

S3 Upload Automation Script

Code
Docs

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

Aa Name

S3 Upload
Automation
Script

```
yum update -y  
yum install git -y
```



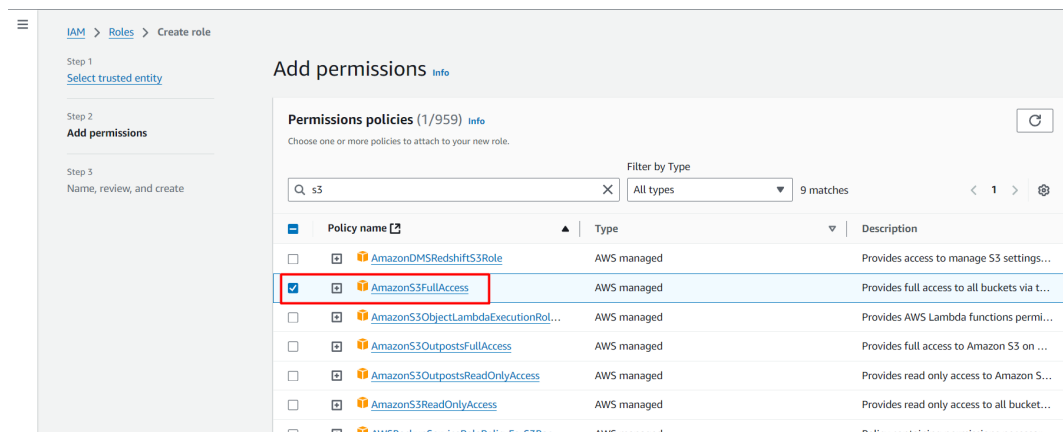
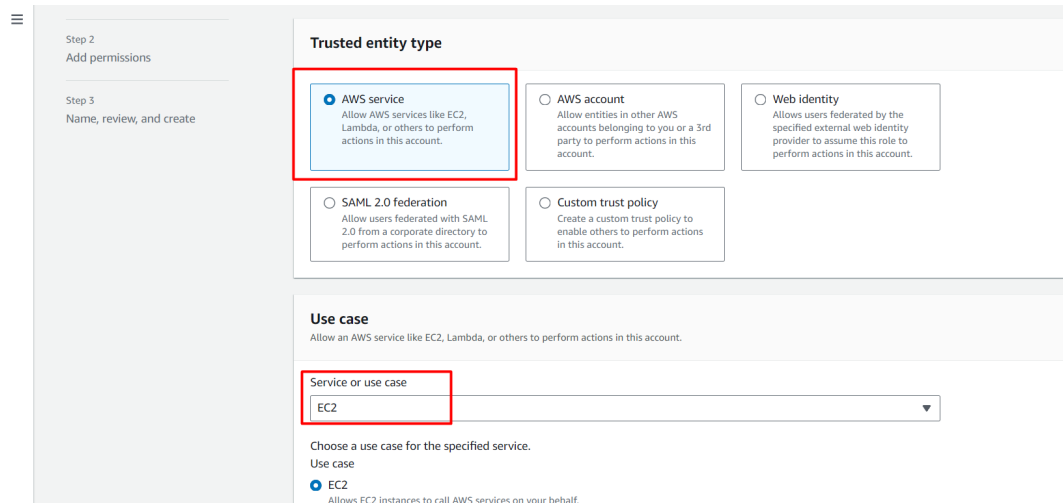
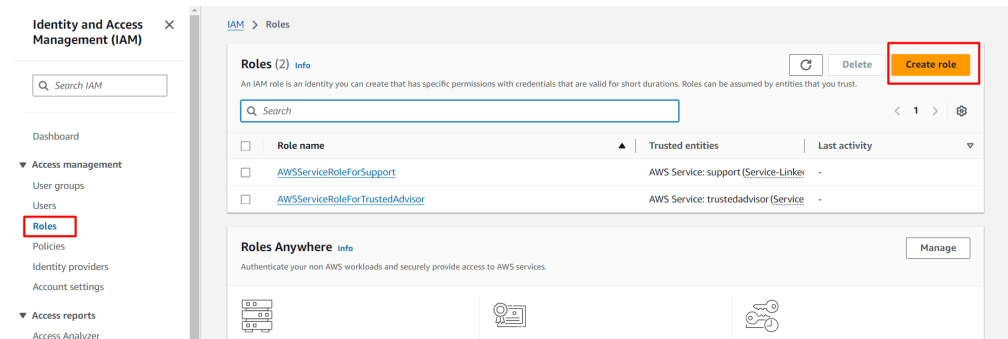
Cloning Git Repo to AWS S3 using Shell Script

```
password:
[root@ip-172-31-14-184 ec2-user]# sudo yum update -y
Last metadata expiration check: 0:10:43 ago on Wed Oct 30 12:52:12 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-14-184 ec2-user]# sudo yum install git -y
Last metadata expiration check: 0:13:52 ago on Wed Oct 30 12:52:12 2024.
Package git-2.40.1-1.amzn2023.0.3.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-14-184 ec2-user]# sudo yum install aws-cli -y
Last metadata expiration check: 0:14:24 ago on Wed Oct 30 12:52:12 2024.
Package awscli-2-2.15.30-1.amzn2023.0.1.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-14-184 ec2-user]# █
```

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**.



