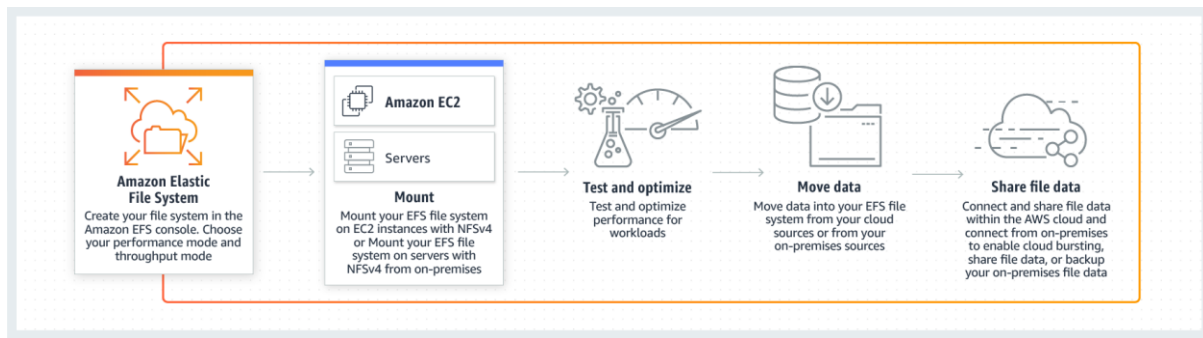


Getting Started with AWS - Amazon Elastic File System



Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources. It is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files, eliminating the need to provision and manage capacity to accommodate growth.

This Scenario is for: **Amazon EC2 mount instructions (from local VPC)**

Step 1: Create your EC2 two AMI Linux instances (using default security groups) to check your Data Online with AWS DataSync

Step 2: Once your Instances are ready you can access using putty [AMI Server 1](#) / [AMI Server 2](#) and setup the below configuration:

Server 1

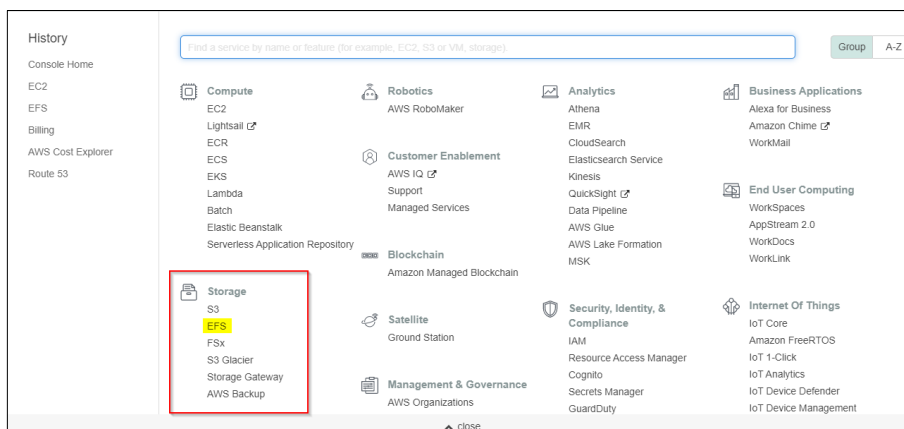
```
#sudo su
#yum update -y (To update the server packages)
#mkdir efs (create a "efs" directory to synch with EFS)
```

Server 2

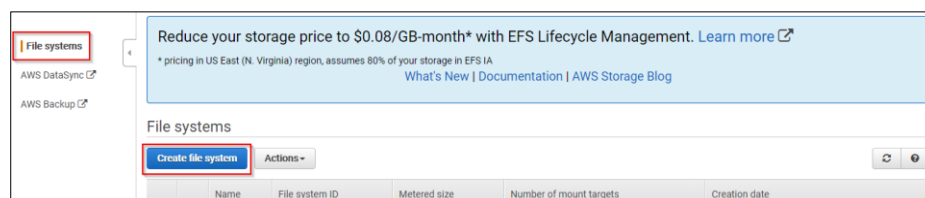
```
#sudo su
#yum update -y (To update the server packages)
#mkdir efs (create a "efs" directory to synch with EFS)
```

Step 3: Click on EFS service in AWS to Create Your Amazon EFS File System.

