

How to host Geoserver on an AWS EC2 Instance

Tif file to cloud optimized GeoTif

Open up **OSGeo4W Shell** from your QGIS folder

<https://www.cogeo.org/developers-guide.html>

Use Shell commands to access the folder your input tif file is located.

Insert the following

```
gdal_translate input.tif output_cog.tif -of COG -co  
COMPRESS=LZW
```

input.tif -name of the input file

output_cog.tif -name of the output file that you want. This file will be cloud optimized .

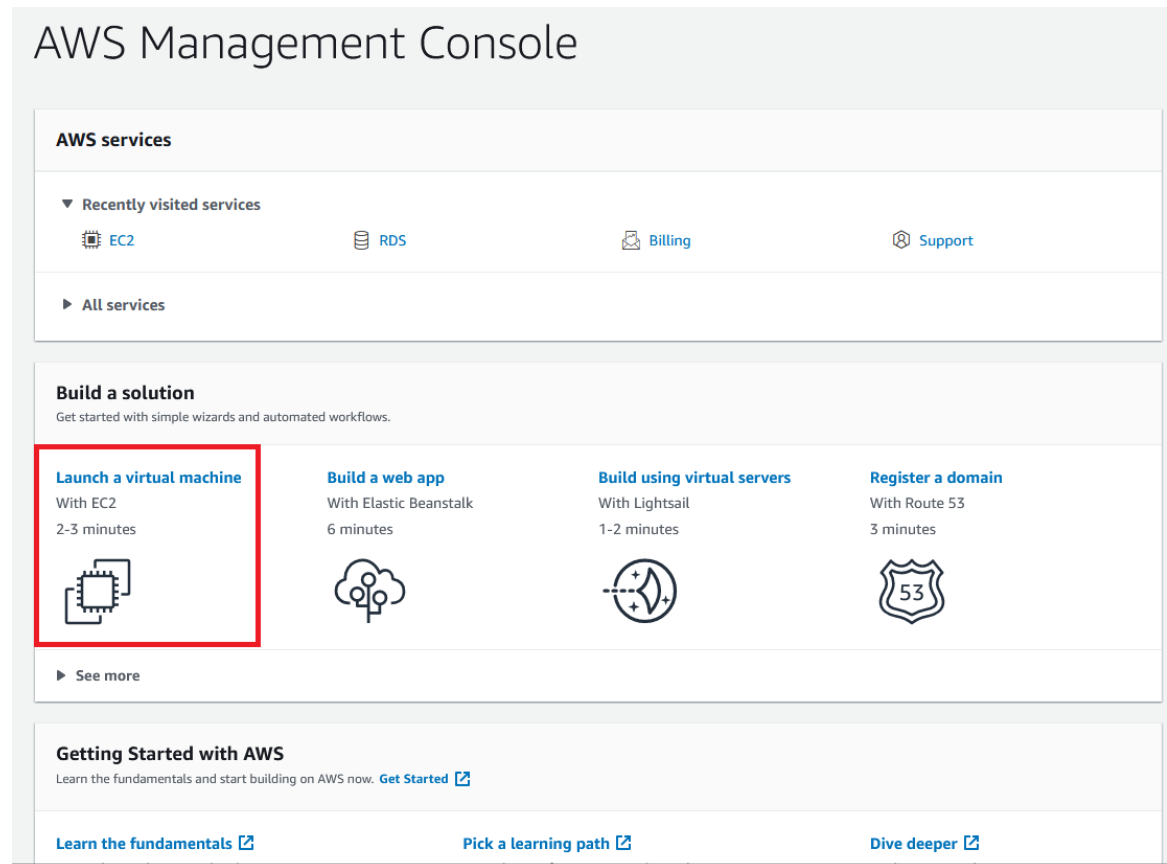
Step 1. Signup for an Amazon Web Services Account

This step is relatively easy. Go to <https://aws.amazon.com/> and sign up for an AWS account. I believe you can use an existing Amazon account as well. In my research, I read that some deals are available to students with a .edu email address or other promotional offers. However, everything in this tutorial will be free-tier eligible.

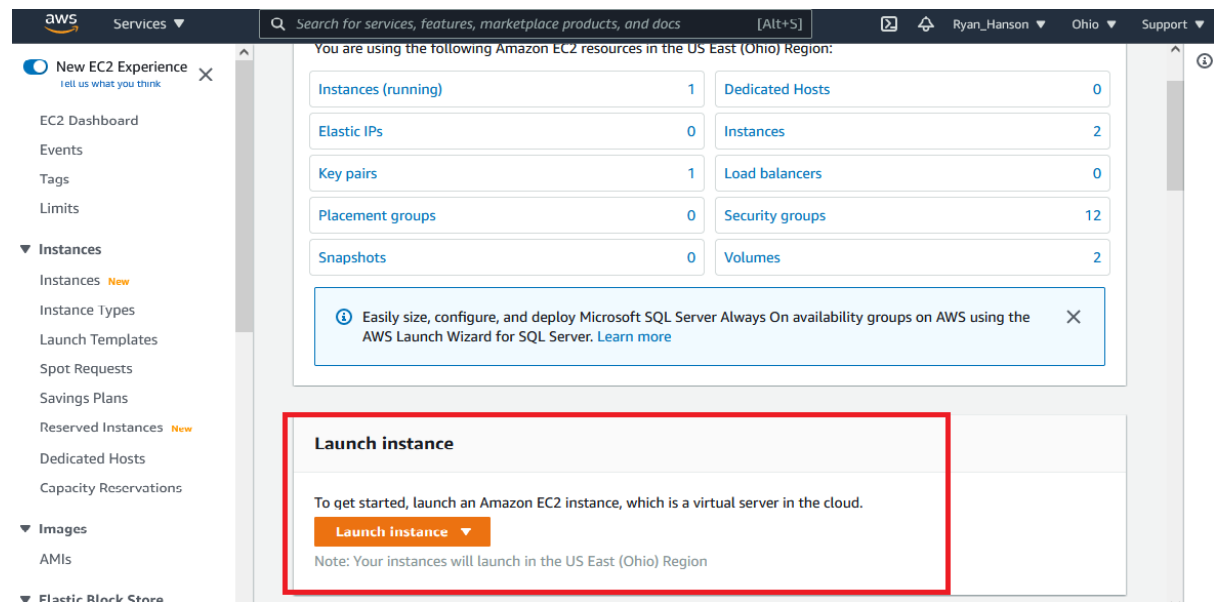
Step 2. Create a New EC2 Instance

EC2 stands for Elastic Cloud Computing, and an instance is a virtual server that lives in the AWS EC2 cloud. There are many advantages to having your infrastructure in the cloud instead of maintaining a physical server that I will not go into here. There are other cloud providers that you could use, such as Microsoft Azure, HostGator, or DigitalOcean. I decided to use AWS because it is an industry standard, and I wanted to have skills transferable to other areas.

