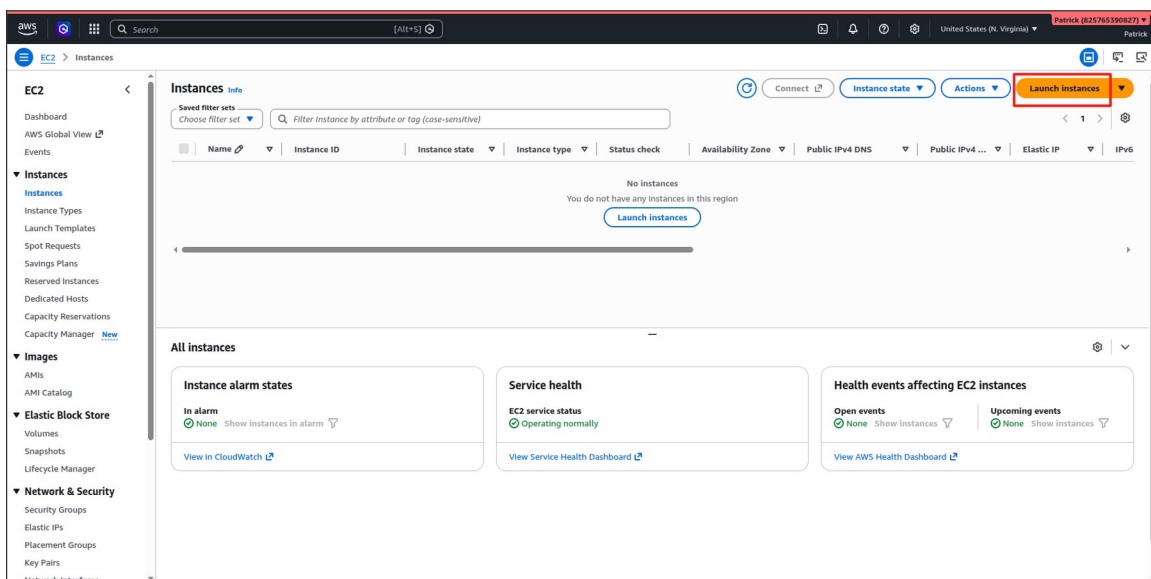
Go to AWS.com and sign in. When you open it, it will look similar to the image to the right.

Select the EC2 service



Step 71

To create an EC2 instance click “**Launch instance**” in the top right.



Step 72

Give the server a name. Helpful if you pick one the is relevant to its purpose.

Launch an instance info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags info

Name: [Add additional tags](#)

Application and OS Images (Amazon Machine Image) info

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

Recents **My AMIs** **Quick Start**

Amazon Linux **macOS** **Ubuntu** **Windows** **Red Hat** **SUSE Linux** **Debian**

Amazon Machine Image (AMI)

Amazon Linux 2023 kernel-6.18 AMI
ami-081b0d6eac00b4f53 (64-bit (x86), uefi-preferred) / ami-07987a01dcdb011ef (64-bit (ARM), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs Free tier eligible

Description

Amazon Linux 2023 (kernel-6.18) is a modern, general purpose Linux-based OS that will be supported until June 2029. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.12.20260831.0 x86_64 HVM kernel-6.18

Summary

Number of instances info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.12... [read more](#)
ami-081b0d6eac00b4f53

Virtual server type (instance type)

t3.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t3.micro instance usage (or t3.micro where t3.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance. Charges may apply depending on your account's free tier status.

[Cancel](#) [Launch instance](#) [Preview code](#)

Step 73

For the networking you want to make sure the “Auto-assign public IP” is enabled.

Network settings info [Edit](#)

Network info

vpc-07523232af657c6e2d

Subnet info

No preference (Default subnet in any availability zone)

Auto-assign public IP info

☒ Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) info

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

☒ Create security group ☐ Select existing security group

We'll create a new security group called "launch-wizard-1" with the following rules:

☒ Allow SSH traffic from Helps you connect to your instance

☐ 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.

Configure storage info [Advanced](#)

1x GiB Root volume, 3000 IOPS, Not encrypted

Summary

Number of instances info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.12... [read more](#)
ami-081b0d6eac00b4f53

Virtual server type (instance type)

t3.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t3.micro instance usage (or t3.micro where t3.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance. Charges may apply depending on your account's free tier status.

[Cancel](#) [Launch instance](#) [Preview code](#)

Step 74

Next you want to make sure that the you select “existing security group”.

- select the security group that you made earlier.

Network settings [Info](#) [Edit](#)

Network [Info](#)
vpc-075232aaf657cde2d

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable
Additional charges apply when outside of free tier allowance

Firewall (security groups) [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.

☐ Create security group ☒ Select existing security group

Common security groups [Info](#) [Compare security group rules](#)

Select security groups

default sg-0e346edf6a3c35a2c X
VPC: vpc-075232aaf657cde2d

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Configure storage [Info](#) [Advanced](#)

1x 8 GIB gp3 Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage X

[Add new volume](#)

Summary [Info](#)

Number of instances 1

Software image (AMI)
Amazon Linux 2023 AMI 2023.12...[read more](#)
ami-081b0a6eac00b4f53

Virtual server type (instance type)
t3.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t3.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance. Charges may apply depending on your account's free tier status.

[Cancel](#) [Launch instance](#) [Preview code](#)

Step 75

Leave default settings for the rest and click on “Launch instance” in the bottom right corner.

☐ Create security group ☒ Select existing security group

Common security groups [Info](#) [Compare security group rules](#)

Select security groups

default sg-0e346edf6a3c35a2c X
VPC: vpc-075232aaf657cde2d

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Configure storage [Info](#) [Advanced](#)

1x 8 GIB gp3 Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage X

[Add new volume](#)

☒ Click refresh to view backup information
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

File systems

☐ S3 Files - new ☐ EFS

☐ FSx ☒ None

Advanced details [Info](#)

Summary [Info](#)

Number of instances 1

Software image (AMI)
Amazon Linux 2023 AMI 2023.12...[read more](#)
ami-081b0a6eac00b4f53

Virtual server type (instance type)
t3.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t3.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet. Data transfer charges are not included as part of the free tier allowance. Charges may apply depending on your account's free tier status.

[Cancel](#) [Launch instance](#) [Preview code](#)

Step 76

After it is launched you will come to this screen. Here you will confirm that the instances are running. Then you will need to make sure the you get 3/3 checks passed.

The screenshot displays the AWS Management Console for the EC2 service. The left-hand navigation pane shows the 'Instances' section under 'EC2'. The main content area is titled 'Instances (1/1)' and features a table with the following columns: Name, Instance ID, Instance state, Instance type, Status check, Availability Zone, Public IPv4 DNS, Public IPv4..., Elastic IP, and IPv6. A single instance, 'Database' (ID: i-0c9319dc88d0f4514), is listed with a 'Running' status and '3/3 checks passed'. Below the table, the 'Details' tab for the instance 'i-0c9319dc88d0f4514 (Database)' is active, showing fields for Instance ID, Public IPv4 address, Private IPv4 addresses, Public DNS, and Instance type (t3.micro).

Name	Instance ID	Instance state	Instance type	Status check	Availability Zone	Public IPv4 DNS	Public IPv4...	Elastic IP	IPv6
Database	i-0c9319dc88d0f4514	Running	t3.micro	3/3 checks passed	us-east-1c	ec2-52-91-171-9.com...	52.91.171.9	-	-

i-0c9319dc88d0f4514 (Database)