(Please note that AWS EFS service is chargeable for \$0.08/GB-Month*)



Step 4: Click on "Create File systems"



Step 5: Configure file system access click “Next”

Select the check boxes for all of the **Availability Zones**. Make sure that they all have the default subnets, automatic IP addresses, and the default security groups chosen. These are your mount targets. For more information, see [Creating Mount Targets](#).

Make sure that you have to select the right Security Group where you would need to mount the EFS on that particular EC2 system.

VPC vpc-41909226 (default) ⓘ

Manage mount targets

Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

	Availability Zone	Subnet	IP address	Security groups	Mount target state
⊗	ap-southeast-1a	subnet-4afb0d2c (default)	172.31.22.72	<div>sg-092e7d1ed126768b6 - SG-1 ✕</div> <div>sg-0c288773 - default ✕</div>	Available
⊗	ap-southeast-1b	subnet-c141a689 (default)	172.31.37.235	<div>sg-092e7d1ed126768b6 - SG-1 ✕</div> <div>sg-0c288773 - default ✕</div>	Available
⊗	ap-southeast-1c	subnet-85cb66dc (default)	172.31.0.128	<div>sg-092e7d1ed126768b6 - SG-1 ✕</div> <div>sg-0c288773 - default ✕</div>	Available

Make Sure you have added same security group that you created for your EC2 system

Cancel Save

Step 6: Configure optional settings just click “Next”

Configure optional settings

Add tags

You can add tags to describe your file system. A tag consists of a case-sensitive key-value pair. (For example, you can define a tag with key-value pair with key = Corporate Department and value = Sales and Marketing.) At a minimum, we recommend a tag with key = Name.

Key	Value	Remove
Name	Add New Value	✕
Add New Key		

Enable lifecycle management **NEW**

Automatically save up to 92% on your EFS bill as your access patterns change by enabling **Lifecycle Management** for your file system. Based on the policy you choose, any files in your file system that are not accessed for a period of time will automatically move to the EFS Infrequent Access (EFS IA) storage class. EFS IA provides price/performance that's cost-optimized for files not accessed every day. [Learn more](#)

Lifecycle policy: None

Choose throughput mode

We recommend **Bursting** throughput mode for most file systems. Use **Provisioned** throughput mode for applications that require more throughput than allowed by **Bursting** throughput. [Learn more](#)

☒ Bursting
☐ Provisioned

Choose performance mode

We recommend **General Purpose** performance mode for most file systems. **Max I/O** performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system – it scales to higher levels of aggregate throughput and operations per second with a tradeoff of slightly higher latencies for file operations.

☒ General Purpose
☐ Max I/O

Enable encryption

If you enable encryption for your file system, all data on your file system will be encrypted at rest. You can select a KMS key from your account to protect your file system, or you can provide the ARN of a key from a different account. Encryption of data at rest can only be enabled during file system creation. Encryption of data in transit is configured when mounting your file system. [Learn more](#)

☐ Enable encryption of data at rest

Cancel Previous **Next Step**

Step 7: Review the file system configuration, and then choose “Create File System.”

Review and create

Review the configuration below before proceeding to create your file system.

