

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

**Arava Jayaram
REG. NO. 39120045**



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

**SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

APRIL 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Arava Jayaram** (Reg. No: 39120045) who carried out the project entitled “**Implementing EFS with Ec2 Instance**” under my supervision from Feb 2022 to April 2022.

Internal Guide

Dr.V.Nirmalarani

=

Head of the Department

Dr.R.Subhashini M.E.,Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Arava Jayaram** hereby declare that the project report entitled **Implementing an EFS with Ec2 Instance** done by me under the guidance of is **Dr.V.Nirmalarani** submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Information Technology

DATE: 11/04/2022

PLACE: Chennai.

SIGNATURE OF THE CANDIDATE:

Arava Jayaram

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean, School of Computing, Dr.Subhashini M.E., Ph.D. and Head of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide for his valuable **DR.V.Nirmalarani** guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



Award of Completion

Arava Jayaram

Has Successfully Completed

Star Cloud Computing

Star Authorized Delivery Partner

Advantage Pro

You can verify authenticity of this certificate by visiting
<http://www.starcertification.org/Verification/Participation>

10-04-2022

DATE

KxvBoBRAAF

CERTIFICATE NUMBER



DONALD WILLIAM
DIRECTOR
GLOBAL CERTIFICATIONS



TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	8
1	INTRODUCTION	9
2	AIM AND SCOPE OF THE PRESENT INVESTIGATION	10
3	EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED	10
3.1	Cloud computing	10
3.2	Steps Required	13
4	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS PREDICTION AND RESULTS PERFORMANCE ANALYSIS	18
5	SUMMARY AND CONCLUSIONS	22

LIST OF PICTURES

FIGURE NO.	PAGE NO
Figure 1	11
Figure 2	19
Figure 3	19
Figure 4	20
Figure 5	20
Figure 6	21
Figure 7	21

ABSTRACT

Amazon Elastic File System (Amazon EFS) provides a simple, serverless, set-and-forget elastic 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. Amazon EFS has a simple web services interface that allows you to create and configure file systems quickly and easily. The service manages all the file storage infrastructure for you, meaning that you can avoid the complexity of deploying, patching, and maintaining complex file system configurations.

Amazon EFS supports the Network File System version 4 (NFSv4.1 and NFSv4.0) protocol, so the applications and tools that you use today work seamlessly with Amazon EFS. Multiple compute instances, including Amazon EC2, Amazon ECS, and AWS Lambda, can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one compute instance or server.

CHAPTER 1: INTRODUCTION

This project focuses on Creating a EFS with EC2 instance using amazon webservises. we can actually create a EFS and can access it from different instances. As by name it is elastic its storage was allocated according to run time. This way efs was very usable and this is the specialty of the EFS.

CHAPTER 2: AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 AIM: Explain various storage option available in Google cloud platform , implement Storage solution .

