

S3 Upload Automation Script

# Step-1

Create an **EC2 instance** in AWS and log in to it.

Update your EC2 instance by running the following commands to install Git

# Step-2

Next, you need to assign **S3 access** to your EC2 instance.

There are two ways to do this:

1. **Create an IAM user**: You can create an IAM user, assign a policy to that user, and configure it in the EC2 instance.
2. **Assign a role**: You can assign a role to the EC2 instance, which will give the entire instance access to S3.

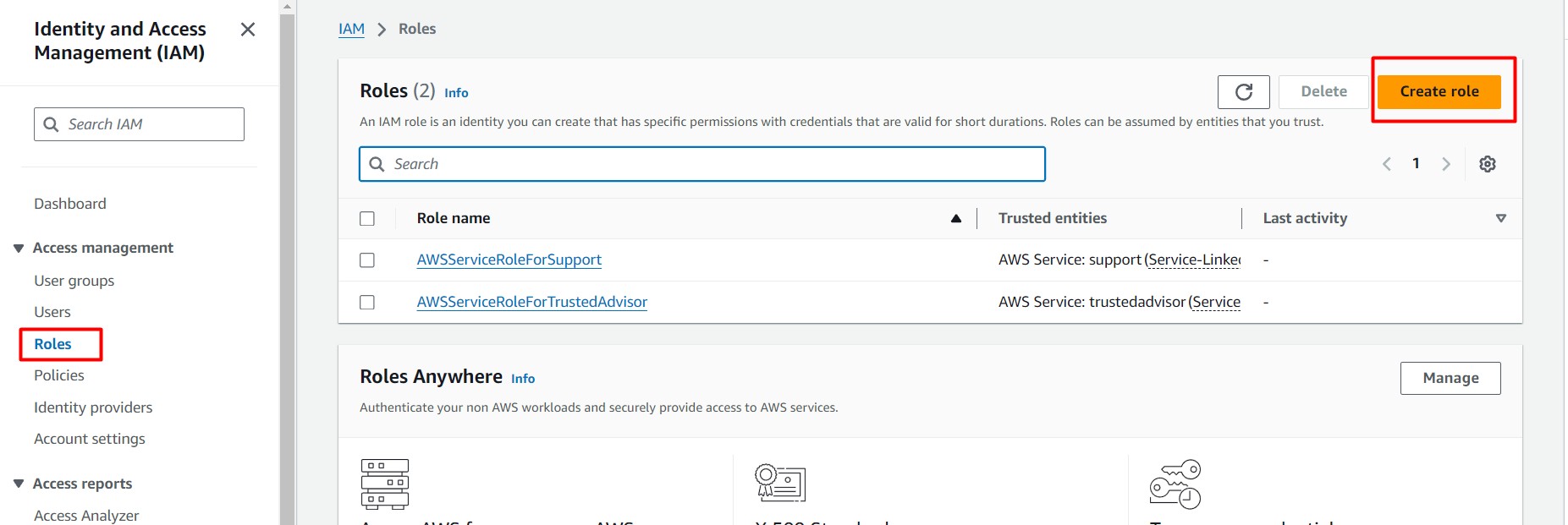
I will use the second method here.

