Sahal Mohamed

35480695

# ICT 171

# Introduction To Server Environments & Architectures

# Project Documentation

IP Address: 51.20.208.213

Domain Name: https://atmosfar.online

## Table of Contents

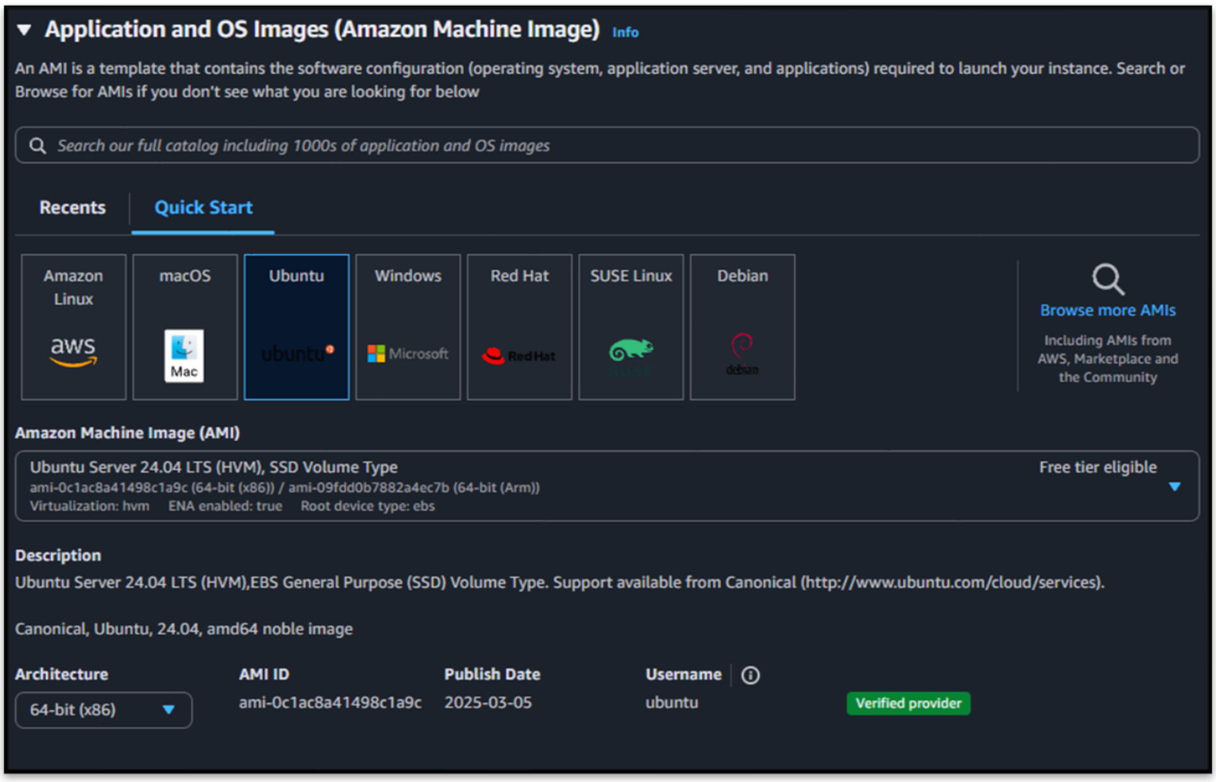
## How To Start An AWS Account?

1. Go To https://aws.amazon.com/ .
2. Click On “Create an AWS Account”.
3. Fill In The Details.

**Now You Should Have An AWS Account!**

## How To Setup A Server Instance?

1. Log Into AWS.
2. Search for “EC2” In The Search Bar Located At Top Left.
3. Once Chosen, Click On Launch “Launch Instance”.
4. Give Your Instance A Name.
5. Choose An Appropriate OS Image. In My Case I Chose Ubuntu Server.



**Application and OS Images (Amazon Machine Image)** Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

*Search our full catalog including 1000s of application and OS images*

**Amazon Machine Image (AMI)**

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type — Free tier eligible
ami-0c1ac8a41498c1a9c (64-bit (x86)) / ami-09fdd0b7882a4ec7b (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

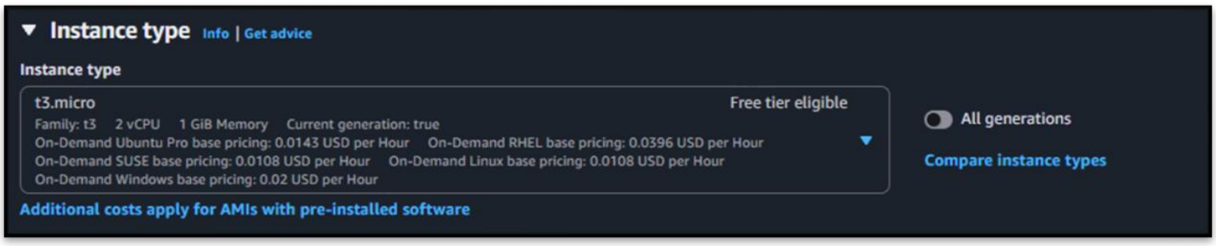
**Description**

Ubuntu Server 24.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Canonical, Ubuntu, 24.04, amd64 noble image

| Architecture | AMI ID | Publish Date | Username | |
|---|---|---|---|---|
| 64-bit (x86) | ami-0c1ac8a41498c1a9c | 2025-03-05 | ubuntu | Verified provider |