2.2 SCOPE OF THE PROJCT : Finding the storage solution of Google cloud platform and hosting a file in google cloud.

CHAPTER 3: EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED

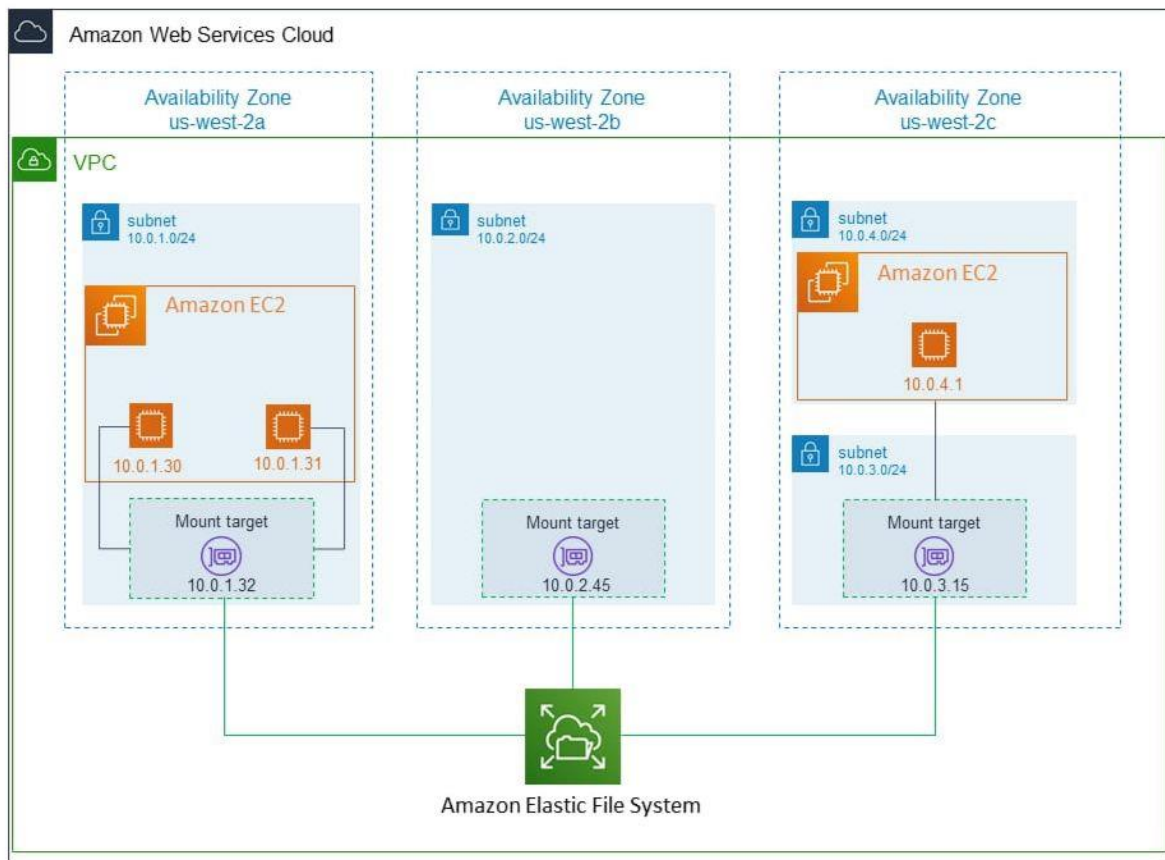
3.1 Cloud Computing :

1) Cloud computing is on-demand access, via the internet, to computing resources—applications, servers(physical servers and virtual servers), data storage, development tools, networking capabilities, and more—hosted at a remote datacenter managed by a cloud services provider (or CSP). The CSP makes these resources available for a monthly subscription fee or bills them according to usage. Compared to traditional onpremises IT, and depending on the cloud services you select,

2) cloud computing helps do the following: i) Lower IT costs ii) Improve agility and time-to-value iii) Scale more easily and cost-effectively

3)Types of cloud : i) Public Cloud ii) Private Cloud iii) Hybrid Cloud iv) Multicloud and hybrid multicloud

ELASTIC FILE SYSTEM



- ❖ EFS is a file-level, fully managed, storage provided by AWS that can be accessed by multiple EC2 instances concurrently. Just like the AWS EBS, EFS is specially designed for high throughput and low latency applications.
- ❖ The aforementioned list, EFS falls under the file storage category. EFS is a file-level, fully managed, storage provided by AWS that can be accessed by multiple EC2 instances concurrently. Just like the AWS EBS, EFS is specially designed for high throughput and low latency applications.
- ❖ Connect To EFS from your EC2 servers. Primarily there are 2 methods of connecting to EFS from EC2 servers
- ❖ Linux NFS Client: This is the old traditional method of connecting to file systems.
- ❖ EFS Mount Helper: This is the AWS recommended and simpler solution to connect to EFS.

METHODOLOGY

Implementing EFS with EC2 Instance

- 1) Create a AWS account with a Gmail id.
- 2) After successful creation of AWS account create 2 Instances.
- 3) Click and open “Launch Instance” to create the instance .
- 4) Select an Amazon Linux 2 64-bit
- 5) Fill all the mandatory fields .
- 6) After successful creation of Instance. Create another instance with same security group .
- 7) Now, create a EFS with a name and make it available.
- 8) Now open command prompt and connect to your one of the EC2 instance .
- 9) Now download the necessary EFS utils for the mounting process
- 10) Now open your EFS and Click on attach button and checkout the mounting command
- 11) And using that command mount that instance
- 12) Now, after mounting create a file and put some content into it.
- 13) Now open another command prompt and connect another instance to it
- 14) Now mount the same EFS into it and
- 15) Now create some file in it and save that
- 16) Now open up your another command prompt and after mounting the same efs in it now u can look at the file which we created earlier in the command prompt
- 17) In this way EFS can be used with Ec2 instance and a file can be used my multiple machines

Figure 1. A reference of how the train and test split works

3.2 Steps Required:

Create an EFS file system using Amazon EFS Quick Create

You can create an EFS file system and mount it to your instance at the time of launch using the Amazon EFS Quick Create feature of the Instance Launch Wizard.