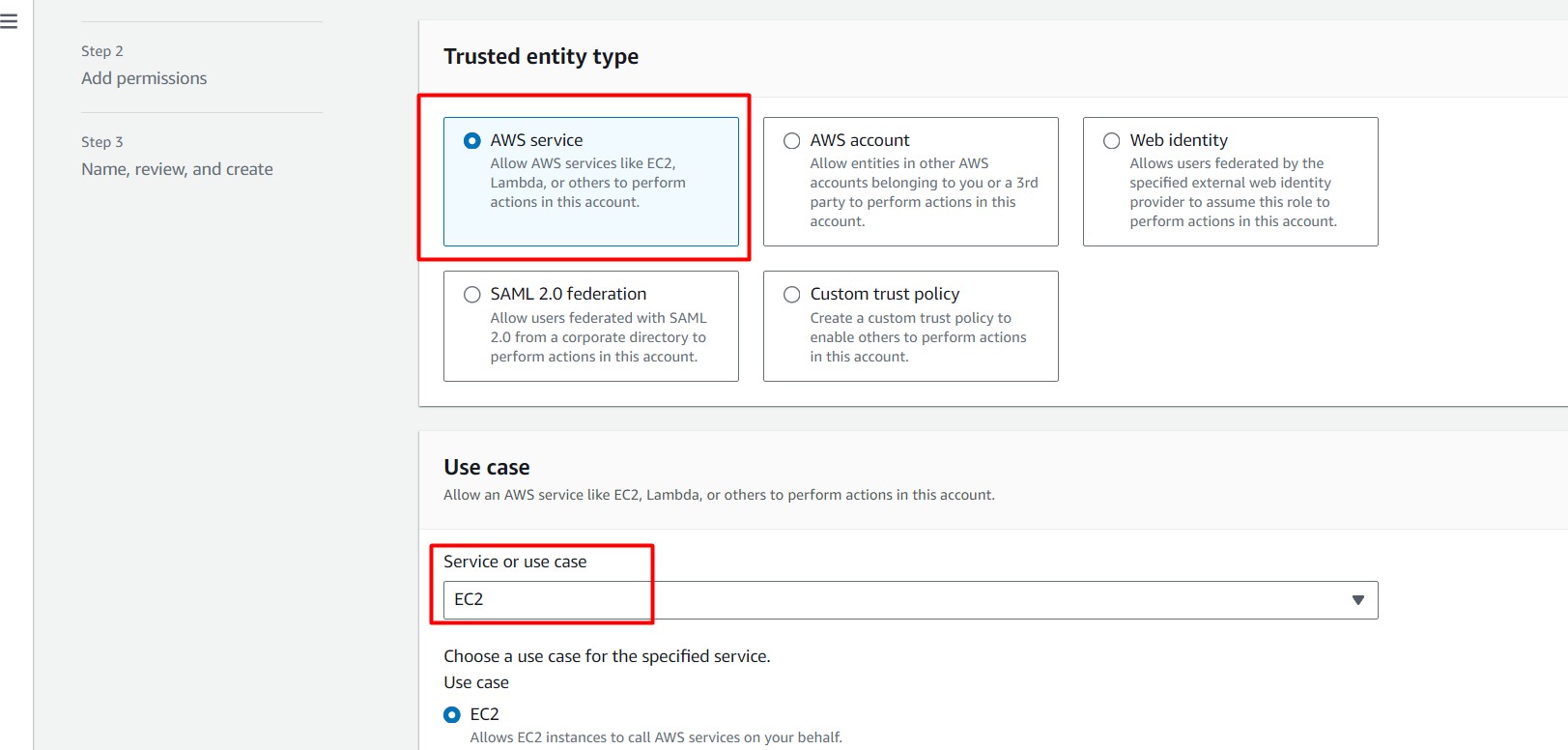
Go to the **AWS Management Console**, search for **IAM**, and click on **Roles**.