6. Choose An Instance Type. Ideally For A Web Server, t3.micro Would Be Enough.



**Instance type** Info | Get advice

Instance type

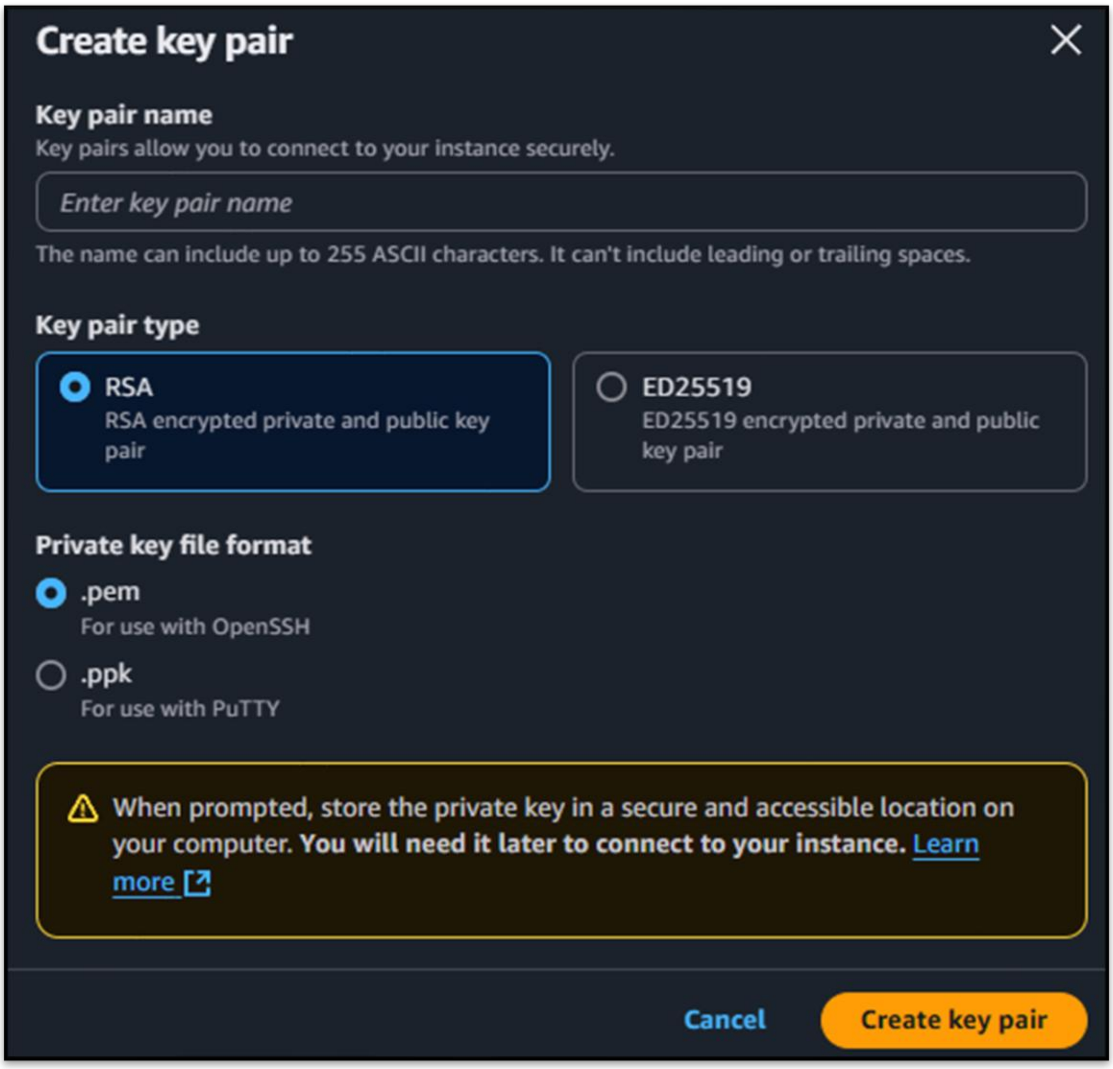
t3.micro Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0143 USD per Hour On-Demand RHEL base pricing: 0.0396 USD per Hour
On-Demand SUSE base pricing: 0.0108 USD per Hour On-Demand Linux base pricing: 0.0108 USD per Hour
On-Demand Windows base pricing: 0.02 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