From the AWS Management Console, select launch and new virtual machine with EC2.



You can also Launch instance from the EC2 Dashboard.




For this tutorial, I used the Ubuntu Server 20.04 LTS, which is free-tier eligible. You could go with another distribution of Linux, but I went with Ubuntu since it seems to be widely used.


Recents

Quick Start


Amazon Linux




macOS




Ubuntu




Windows




Red Hat



SUSE Li





Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-0f5ee92e2d63afc18 (64-bit (x86)) / ami-077053fb4029de92f (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-05-16

Architecture

64-bit (x86)

AMI ID

ami-0f5ee92e2d63afc18

Verified provider

Next you will select your instance type. Again, we are going with the t2.micro that is free tier eligible.

▼ Instance type

Info

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Linux base pricing: 0.0124 USD per Hour

On-Demand Windows base pricing: 0.017 USD per Hour

On-Demand RHEL base pricing: 0.0724 USD per Hour

On-Demand SUSE base pricing: 0.0124 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

Also select the option to create key pair password and save it as a .pem file with a unique file name.

▼ 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*

Select

 [Create new key pair](#)

Create key pair



Key pair name

Key pairs allow you to connect to your instance securely.

geo1

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

Key pair type



RSA

RSA encrypted private and public key pair



ED25519

ED25519 encrypted private and public key pair

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

This part is important. This server will host GeoServer, which will serve your GIS data to clients such as desktop GIS and websites. You are going to create a security rule that will allow GeoServer to communicate through port 8080. Select the Add Rule button. Choose Custom TCP. Enter port 8080. Choose Anywhere. Give the rule a name—select Review and Launch.

▼ Network settings [Info](#)

Edit

Network [Info](#)

vpc-015629d2275fed5fc

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-14' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

Anywhere
0.0.0.0/0

☐ Allow HTTPS traffic from the internet

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

☐ Allow HTTP traffic from the internet

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

Scroll down until you see Add Security Group Rule

Security group name required

launch-wizard-14

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!\$*

Description - required [Info](#)

launch-wizard-14 created 2023-08-22T06:57:16.822Z

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type [Info](#)

ssh ▼

Protocol [Info](#)

TCP

Port range [Info](#)

22

Source type [Info](#)

Anywhere ▼

Source [Info](#)

Q Add CIDR, prefix list or security group ID

0.0.0.0/0 ✕

Description - optional [Info](#)

e.g. SSH for admin desktop

⚠ 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. ✕

Add security group rule

Fill out the security rule according to the image below

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Remove

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

ssh ▼

TCP

22

Source type [Info](#)

Source [Info](#)

Description - optional [Info](#)

Anywhere ▼

Q Add CIDR, prefix list or securi

0.0.0.0/0 ✕

e.g. SSH for admin desktop

▼ Security group rule 2 (TCP, 8080, 0.0.0.0/0, Geoserver)

Remove

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

Custom TCP ▼

TCP

8080

Source type [Info](#)

Source [Info](#)

Description - optional [Info](#)

Anywhere ▼

Q Add CIDR, prefix list or securi

0.0.0.0/0 ✕

Geoserver

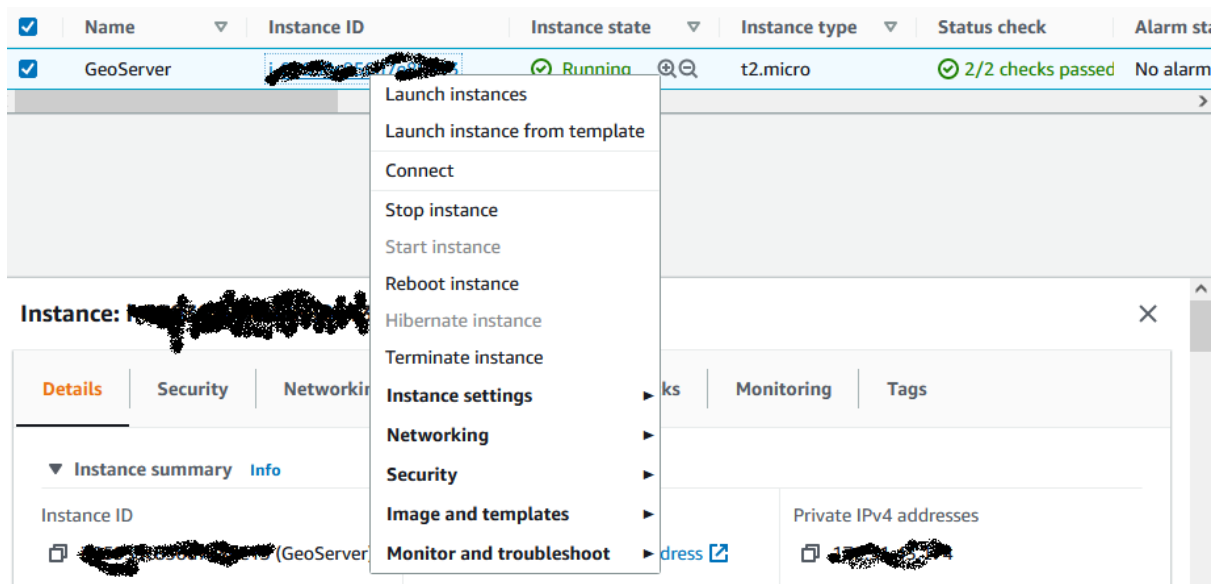
⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

✕

Add security group rule

Step 3. Log in to Your Ubuntu Linux Server and Install GeoServer

Go to your EC2 instances page, and you will see the instance you created. Right Click on the Instance ID link and select Connect.



There are several different ways to connect to your instance. I chose to connect through the EC2 Instance Connect. Select that option and click the orange Connect button on the bottom of the screen. A new browser window will launch with a command-line interface of your Linux server. This might be scary to those unfamiliar with the command line, but don't be frightened. It is easy, and this is an excellent opportunity to learn some Linux commands.

Once Linux is loaded, you will begin by getting updates and do an upgrade. These steps are typical before you start working with any new Linux server. The commands are below. Type 'Y' or 'y' when the option of Yes or No appears.