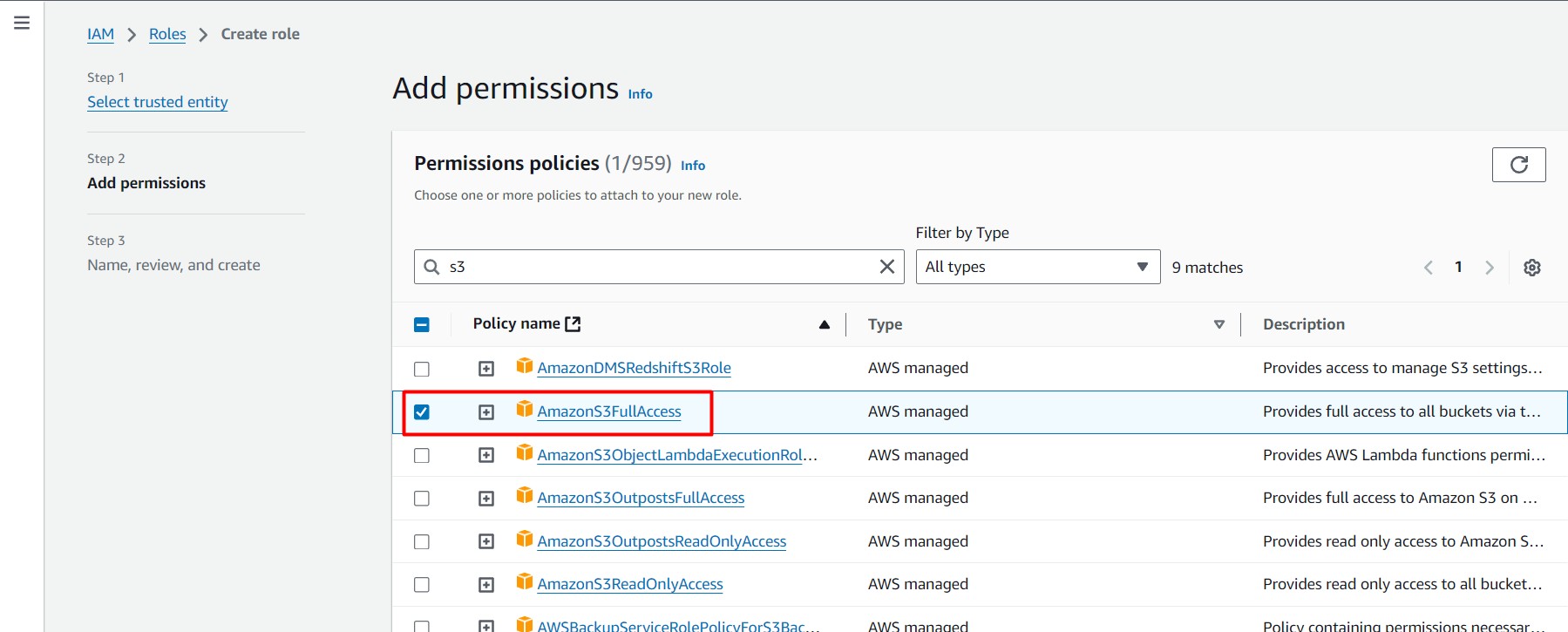
Click on **Create Role**.

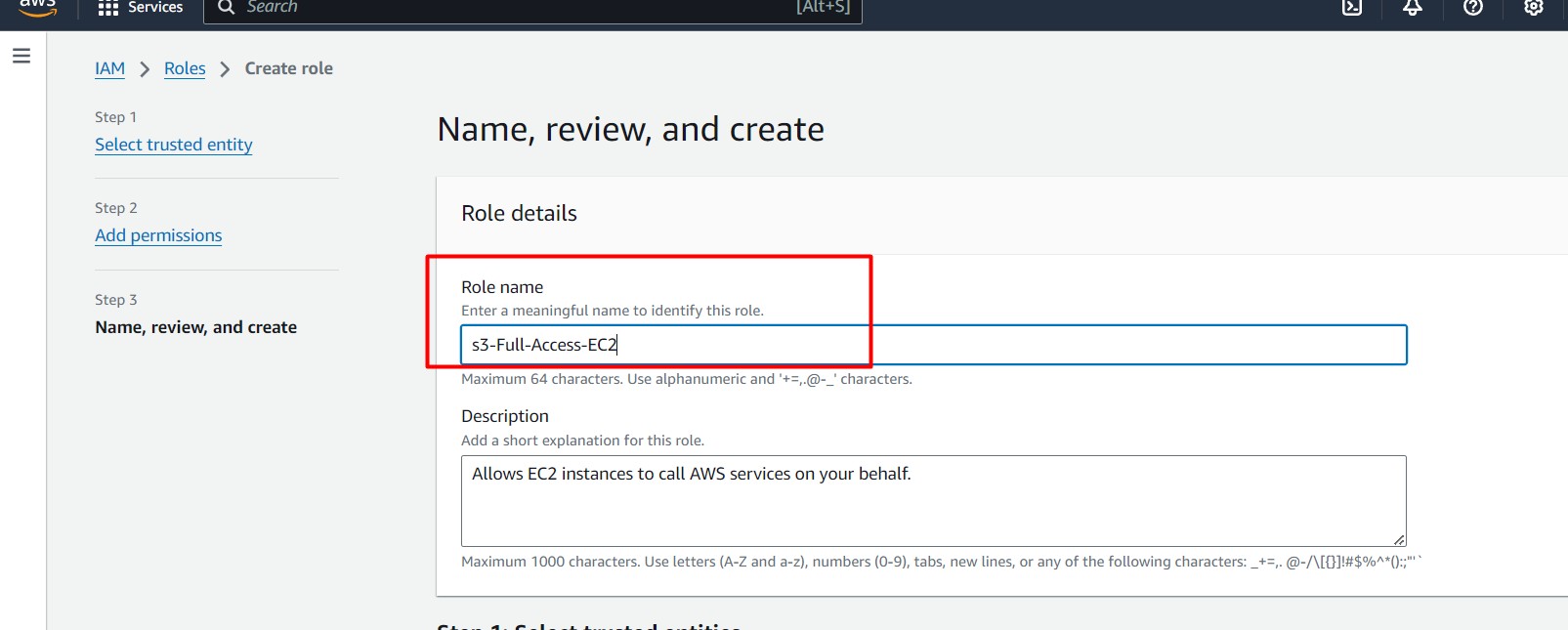
Select **AWS Service**, choose **EC2**, give your role a name, add permissions for **S3 Full Access**, and then