7. Create A New Key Pair & Choose It. This Is Needed In Order To SSH Login To The Server.



**Create key pair**

**Key pair name**
Key pairs allow you to connect to your instance securely.

*Enter key pair name*

The name can include up to 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

**Private key file format**

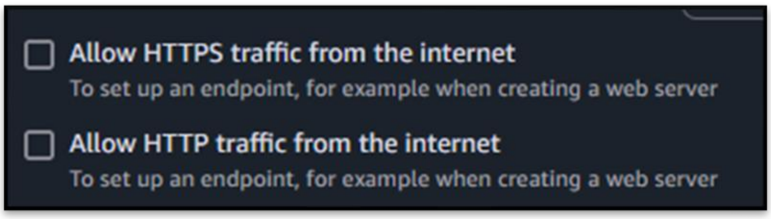
[✓ .pem] For use with OpenSSH

.ppk For use with PuTTY

⚠ 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

Cancel Create key pair

8. Configure Network Settings. Make Sure To Enable "Allow HTTPS/HTTP traffic from the internet".



☐ 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

9. Configure Storage Based On Your Needs.

**That's It! You Should Have A Instance Now.**

## How To Setup Apache In Your Instance?

1. Connect To Your Server Using SSH. Follow This Command And Replace As Needed

   ```
   ssh -i "path\to\keypair" ubuntu@<server-ip-address-without-brackets>
   ```

2. Make Sure Your Server Is Up To Date. Use These Following Commands To Update Your Server.

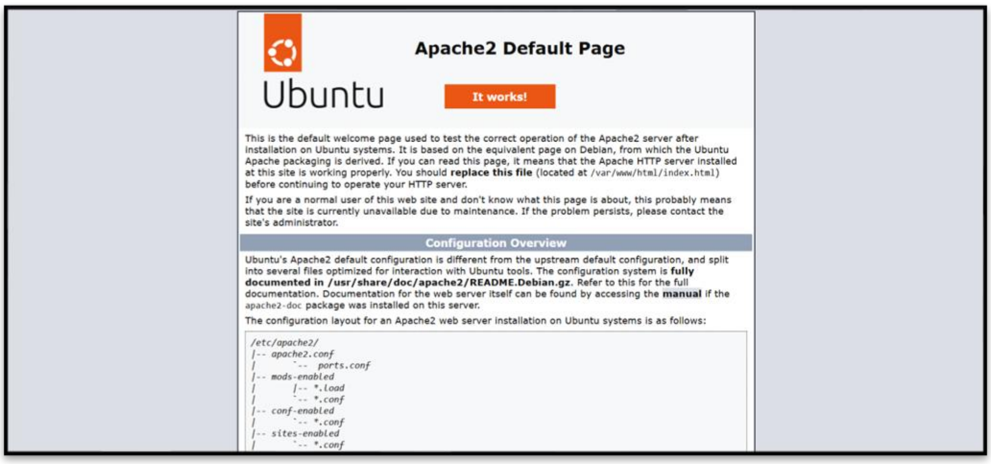
   ```
   sudo apt update
   ```

3. Install Apache.
   a. `sudo apt install apache2`

      You Will Be Prompted To Confirm The Installation. Press Y To Confirm.

   b. Make Sure Your Server Is Running By Typing In "`sudo systemctl status apache2`"

c. Once Done You Can Type In Your IP Address Into The Browser & You Will Have Something Like This.



This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

**Configuration Overview**

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

4. Setup Your Server Firewall By Typing In These Commands.

   ```
   sudo ufw allow in “Apache Full”
   sudo ufw enable
   ```

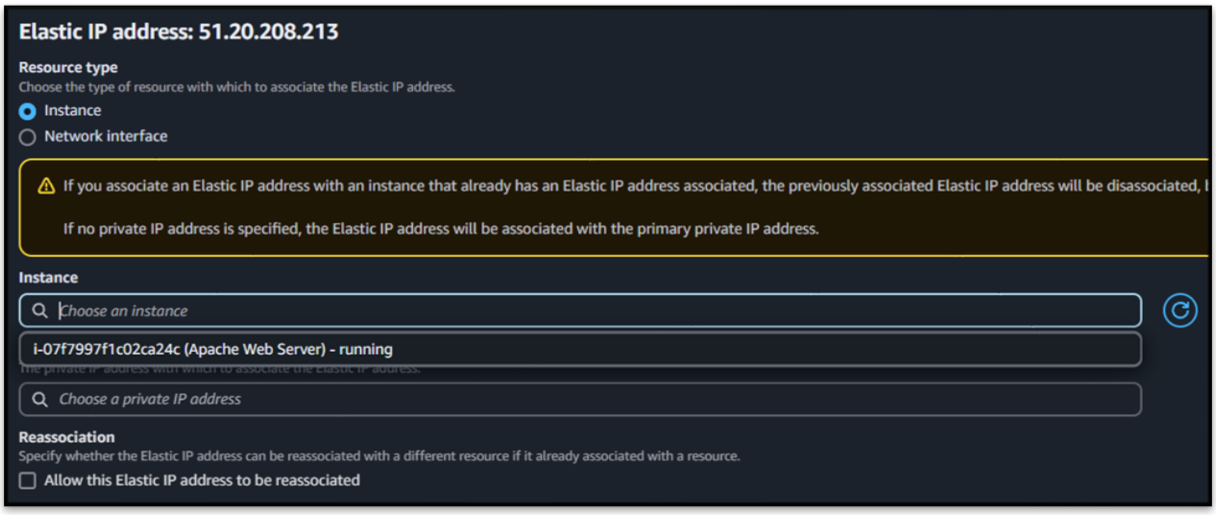
   You Should Be Able To See The Status Of Them By Typing “`sudo ufw status`”.

