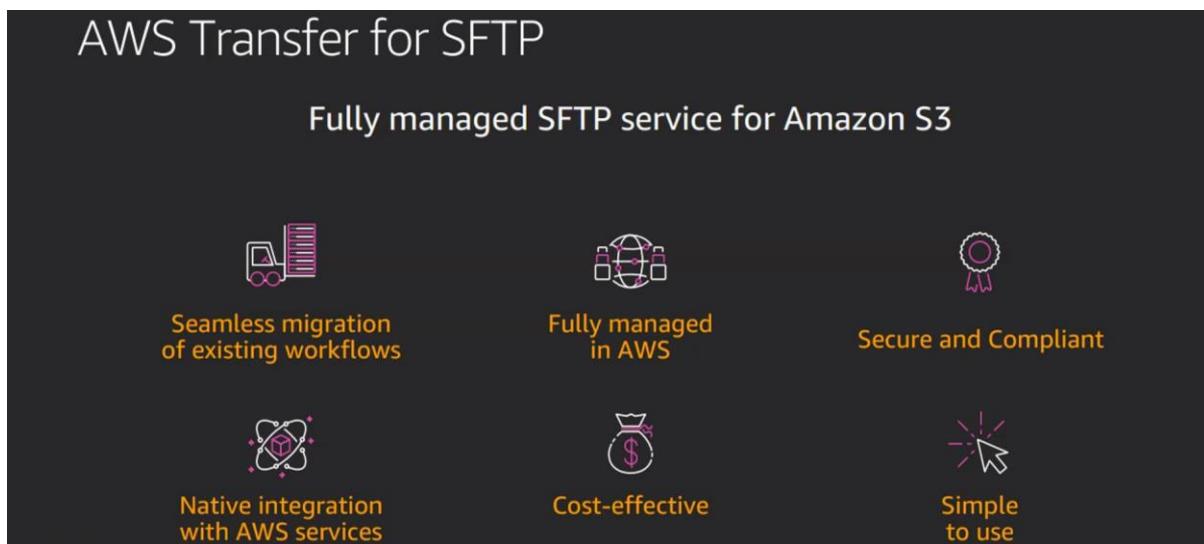


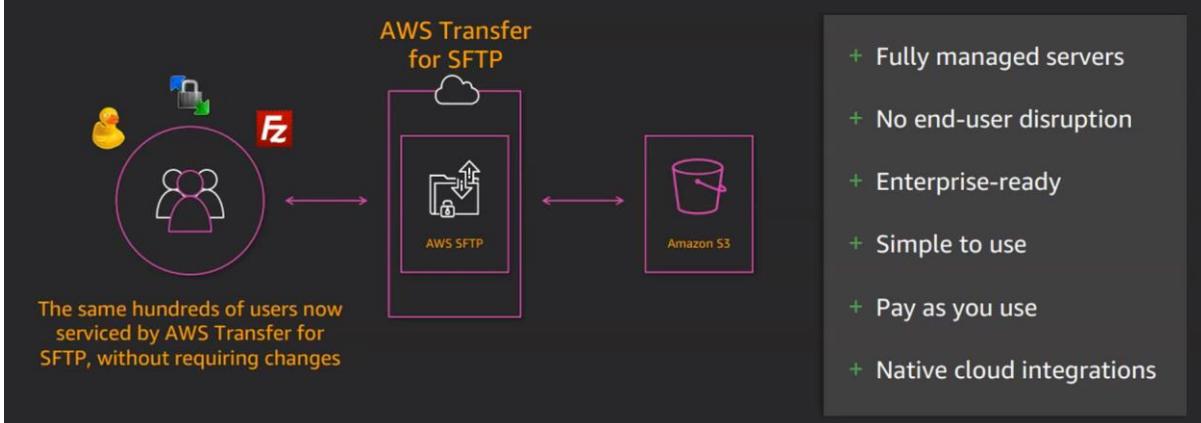
AWS - Transfer Family | Managed SFTP Service from AWS

⌚ AWS Transfer Family offers fully managed support for the transfer of files over SFTP, AS2, FTPS, and FTP directly into and out of Amazon S3 or Amazon EFS. You can seamlessly migrate, automate, and monitor your file transfer workflows by maintaining existing client-side configurations for authentication, access, and firewalls — so nothing changes for your customers, partners, and internal teams, or their applications.



1. If we use this service our data actually store in S3 bucket.
2. Using S3 is reliable because multiple copies can be maintained.
3. AWS transfer service is managed we don't need to worry about any servers.

Your SFTP architecture now



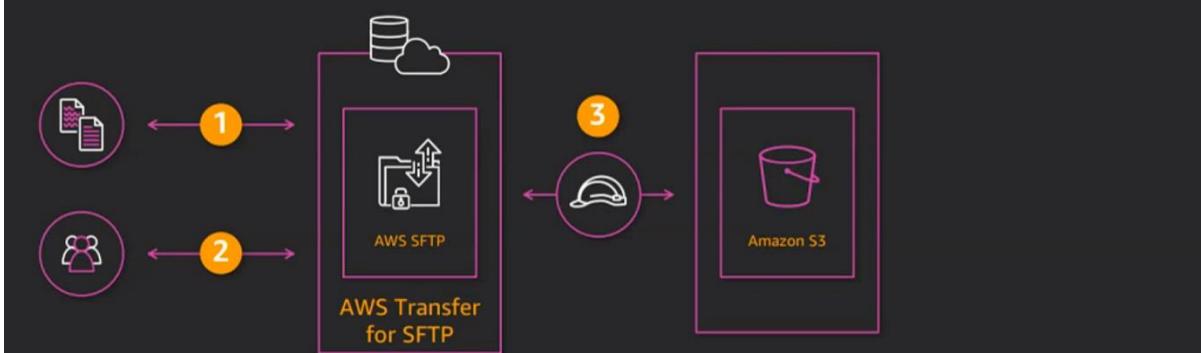
Key features

Time saved from managing SFTP servers	HIPAA-eligible and PCI-compliant
Automatically scales to meet your needs in real-time	Encryption at rest options such as SSE-S3 or SSE-KMS
Redundant across Availability Zones in Region	End-user activity tracking in Amazon CloudWatch
Store data in your Amazon S3 buckets for archiving, processing, or analyzing	SFTP endpoint fee @ \$0.30/hour
Automate post-upload processing with Amazon S3 events	SFTP uploads and downloads @ \$0.04/GB of transfer
Control end user access to resources by using IAM	Easy to set up and configure using the AWS Management Console or service API
Encrypt your data using server-side encryption using Amazon S3 or AWS KMS	No IT expertise required for SFTP server or user access configuration