click **Create Role**.









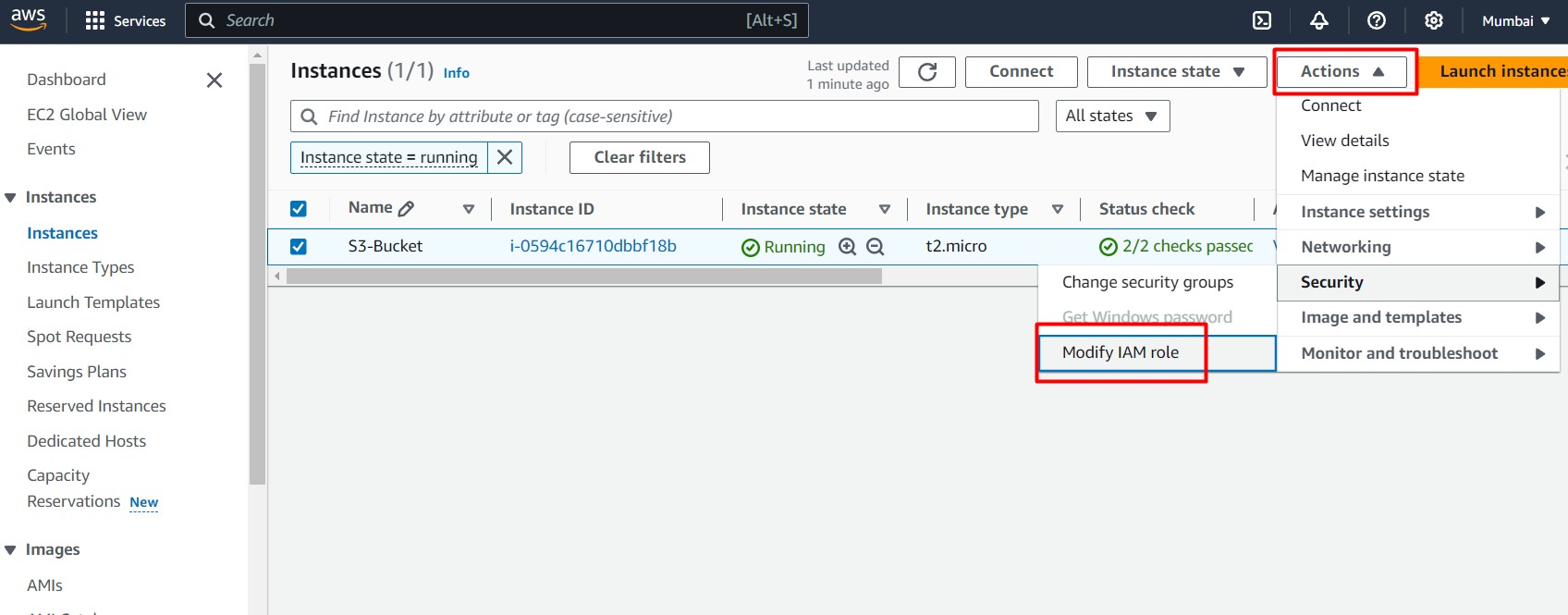
# Step-3

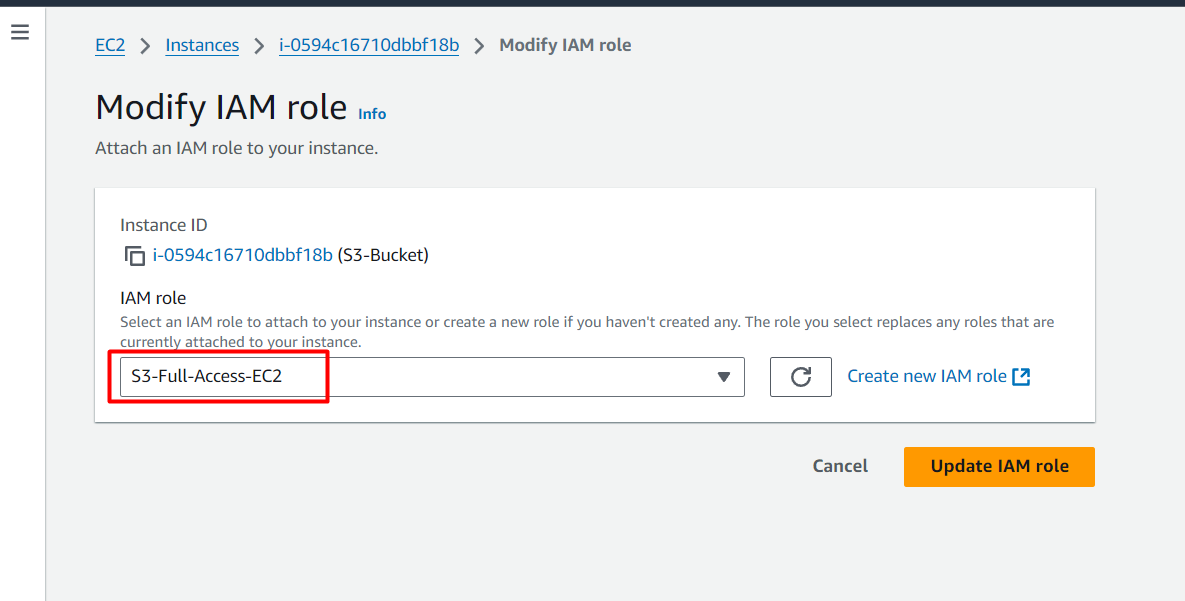
Next, assign the role to your EC2 instance.

Go to **EC2** in the console and select your EC2 instance.

Click on **Actions** > **Security** > **Modify IAM Role**.

Select the created role from the dropdown menu and click **Update IAM Role**.





# Step-4

Next, create an S3 bucket in AWS. Search for **S3** in the console, click **Create Bucket**, and give it a name. You can leave the rest of the settings as default.

