Set up an SFTP server on AWS using the AWS Transfer Family service. Ensure the SFTP site is accessible using a username and password for authentication

prerequisites

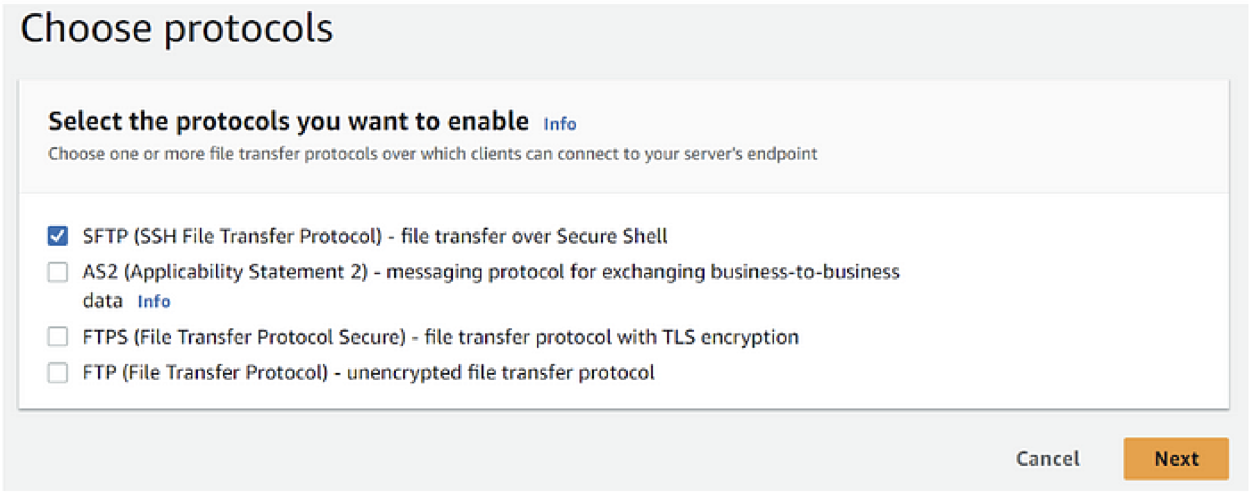
1. An S3 Bucket
2:IAM Role with Permissions
3. AWS Transfer Family Endpoint
4. User Configuration
5. Networking and Security
6. Testing SFTP Access Use an SFTP client (e.g., FileZilla, WinSCP, or CLI `sftp` command)

What is **AWS Transfer Family**

AWS Transfer Family is a managed file transfer service that supports the transfer of files over SFTP, AS2, FTPS, and FTP directly into and out of Amazon S3 or Amazon EFS. In essence, it is a managed file server service that makes it very easy to set up a server to facilitate the supported protocols.

Step1: go to AWS account —--> navigate to AWS Transfer Family click on create then it shows like this

Select the SFTP protocol as shown below and click next.



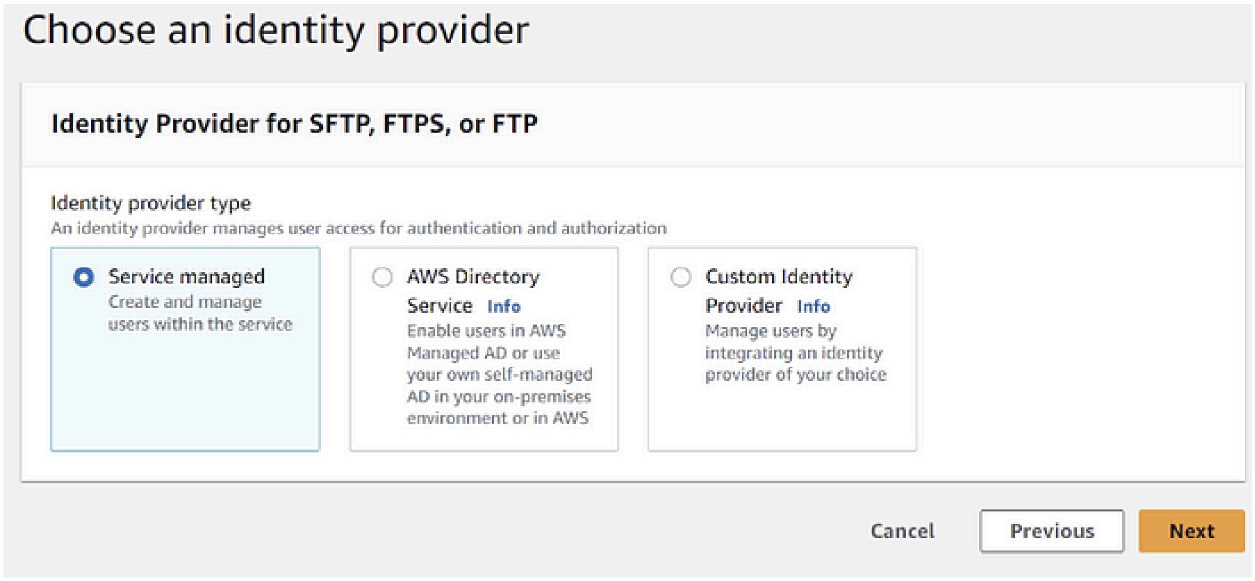
Choose protocols

Select the protocols you want to enable Info

Choose one or more file transfer protocols over which clients can connect to your server's endpoint