Service managed user access

Store and manage user identities and keys inside the service

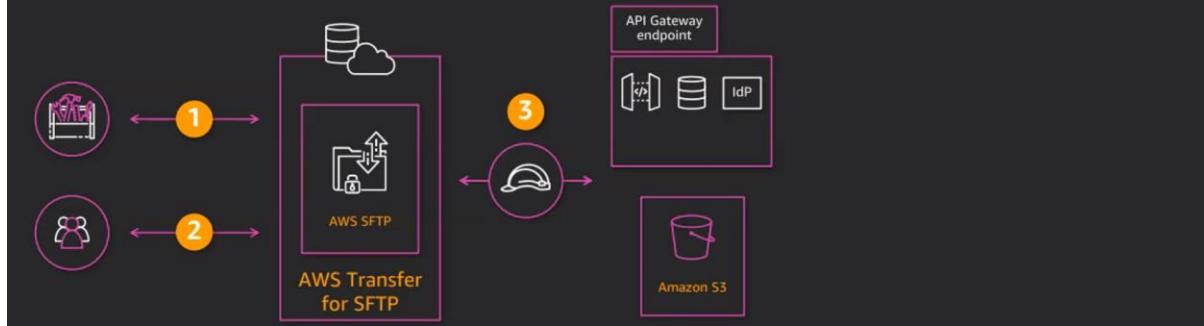
1. Configure your users credentials and keys using the AWS Management Console
2. Users serviced using their existing clients and credentials
3. Amazon S3 accessed by using IAM during file transfers



Plug in your custom identity provider (IdP)

If you have custom-built IdP, integrate by using Amazon API Gateway

1. Use API Gateway method to integrate your IdP
2. The service authenticates users using your IdP via API Gateway
3. The service assumes the IAM role to access bucket during file transfers



STEP 1:

1. Log in to AWS Console.
2. Go to S3 and create 2 buckets or use any empty bucket if available.
3. Then search AWS Transfer Family and create a server.

The screenshot shows the AWS Transfer Family console with the 'New server' wizard open. The wizard steps are:

1. Choose Protocols
2. Map your hostname
3. Set up your users
4. Select your S3 bucket(s) or EFS Filesystem(s)

On the right, there's a 'Pricing' section detailing costs for server endpoints, data uploads/downloads, and messages sent/received. The 'Benefits and features' section highlights operational burdens, workflow migrations, and processing management.

4. So here we get more than one option for the time being we'll use only SFTP.

The screenshot shows the 'Choose protocols' step of the 'Create server' wizard. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Choose protocols' (selected). Step 2 is 'Choose an identity provider'. Step 3 is 'Choose an endpoint'. Step 4 is 'Choose a domain'. Step 5 is 'Configure additional details'. Step 6 is 'Review and create'. The main content area is titled 'Choose protocols' and contains a section titled 'Select the protocols you want to enable'. It lists four options: SFTP (selected with a checked checkbox), AS2, FTPS, and FTP. Below each option is a small description and an 'Info' link. At the bottom right are 'Cancel' and 'Next' buttons.

5. On the next page select Service Managed.

The screenshot shows the 'Choose an identity provider' step of the 'Create server' wizard. The sidebar shows steps 1 through 6. Step 1 is 'Choose protocols' (selected). Step 2 is 'Choose an identity provider' (selected). Step 3 is 'Choose an endpoint'. Step 4 is 'Choose a domain'. Step 5 is 'Configure additional details'. Step 6 is 'Review and create'. The main content area is titled 'Choose an identity provider' and contains a section titled 'Identity Provider for SFTP, FTPS, or FTP'. It asks 'Identity provider type' and says 'An identity provider manages user access for authentication and authorization'. Three options are shown: 'Service managed' (selected with a blue circle), 'AWS Directory Service' (with a grey circle), and 'Custom Identity Provider' (with a grey circle). Each option has a brief description. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

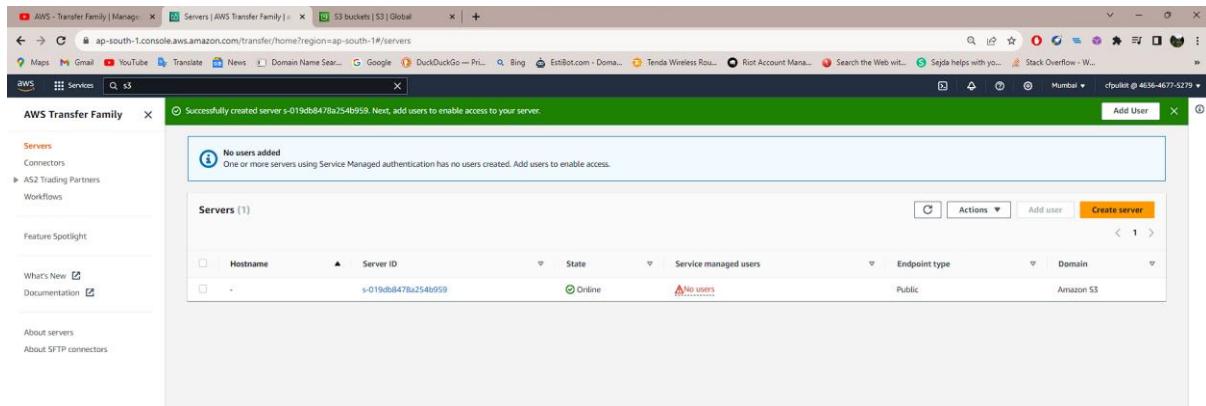
6. The next page we've two options either we can select publicly accessible or VPC hosted. We can use VPC only if we want to keep our data private but for now we'll go with publicly accessible.