File system access

VPC	Availability Zone	Subnet	IP address	Security groups
vpc-92a5e5e8 (default)	us-east-1a	subnet-c0d8ecab (default)	Automatic	sg-0639eb57 - default
	us-east-1b	subnet-9959bb7 (default)	Automatic	sg-0639eb57 - default
	us-east-1c	subnet-f9f58b2 (default)	Automatic	sg-0639eb57 - default
	us-east-1d	subnet-bf586e3 (default)	Automatic	sg-0639eb57 - default
	us-east-1e	subnet-6770459 (default)	Automatic	sg-0639eb57 - default
	us-east-1f	subnet-9e53bd90 (default)	Automatic	sg-0639eb57 - default

Optional settings

Tags: No tags added

Performance mode: General Purpose

Throughput mode: Bursting

Encrypted: No

Lifecycle policy: None

Cancel Previous **Create File System**

Step 8: Click on **"File System ID"** and navigate settings for mount the Amazon EFS file system.

File systems

Create file system Actions

Name	File system ID	Metered size	Number of mount targets	Creation date
EFS-EVA	fs-c5601984	12.0 KiB	3	11/08/2019, 10:32:29 UTC

Other details **Tags** [Manage tags](#)

Owner ID: 066850415851

File system state: Available

Performance mode: General Purpose

Throughput mode: Bursting

Encrypted: No

Lifecycle policy: None

File system access [Manage file system access](#)

DNS name: fs-c5601984.efs.ap-southeast-1.amazonaws.com

Annotations:

- This option work for Local VPC EC2 System Only (Same Region and availability Zone)
- This option work for Across VPC peering EC2 System Only (across Region and availability Zone)
- This option work for On-premises to cloud EC2 System Only (On-Premises to EC2 cloud)

Mount instructions:

- Amazon EC2 mount instructions (from local VPC)
- Amazon EC2 mount instructions (across VPC peering connection)
- On-premises mount instructions

Step 9: Please connect to your Amazon EC2 instance **Server 1** and **server 2** and mount the Amazon EFS file system, **copy the command** and **mount target** for your Amazon EFS file system in your both the servers.

Amazon EC2 mount instructions (from local VPC)

To set up your EC2 instance:

- Using the [Amazon EC2 console](#), associate your EC2 instance with a VPC security group that enables access to your mount target. For example, if you assigned the 'default' security group to your mount target, you should assign the 'default' security group to your EC2 instance. [Learn more](#)
- Open an SSH client and connect to your EC2 instance. (Find out [how to connect](#).)
- If you're using an Amazon Linux EC2 instance, install the EFS mount helper with the following command:


```
sudo yum install -y amazon-efs-utils
```

 You can still use the EFS mount helper if you're not using an Amazon Linux instance. [Learn more](#)

If you're not using the EFS mount helper, install the NFS client on your EC2 instance:

- On a Red Hat Enterprise Linux or SUSE Linux instance, use this command:


```
sudo yum install -y nfs-utils
```
- On an Ubuntu instance, use this command:


```
sudo apt-get install nfs-common
```

Mounting your file system

- Open an SSH client and connect to your EC2 instance. (Find out [how to connect](#)).
- Create a new directory on your EC2 instance, such as 'efs'.


```
sudo mkdir efs
```
- Mount your file system with a method listed following. If you need encryption of data in transit, use the EFS mount helper and the TLS mount option. [Mounting considerations](#)
 - Using the EFS mount helper:


```
sudo mount -t efs fs-3f60afbe:/ efs
```
 - Using the EFS mount helper and the TLS mount option:


```
sudo mount -t efs -o tls fs-3f60afbe:/ efs
```
 - Using the NFS client:


```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,hard,timeo=600,retrans=2,noresvport fs-3f60afbe.efs.us-east-1.amazonaws.com:/ efs
```

Annotations:

- The local directory that you created in your EC2 Server 1 and EC2 Server 2 should be the same. Example: efs

If you can't to connect, see our [troubleshooting documentation](#).

1 **2** **Close**

To update the /etc/fstab file in your EC2 instance

1. Connect to your EC2 instance, and open the `/etc/fstab` file in an editor.
2. Add the following line to the `/etc/fstab` file.

```
fs-12345678:/mnt/efs efs defaults,_netdev 0 0
```

Warning

Use the `_netdev` option, used to identify network file systems, when mounting your file system automatically. If `_netdev` is missing, your EC2 instance might stop responding. This result is because network file systems need to be initialized after the compute instance starts its networking. For more information, see [Automatic Mounting Fails and the Instance Is Unresponsive](#).