**Now You Have A Web Server Running!.**

## How To Connect Domain To Your Web Server?

1. Buy A Domain. I Used https://www.godaddy.com/.
2. Now Go To AWS EC2 Dashboard & Find “Elastic IPs”.
3. Click On “Allocate Elastic IP Address” And Then Click On “Allocate”
4. Once That’s Done, You Should See A Button Saying “Associate This Elastic IP”

5. Now You Should Be Able To Choose Your Instance


**Elastic IP address: 51.20.208.213**

**Resource type**
Choose the type of resource with which to associate the Elastic IP address.
● Instance
○ Network interface

⚠ If you associate an Elastic IP address with an instance that already has an Elastic IP address associated, the previously associated Elastic IP address will be disassociated, I

If no private IP address is specified, the Elastic IP address will be associated with the primary private IP address.

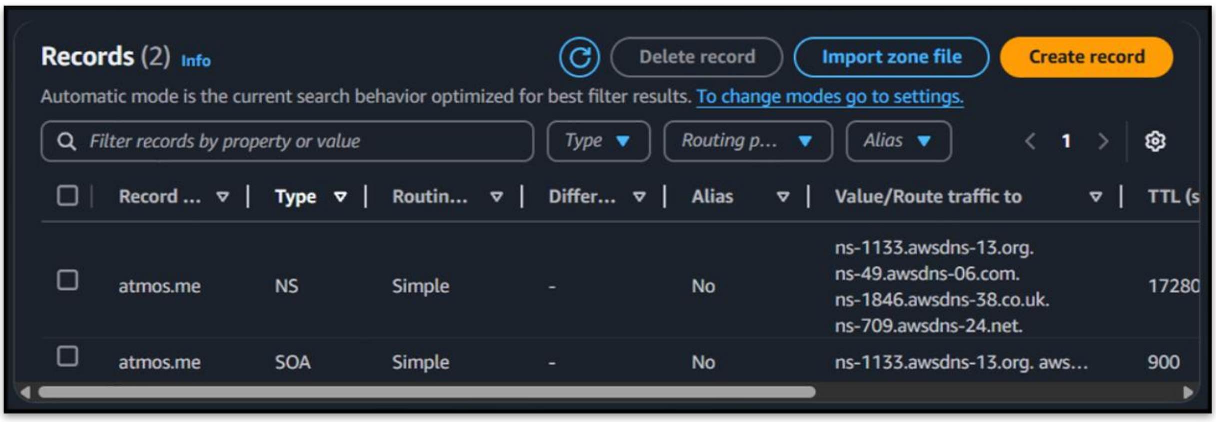
**Instance**
Choose an instance
i-07f7997f1c02ca24c (Apache Web Server) - running

Choose a private IP address

**Reassociation**
Specify whether the Elastic IP address can be reassociated with a different resource if it already associated with a resource.
☐ Allow this Elastic IP address to be reassociated

Make Sure To Copy Your New Elastic IP

6. Now Go To AWS & Search Up “Route 53”.
7. Once Chosen, Find “Hosted Zones”.
8. Create A Hosted Zone With Your Domain Name.
9. Once That’s Done, You Should See Something Like This.