The screenshot shows the 'Create server' wizard in the AWS Transfer Family console. The current step is 'Step 1 Choose protocols'. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Choose protocols' (selected). Step 2 is 'Choose an identity provider'. Step 3 is 'Choose an endpoint' (current step). Step 4 is 'Choose a domain'. Step 5 is 'Configure additional details'. Step 6 is 'Review and create'. The main content area is titled 'Choose an endpoint' and contains 'Endpoint configuration' settings. Under 'Endpoint type', 'Publicly accessible' is selected (radio button is checked). Under 'Custom hostname', a dropdown menu is set to 'None'. Under 'FIPS Enabled', a checkbox is unchecked. Navigation buttons at the bottom right are 'Cancel', 'Previous', and 'Next'.

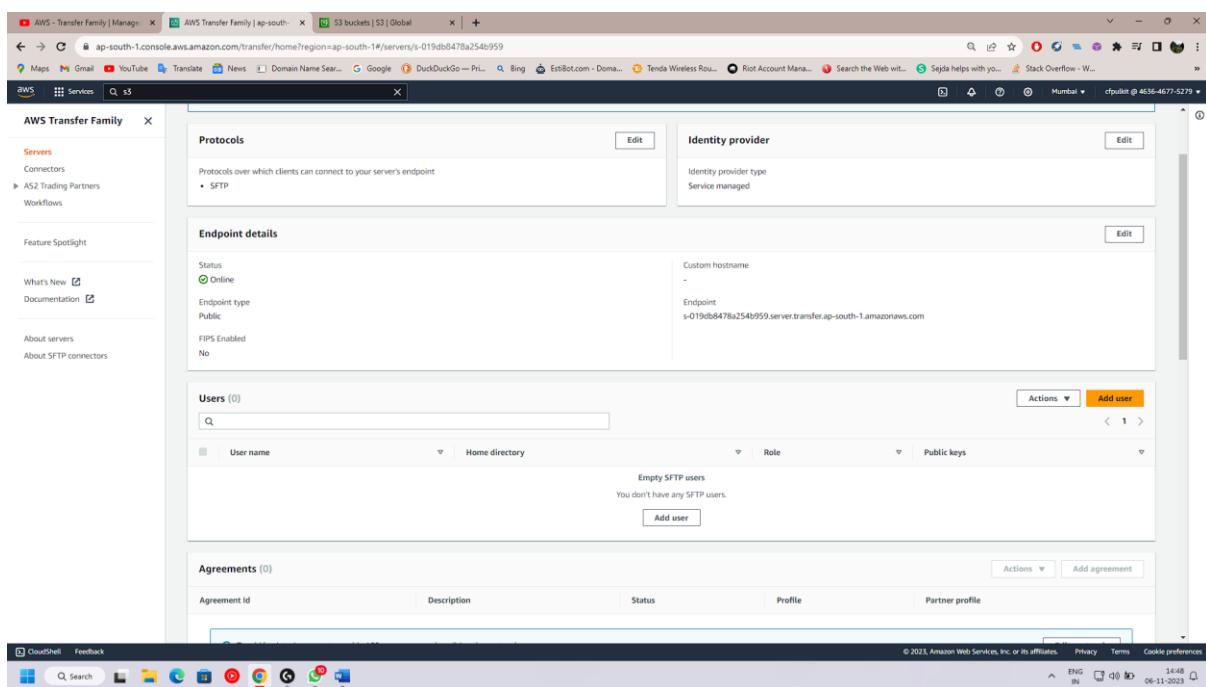
7. So, here we see two options again and we're going to select Amazon S3 because we're going to upload and download our files from here.

The screenshot shows the 'Create server' wizard in the AWS Transfer Family console. The current step is 'Step 2 Choose an identity provider'. On the left, a sidebar lists steps from 1 to 6. Step 1 is 'Choose protocols'. Step 2 is 'Choose an identity provider' (selected). Step 3 is 'Choose an endpoint'. Step 4 is 'Choose a domain' (current step). Step 5 is 'Configure additional details'. Step 6 is 'Review and create'. The main content area is titled 'Choose a domain' and contains 'Domain' settings. Under 'Domain', 'Amazon S3' is selected (radio button is checked). Under 'Amazon EFS', a radio button is unchecked. Navigation buttons at the bottom right are 'Cancel', 'Previous', and 'Next'.

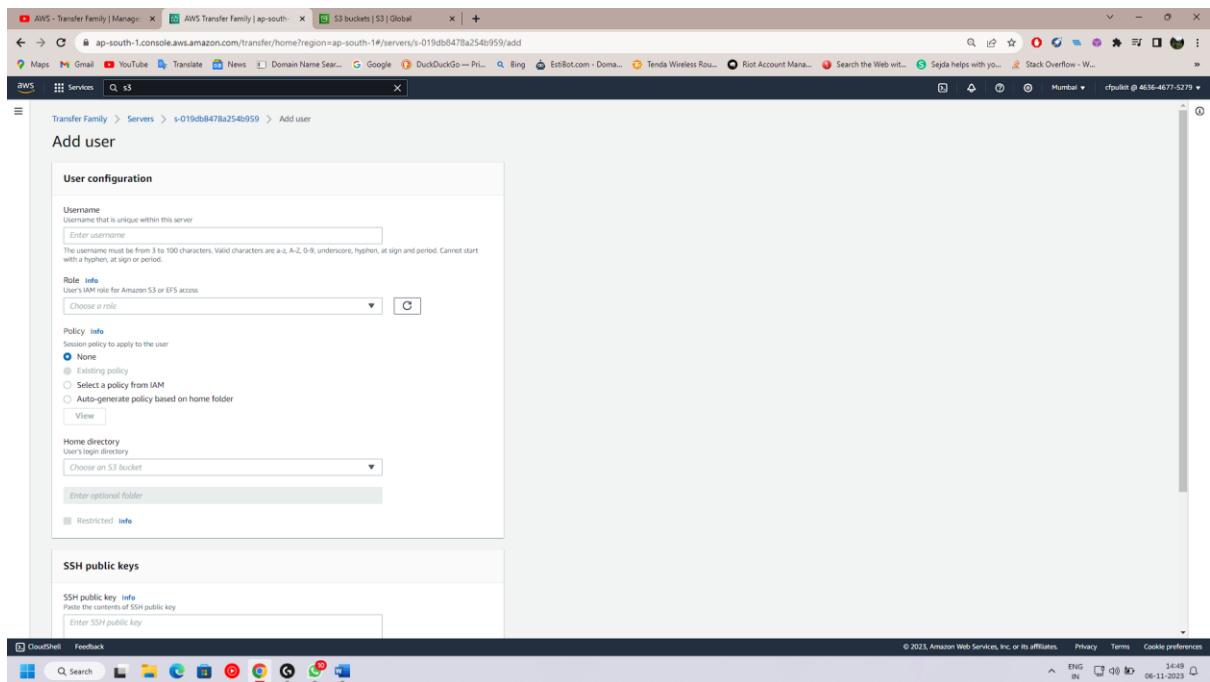
8. The next we can keep as default and move to the last page where we'll review the server details and then create server and wait for it to start and wait for the server to get online.