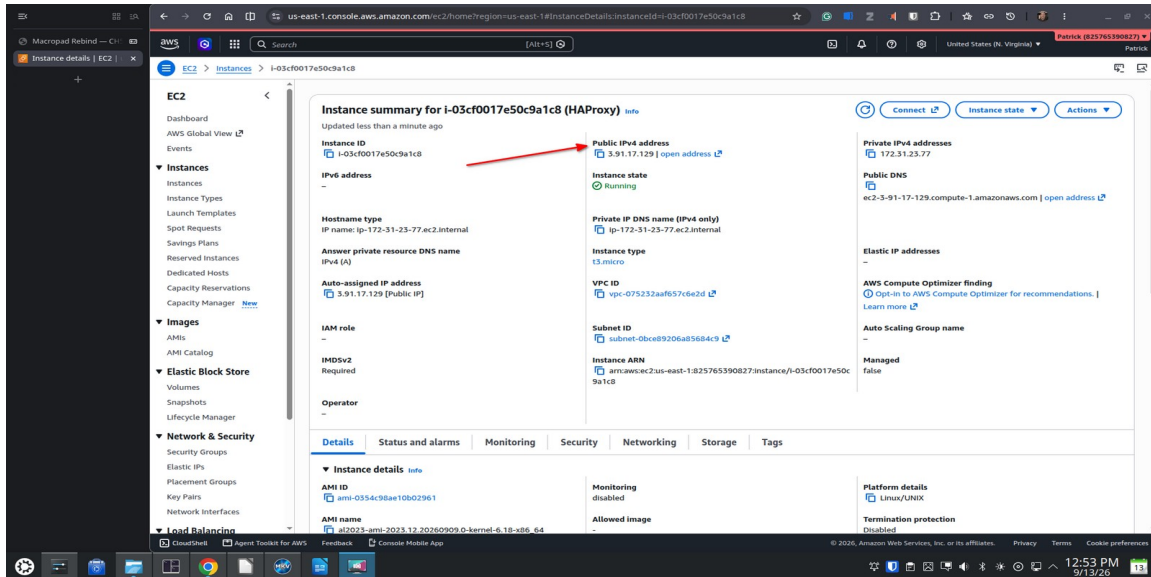
Instance summary

Instance ID i-0c9319dc88d0f4514	Public IPv4 address 52.91.171.9 [open address]	Private IPv4 addresses 172.31.21.133
IPv6 address -	Instance state Running	Public DNS ec2-52-91-171-9.compute-1.amazonaws.com [open address]
Hostname type IP name: ip-172-31-21-133.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-21-133.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t3.micro	

SSH Into Instance

Step 77

Need to get a copy of the public IP so that we can SSH into the instance.

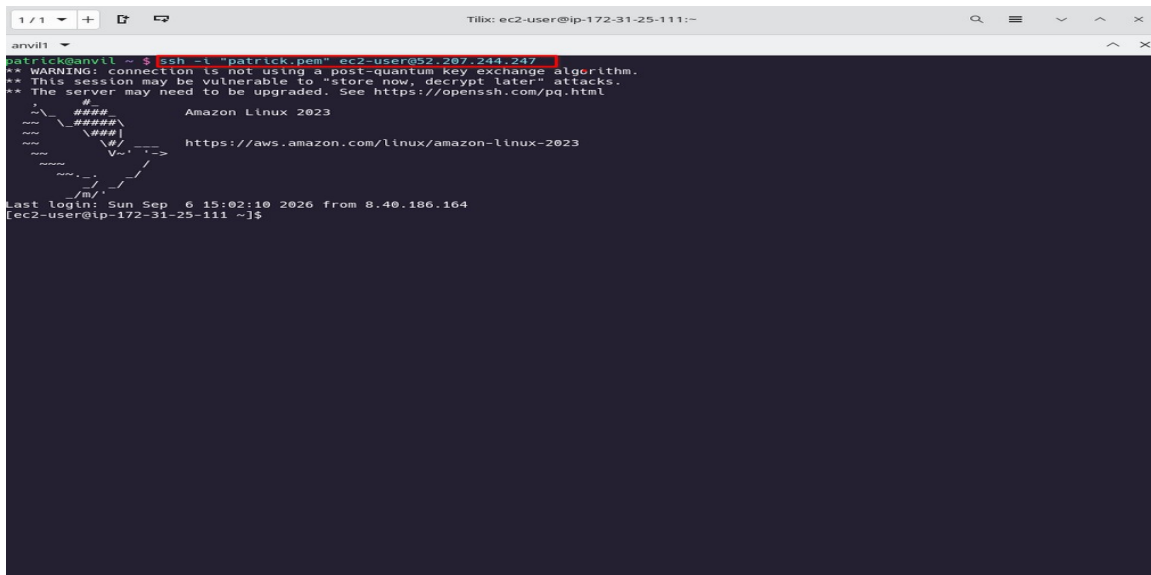


Step 78

To SSH into the instance use command:

ssh -i ~/your key.pem ec2-user@public IP.

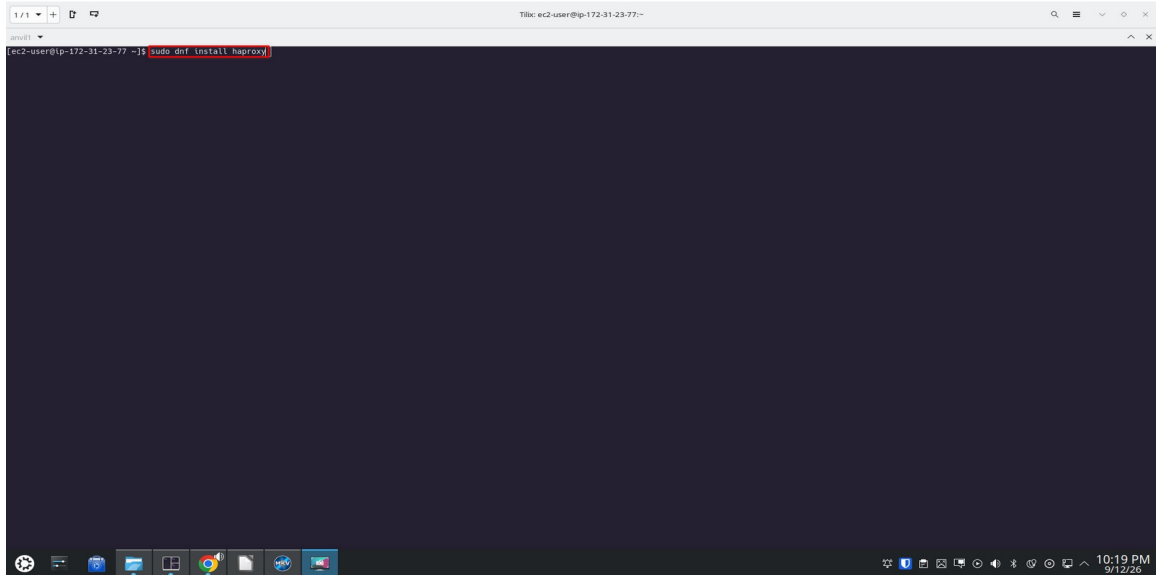
After your in we are going to update the system with: **sudo dnf update -y**



Installation of HAProxy

Step 79

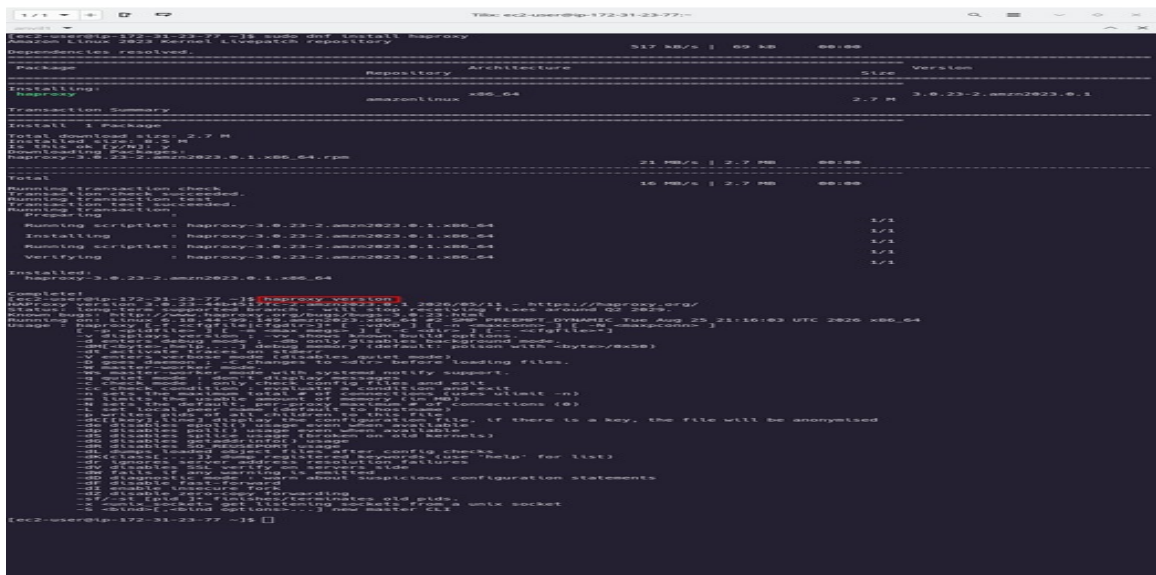
To install HAProxy. **sudo dnf install haproxy**