- [x] SFTP (SSH File Transfer Protocol) - file transfer over Secure Shell
- [ ] AS2 (Applicability Statement 2) - messaging protocol for exchanging business-to-business data Info
- [ ] FTPS (File Transfer Protocol Secure) - file transfer protocol with TLS encryption
- [ ] FTP (File Transfer Protocol) - unencrypted file transfer protocol

Step2: Select "Service managed" as the identity provider type and click next. We will create the necessary users to access the SFTP service at a later stage.



Choose an identity provider

Identity Provider for SFTP, FTPS, or FTP

Identity provider type

An identity provider manages user access for authentication and authorization

[✓ Service managed] Create and manage users within the service

AWS Directory Service Info — Enable users in AWS Managed AD or use your own self-managed AD in your on-premises environment or in AWS

Custom Identity Provider Info — Manage users by integrating an identity provider of your choice

Service-Managed Identities:

- Add users through the AWS Transfer Family console or API.
- Assign each user an SSH public key (for SFTP) or a password (for FTPS/FTP).

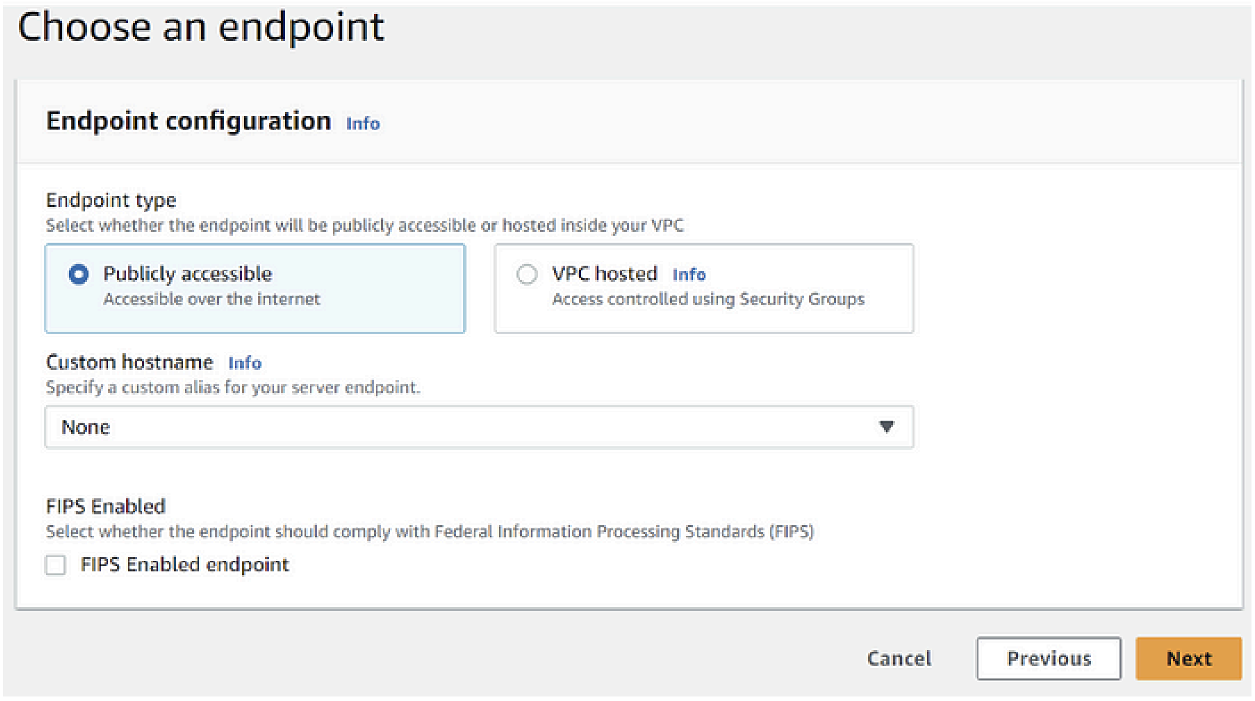
Custom Identity Provider:

- Create an **AWS Lambda function** for user authentication and directory mapping.
- Register the Lambda function with the Transfer Family server.

AWS Directory Service:

- Configure an AWS-managed Microsoft AD or link your existing on-premises AD.
- Associate the AD with your Transfer Family server.