3. Save the changes to the file.

Your EC2 instance is now configured to mount the EFS file system whenever it restarts.

Note

If your Amazon EC2 instance needs to start regardless of the status of your mounted Amazon EFS file system, add the `nofail` option to your file system's entry in your `/etc/fstab` file.

The line of code you added to the `/etc/fstab` file does the following.

Field	Description
<code>fs-12345678:/</code>	The ID for your Amazon EFS file system. You can get this ID from the console or programmatically from the CLI or an AWS SDK.
<code>/mnt/efs</code>	The mount point for the EFS file system on your EC2 instance.
<code>efs</code>	The type of file system. When you're using the mount helper, this type is always <code>efs</code> .
<code>mount options</code>	Mount options for the file system. This is a comma-separated list of the following options: <ul style="list-style-type: none">• <code>defaults</code> – This value tells the operating system to use the default mount options, which you can list after the file system has been mounted by viewing the output of the <code>mount</code> command.• <code>_netdev</code> – The value tells the operating system that the file system resides on a device that requires network access. This option prevents the instance from mounting the file system until the network has been enabled on the client.• You can replace <code>defaults</code> here with <code>tls</code> to enable encryption of data in transit.
<code>0</code>	A nonzero value indicates that the file system should be backed up by <code>dump</code> . For EFS, this value should be <code>0</code> .
<code>0</code>	The order in which <code>fsck</code> checks file systems at boot. For EFS file systems, this value should be <code>0</code> to indicate that <code>fsck</code> should not run at startup.

Step 10: AMI Server 1

```
[root@ip-172-31-85-194 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-3f60afbe.efs.us-east-1.amazonaws.com:/ efs
```

```
Complete!
[root@ip-172-31-85-194 ec2-user]# mkdir efs
[root@ip-172-31-85-194 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-3f60afbe.efs.us-east-1.amazonaws.com:/ efs
```

Step 11: AMI Server 2

```
[root@ip-172-31-89-235 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-3f60afbe.efs.us-east-1.amazonaws.com:/ efs
```

Now our EC2 Server 1 and Server 2 instances is now configured to mount the EFS file system.

Step 12: Now you can create a directory or two files on Server 1 under “efs” directory

```
#touch abc pqr
#mkdir ethans
```

Step 13: Now you can check on EC2 Server 2 under “efs” directory If directory and files have been synced automatically from Server 1

FYI – See the output

Server 1

```
drwxr-xr-x 3 root root 6144 Oct 16 04:15 efs
[root@ip-172-31-85-194 ec2-user]# cd efs/
[root@ip-172-31-85-194 efs]# ll
total 16
-rw-r--r-- 1 root root 0 Oct 16 04:14 abc
drwxr-xr-x 2 root root 6144 Oct 16 04:15 ethans
-rw-r--r-- 1 root root 0 Oct 16 04:14 pqr
-rw-r--r-- 1 root root 14 Oct 16 04:15 temp.txt
[root@ip-172-31-85-194 efs]#
```

Server2

```
drwxr-xr-x 3 root root 6144 Oct 16 04:15 efs
[ec2-user@ip-172-31-84-96 ~]$ cd efs/
[ec2-user@ip-172-31-84-96 efs]$ ll
total 16
-rw-r--r-- 1 root root 0 Oct 16 04:14 abc
drwxr-xr-x 2 root root 6144 Oct 16 04:15 ethans
-rw-r--r-- 1 root root 0 Oct 16 04:14 pqr
-rw-r--r-- 1 root root 14 Oct 16 04:15 temp.txt
[ec2-user@ip-172-31-84-96 efs]$
```

Done: Transfer Files to Amazon EFS Using AWS DataSync Successfully

To transfer files from a source location to a destination location using AWS DataSync.