Verification of Correct Installation

Step 80

Use the command **haproxy version**



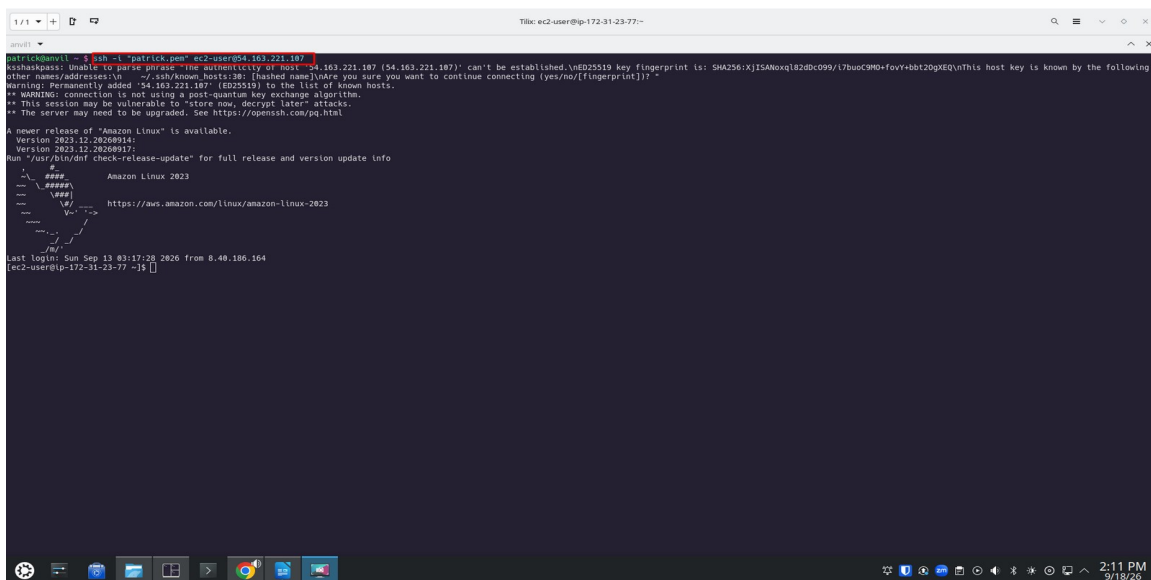
Creation of Self-Signed SSL Certification

The following section is about how to create and combine an SSL certification.

Step 81

If not already, SSH into the HAProxy server. Use command:

ssh -i “file.pem” ec2-user@public IP.



```
patrick@mac: ~ % ssh -i "patrick.pem" ec2-user@54.163.221.107
Warning: Permanently added '54.163.221.107' (ED25519) to the list of known hosts.
ec2-user@ip-172-31-23-77:~$
A newer release of "Amazon Linux" is available.
Version 2023.12.20206914:
Version 2023.12.20206917:
Run "rpm/anaconda check-release-update" for full release and version update info
#
# Amazon Linux 2023
#
# https://aws.amazon.com/linux/amazon-linux-2023
#
Last login: Sun Sep 13 03:17:20 2020 from 8.40.186.164
[ec2-user@ip-172-31-23-77 ~]$
```

Enable and start HAProxy

Step 82

First we need to check the status of HAProxy use the command:

sudo systemctl status haproxy

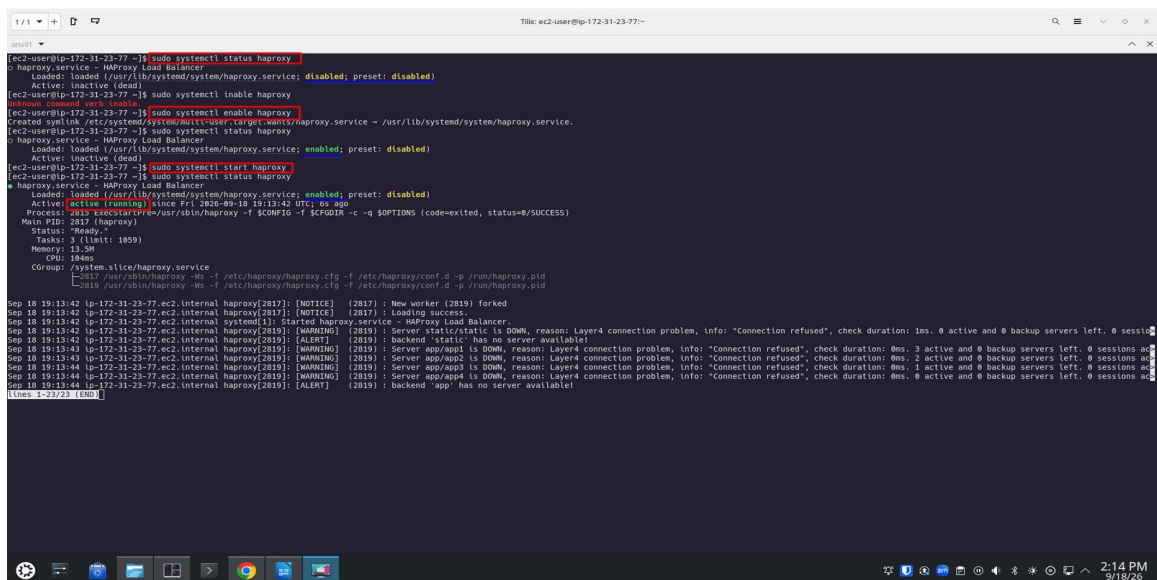
If your HAProxy in disabled then you need to enable it. Use the command:

sudo systemctl enable haproxy

After you enable you need to start HAProxy

sudo systemctl start haproxy

Then if you run another status command you want to see: **active (running)**



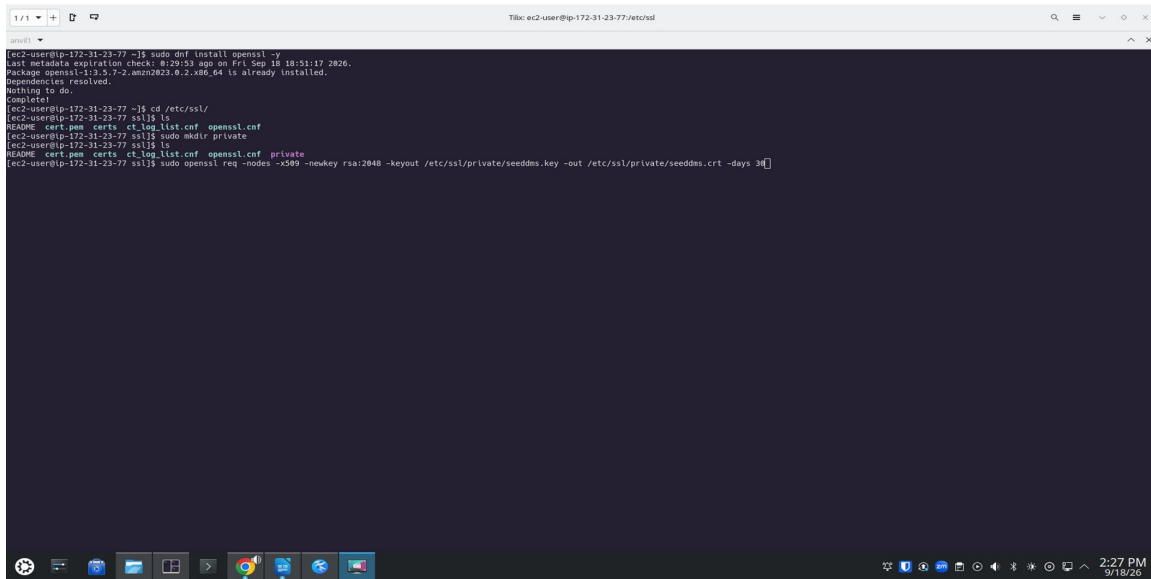
```
[ec2-user@ip-172-31-23-77 ~]$ sudo systemctl status haproxy
o haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/usr/lib/systemd/system/haproxy.service; disabled; preset: disabled)
   Active: inactive (dead)
[ec2-user@ip-172-31-23-77 ~]$ sudo systemctl inable haproxy
Unknown command (see man!)
[ec2-user@ip-172-31-23-77 ~]$ sudo systemctl enable haproxy
Created symlink /etc/systemd/system/multi-user.target.wants/haproxy.service → /usr/lib/systemd/system/haproxy.service.
[ec2-user@ip-172-31-23-77 ~]$ sudo systemctl status haproxy
o haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; preset: disabled)
   Active: inactive (dead)
[ec2-user@ip-172-31-23-77 ~]$ sudo systemctl start haproxy
[ec2-user@ip-172-31-23-77 ~]$ sudo systemctl status haproxy
o haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; preset: disabled)
   Active: active (running) since Fri 2020-09-18 19:13:42 UTC; 6s ago
     Process: 2019 execstart=/usr/sbin/haproxy -f $CONFIG -f $CMDLINE -c -q $OPTIONS (code=exited, status=0/SUCCESS)
    Main PID: 2817 (haproxy)
       State: 'ready'
        Tasks: 3 (limit: 1859)
         CPU: 104ms
    CGroup: /system.slice/haproxy.service
            └─2817 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d -p /run/haproxy.pid
              └─2819 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -f /etc/haproxy/conf.d -p /run/haproxy.pid

Sep 18 19:13:42 ip-172-31-23-77.ec2.internal haproxy[2817]: [NOTICE] (2817) : New worker (2819) forked
Sep 18 19:13:42 ip-172-31-23-77.ec2.internal haproxy[2817]: [NOTICE] (2817) : Loading success.
Sep 18 19:13:42 ip-172-31-23-77.ec2.internal systemd[1]: Started haproxy.service - HAProxy Load Balancer.
Sep 18 19:13:42 ip-172-31-23-77.ec2.internal haproxy[2819]: [WARNING] (2819) : Server static/static is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 1ms. 0 active and 0 backup servers left. 0 sessions active, 0 waiting connections are in queue.
Sep 18 19:13:42 ip-172-31-23-77.ec2.internal haproxy[2819]: [ALERT] (2819) : backend 'static' has no server available!
Sep 18 19:13:43 ip-172-31-23-77.ec2.internal haproxy[2819]: [WARNING] (2819) : Server app/app1 is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 0ms. 3 active and 0 backup servers left. 0 sessions active, 0 waiting connections are in queue.
Sep 18 19:13:43 ip-172-31-23-77.ec2.internal haproxy[2819]: [WARNING] (2819) : Server app/app2 is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 0ms. 2 active and 0 backup servers left. 0 sessions active, 0 waiting connections are in queue.
Sep 18 19:13:44 ip-172-31-23-77.ec2.internal haproxy[2819]: [WARNING] (2819) : Server app/app3 is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 0ms. 1 active and 0 backup servers left. 0 sessions active, 0 waiting connections are in queue.
Sep 18 19:13:44 ip-172-31-23-77.ec2.internal haproxy[2819]: [WARNING] (2819) : Server app/app4 is DOWN, reason: Layer4 connection problem, info: "Connection refused", check duration: 0ms. 0 active and 0 backup servers left. 0 sessions active, 0 waiting connections are in queue.
Sep 18 19:13:44 ip-172-31-23-77.ec2.internal haproxy[2819]: [ALERT] (2819) : backend 'app' has no server available!
```