If any pop-ups appear after using a command then just press '**Esc**' key

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install openjdk-8-jre
```

```
sudo apt-get install tomcat9
```

```
sudo mkdir Downloads
```

```
cd /home/ubuntu/Downloads
```



```
sudo wget http://sourceforge.net/projects/geoserver/files/GeoServer/2.18.0/geoserver-2.18.0-war.zip
```

```
sudo apt-get install unzip
```

```
sudo unzip geoserver-2.18.0-war.zip
```

```
cd /home/
```

```
sudo mkdir raster
```

```
cd /home/raster
```

```
sudo chmod a+rwX /home/raster
```

```
cd /home/ubuntu/Downloads
```

```
sudo mv geoserver.war /var/lib/tomcat9/webapps/
```

```
cd /var/lib/tomcat9/webapps/
```

```
sudo service tomcat9 restart
```

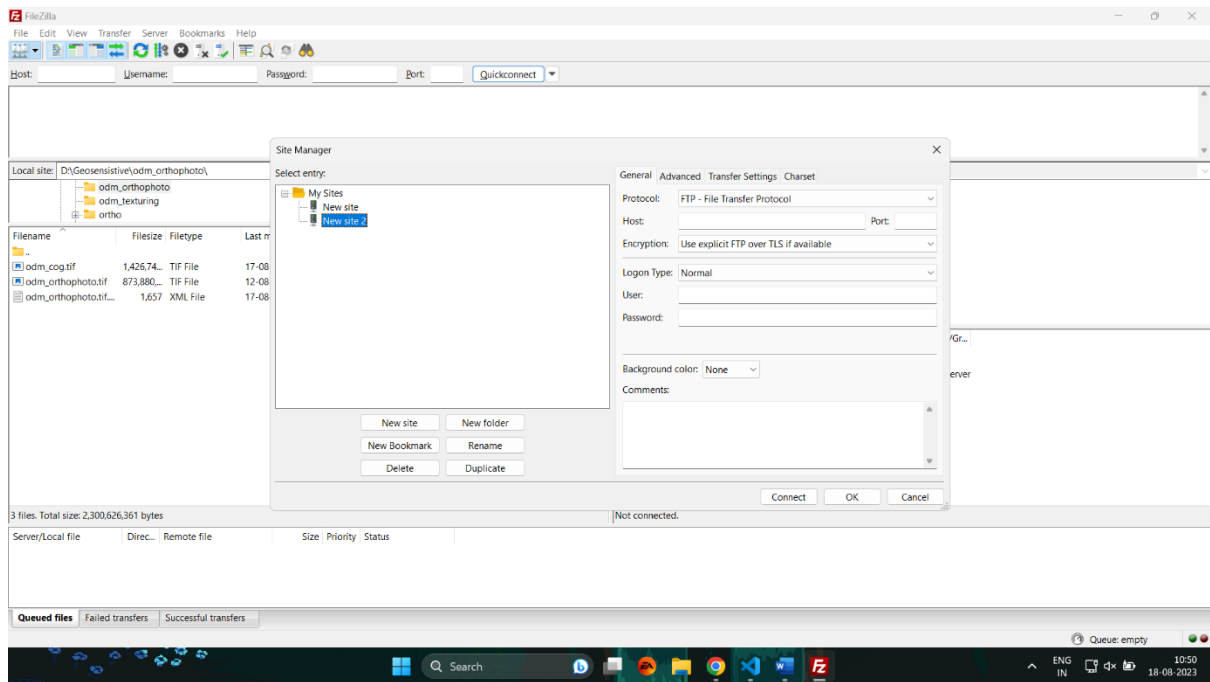
- /home/raster is given all the permissions(read and write) as it will be the folder that will have the transferred raster files .
- cd /var/lib/tomcat9/webapps/
sudo service tomcat9 restart-restarts the tomcat service
- geoserver war file(java) will be moved in the webapps folder of tomcat
- mkdir raster-can insert any name for the directory as you please

//Moving a raster file to an AWS Instance

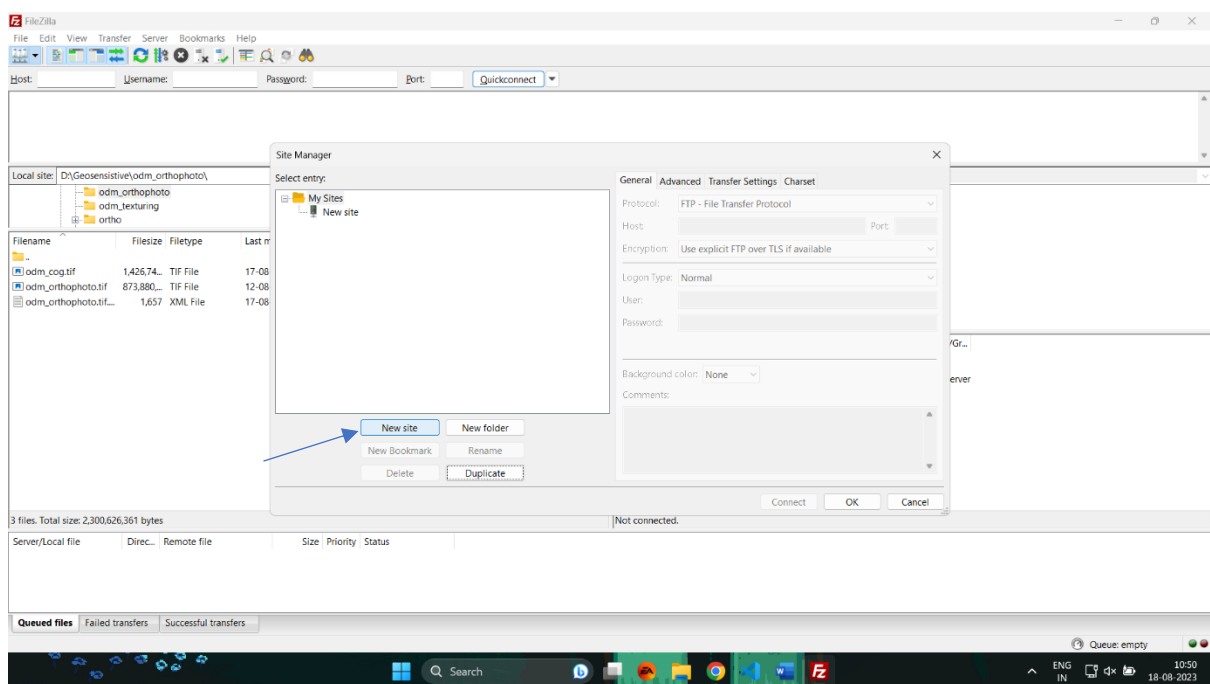
<https://filezilla-project.org/download.php?type=client>