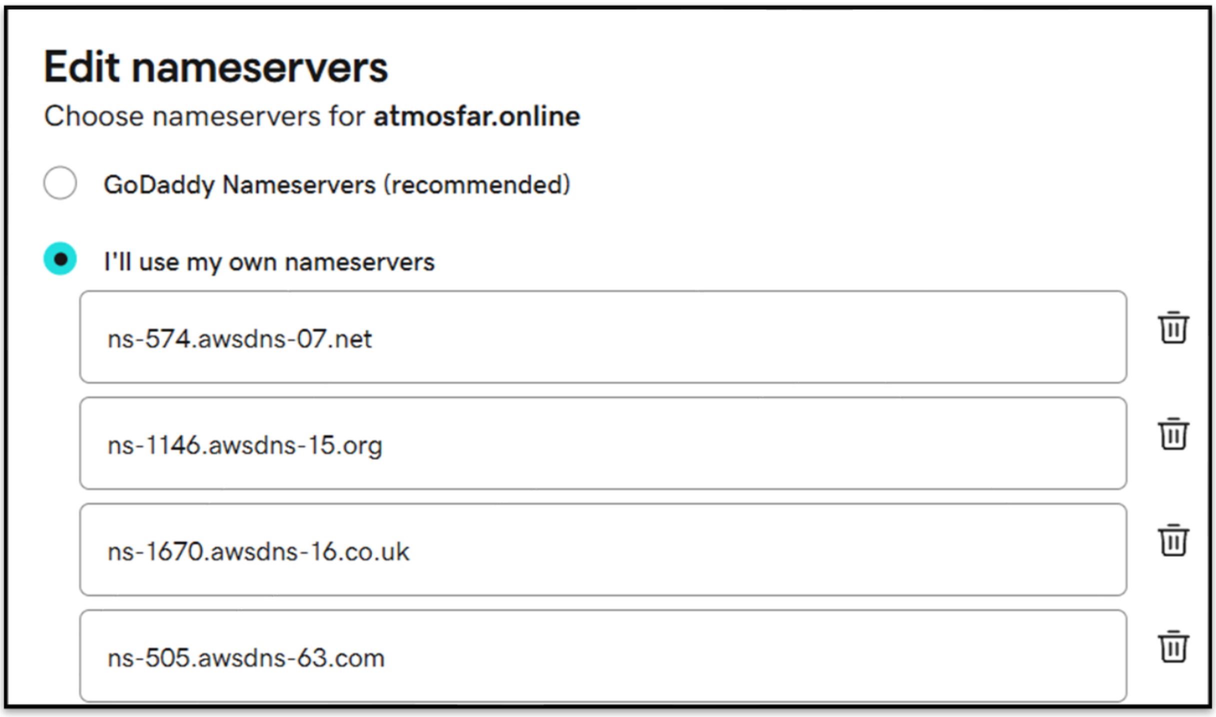

**Records (2)** Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

| Record ... | Type | Routin... | Differ... | Alias | Value/Route traffic to | TTL (s |
|---|---|---|---|---|---|---|
| atmos.me | NS | Simple | - | No | ns-1133.awsdns-13.org. ns-49.awsdns-06.com. ns-1846.awsdns-38.co.uk. ns-709.awsdns-24.net. | 17280 |
| atmos.me | SOA | Simple | - | No | ns-1133.awsdns-13.org. aws... | 900 |

10. Copy The Contents/Values Of NS Record.
11. Go Back To GoDaddy.
12. Find “DNS” Then “Nameservers”
13. Click On “Change Nameservers”

14. Choose "I'll Use My Own Nameservers" Then Paste In The Values Of The NS Record You Copied. Make Sure To Save Them.



**Edit nameservers**

Choose nameservers for **atmosfar.online**

GoDaddy Nameservers (recommended) [✓ I'll use my own nameservers]

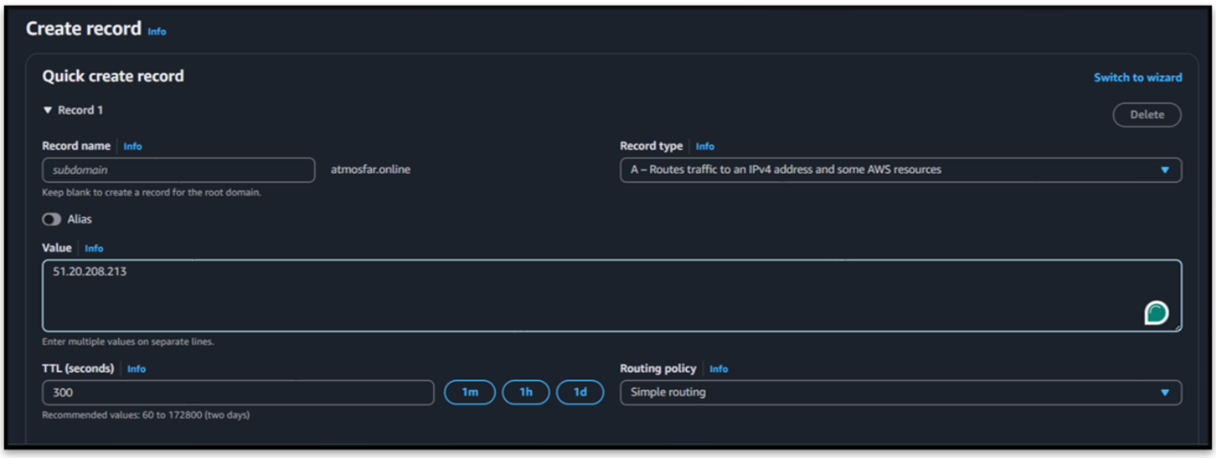
ns-574.awsdns-07.net

ns-1146.awsdns-15.org

ns-1670.awsdns-16.co.uk

ns-505.awsdns-63.com

15. Now Go Back To AWS Route 53 Hosted Zone And Create An "A" Type Record With The Value Being Your Elastic IP



**Create record** Info

**Quick create record** Switch to wizard

Record 1 Delete

Record name: subdomain atmosfar.online
Keep blank to create a record for the root domain.