9. Once the server is online then we need to Add users to this.

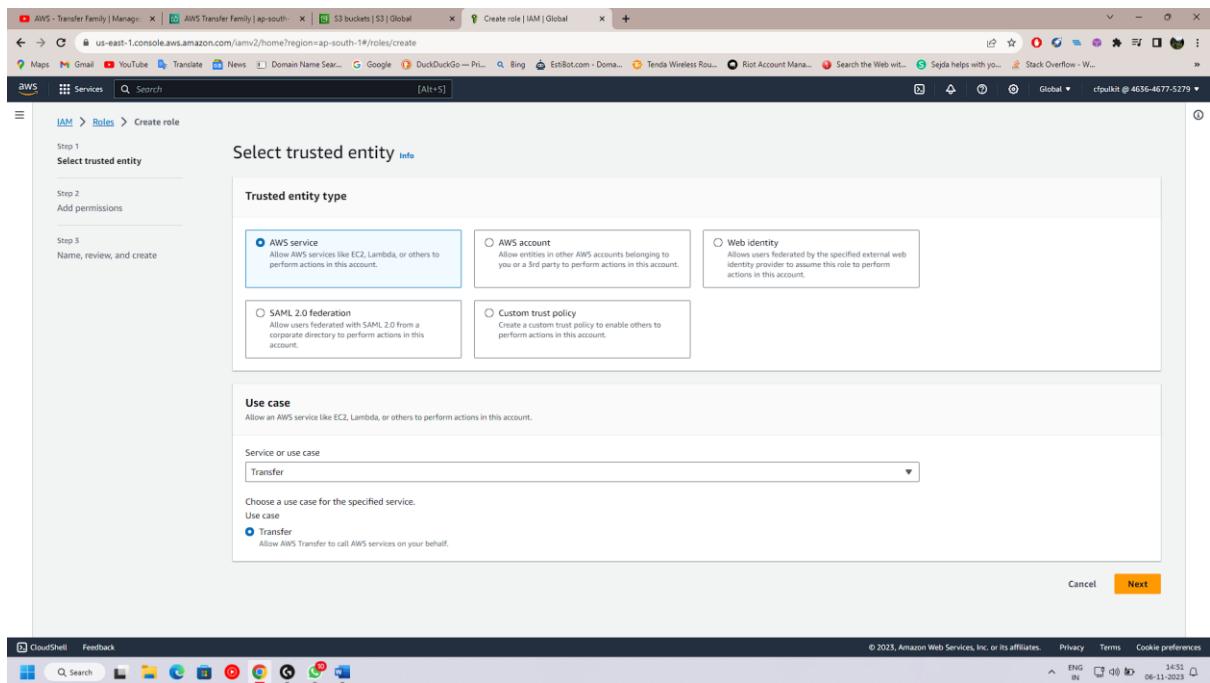


10. We'll see this page on our screen and we've to give it a name then we need to create a IAM role for this to move forward.



CREATE an IAM ROLE

1. Go to IAM and create role select Use Case as “Transfer”



2. Before giving it the permission we need to create a policy of our own.
 - a. First go to policies and click on create policy.
 - b. Select service as S3 and tick on all S3 actions.

The screenshot shows the AWS IAM Policy Editor for an S3 policy. At the top, there's a dropdown for 'S3' and a 'Allow All actions' button. Below that, a note says 'Specify what actions can be performed on specific resources in S3.' Under 'Actions allowed', there's a 'Filter Actions' input field and a 'Effect' dropdown set to 'Allow'. A list of actions is shown, with 'All S3 actions (s3:*)' checked. To the right, there are buttons for 'Expand all' and 'Collapse all'. The list includes: List (Selected 10/10), Read (Selected 53/53), Write (Selected 42/42), Permissions management (Selected 15/15), and Tagging (Selected 10/10).

c. Then scroll down a little on the resources section select add ARN to bucket.

The screenshot shows the 'Add ARNs to restrict access' section for an S3 bucket. It has a note 'Add ARNs to restrict access.', a text input field containing 'arn:aws:s3:::pulkitbucket1', and a 'Any' checkbox.

The screenshot shows the 'Specify ARN(s)' dialog box. It has tabs for 'Visual' (selected) and 'Text'. Under 'Resource bucket name', there's a checkbox 'Any bucket name' and an empty input field. Under 'Resource ARN', there's an input field containing 'arn:aws:s3:::pulkitbucket1'. At the bottom are 'Cancel' and 'Add ARNs' buttons.

d. Then go to the top and select JSON on it to change something in the code and delete the section marked as blue below.