Download FileZilla and install it.(the free version will do).

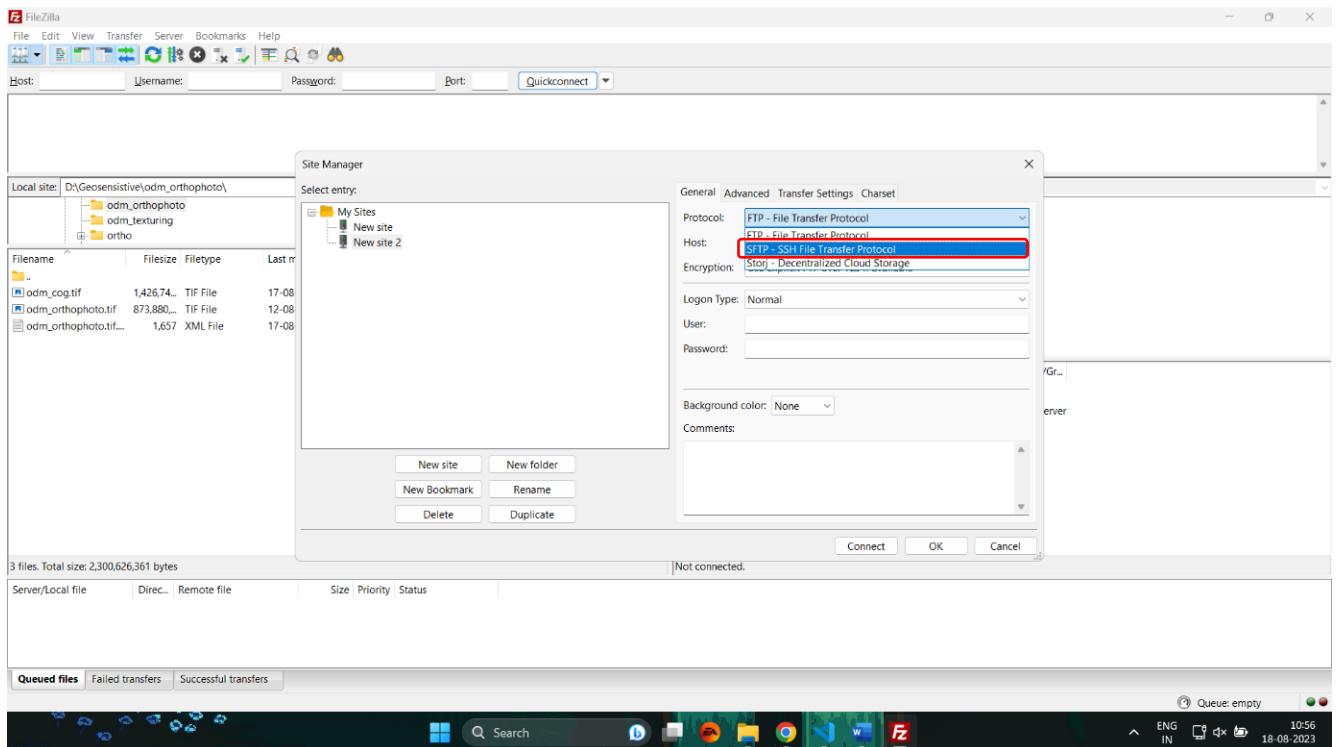
Click on the top-left option of the toolbar called Site Manager.



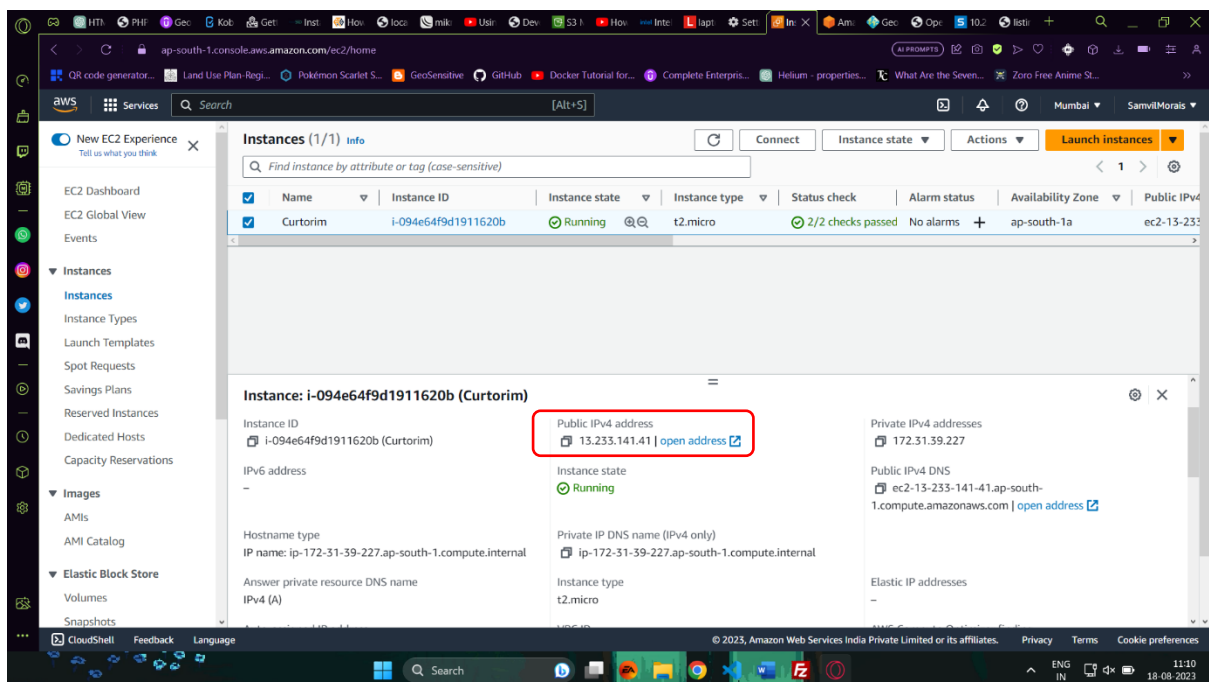
Click on **New Site**



In the General tab select the Protocol and select SFTP from the dropdown



For **Host**, the input is your AWS EC2 instance Public IPv4 address or Public IPv4 DNS

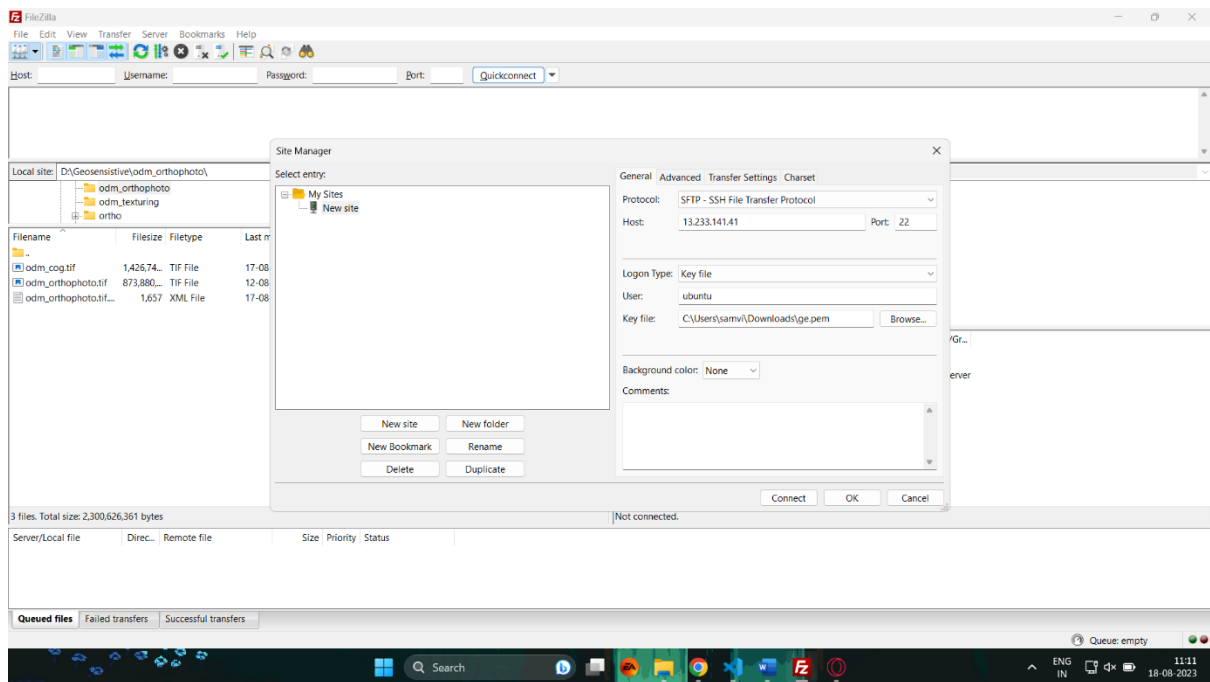
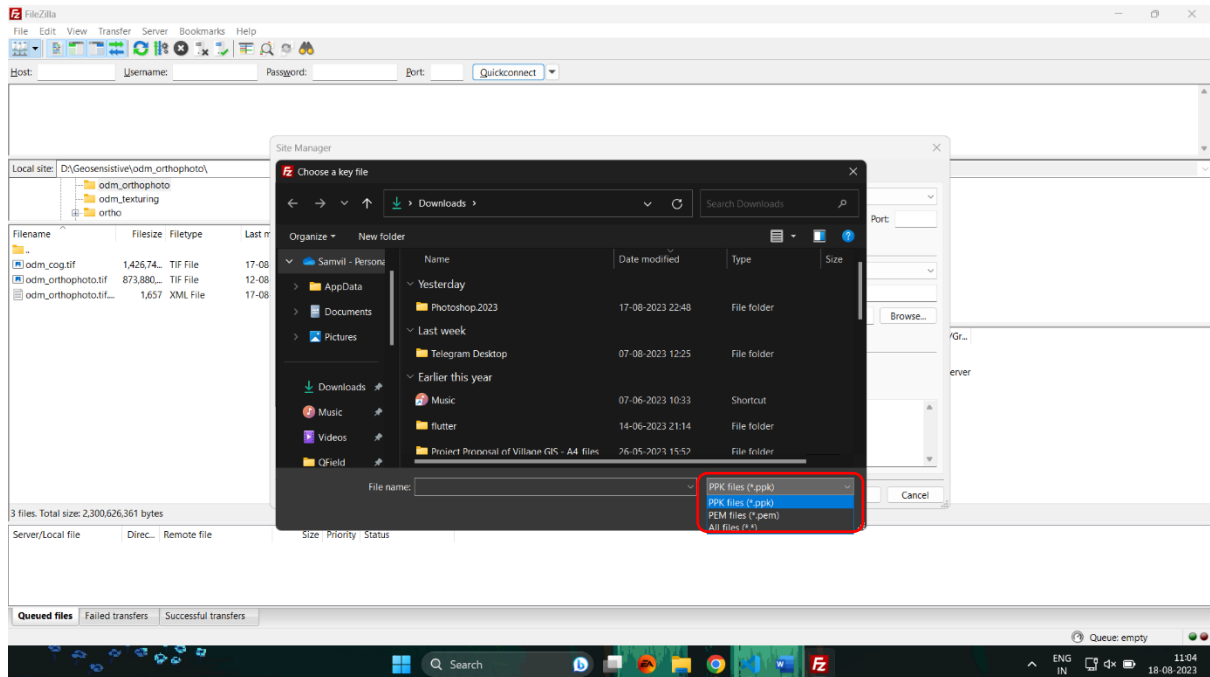


Port: 22

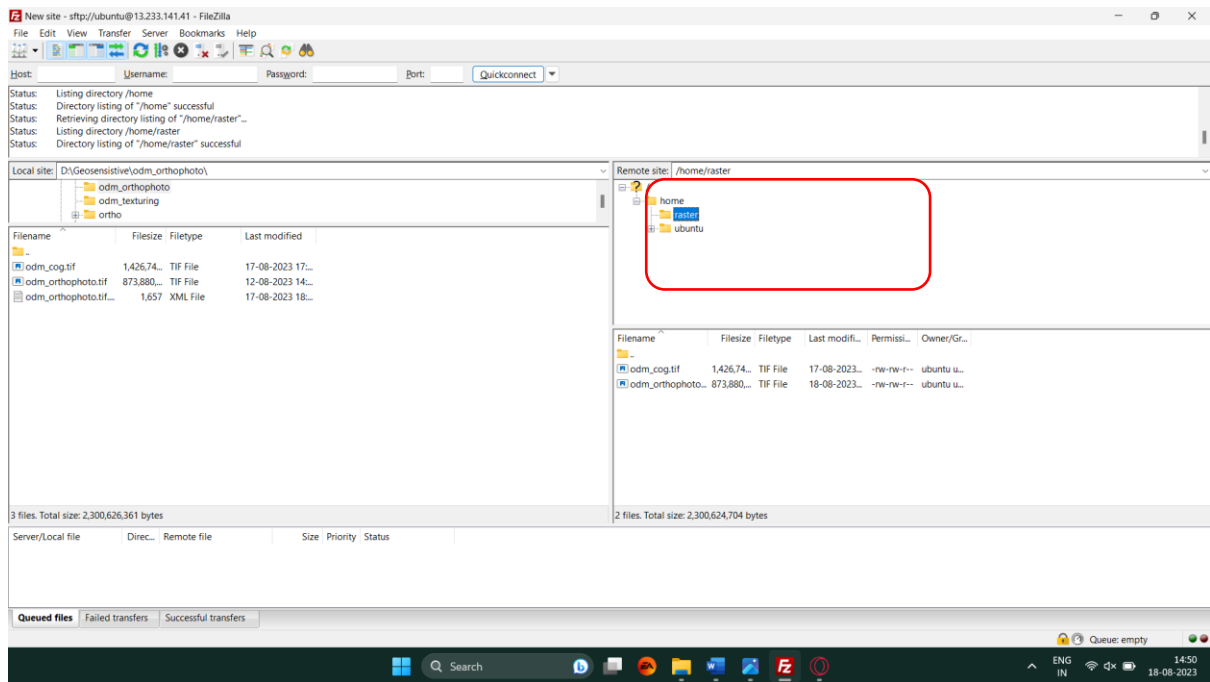
Logon type: Select '**Key file**' from the dropdown list

User: ubuntu