Uncheck **Block all public access** and click on

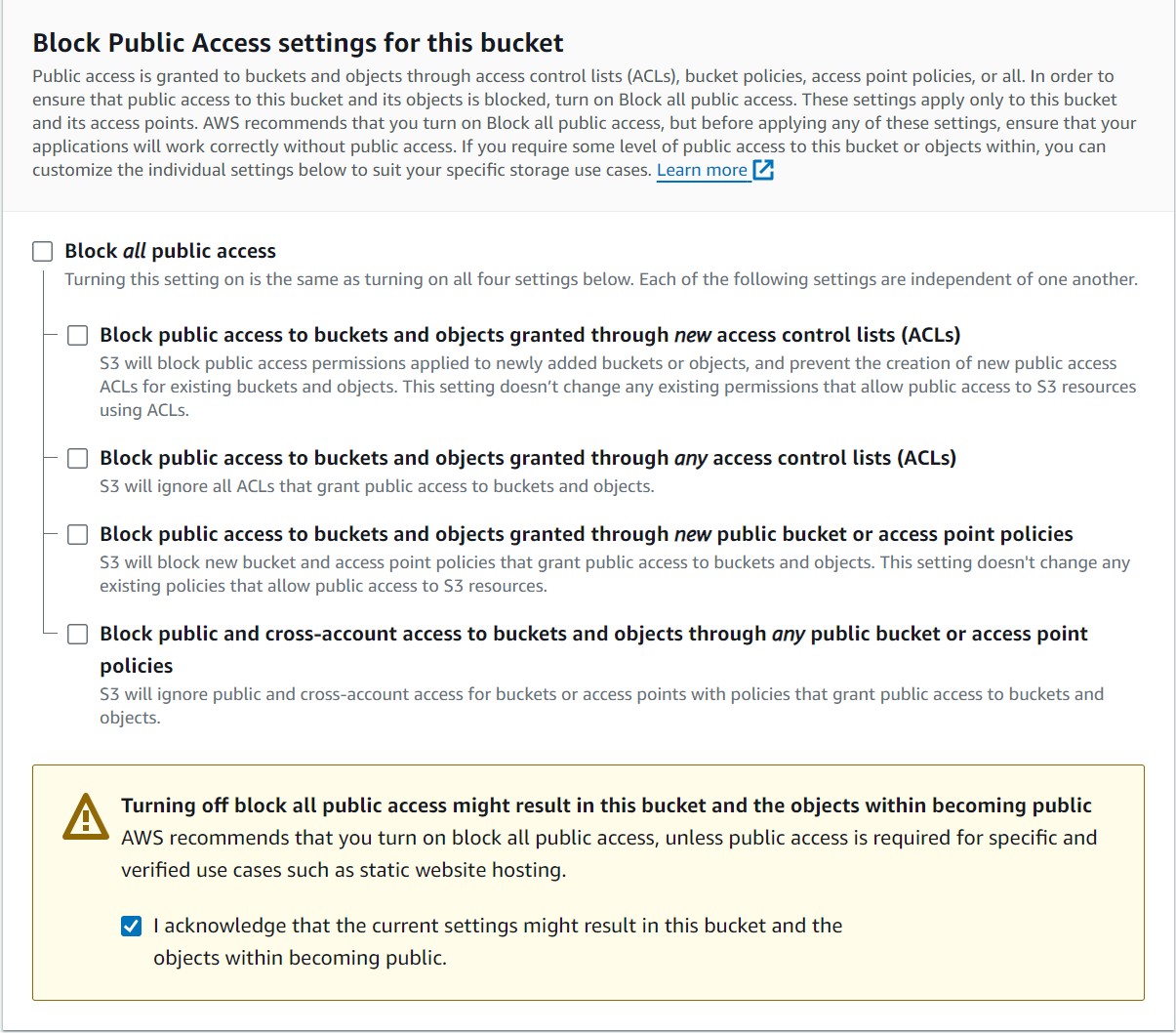
**Create Bucket** to create the bucket.

After creating the bucket, go to your EC2 instance and check if the S3 role is assigned

properly by running the command command will show the list of buckets.

aws s3 ls

. This





# Step-4

Now, we need to connect GitHub to our EC2 instance by generating an SSH key.Run the following command to generate the SSH key:

bash

Copy code

ssh-keygen -t rsa -b 4096 -C "your\_email@exampl e.com"

#Replace ["your\_email@example.com"](mailto:your_email@example.com) with your actu al email address.

Press Enter to save the key. To check the keys,

run

, and to see your public key, use

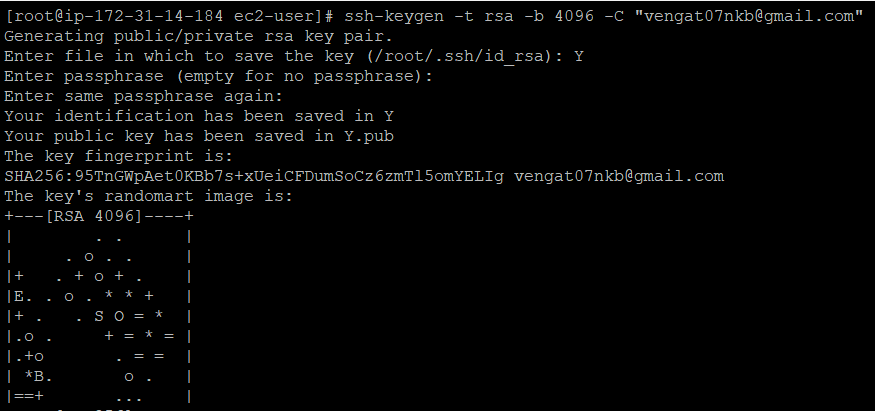
.