Step 3: Now we choose an endpoint. We are going to make ours publicly accessible, but we could attach it to a VPC and make it more private. appropriate sectionforthecustomhostname.


Choose an endpoint

Endpoint configuration Info

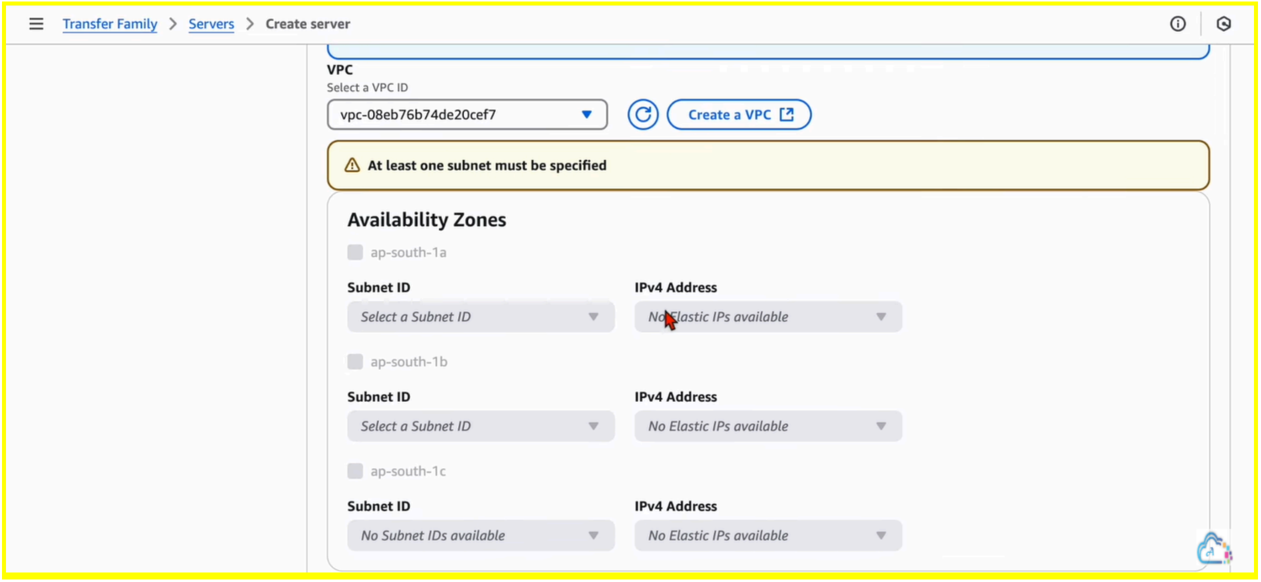
Endpoint type
Select whether the endpoint will be publicly accessible or hosted inside your VPC

[✓ Publicly accessible — Accessible over the internet]
VPC hosted Info — Access controlled using Security Groups

Custom hostname Info
Specify a custom alias for your server endpoint.
None

FIPS Enabled
Select whether the endpoint should comply with Federal Information Processing Standards (FIPS)
☐ FIPS Enabled endpoint

NOTE: now I have public access and a more secure VPC host we can launch the subnet and security group but we need to asinine the Elastic IP to it is a cost-effective way we will create costume hoster zone also



Transfer Family > Servers > Create server

VPC
Select a VPC ID
vpc-08eb76b74de20cef7

Create a VPC

At least one subnet must be specified

Availability Zones

ap-south-1a

Subnet ID	IPv4 Address
Select a Subnet ID	No Elastic IPs available

ap-south-1b

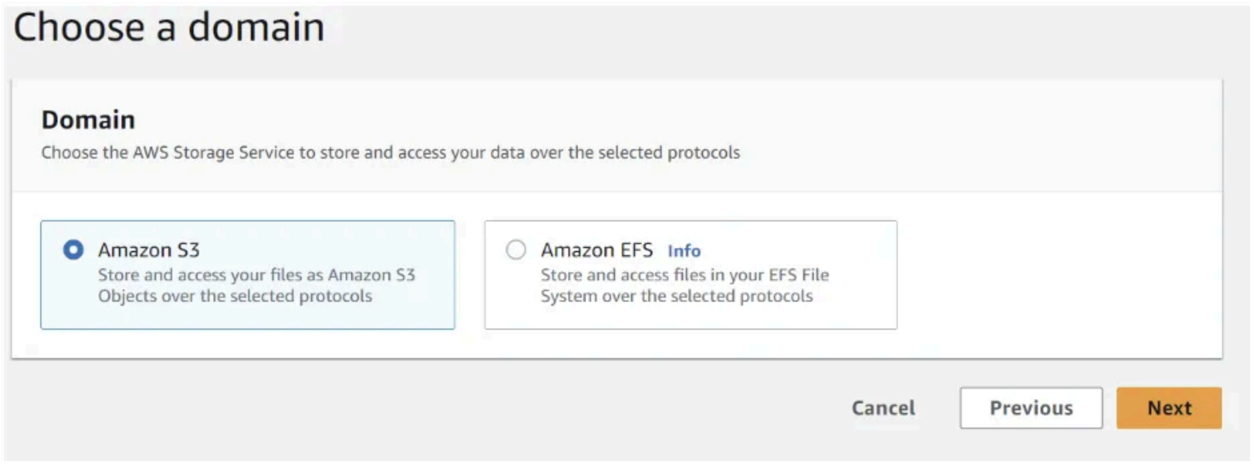
Subnet ID	IPv4 Address
Select a Subnet ID	No Elastic IPs available