Creating a Single-Sign SSL

Step 83

Now that HAProxy is up and running, next we need to install openssl run the command:
sudo dnf install openssl -y



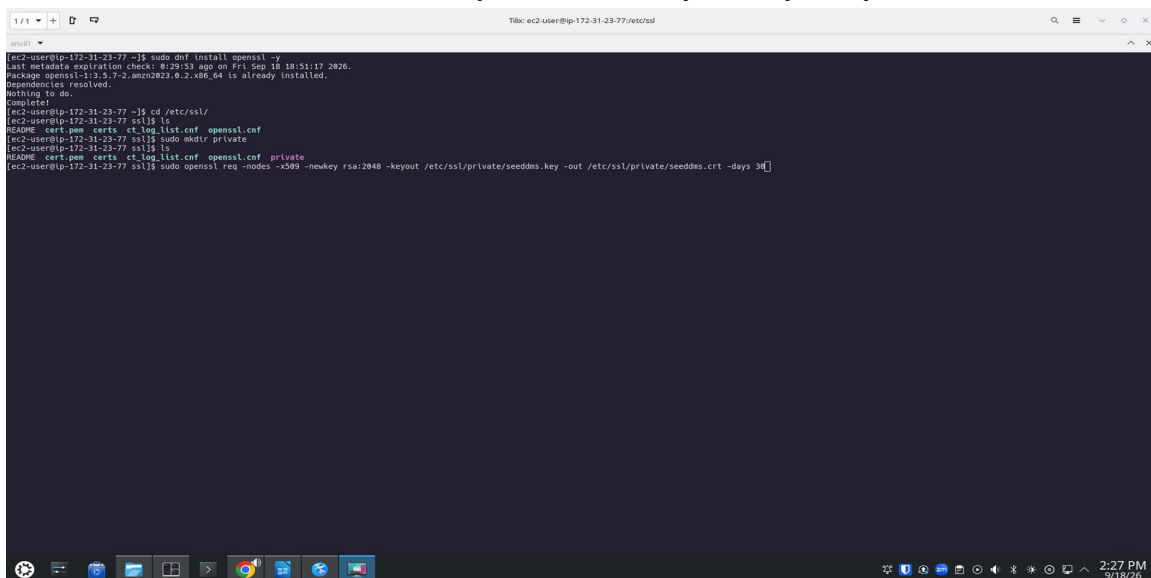
```
ec2-user@ip-172-31-23-77:~$ sudo dnf install openssl -y
Last metadata expiration check: 8:29:53 ago on Fri Sep 18 18:51:17 2020.
Package openssl-1:3.0.7-2.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
ec2-user@ip-172-31-23-77:~$ cd /etc/ssl/
ec2-user@ip-172-31-23-77:~$ ls
README  cert.pem  certs  ct_log_list.cnf  openssl.cnf
ec2-user@ip-172-31-23-77:~$ sudo mkdir private
ec2-user@ip-172-31-23-77:~$ ls
README  cert.pem  certs  ct_log_list.cnf  openssl.cnf  private
ec2-user@ip-172-31-23-77:~$ sudo openssl req -nodes -x509 -newkey rsa:2048 -keyout /etc/ssl/private/seeddms.key -out /etc/ssl/private/seeddms.crt -days 365
```

Step 84

With openssl installed we now need to create a directory for the SSL certificate to stay in. First we need to move to the ssl directory: **cd /etc/ssl/**

Next to make the directory us: **sudo mkdir private** Now for the key and the certification. To create the key and the certification run this command: **openssl req -nodes -x509 -newkey rsa:2048 -keyout /etc/ssl/private/your.key -out /etc/ssl/private/your.crt -days 30**

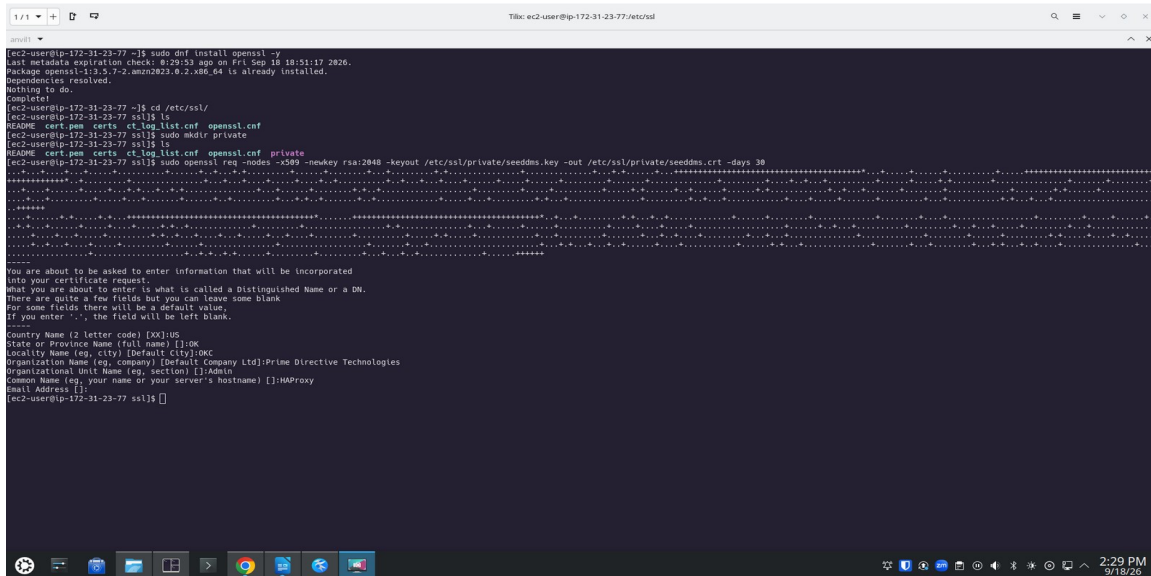
Note: Make sure that you rename the **your.key** and **your.crt**



```
ec2-user@ip-172-31-23-77:~$ sudo dnf install openssl -y
Last metadata expiration check: 8:29:53 ago on Fri Sep 18 18:51:17 2020.
Package openssl-1:3.0.7-2.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
ec2-user@ip-172-31-23-77:~$ cd /etc/ssl/
ec2-user@ip-172-31-23-77:~$ ls
README  cert.pem  certs  ct_log_list.cnf  openssl.cnf
ec2-user@ip-172-31-23-77:~$ sudo mkdir private
ec2-user@ip-172-31-23-77:~$ ls
README  cert.pem  certs  ct_log_list.cnf  openssl.cnf  private
ec2-user@ip-172-31-23-77:~$ sudo openssl req -nodes -x509 -newkey rsa:2048 -keyout /etc/ssl/private/seeddms.key -out /etc/ssl/private/seeddms.crt -days 365
```

Step 85

Follow and answer the question that you deem important. Keep in mind that this information is what is going to be in the key.