ls -l /root/.ssh

cat /root/.ssh/id\_rsa.pub

ls -l /root/.ssh

cat /root/.ssh/id\_rsa.pub

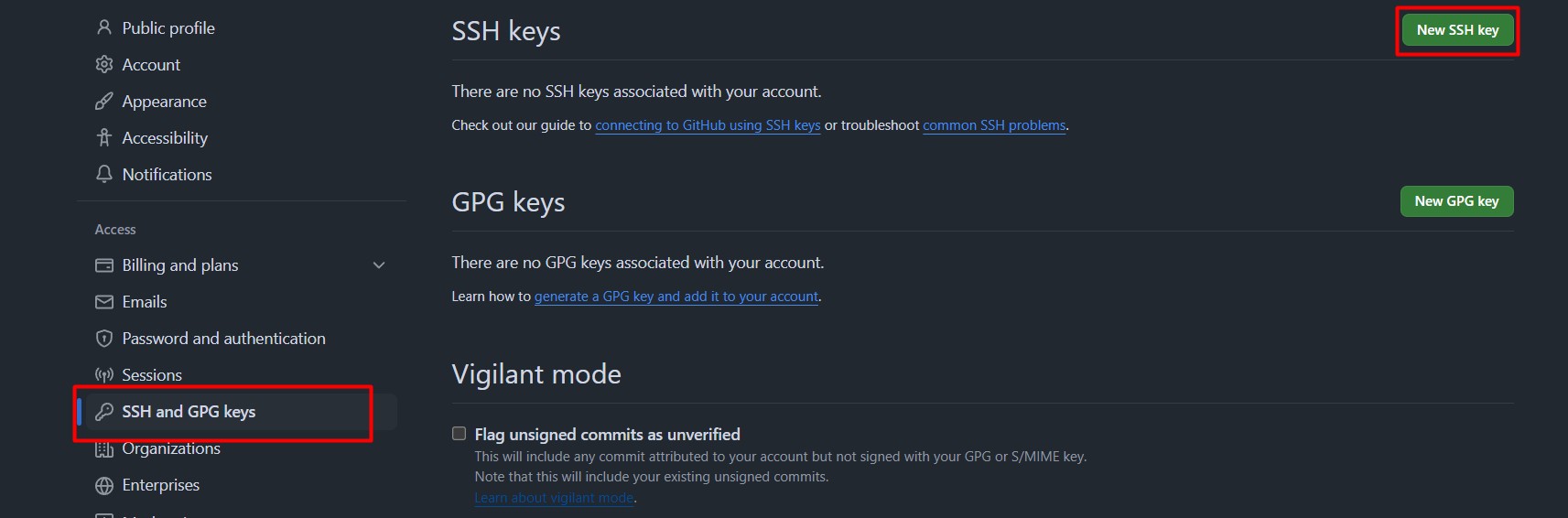


# Step-5

Go to your GitHub account, click on **Settings**, and select **New SSH key**.

Paste the key you copied from EC2.

***(Note:Make sure to specify which repository in Github you want to upload to S3)***



# Step-6

Now, we need to create a script to automate cloning the repository and uploading to S3.

Below is the script code to execute this.

**Important**: After creating the file, you need to give it executable permissions using

chmod +x

for it to run.

<filename>

#!/bin/bash

# Variables GIT\_REPO\_URL="git@github.com:Vengatesh-Bala/Falc S3\_BUCKET\_NAME="bucket-script-ec2" # Replace wit

LOCAL\_DIR="/tmp/Falcon-Fighters" # Temporary dir

# Step 1: Clone the Git Repository

echo "Cloning the repository from $GIT\_REPO\_URL if [ -d "$LOCAL\_DIR" ]; then

rm -rf "$LOCAL\_DIR" # Remove the existing dir

fi

git clone "$GIT\_REPO\_URL" "$LOCAL\_DIR"

if [ $? -ne 0 ]; then

echo "Error: Failed to clone the Git repositor

exit 1

fi

echo "Successfully cloned the repository."

# Step 2: Upload Files to S3

echo "Uploading files to S3 bucket $S3\_BUCKET\_NA aws s3 cp "$LOCAL\_DIR" "s3://$S3\_BUCKET\_NAME/" -

if [ $? -ne 0 ]; then

echo "Error: Failed to upload files to S3." exit 1

fi

echo "Successfully uploaded files to S3 bucket $

# Clean up

rm -rf "$LOCAL\_DIR"

echo "Deleted the local repository."



# Step-7

Run the script using .

./filename

You should see that the files are uploaded to your S3 bucket.