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```

1▼ {
2    "Version": "2012-10-17",
3    "Statement": [
4        {
5            "Sid": "VisualEditor0",
6            "Effect": "Allow",
7            "Action": [
8                "s3>ListStorageLensConfigurations",
9                "s3>ListAccessPointsForObjectLambda",
10               "s3>GetAccessPoint",
11               "s3>PutAccountPublicAccessBlock",
12               "s3>GetAccountPublicAccessBlock",
13               "s3>ListAllMyBuckets",
14               "s3>ListAccessPoints",
15               "s3>PutAccessPointPublicAccessBlock",
16               "s3>ListJobs",
17               "s3>PutStorageLensConfiguration",
18               "s3>ListMultiRegionAccessPoints",
19               "s3>CreateJob"
20            ],
21            "Resource": "*"
22        },
23        {
24            "Sid": "VisualEditor1",
25            "Effect": "Allow",
26            "Action": "s3:*",
27            "Resource": "arn:aws:s3:::pulkitbucket1"
28        }
29    ]
30}

```

+ Add new statement

Visual **JSON**

Edit statement VisualEditor0
Remove

Add actions
Choose a service Filter services

Included S3

Available AMP API Gateway API Gateway V2 ASC Access Analyzer Account Activate Alexa for Business

Add a resource

Add a condition

- e. Then in the last add this line to the code and change the bucket name accordingly to yours.

"Resource": ["arn:aws:s3:::pulkitbucket1", "arn:aws:s3:::pulkitbucket1/*"]

IAM > Policies > Create policy

Step 1 Specify permissions

Step 2 Review and create

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

```

1▼ {
2    "Version": "2012-10-17",
3    "Statement": [
4        {
5            "Sid": "VisualEditor1",
6            "Effect": "Allow",
7            "Action": "s3:*",
8            "Resource": "arn:aws:s3:::pulkitbucket1", "arn:aws:s3:::pulkitbucket1/*"
9        }
10    ]
11}

```

Visual **JSON** Actions ▾

Edit statement VisualEditor1 Remove

Add actions
Choose a service Filter services

Included S3

Available AMP

- f. Give the policy a name and then go back to roles then add two permissions.
g. First S3 Read only Access and the policy we created.
h. Then create role.

[Go back to Amazon Transfer Family and add user now.](#)

Transfer Family > Servers > s-019db8478a254b959 > Add user

Add user

User configuration

Username
Username that is unique within this server
The username must be from 3 to 100 characters. Valid characters are a-z, A-Z, 0-9, underscore, hyphen, at sign and period. Cannot start with a hyphen, at sign or period.

Role [Info](#)
User's IAM role for Amazon S3 or EFS access
▼ [C](#)

Policy [Info](#)
Session policy to apply to the user
 None
 Existing policy
 Select a policy from IAM
 Auto-generate policy based on home folder
[View](#)

Home directory
User's login directory
▼

 Restricted [Info](#)

- Above we can see that we gave it a name then we selected the role for it which we created and then on the home directory select the bucket which we want to have the data for the demo.

💡 As we can see, I've selected Restricted, which implies that if we select this option, the user won't be able to navigate anywhere else.

💡 After this we need to give it a SSH Public Key which we can create through a online website. Then add user.

<https://8gwifi.org/sshfunctions.jsp>

SSH public keys

SSH public key [Info](#)
Paste the contents of SSH public key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAQABAAQCB4x2Fr+zjG7uVtDlfM/k6k0qvMvviGvAi
69IGSei3eLfD03Zoen9FIMhi/5LxAzaPdgqTZAjHE3Zci/cTLy85+vEE9J+kWtUxTvF1
```

Users (1)					Actions ▾	Add user
					< 1 >	
User name	Home directory	Role	Public keys			
testpulkitbucket1	Restricted	pulkitRole1	1			

- Then we need to open Win SCP app which we can download it through the internet. In this app we need to give it the log in credentials which we can find on the server.