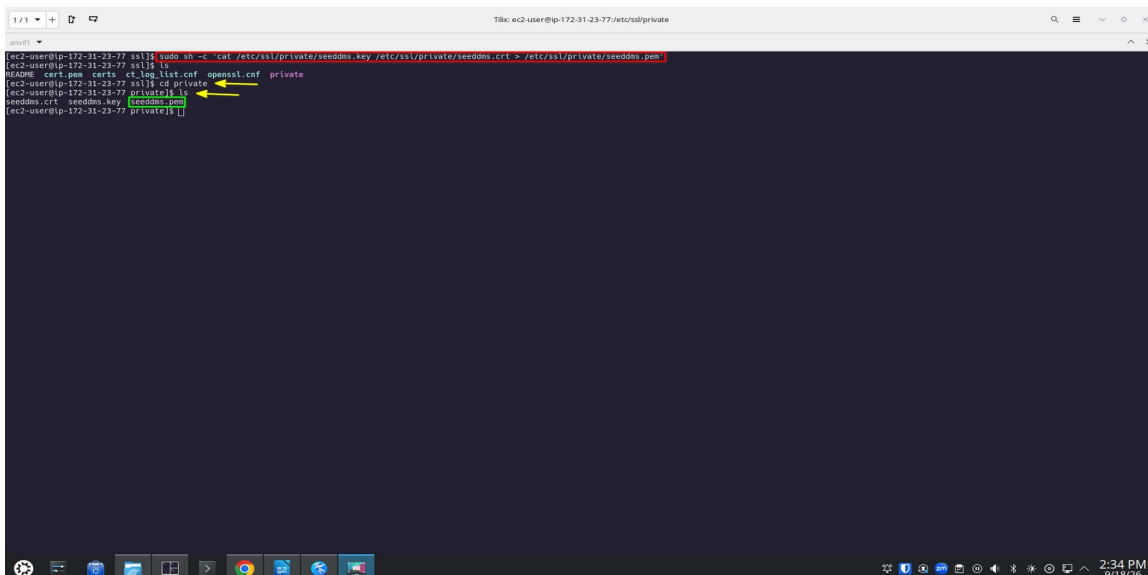


```
ec2-user@ip-172-31-23-77:~$ sudo dnf install openssl -y
Last metadata expiration check: 8:29:53 ago on Fri Sep 18 18:51:17 2020.
Package openssl-1:3.5.7-2.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
ec2-user@ip-172-31-23-77:~$ cd /etc/ssl/
ec2-user@ip-172-31-23-77:~$ ls
README  cert.pem  certs  ct_log_list.cnf  openssl.cnf
ec2-user@ip-172-31-23-77:~$ sudo mkdir private
ec2-user@ip-172-31-23-77:~$ ls
README  cert.pem  certs  ct_log_list.cnf  openssl.cnf  private
ec2-user@ip-172-31-23-77:~$ cd private
ec2-user@ip-172-31-23-77:~$ sudo openssl req -nodes -x509 -newkey rsa:2048 -keyout /etc/ssl/private/seeddns.key -out /etc/ssl/private/seeddns.crt -days 30
.....
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank.
For some fields there will be a default value,
If you enter -, the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:OK
Locality Name (eg, city) [Default: City]:OK
Organization Name (eg, company) [Default: Company Ltd]:Prime Directive Technologies
Organizational Unit Name (eg, section) []:Admin
Common Name (eg, your name or your server's hostname) []:HAPProxy
Email Address []:
ec2-user@ip-172-31-23-77:~$ ls
seeddns.crt  seeddns.key  [seeddns.pem]
```

Step 86

Next we are going to combine the key and the certificate into a single .pem file. Use the command:

cat /etc/ssl/private/test.key /etc/ssl/private/your.crt > /etc/ssl/private/your.pem



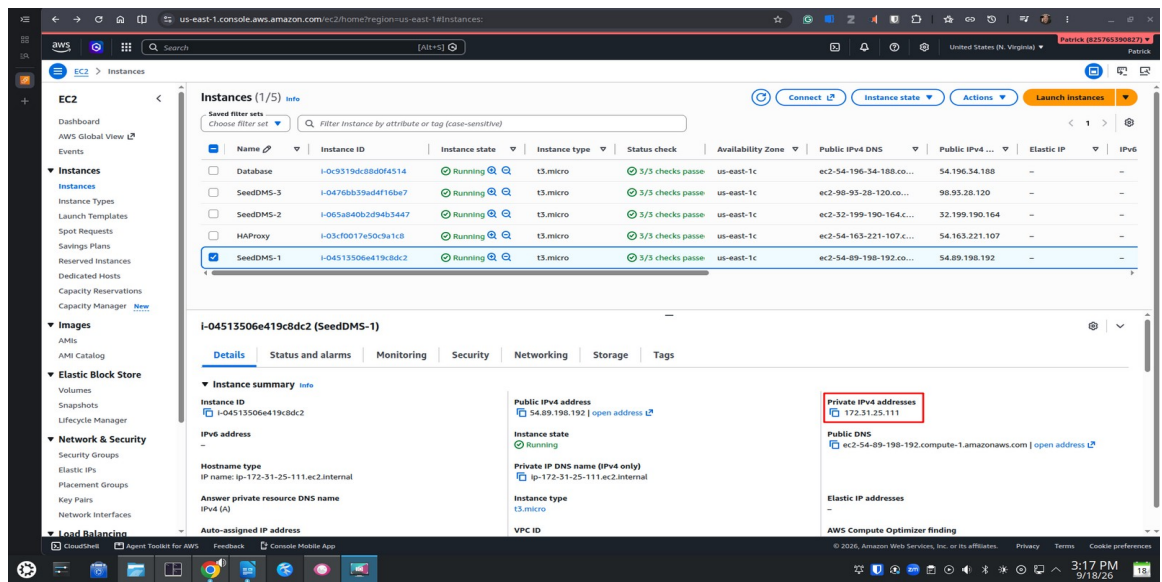
```
ec2-user@ip-172-31-23-77:~$ sudo sh -c 'cat /etc/ssl/private/seeddns.key /etc/ssl/private/seeddns.crt > /etc/ssl/private/seeddns.pem'
ec2-user@ip-172-31-23-77:~$ ls
README  cert.pem  certs  ct_log_list.cnf  openssl.cnf  private
ec2-user@ip-172-31-23-77:~$ cd private
ec2-user@ip-172-31-23-77:~$ ls
seeddns.crt  seeddns.key  [seeddns.pem]
```

Configuring HAProxy to use the SSL Certificate

Step 87

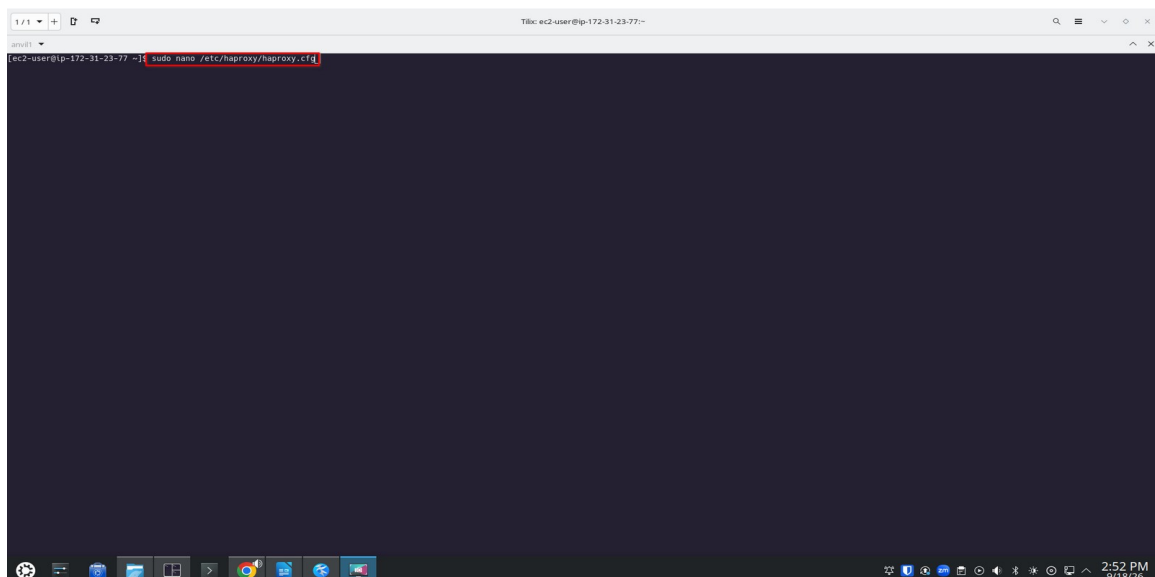
The modification that we are going to do next is going to need the **private** IP addresses for all 3 instances.