ap-south-1c

Subnet ID	IPv4 Address
No Subnet IDs available	No Elastic IPs available

Create a hosted zone for the identification or take option we have any a hoster zone go to hoster stone based on the required in my case taking non

Step 4: Select S3 as the storage service to use on the backend and click Next. You can also select EFS if it aligns with your specific business use case.



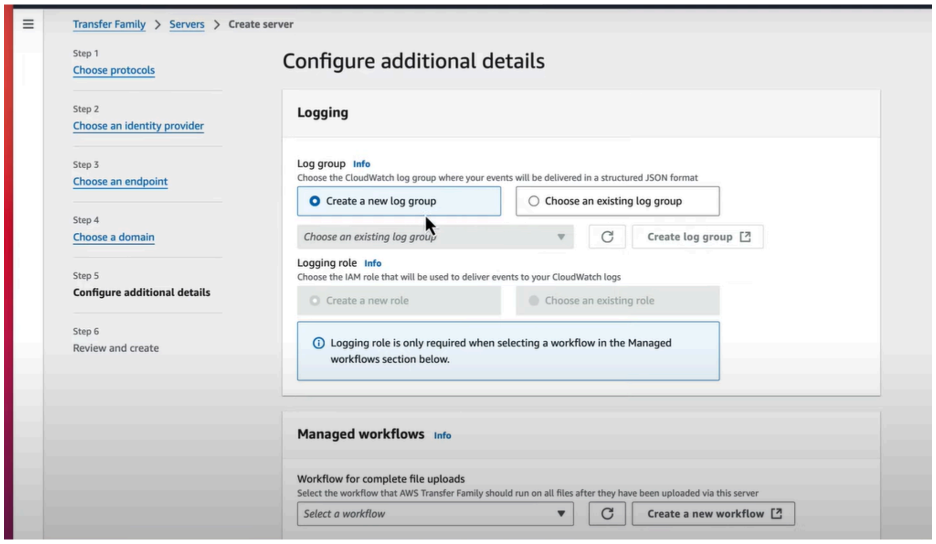
Choose a domain

Domain
Choose the AWS Storage Service to store and access your data over the selected protocols

[✓ Amazon S3] Store and access your files as Amazon S3 Objects over the selected protocols

Amazon EFS Info — Store and access files in your EFS File System over the selected protocols

Step5: create the cloud with Loge group and mange the work flow how the file come to hear private key and public for migration can click and review the


Transfer Family > Servers > Create server

Step 1
Choose protocols

Step 2
Choose an identity provider

Step 3
Choose an endpoint

Step 4
Choose a domain

Step 5
Configure additional details

Step 6
Review and create

Configure additional details

Logging

Log group Info
Choose the CloudWatch log group where your events will be delivered in a structured JSON format
(●) Create a new log group ( ) Choose an existing log group
Choose an existing log group | Create log group

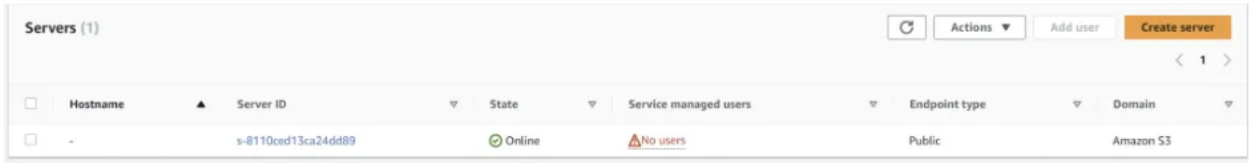
Logging role Info
Choose the IAM role that will be used to deliver events to your CloudWatch logs
( ) Create a new role ( ) Choose an existing role

Logging role is only required when selecting a workflow in the Managed workflows section below.