Record type: A – Routes traffic to an IPv4 address and some AWS resources

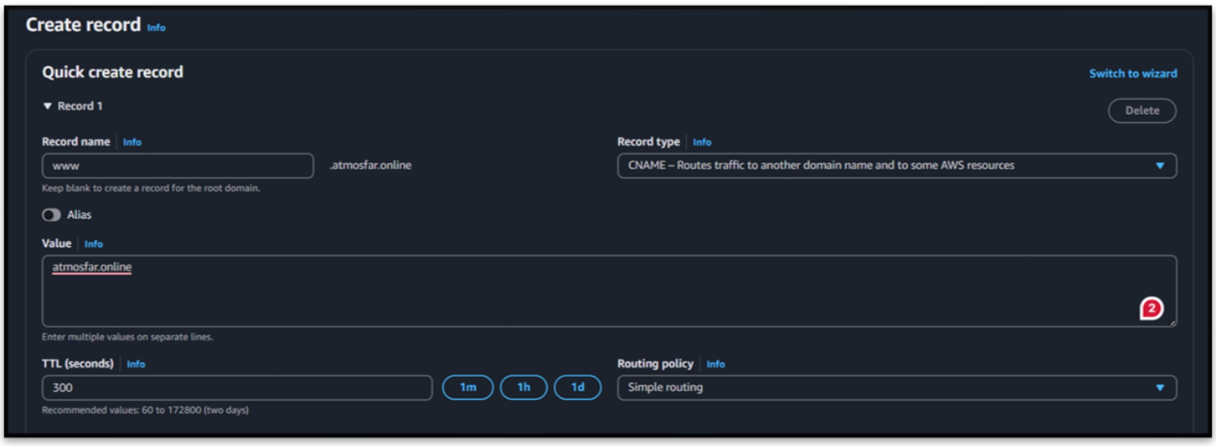
Alias

Value: 51.20.208.213
Enter multiple values on separate lines.

TTL (seconds): 300 1m 1h 1d
Recommended values: 60 to 172800 (two days)

Routing policy: Simple routing

16. Now Create An Another Record, This Time It Being A "CNAME" Type Record. Make The Record Name As "www" And Value As Your Domain Name.



Create record Info

Quick create record

Record 1

Record name: www .atmosfar.online

Keep blank to create a record for the root domain.

Record type: CNAME – Routes traffic to another domain name and to some AWS resources

Alias

Value: atmosfar.online

Enter multiple values on separate lines.

TTL (seconds): 300

Recommended values: 60 to 172800 (two days)

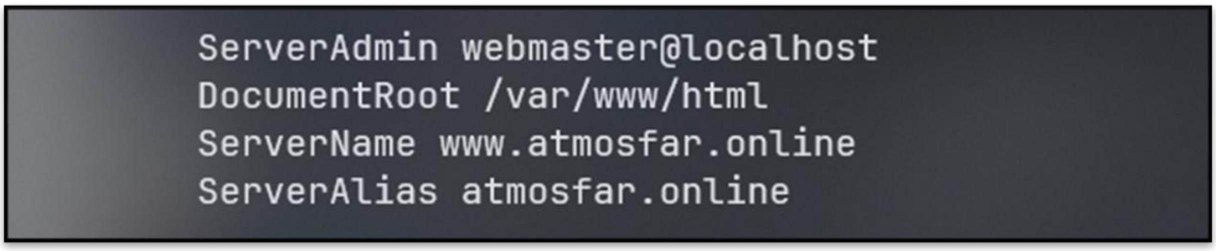
Routing policy: Simple routing

17. Now Go Back To Your Server And Type In This Command To Edit Apache2 Configuration.

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

18. Once You're In, Type In This Code

```
ServerName www.your_domain
ServerAlias your_domain
```



```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
ServerName www.atmosfar.online
ServerAlias atmosfar.online
```