So if not already, sign in to AWS and navigate to your instances. Select one of the SeedDMS instances and on the right side is the **private** key.



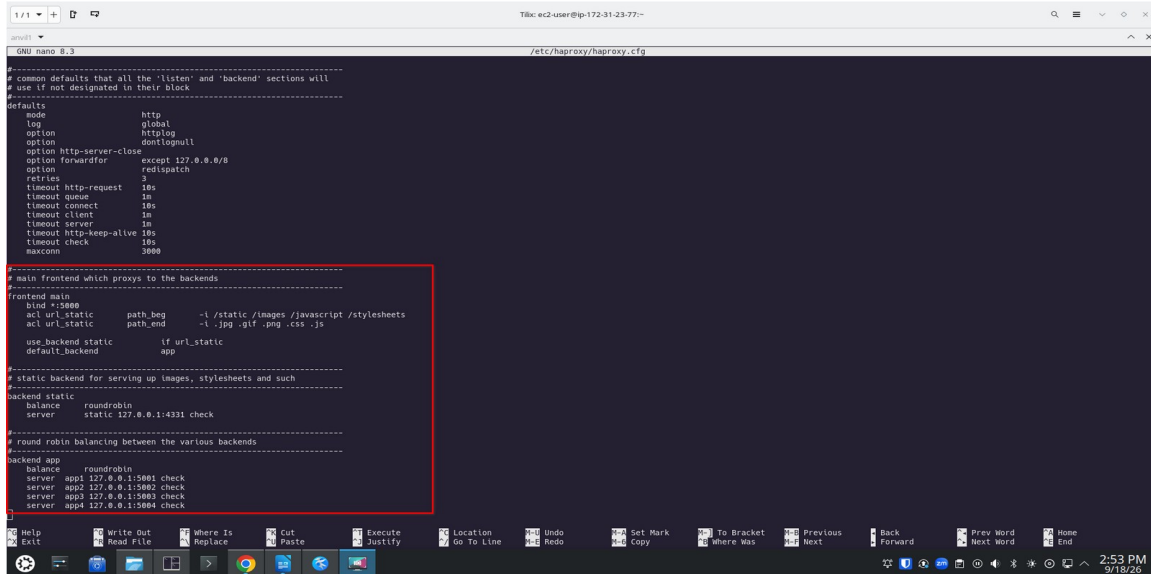
Step 88

First lets open the configuration file. Use the command: **sudo nano /etc/haproxy/haproxy.cfg**



Step 89

Once inside the config file need to scroll down the front end and back end section.



```
GNU nano 8.3 /etc/haproxy/haproxy.cfg
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
-----
defaults
    mode                http
    log                 global
    option              httplog
    option              dontlognull
    option http-server-close
    option forwardfor   except 127.0.0.0/8
    option               redispatch
    retries              3
    timeout http-request 10s
    timeout queue        1m
    timeout connect      10s
    timeout client       1m
    timeout server       1m
    timeout http-keep-alive 10s
    timeout check        10s
    maxconn             3000
-----

# main frontend which proxys to the backends
frontend main
    bind *:5000
    acl url_static      path_beg       -l /static /images /javascript /stylesheets
    acl url_static      path_end       -l .jpg .gif .png .css .js
    use_backend static   if url_static
    default_backend      app

# static backend for serving up images, stylesheets and such
backend static
    balance roundrobin
    server      static 127.0.0.1:4331 check

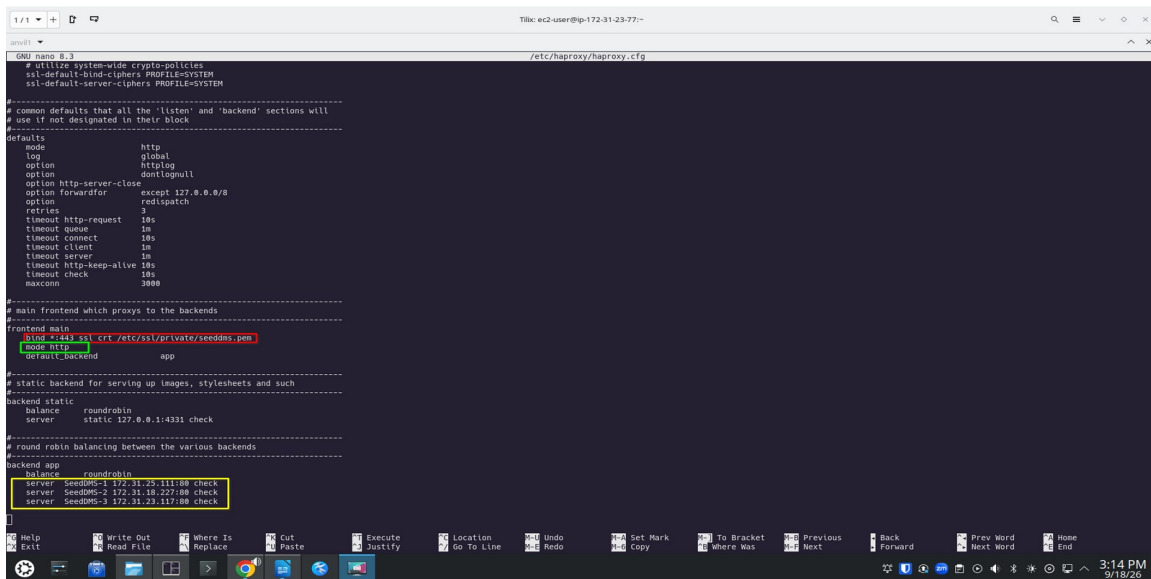
# round robin balancing between the various backends
backend app
    balance roundrobin
    server app1 127.0.0.1:5001 check
    server app2 127.0.0.1:5002 check
    server app3 127.0.0.1:5003 check
    server app4 127.0.0.1:5004 check
```

Step 90

Now lets start with the front end. First thing is binding. You should see that it is binded to port 5000. We need to change that to port 443 and then add are .pem file that you created. So replace the bind section with: **bind *:443 ssl crt /etc/ssl/private/seaddedms.pem** Next you can delete the lines that reference static. Next you will put in **mode http** Now for the back end. For the back end need to change the server names and IP to match that of each of your instances. Then instead of :500? check, you'll put :80 check.

After your all done your front and back end should look similar to the image below.