Managed workflows Info

Workflow for complete file uploads
Select the workflow that AWS Transfer Family should run on all files after they have been uploaded via this server
Select a workflow | Create a new workflow

Step6: Review the configuration once more and click “Create” Please note that it may take some time for the server to launch. Once it’s ready,

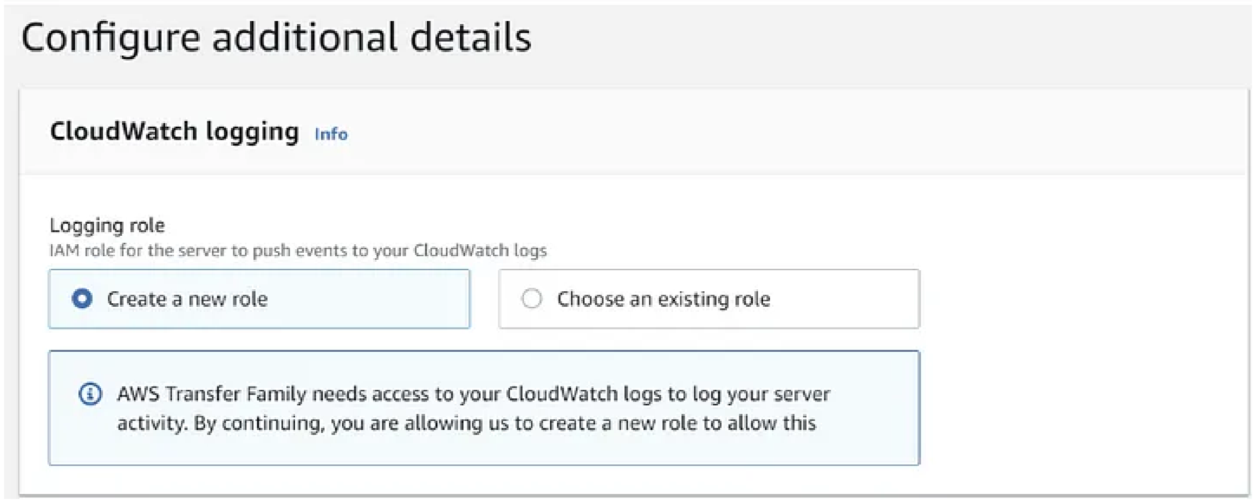

Servers (1)

Actions | Add user | Create server

Hostname	Server ID	State	Service managed users	Endpoint type	Domain
-	s-8110ced13ca24dd89	Online	No users	Public	Amazon S3

Add Users

Select the server and click on “Add User” and it will open the user creation screen as shown below. Enter the desired username and choose the appropriate IAM role to associate with this user.
Create a role select a new role and create a custom role