19. Restart Apache To Implement Changes.

```
sudo systemctl restart apache2
```

20. Now To Implement SSL Certificate To Your Website, Type In These Commands

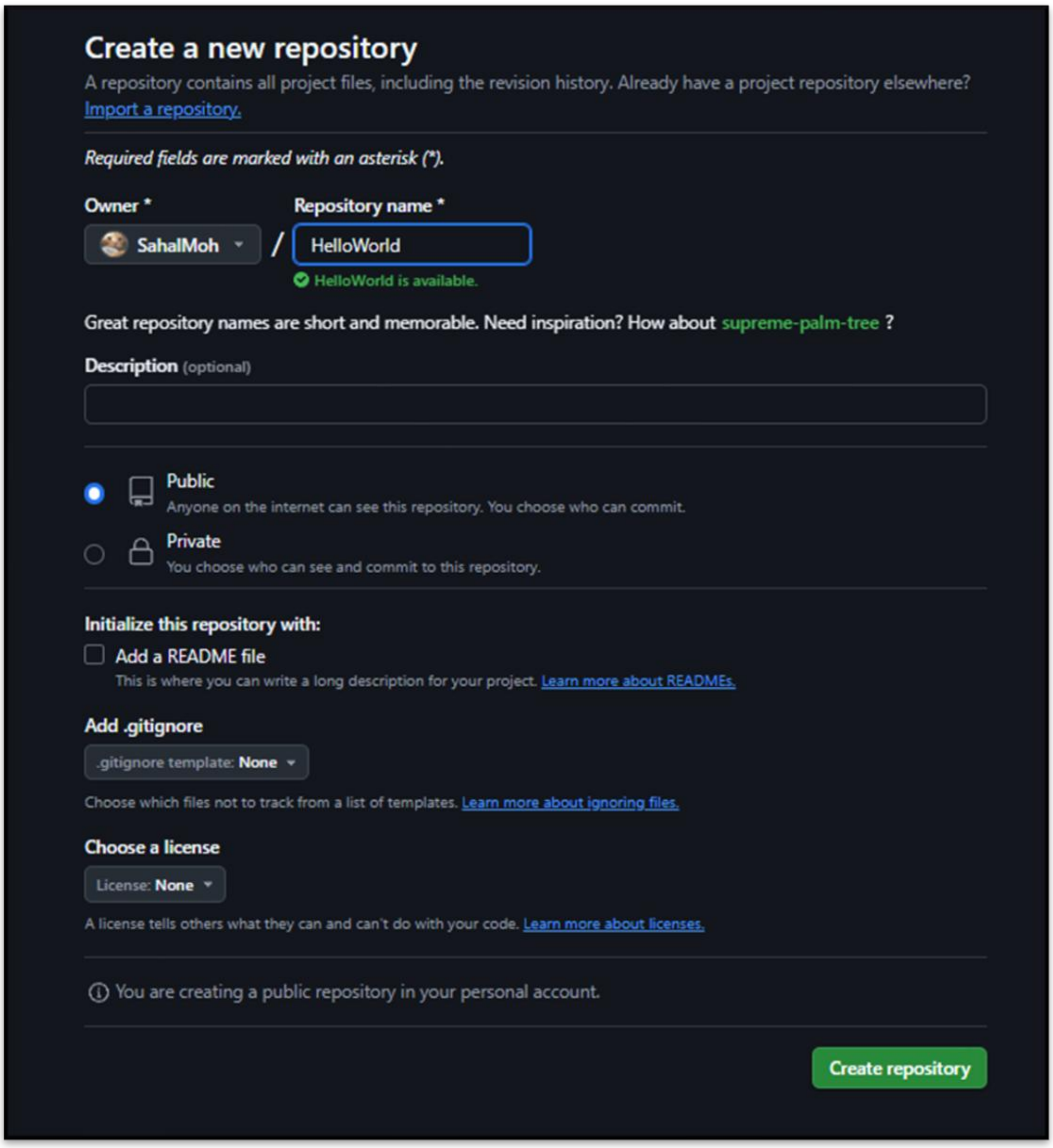
```
sudo snap install --classic certbot
sudo ln -s /snap/bin/certbot /usr/bin/certbot
sudo certbot –apache
```

Just Click "Y"

**Your Domain Is Now Connected To Your Website.**

## How To Configure Your Website?

1. Create A GitHub Account.
2. Create A GitHub Repository.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

*Required fields are marked with an asterisk (*).*

Owner *: SahalMoh / Repository name *: HelloWorld

HelloWorld is available.

Great repository names are short and memorable. Need inspiration? How about supreme-palm-tree ?

Description (optional)

● Public — Anyone on the internet can see this repository. You choose who can commit.

○ Private — You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file — This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