Then to save and exit use: **CTRL+S** to save and then **CTRL+X** to exit the config file.



```
GNU nano 2.9.3 /etc/haproxy/haproxy.cfg
# Utilize system-wide crypto policies
ssl-default-bind-ciphers PROFILE=SYSTEM
ssl-default-server-ciphers PROFILE=SYSTEM

#-----
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#-----
defaults
    mode                http
    log                  global
    option               httplog
    option               dontlognull
    option http-server-close
    option forwardfor    except 127.0.0.0/8
    option               redispatch
    retries              3
    timeout http-request 10s
    timeout queue        1m
    timeout connect      10s
    timeout client       1m
    timeout server       1m
    timeout http-keep-alive 10s
    timeout check        10s
    maxconn              2000

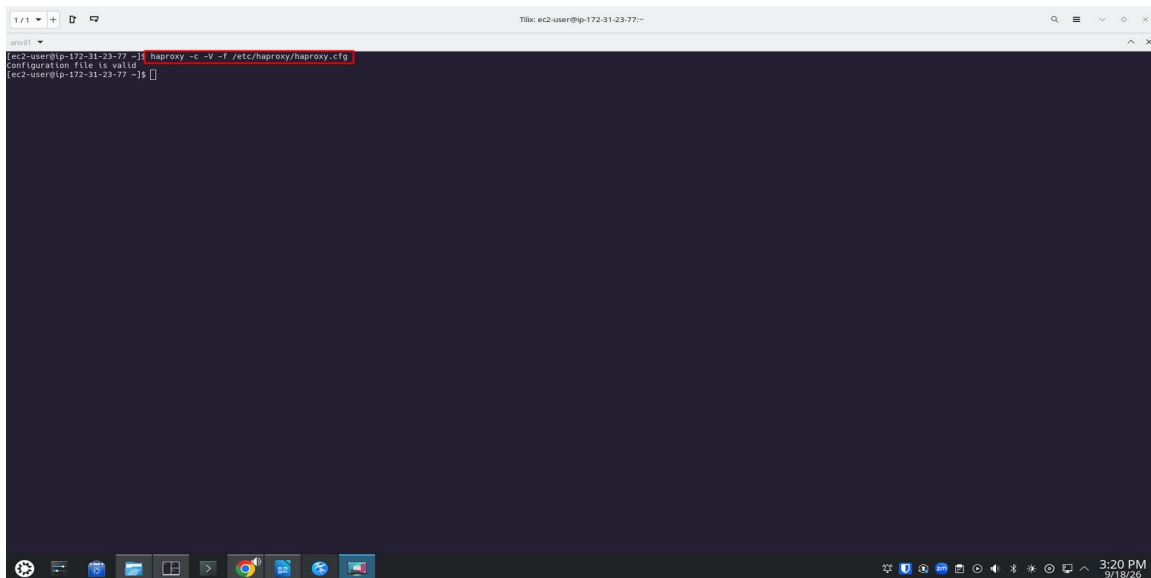
#-----
# main frontend which proxys to the backends
#-----
frontend main
    bind *:443 ssl crt /etc/ssl/private/seedns.pem
    mode http
    default_backend app

#-----
# static backend for serving up images, stylesheets and such
#-----
backend static
    balance roundrobin
    server static 127.0.0.1:4331 check

#-----
# round robin balancing between the various backends
#-----
backend app
    balance roundrobin
    server SeedNS-1 172.31.25.111:80 check
    server SeedNS-2 172.31.25.227:80 check
    server SeedNS-3 172.31.23.117:80 check
```

Step 91

Then lets verify that it is there. Run **haproxy -c -V /etc/haproxy/haproxy.cfg** What your looking for is that your configuration file is valid.



```
ec2-user@ip-172-31-23-77:~$ haproxy -c -V /etc/haproxy/haproxy.cfg
haproxy -c -V /etc/haproxy/haproxy.cfg
Configuration file is valid
ec2-user@ip-172-31-23-77:~$
```

Failover and Cookies Test

Now we are going to perform a failover test to confirm the sticky session dose in fact work as it should.

If not already at the website by using the HAProxy instance follow the next step to open the website.

Step 92

In AWS, select the HAProxy instance. Then go down to where it says **Public IPv4 address**. Then click on the link that says **open address**.

The screenshot shows the AWS Management Console interface for the EC2 Instances page. The left sidebar contains navigation links for EC2, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area displays a table of instances. The 'HAProxy' instance is selected, and its details are shown below the table. The 'Public IPv4 address' is highlighted with a red box, and the 'open address' link is also highlighted.

Name	Instance ID	Instance state	Instance type	Status check	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6
Database	i-0c9319dc88d0f4514	Running	t3.micro	3/3 checks passed	us-east-1c	ec2-54-234-208-156.c...	54.234.208.156	-	-
SeedDMS-3	i-0476bb39ad4f16be7	Running	t3.micro	3/3 checks passed	us-east-1c	ec2-100-61-145-190.c...	100.61.145.190	-	-
SeedDMS-2	i-065ab40b2d94b3447	Running	t3.micro	3/3 checks passed	us-east-1c	ec2-3-80-89-51.comp...	3.80.89.51	-	-
HAProxy	i-03cf0017e50c9a1c8	Running	t3.micro	3/3 checks passed	us-east-1c	ec2-18-209-8-196.co...	18.209.8.196	-	-
SeedDMS-1	i-04513506e419c8d2	Running	t3.micro	3/3 checks passed	us-east-1c	ec2-54-226-221-140.c...	54.226.221.140	-	-

I-03cf0017e50c9a1c8 (HAProxy)

Instance summary

- Instance ID: i-03cf0017e50c9a1c8
- IPv6 address: -
- Hostname type: IP name: ip-172-31-23-77.ec2.internal
- Answer private resource DNS name: IPv4 (A)
- Auto-assigned IP address: -
- VPC ID: -

Public IPv4 address
18.209.8.196 [open address](#)

Private IPv4 addresses
172.31.23.77

Public DNS
ec2-18-209-8-196.compute-1.amazonaws.com [open address](#)

Elastic IP addresses
-

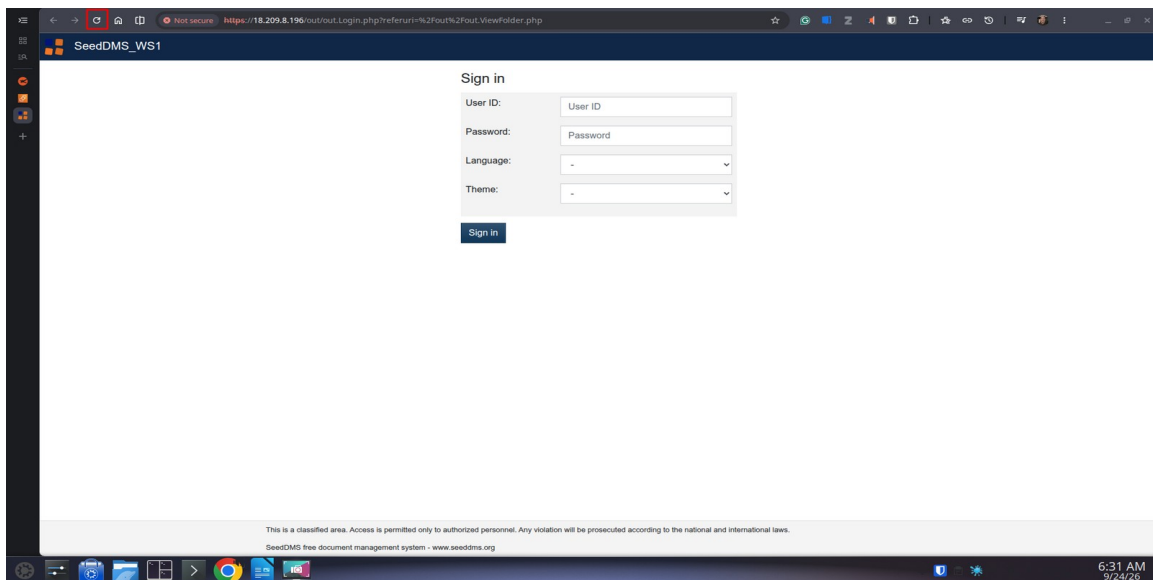
AWS Compute Optimizer finding

Testing the Cookies Persistence

Step 93

Next to test that the cookies persistence. Simply refresh the page 2-3 times. Watch which SeedDMS_WS? It is using. As long as the website does not change then the test passes.

The purpose of this test is to make sure that the HAProxy cookie persistence settings keeps the user on the same server if the page was to refresh.



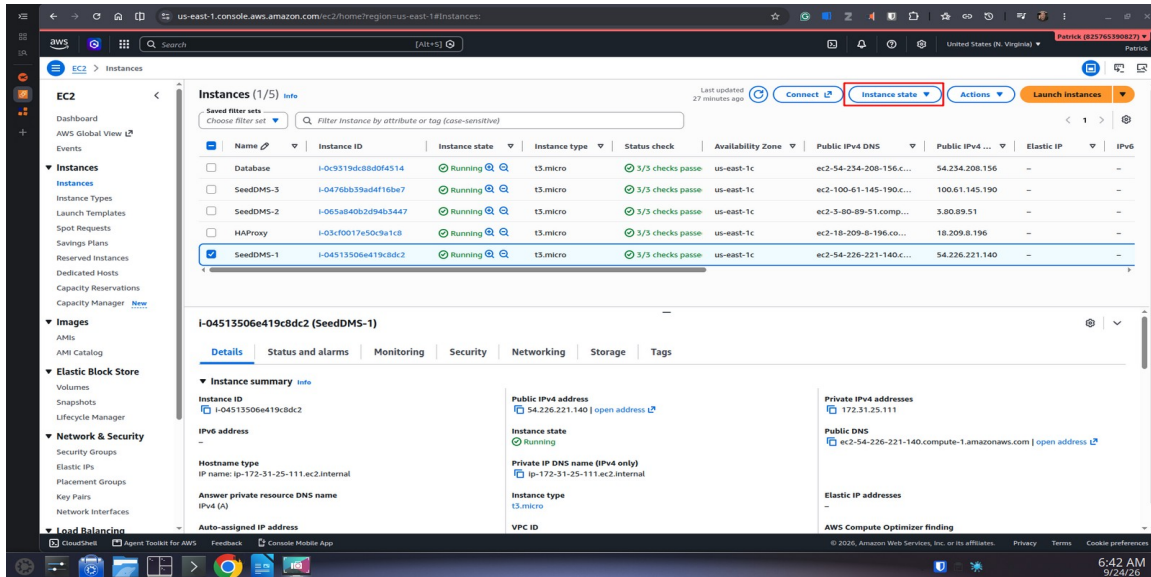
Failover Test

Next back on AWS you will select the SeedDMS sever instance that you use in the previous step.