1. We need to copy the end point to the host name.

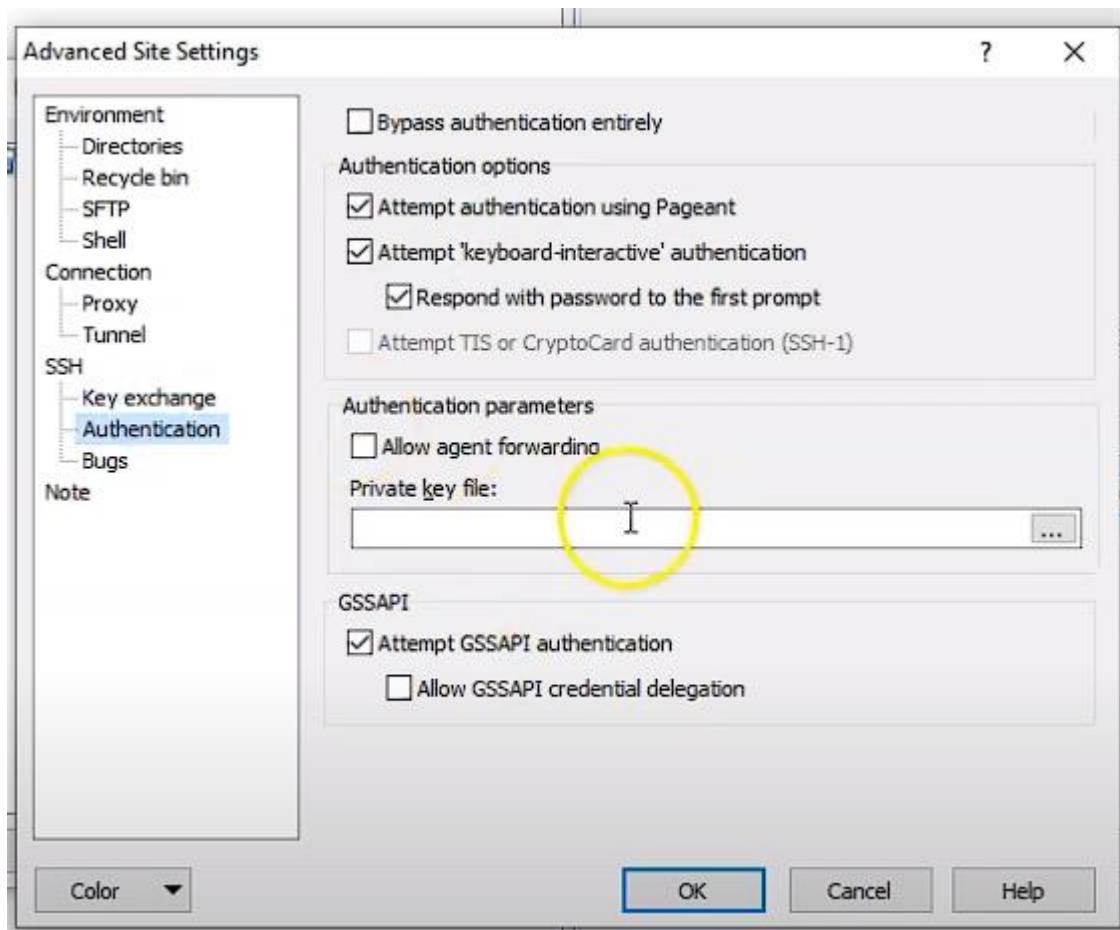
Endpoint details

Status Online	Custom hostname -
Endpoint type Public	Endpoint s-019db8478a254b959.server.transfer.ap-south-1.amazonaws.com
FIPS Enabled No	

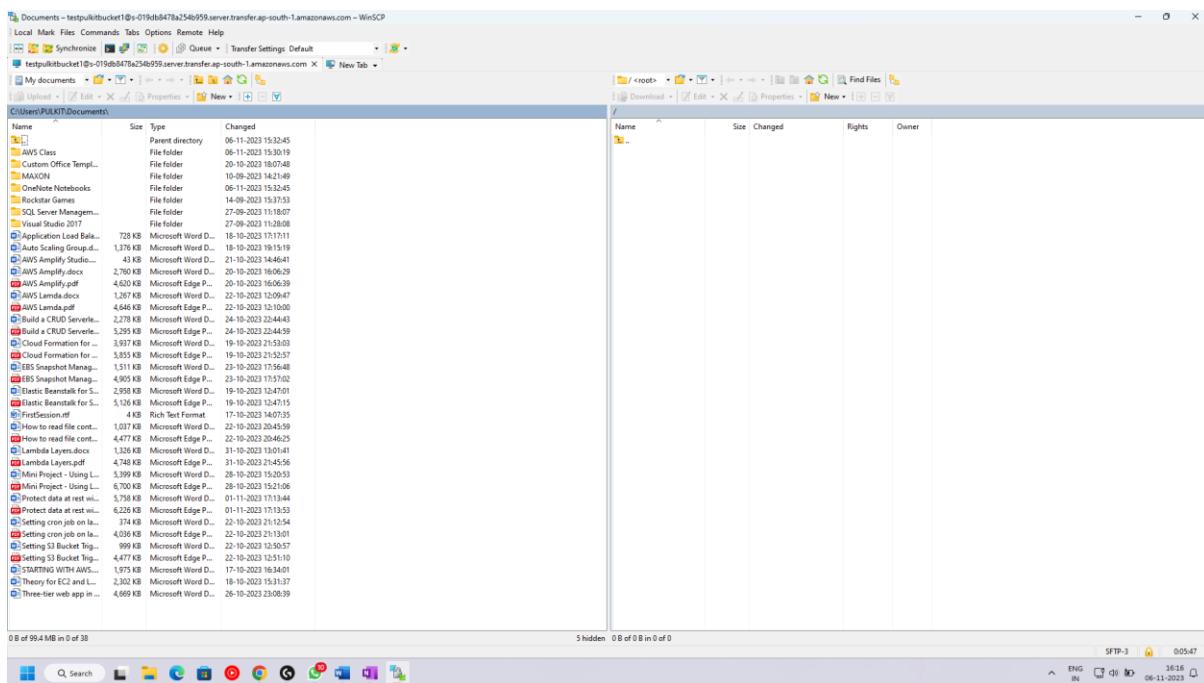
2. Then we need to write our username

Users (1)					Actions ▾	Add user
					< 1 >	
User name	Home directory	Role	Public keys			
testpulkitbucket1	Restricted	pulkitRole1	1			

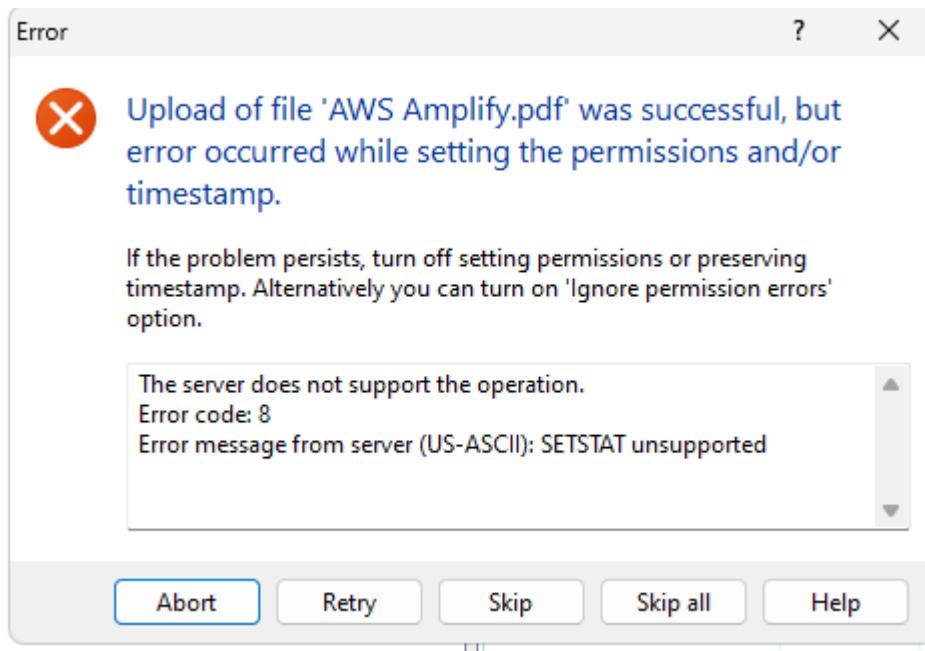
3. For the password go to advanced settings and select the private key file from the system and run it.