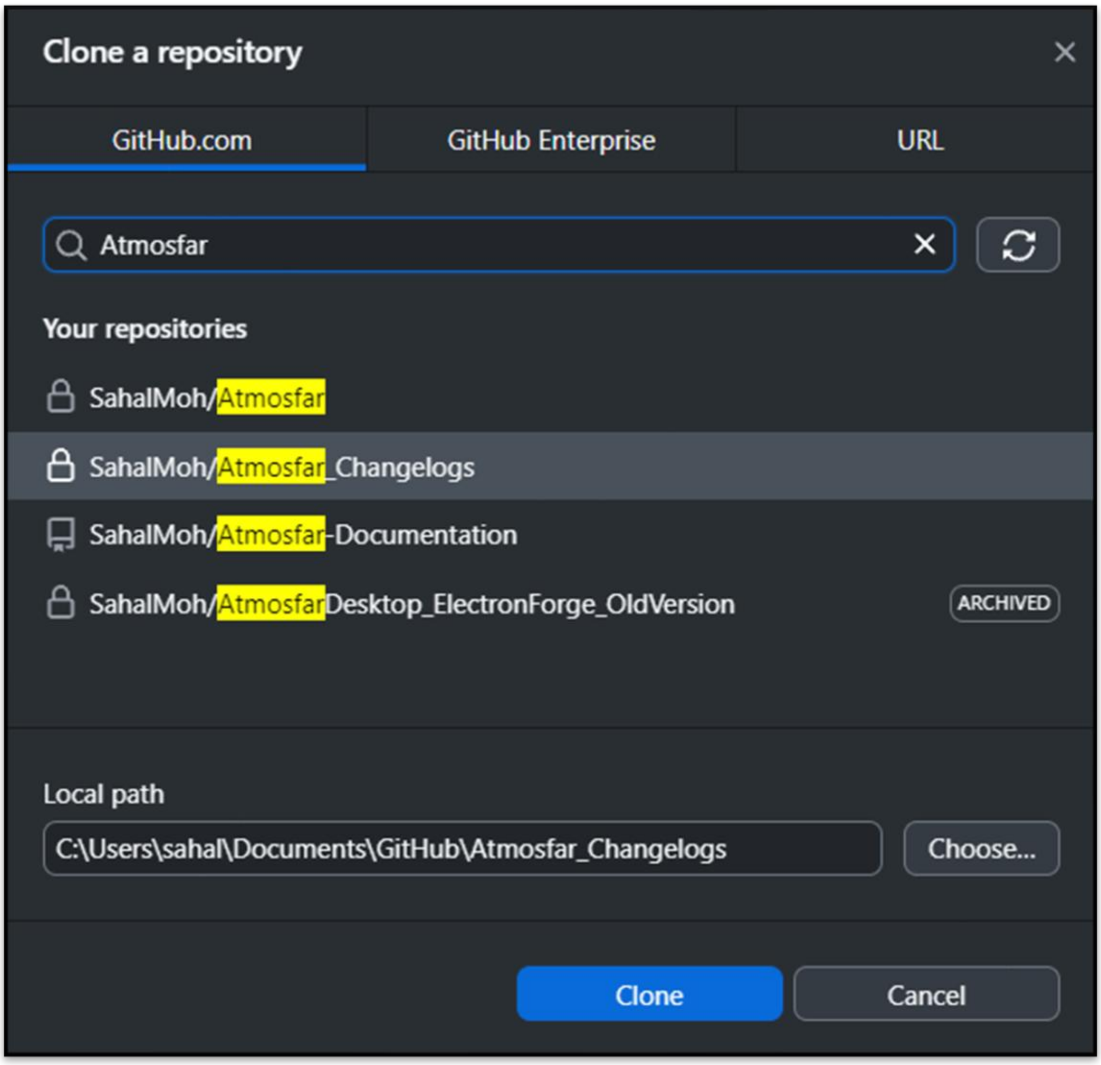
License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

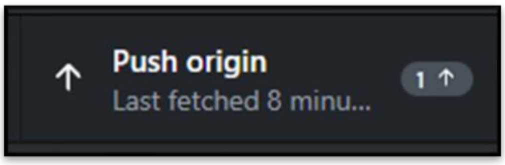
You are creating a public repository in your personal account.

3. Download GitHub Desktop.
4. Log Into GitHub Desktop Using GitHub Account.

5. Clone Your GitHub Repository.


Clone a repository
GitHub.com
GitHub Enterprise
URL
Atmosfar
Your repositories
SahalMoh/Atmosfar
SahalMoh/Atmosfar_Changelogs
SahalMoh/Atmosfar-Documentation
SahalMoh/AtmosfarDesktop_ElectronForge_OldVersion
ARCHIVED
Local path
C:\Users\sahal\Documents\GitHub\Atmosfar_Changelogs
Choose...
Clone
Cancel

6. Move Your Website Files Into The GitHub Repository Folder.
7. Push The Files Into Your GitHub Repository



8. Go To Your Web Server
9. Go Into /var/www/ Using `cd /var/www`
10. Clone Your GitHub Repository With Website Files By Typing In This Command

```
sudo git clone
https://github.com/<UserName>/<RepoName>.git
```

11. Once That Has Been Done, Move Your Files To /var/www/html By Doing This Command

    ```
    sudo mv -f /source/path/* /destination/path
    ```

12. Restart Apache By Running This Command

    ```
    sudo systemctl reload apache2
    ```

**Now You Should See Your Website!**

## References

1. Firewall Setup On Ubuntu: https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu
2. Apache Server Setup: https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04
3. SSL Certificate: https://certbot.eff.org/instructions?ws=apache&os=snap
4. Domain Setup On AWS: https://youtu.be/1hpHn1uOEeI?si=IXIccA6Tb9tuUSZu