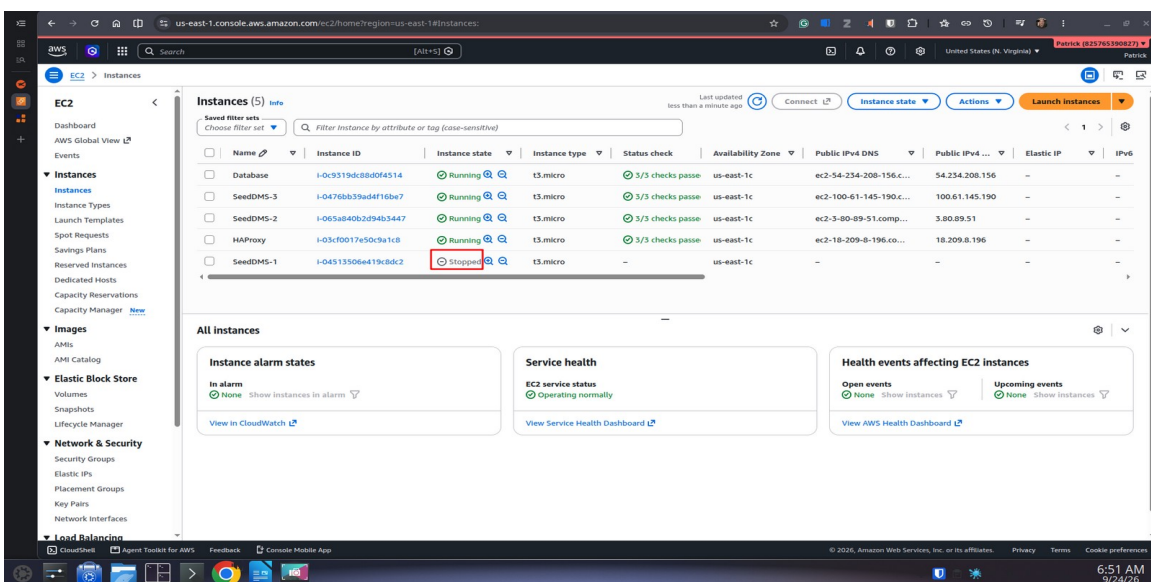
Step 94

Select the instance that you are using and go to **Instance state** in the top right.

Then scroll down to stop instance.

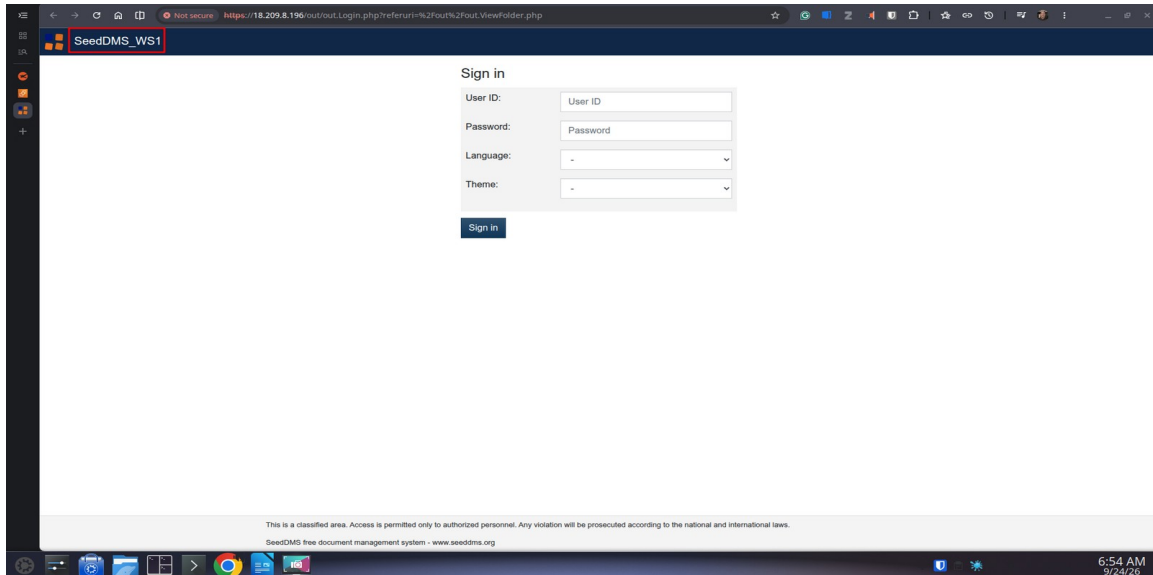


Wait about 2-3 minutes and then refresh the page. You want to make sure the instance shows the it is stopped.



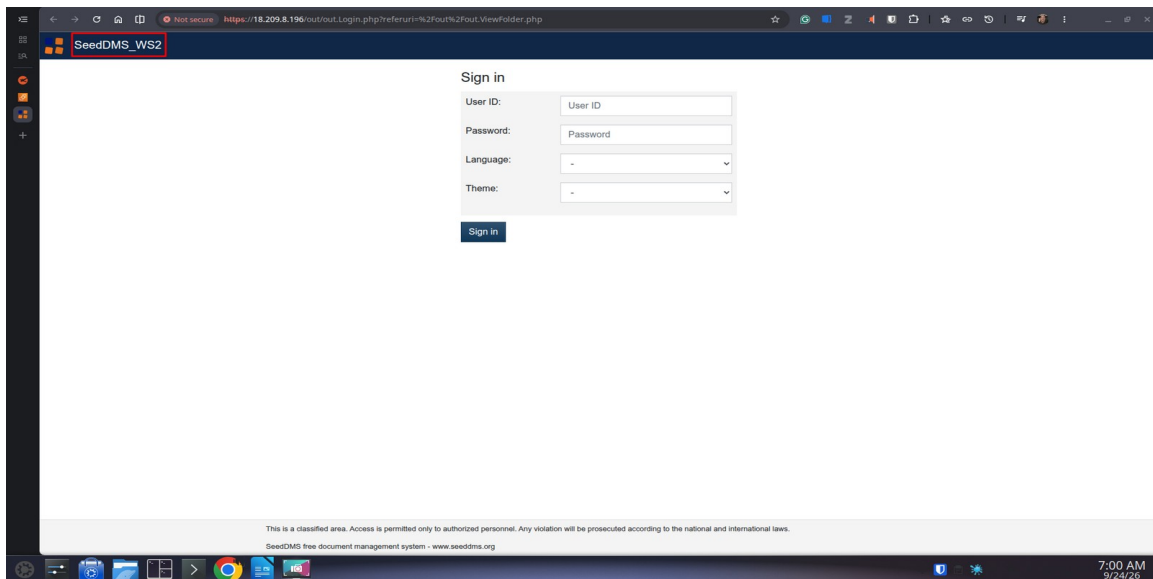
Step 95

Now go back to the page and confirm the instance that you stopped is the one your on.



Step 96

Next you will refresh the page. You will see that the sever changed. You can then refresh it a few more times and see the cookies will keep you on the new server.



Note: In the event that the browser is showing you on the same server, you will need to clear your browser cache or you can press Ctrl+Shift+R (Command+Shift+R on Mac) which will bypass the cache file and refresh the page.

For more detailed information on installation and configuration, see the sources in the References section below.

References

- [How to Create MySQL User](#)
- [Installing MariaDB on Amazon Linux 2023](#)
- [How to Install Apache Web Server on Amazon Linux 2023](#)
- [How to install PHP on Amazon Linux 2023 server](#)
- [HAProxy with SSL and Let's Encrypt](#)
- [HAproxy Sticky Sessions](#)