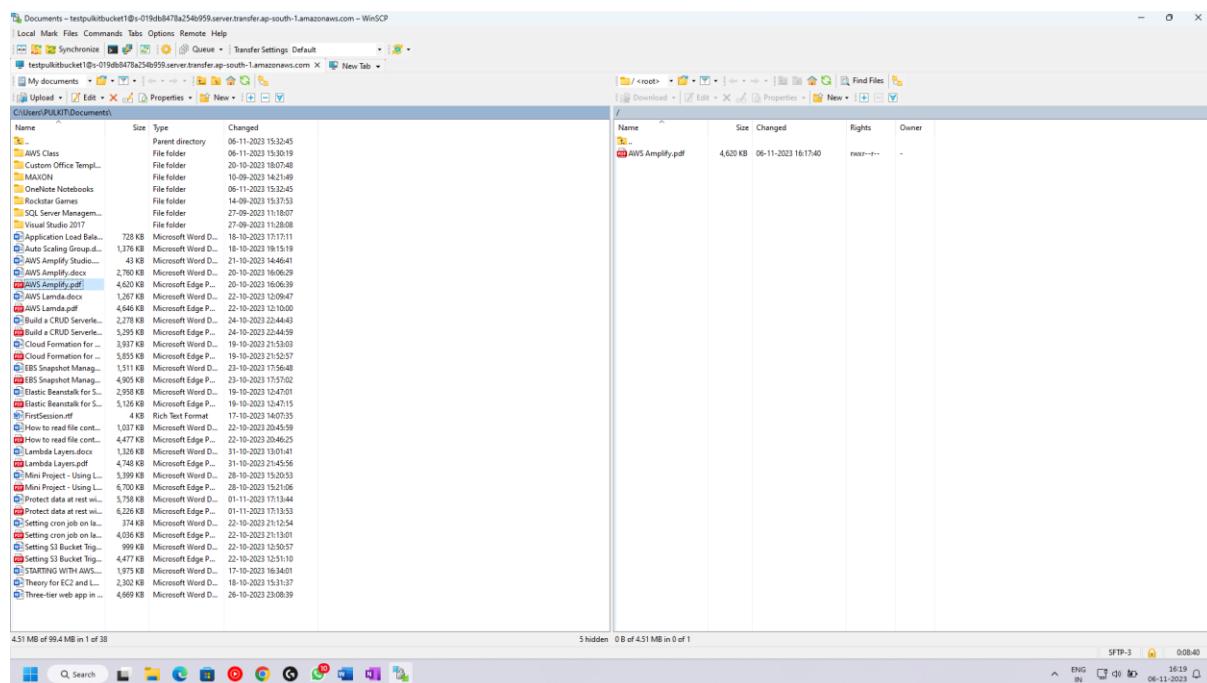
4. Here we'll see two windows side by side left side is our window and the right side is root window or S3 part.



5. While uploading we'll see a error but just skip it because our file is uploaded



6. Hence, we can see that our file is uploaded



7. And now if we go to S3 we can see the same file is uploaded is here too.

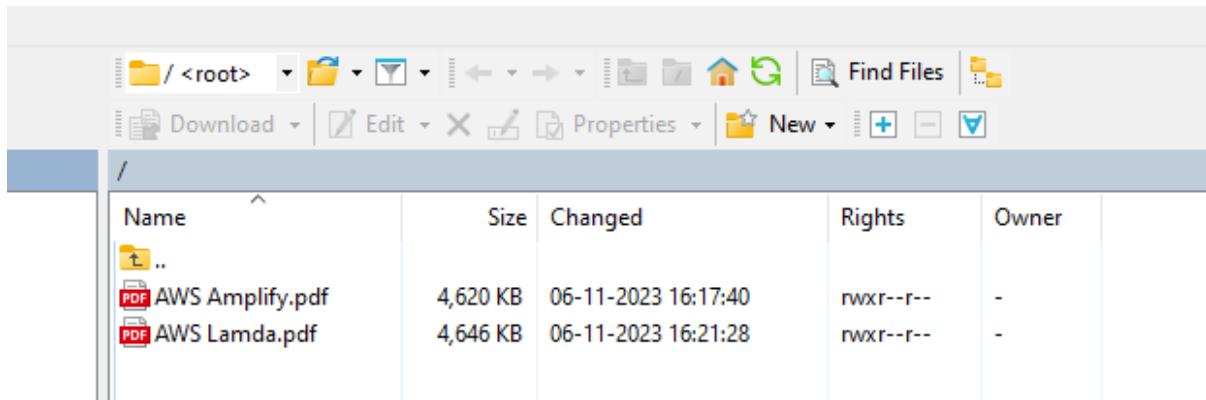
The screenshot shows the AWS S3 console interface. The top navigation bar includes tabs for 'AWS Transfer Family | Manage', 'AWS Transfer Family | ap-south...', 'pulkitbucket1 - S3 bucket | S3', 'Roles | IAM | Global', 'Policies | IAM | Global', 'Online Generate SSH keys alg...', 'Paraphrasing Tool - QuillBot AI', and a '+' button. Below the navigation bar, the AWS logo and 'Services' link are visible. A search bar with the placeholder '[Alt+S]' is present. The main content area shows the 'Amazon S3 > Buckets > pulkitbucket1' path. Under the 'Objects' tab, there is one object listed: 'testpulkitbucket1/' which is a folder. The table below lists the object with columns for Name, Type, Last modified, Size, and Storage class.