Key File : Click on Browse and select .pem files from file explorer, find and select then .pem file created while launching the instance.



Click Connect

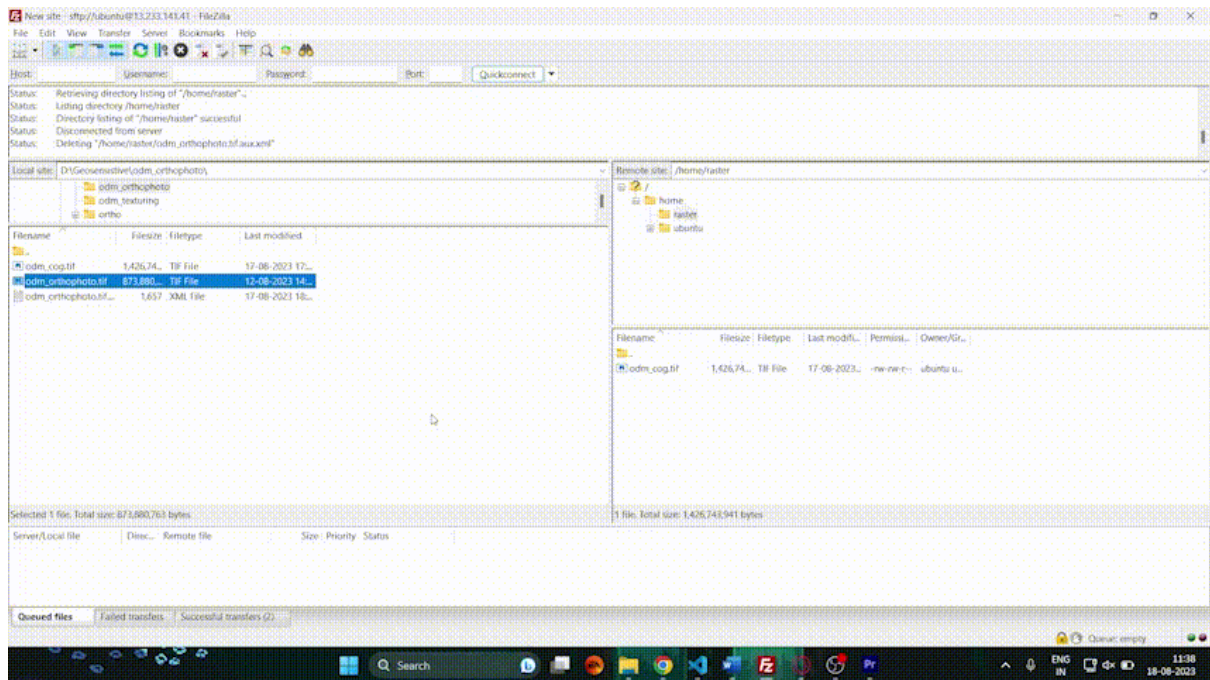


On the left side is the local directories

Browse to find the file you want to transfer(.tif)

On the right side with AWS directories, browse to home/raster directory as shown in the above image.

Just click on the file and drag it to the right side, which will be the AWS. With Directories location as(/home/raster).



The transfer process will start now.

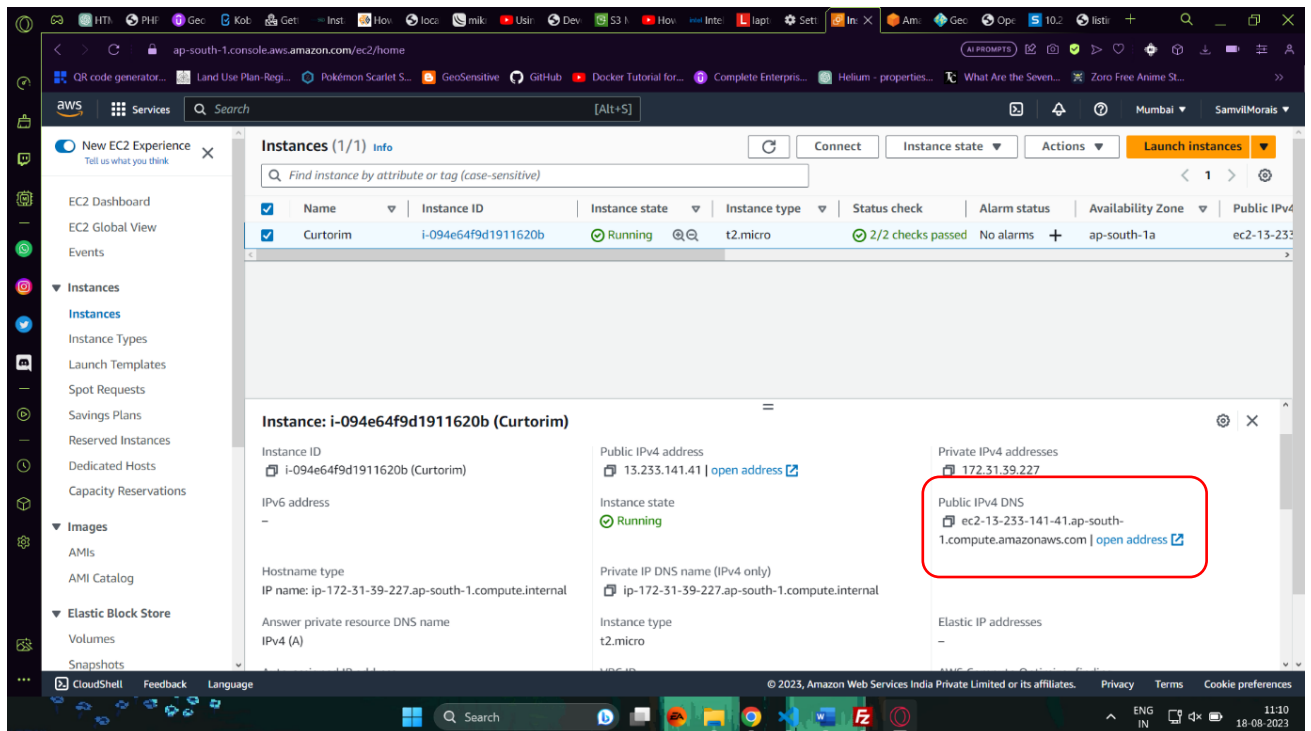
//AWS(restart tomcat)