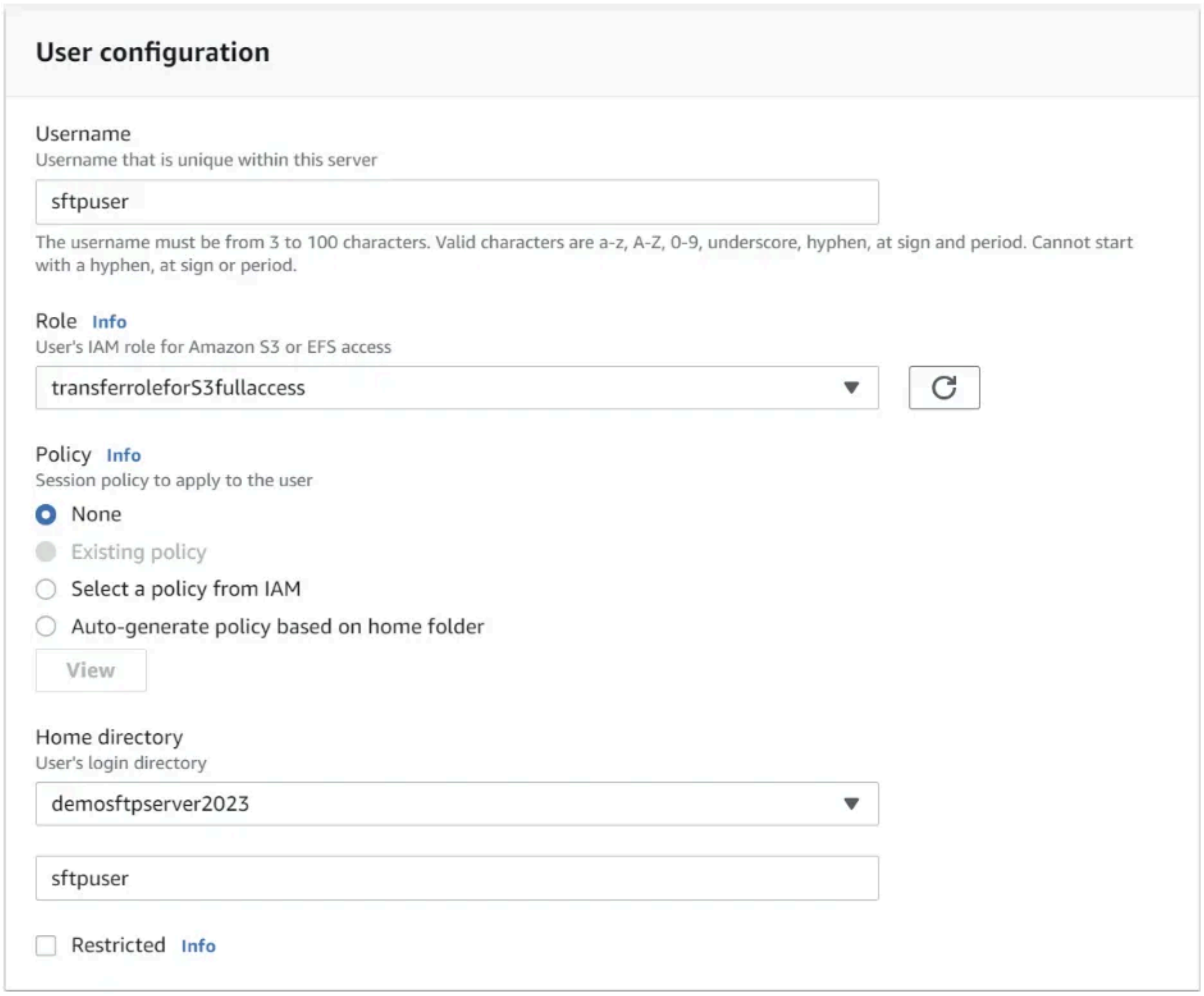

Configure additional details

CloudWatch logging Info

Logging role
IAM role for the server to push events to your CloudWatch logs
(●) Create a new role ( ) Choose an existing role

AWS Transfer Family needs access to your CloudWatch logs to log your server activity. By continuing, you are allowing us to create a new role to allow this

Note: give the top role and give bucker name you want the restick the bucket in a single folder



User configuration

Username
Username that is unique within this server

sftpuser

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

transferroleforS3fullaccess

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

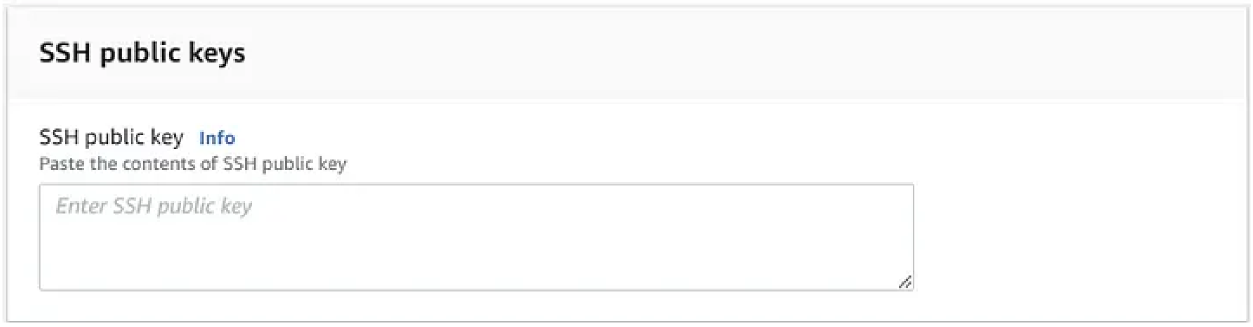
demosftpserver2023

sftpuser

☐ Restricted Info

Enable restricted sepic folder user has one folder

- Paste SSH public key content in the SSH section as shown below. We will use the SSH private key to connect to the SFTP server.



SSH public keys

SSH public key Info
Paste the contents of SSH public key

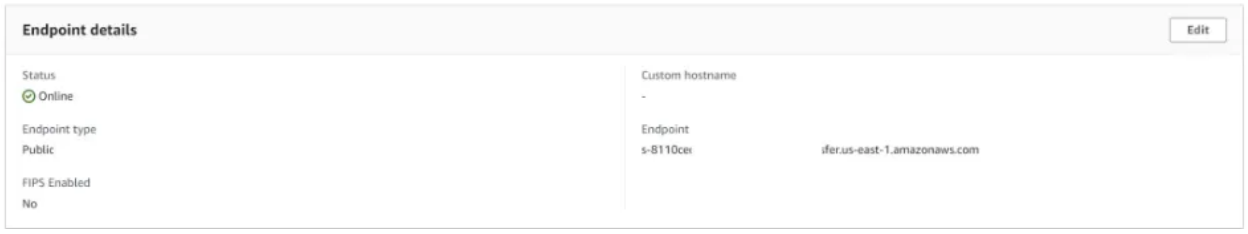
Enter SSH public key

Once the user is created as shown below we are ready to connect to the SFTP server. ssh-keygen command

Now we can use any tool to connect Sftp

SFTP Access Use an SFTP client (e.g., FileZilla, WinSCP, or CLI `sftp` command)