EC2 > Instances > i-0594c16710dbbf18b > Modify IAM role

Modify IAM role [Info](#)

Attach an IAM role to your instance.

Instance ID
i-0594c16710dbbf18b (S3-Bucket)

IAM role
Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

S3-Full-Access-EC2

[Create new IAM role](#)

Cancel [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 `aws s3 ls`. This command will show the list of buckets.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

☐ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

☐ Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

☐ Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

☐ Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

☐ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings might result in this bucket and the objects within becoming public.

```
[root@ip-172-31-14-184 ec2-user]# aws s3 ls
2024-10-30 16:41:59 bucket-script-ec2
```

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@example.com"
```

```
#Replace "your_email@example.com" with your actual email address.
```

- Press Enter to save the key. To check the keys, run `ls -l /root/.ssh`, and to see your public key, use `cat /root/.ssh/id_rsa.pub`.

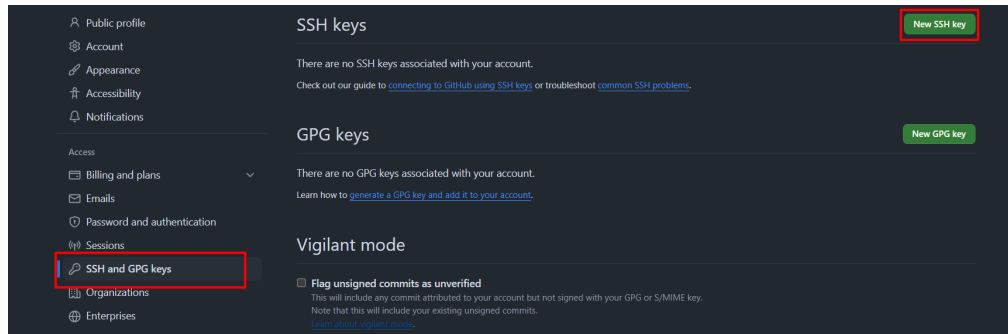
```
ls -l /root/.ssh
cat /root/.ssh/id_rsa.pub
```

```
[root@ip-172-31-14-184 ec2-user]# ssh-keygen -t rsa -b 4096 -C "vengat07nkb@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): Y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in Y
Your public key has been saved in Y.pub
The key fingerprint is:
SHA256:95TnGWpAet0KBb7s+xUeiCFDumSoCz6zmTl5omYELIg vengat07nkb@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|      . .      |
|     . o . .   |
|+    . + o + .  |
|E.   . o . * * + |
|+ .   . S O = *  |
|.o .   + = * =   |
|. +o    . = =    |
| *B.      o .    |
|==+       ...    |
```

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 <filename>` for it to run.

```
#!/bin/bash

# Variables
GIT_REPO_URL="git@github.com:Vengatesh-Bala/Falco"
S3_BUCKET_NAME="bucket-script-ec2" # Replace with your bucket name
LOCAL_DIR="/tmp/Falcon-Fighters" # Temporary directory

# 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 directory
fi

git clone "$GIT_REPO_URL" "$LOCAL_DIR"

if [ $? -ne 0 ]; then
```



```

    echo "Error: Failed to clone the Git repository."
    exit 1
fi

echo "Successfully cloned the repository."

# Step 2: Upload Files to S3
echo "Uploading files to S3 bucket $S3_BUCKET_NAME"
aws s3 cp "$LOCAL_DIR" "s3://$S3_BUCKET_NAME/" --recursive

if [ $? -ne 0 ]; then
    echo "Error: Failed to upload files to S3."
    exit 1
fi

echo "Successfully uploaded files to S3 bucket $S3_BUCKET_NAME"

# Clean up
rm -rf "$LOCAL_DIR"
echo "Deleted the local repository."

```

```

[root@ip-172-31-14-184 ec2-user]# chmod +x upload-S3.sh
[root@ip-172-31-14-184 ec2-user]#

```

Step-7

- Run the script using `./filename`.
- You should see that the files are uploaded to your S3 bucket.

Amazon S3

Buckets

Access Grants

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

Dashboards

Storage Lens groups

AWS Organizations settings

Feature spotlight

Amazon S3 > Buckets > bucket-script-ec2

bucket-script-ec2

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (10) Info

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	.git/	Folder	-	-	-
<input type="checkbox"/>	assets/	Folder	-	-	-
<input type="checkbox"/>	changelog.txt	txt	October 30, 2024, 22:59:44 (UTC+05:30)	3.6 KB	Standard
<input type="checkbox"/>	desktop.ini	ini	October 30, 2024, 22:59:44 (UTC+05:30)	159.0 B	Standard
<input type="checkbox"/>	Falcon Logo/	Folder	-	-	-