Now that we have created a functioning Amazon EFS file system, we can use AWS DataSync to transfer files from an existing file system to Amazon EFS. AWS DataSync is a data transfer service that simplifies, automates, and accelerates moving and replicating data between on-premises storage systems and AWS storage services over the internet or AWS Direct Connect. AWS DataSync can transfer our file data, and also file system metadata such as ownership, time stamps, and access permissions.

Done

Amazon EC2 mount instructions (across a VPC peering connection)

You can mount an EFS file system on an Amazon EC2 instance over a VPC peering connection. [Learn more](#)

You can use an Amazon EFS file system in one VPC based on the Amazon VPC service at a time. That is, you create mount targets in a VPC for your file system, and use those mount targets to provide access to the file system.

SIDE A - ACCOUNT ID: 786125941515 (EFS SYSTEM) SINGAPORE REGION VPC ID: VPC-05c2e3005d9c99c1c

SIDE B - ACCOUNT ID: 268776714427 (EC2 SYSTEM) SINGAPORE REGION

1. Create a new VPC Side A ESF System. IPv4 CIDR (10.0.0.0/16)

VPCs > Create VPC

Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an Amazon-provided IPv6 CIDR block with the VPC.

Name tag

VPC EFS

IPv4 CIDR block*

10.0.0.0/16

IPv6 CIDR block

☒ No IPv6 CIDR Block

☐ Amazon provided IPv6 CIDR block

Tenancy

Default

* Required

Cancel Create

2. Create subnet and associated with VPC that I created above Side A.

Subnets > Create subnet

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag

VPC EFS Subnet

VPC*

vpc-05c2e3005d9c99c1c

VPC CIDRs	CIDR	Status	Status Reason
	10.0.0.0/16	associated	

Availability Zone

ap-southeast-1a

IPv4 CIDR block*

10.0.0.0/24

* Required

Cancel Create

3. Go to side B and create a VPC Peering request as below: Provide A/C ID & VPC ID A Side.

Peering Connections > Create Peering Connection

Create Peering Connection

Peering connection name tag

PCX A

Select a local VPC to peer with

VPC (Requester)*

vpc-6680ad01

CIDRs	CIDR	Status	Status Reason
	172.31.0.0/16	associated	

Select another VPC to peer with

Account

☐ My account

☒ Another account

Account ID*

786125941515

Region

☒ This region (ap-southeast-1)

☐ Another Region

VPC (Acceptor)*

vpc-05c2e3005d9c99c1d

* Required

Cancel Create Peering Connection

4. Go To Side A 'Peering Connection and accept the request.

5. Go To Side A 'Route Table' and create a Route as below and edit the route as below

Route Tables > Create route table

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag:

VPC*:

* Required

Cancel Create

Route Tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
172.31.0.0/16	pcx-0df1f26a2cd726d6b	active	No

Add route

* Required

Cancel Save routes

Route Tables > Edit subnet associations

Edit subnet associations

Route table: rtb-0fe58ca2ac23b9839 (VPC EFS RT)

Associated subnets:

Filter by attributes or search by keyword

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-08d22f02972082496 VPC EFS ...	10.0.0.0/24	-	rtb-0fe58ca2ac23b9839

Now Go To Side B and add the 'Route table' as below (We need to add the destination IP for side A) i.e 10.0.0.0/16

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Name: rtb-efda6a89

Route Table ID: rtb-efda6a89

Explicit subnet association: -

Main: Yes

VPC ID: vpc-6680ad01

Route Table: rtb-efda6a89

Summary Routes Subnet Associations Route Propagation Tags

Edit routes

View: All routes

Destination	Target	Status
172.31.0.0/16	local	active
0.0.0.0/0	igw-6b064a0f	active
10.0.0.0/16	pcx-0df1f26a2cd726d6b	active

Now Go to Side A and Create a new EFS adding under new VPC that we created for EFS. Add Security group.