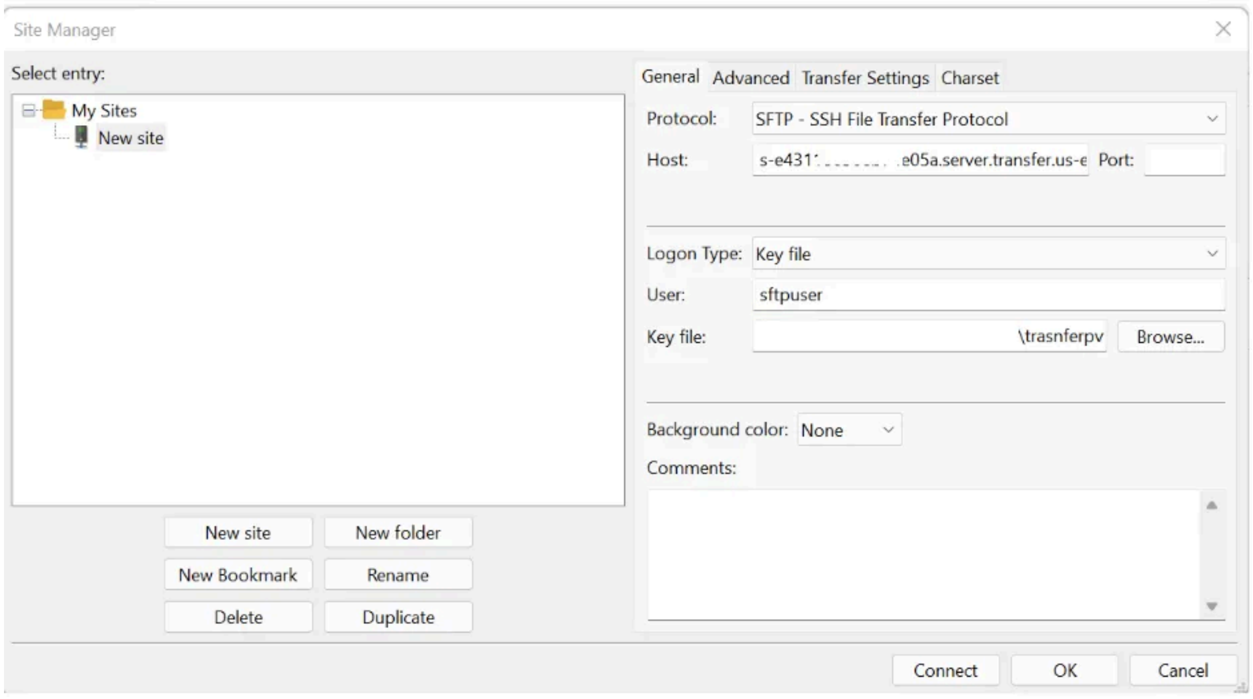
Copy the endpoint from the server we created before as shown below. We need this endpoint to connect any tool.



Endpoint details

Status Online	Custom hostname -
Endpoint type Public	Endpoint s-8110ce sfer.us-east-1.amazonaws.com
FIPS Enabled No	

Step7: Now copy the endpoint


Site Manager

Select entry:

My Sites
New site

Protocol: SFTP - SSH File Transfer Protocol

Host: s-e431`. . . . . .e05a.server.transfer.us-e Port:

Logon Type: Key file

User: sftpuser

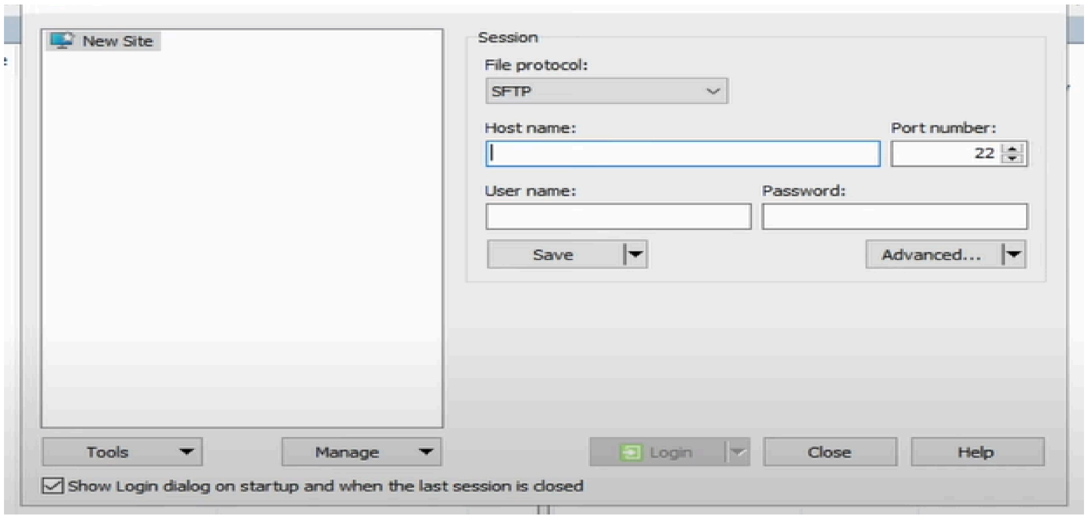
Key file: \trasnferpv

Background color: None

Comments:

Give the host endpoint, and username in sftp we create, and also give your PEM key you past in the user

Click the "Connect" button, and FileZilla will establish a connection with the SFTP server, WinSCP gives the endpoint user, password SSH key 22 use the drag and group



File protocol: SFTP

Host name:

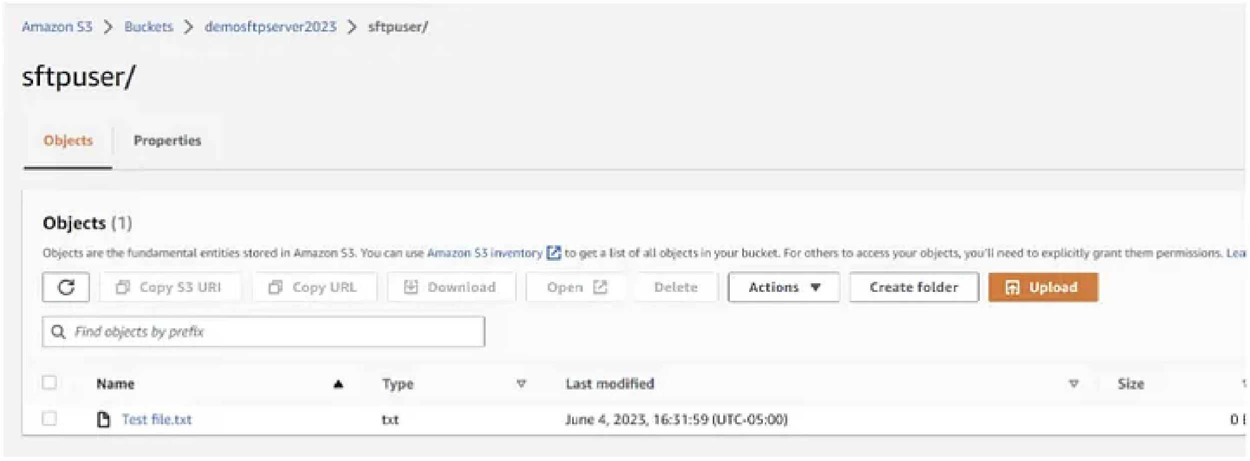
Port number: 22

User name:

Password:

☑ Show Login dialog on startup and when the last session is closed

After completing the it will upload the s3 bucket



sftpuser/

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Lea

Name	Type	Last modified	Size
Test file.txt	txt	June 4, 2023, 16:31:59 (UTC-05:00)	0

NOTE: we can use 2 methods normal best pratic and using lambda i recommend got for without lambda

Cost Optimization Comparison

Feature	Lambda-Based IdP	Service-Managed IdP
Setup Cost	Higher (development of Lambda function)	Low (simple configuration)
Operational Cost	Includes Lambda execution costs	No additional cost (part of Transfer Family)
User Scaling	Scales well for large, dynamic user bases	May require manual intervention
Maintenancefort	High (manage Lambda function, external systems)	Low

Lamda script :

```
import json
import paramiko
import boto3
import os

def lambda_handler(event, context):
    S3Client = boto3.client('s3')
    S3Client.download_file ('sftp-process-bucket', 'key-file/sftp.pem', '/tmp/keyname.pem')
    pem_key = paramiko.RSAKey.from_private_key_file("/tmp/keyname.pem")

    #Create a new client
    SSHClient = paramiko.SSHClient()
    SSHClient.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    host ="54.173.254.242"
    SSHClient.connect (hostname = host, username = "ec2-user", pkey = pem_key)

    print("Connected to: " + host)

    s_path = '/home/ec2-user/source_dir/'
    s_pattern = "'Trigger*'"
    rawcommand = 'find {path} -name {pattern}'
    command = rawcommand.format(path = s_path, pattern = s_pattern)
    stdin, stdout, stderr = SSHClient.exec_command(command)
    FileList = stdout.read().splitlines()

    SFTPClient = SSHClient.open_sftp()
    FileCount = 0
    for TrigFile in FileList:
        (head, filename) = os.path.split(TrigFile)
        FileName = filename.decode('utf-8')
        print(FileName)
        TempFile = '/tmp/' + FileName
        S3File = 'sftp-files/' + FileName
        SFTPClient.get(TrigFile, TempFile)
        S3Client.upload_file(TempFile, 'sftp-process-bucket', S3File)
        SFTPClient.remove(TrigFile)
        FileCount += 1
    SFTPClient.close()
    SSHClient.close()

    return str(FileCount) + " file(s) have been uploaded to the S3 bucket."
```