Name	Type	Last modified	Size	Storage class
testpulkitbucket1/	Folder	-	-	-

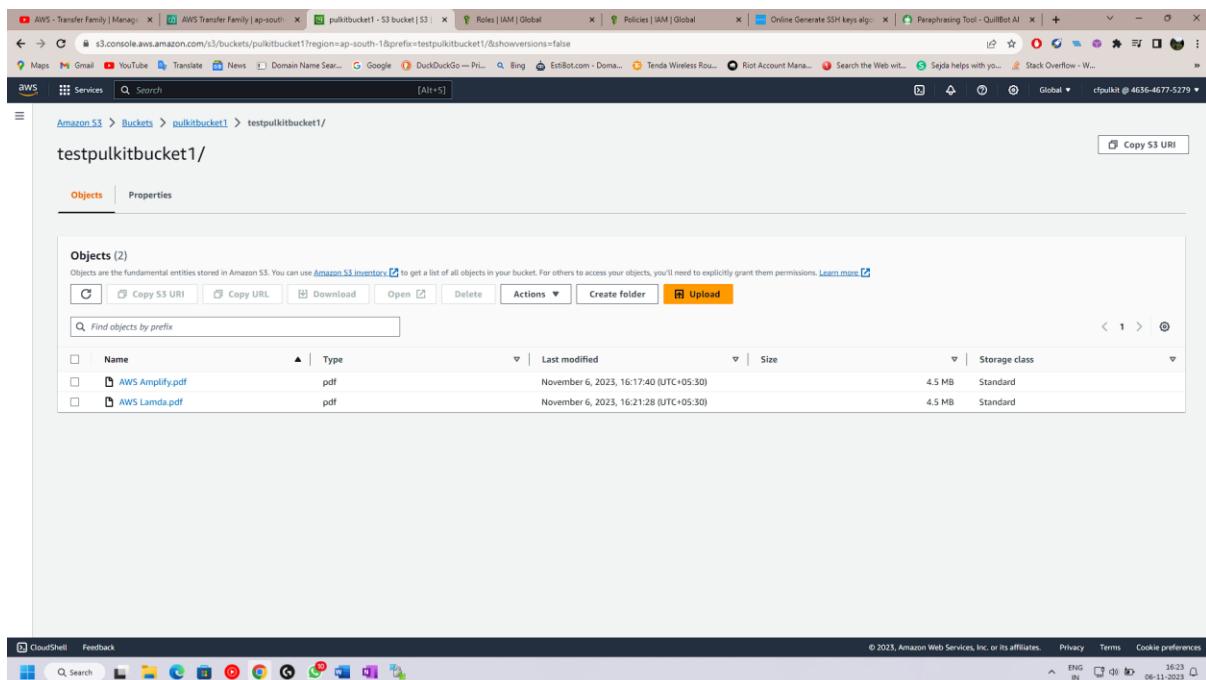
This screenshot shows the AWS S3 console after a file has been uploaded. The path 'Amazon S3 > Buckets > pulkitbucket1 > testpulkitbucket1/' is visible. A 'Copy S3 URI' button is present in the top right. Under the 'Objects' tab, there is one object listed: 'AWS Amplify.pdf'. The table below lists the object with columns for Name, Type, Last modified, Size, and Storage class.

Name	Type	Last modified	Size	Storage class
AWS Amplify.pdf	pdf	November 6, 2023, 16:17:40 (UTC+05:30)	4.5 MB	Standard

8. Now if we upload another file in Win SCP to see if it is working properly or not. If it gets uploaded then the application is working fine and we can proceed further. The same file is now available in S3 bucket too.



Name	Size	Changed	Rights	Owner
..				
AWS Amplify.pdf	4,620 KB	06-11-2023 16:17:40	rwxr--r--	-
AWS Lambda.pdf	4,646 KB	06-11-2023 16:21:28	rwxr--r--	-



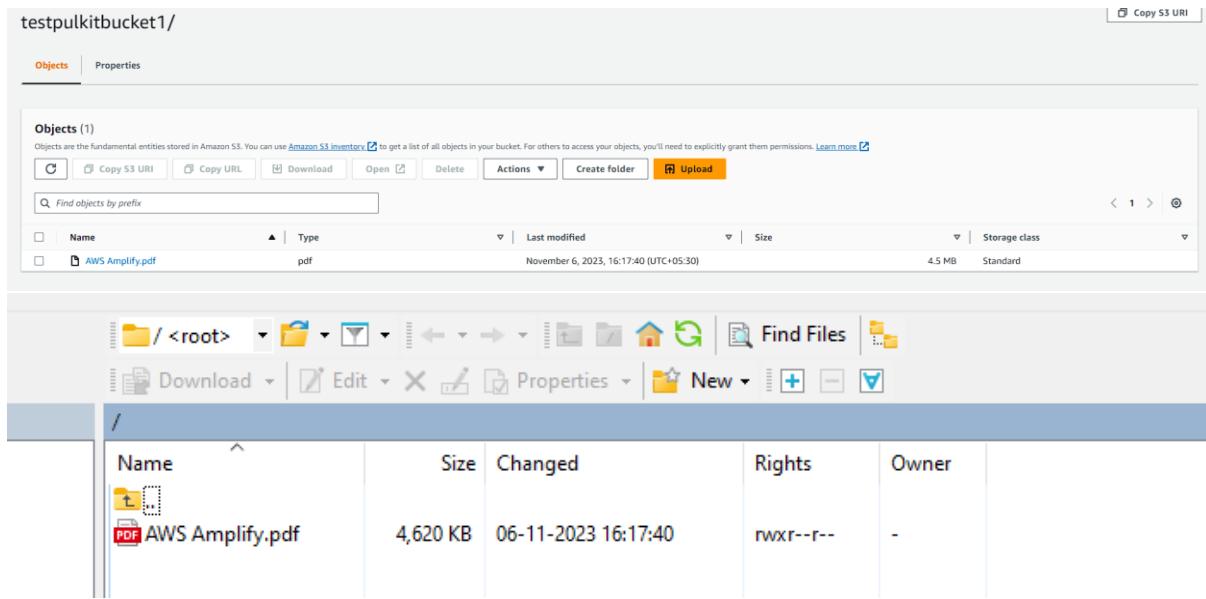
Amazon S3 > Buckets > testpulkitbucket1 > testpulkitbucket1/

testpulkitbucket1/

Objects (2)

Name	Type	Last modified	Size	Storage class
AWS Amplify.pdf	pdf	November 6, 2023, 16:17:40 (UTC+05:30)	4.5 MB	Standard
AWS Lambda.pdf	pdf	November 6, 2023, 16:21:28 (UTC+05:30)	4.5 MB	Standard

😊 Therefore, if we would like to verify the deletion process, we can do so by deleting a file from S3, which would also destroy it from Win SCP.



💡 The catch is that we can see every bucket that is available in the S3 if we go to the AWS transfer family, modify user configuration, deselect "Restricted," save it, and open a new Win SCP session.