```
cd /home/ubuntu/Downloads
```

```
sudo mv geoserver.war /var/lib/tomcat9/webapps/
```

```
cd /var/lib/tomcat9/webapps/
```

```
sudo service tomcat9 restart
```



Copy the Public IPv4 DNS of your instance;

Open a web browser and in the address bar type :

IPv4 DNS:8080/geoserver/web

In this example:

**ec2-13-233-141-41.ap-south-1.compute.amazonaws.com
/geoserver/web**

//Geoserver

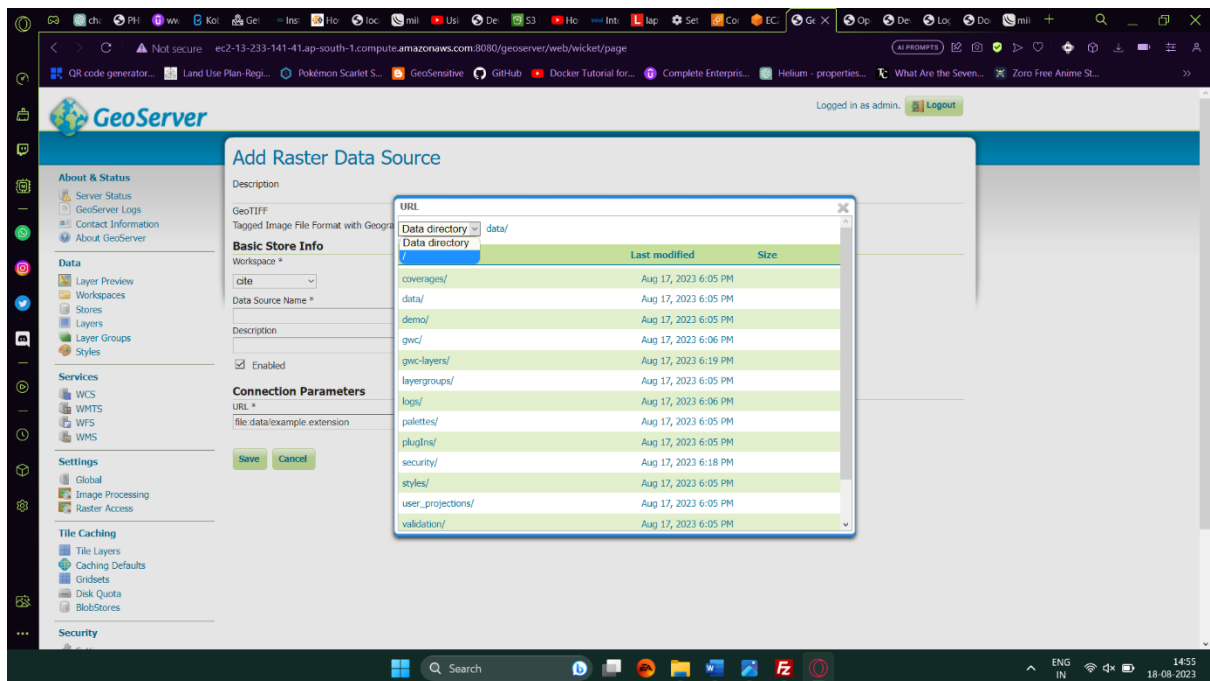