File systems

AWS DataSync

AWS Backup

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system by using a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

VPC: vpc-05c2e3005dcc99c1c - VPC EFS

Manage mount targets

Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

Availability Zone	Subnet	IP address	Security groups	Mount target state
ap-southeast-1a	subnet-08d22f02972082496 - VPC EFS - Public Subnet	10.0.0.34	sg-04ded12bd3c0a8e37 - EFS_NFS sg-0a0724cabfae9120a - default	Available
ap-southeast-1b	No subnets in this Availability Zone			
ap-southeast-1c	No subnets in this Availability Zone			

Now access EFS file sys from Side B EC2 system :

Access my EC2 Instance **Side B**

Create a new directory on your EC2 instance, such as "appserver_efs"

Mount your file system.

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2  
10.0.0.34:/ appserver_efs
```

```
root@ip-172-31-17-127:/# sudo mount -t nfs4 -o  
nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2 10.0.0.34:/ appserver_efs
```

Done

How to set up auto rsync backups in AWS EC2 using ssh.

Rsync (Remote Sync) is a most commonly used command for copying and synchronizing files and directories remotely as well as locally in Linux/Unix systems. With the help of rsync command you can copy and synchronize your data remotely and locally across directories, across disks and networks, perform data backups and mirroring between two Linux machines.

Step 1: Create your EC2 two AMI Linux instances to sync your Data One server to another server Online with rsync command.

Step 2: Server 1 configuration with the steps below

```
#rm -rf .ssh  
#ssh-keygen -t rsa  
#cd .ssh/  
#vi id_rsa.pub (Copy the ssh code and paste it over at the EC2 server 2)  
#cd  
#ssh 172.31.95.224 (Private IP of your server 2 Instance)  
(Check the ssh permitted server 1 to access the server 2 using ssh putty)  
If yes, then go ahead.  
#Exit  
#mkdir /mydir  
#cd /mydir/  
#touch abc pqr  
#ls -lart
```

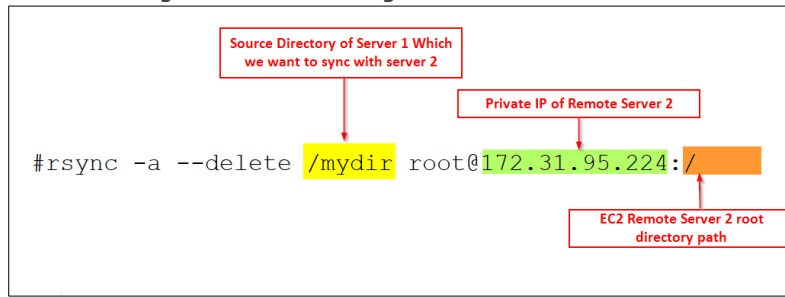
Step 3: Server 2 configuration with the steps below

```
#rm -rf .ssh  
#ssh-keygen -t rsa  
#cd .ssh/  
#vi authorized_keys (Paste the ssh code here which you copied from EC2 server 1)  
Save and Exit  
#chmod 600 authorized_keys  
#mkdir /mydir  
#ls -lart
```

Step 4: Use the following command on Server 1 and will sync files on Server 1 to Server 2.

```
#rsync -a --delete /mydir root@172.31.95.224:/
```


Basic syntax of rsync command



Some common options used with rsync commands

- v : verbose
- r : copies data recursively (but don't preserve timestamps and permission while transferring data)
- a : archive mode, archive mode allows copying files recursively and it also preserves symbolic links, file permissions, user & group ownerships and timestamps
- z : compress file data
- h : human-readable, output numbers in a human-readable format

Step 5: You can schedule "crontab" for auto sync

If you need the backup script run at a specific time daily at 7am, you'll have to manually create a cron job by issuing the command `crontab -e` and then adding a line such as:

```
00 07 * * * rsync -a --delete /mydir root@172.31.95.224:/. 
```

Output: Now you can check at **Server 2** under "mydir" directory If files and directory have been synced automatically from **Server 1**

Done!