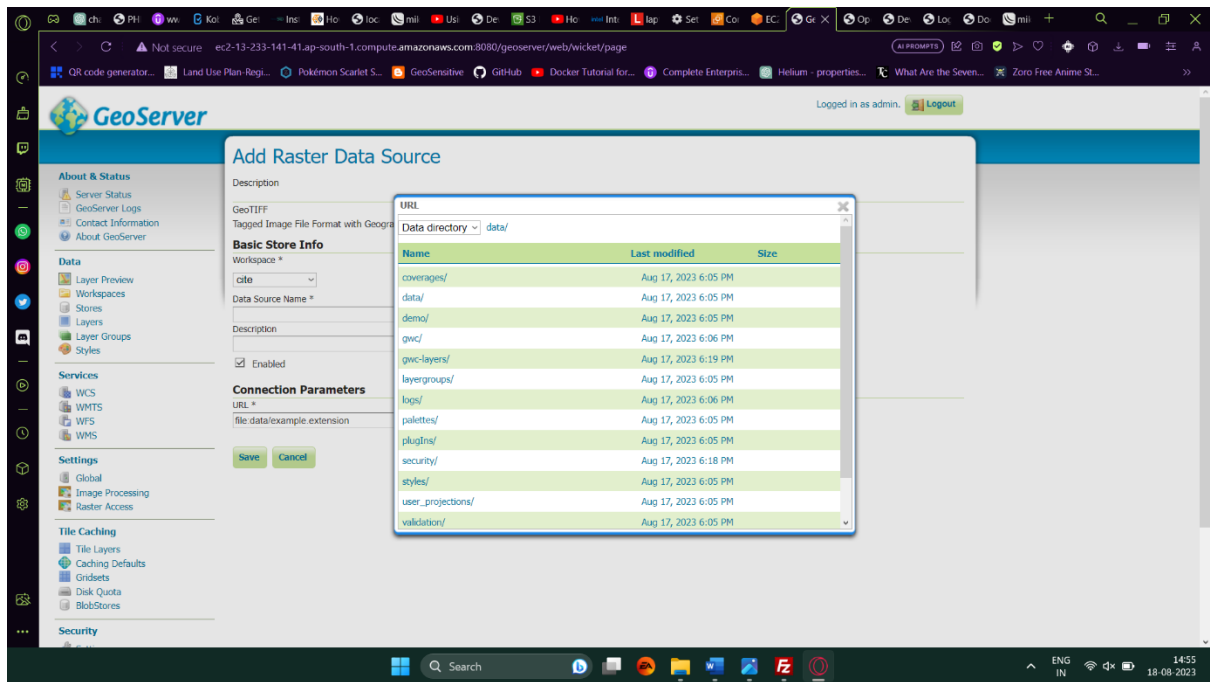
User: admin

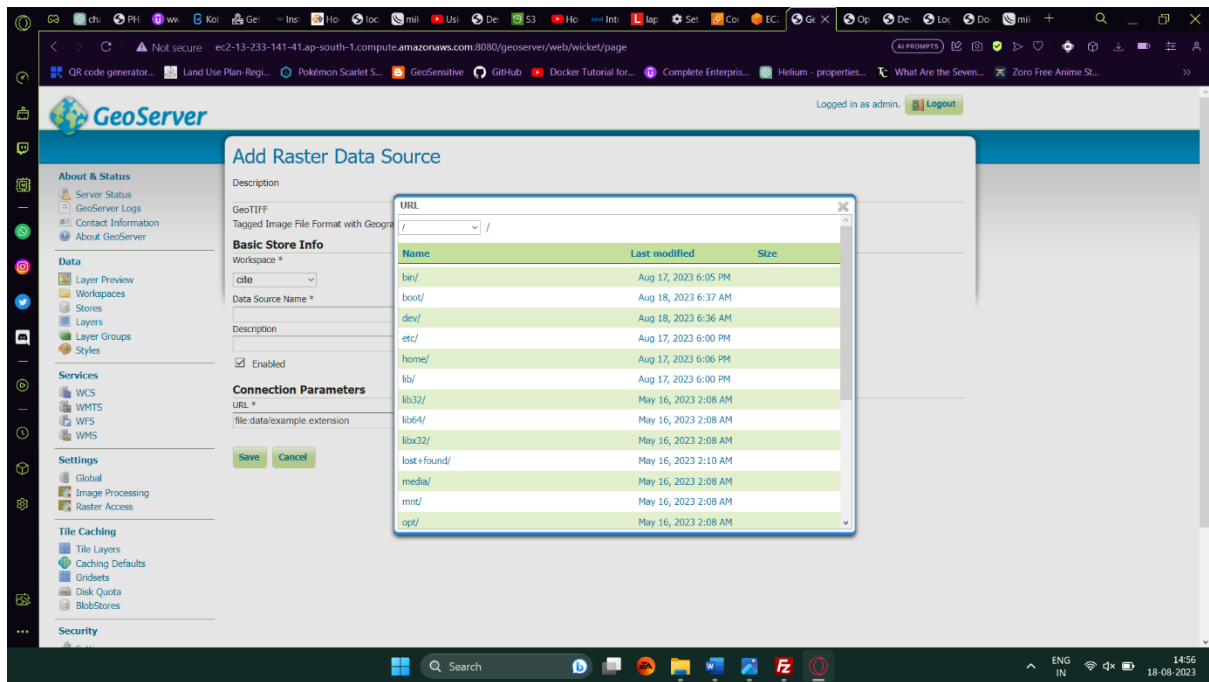
Password: geoserver(change it to make it more secure)

Create a Workspace

Add a store ->GeoTiff->Connection Parameters :URL->Browse:

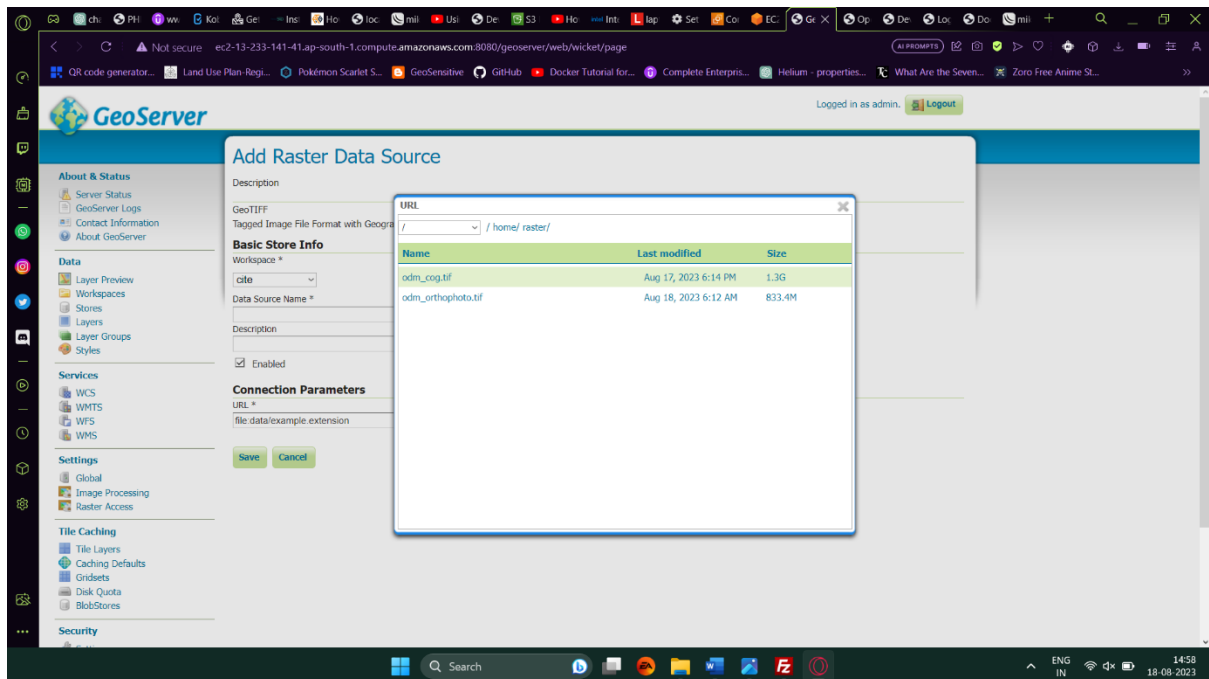
Select "/" from the dropdown





Now navigate to the directory the raster file is stored. In this case:

home/-→raster/→odm_cog.tif



Select the .tif file and save the store and publish the raster layer.