When you create an EFS file system using EFS Quick Create, the file system is created with the following service recommended settings:

- Automatic backups turned on. For more information, see [Using AWS Backup with Amazon EFS](#) in the *Amazon Elastic File System User Guide*.
- Mount targets in each default subnet in the selected VPC, using the VPC's default security group. For more information, see [Managing file system network accessibility](#) in the *Amazon Elastic File System User Guide*.
- General Purpose performance mode. For more information, see [Performance Modes](#) in the *Amazon Elastic File System User Guide*.
- Bursting throughput mode. For more information, see [Throughput Modes](#) in the *Amazon Elastic File System User Guide*.
- Encryption of data at rest enabled using your default key for Amazon EFS (aws/elasticfilesystem). For more information, see [Encrypting Data at Rest](#) in the *Amazon Elastic File System User Guide*.
- Amazon EFS lifecycle management enabled with a 30-day policy. For more information, see [EFS lifecycle management](#) in the *Amazon Elastic File System User Guide*.

To create an EFS file system using Amazon EFS Quick Create

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. On the **Choose an AMI** page, choose a Linux AMI.
4. On the **Choose an Instance Type** page, select an instance type and then choose **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, for **File systems**, choose **Create new file system**, enter a name for the new file system, and then choose **Create**.

To enable access to the file system, the following security groups are automatically created and attached to the instance and the mount targets of the file system.

- **Instance security group**—Includes no inbound rules and an outbound rule that allows traffic over the NFS 2049 port.
- **File system mount targets security group**—Includes an inbound rule that allows traffic over the NFS 2049 port from the instance security group (described above), and an outbound rule that allows traffic over the NFS 2049 port.

You can also choose to manually create and attach the security groups. To do this, clear **Automatically create and attach the required security groups**.

Configure the remaining settings as needed and choose **Next: Add Storage**.

6. On the **Add Storage** page, specify the volumes to attach to the instances, in addition to the volumes specified by the AMI (such as the root device volume). Ensure that you provision enough storage for the Nvidia CUDA Toolkit. Then choose **Next: Add Tags**.
7. On the **Add Tags** page, specify a tag that you can use to identify the temporary instance, and then choose **Next: Configure Security Group**.
8. On the **Configure Security Group** page, review the security groups and then choose **Review and Launch**.
9. On the **Review Instance Launch** page, review the settings, and then choose **Launch** to choose a key pair and to launch your instance.

Create an EFS file system and mount it to your instance

In this tutorial, you create an EFS file system and two Linux instances that can share data using the file system.

Tasks

- [Prerequisites](#)
- [Step 1: Create an EFS file system](#)
- [Step 2: Mount the file system](#)
- [Step 3: Test the file system](#)
- [Step 4: Clean up](#)

Prerequisites

- Create a security group (for example, efs-sg) to associate with the EC2 instances and EFS mount target, and add the following rules:
 - Allow inbound SSH connections to the EC2 instances from your computer (the source is the CIDR block for your network).
 - Allow inbound NFS connections to the file system via the EFS mount target from the EC2 instances that are associated with this security group (the source is the security group itself). For more information, see [Amazon EFS rules](#), and [Creating security Groups](#) in the *Amazon Elastic File System User Guide*.
- Create a key pair. You must specify a key pair when you configure your instances or you can't connect to them. For more information, see [Create a key pair](#).

Step 1: Create an EFS file system

Amazon EFS enables you to create a file system that multiple instances can mount and access at the same time. For more information, see [Creating Resources for Amazon EFS](#) in the *Amazon Elastic File System User Guide*.

To create a file system

1. Open the Amazon Elastic File System console at <https://console.aws.amazon.com/efs/>.
2. Choose **Create file system**.
3. (Optional) For **Name**, enter a name for the file system. This creates a tag with **Name** as the key and the name of the file system as the value.
4. For **Virtual Private Cloud (VPC)**, select the VPC to use for your instances.
5. Choose **Create**.
6. After the file system is created, note the file system ID. It is used later in this tutorial.
7. Choose the file system ID.
8. On the file systems page, choose **Network, Manage**. View the mount targets that Amazon EFS creates in each Availability Zone in the Region in which your VPC resides. For each Availability Zone for your instances, ensure that the value for **Security groups** is the security group that you created in [Prerequisites](#).
9. Choose **Save**.

Step 2: Mount the file system

Use the following procedure to launch two `t2.micro` instances. Note that T2 instances must be launched in a subnet. You can use a default VPC or a nondefault VPC.

Note

There are other ways that you can mount the volume (for example, on an already running instance). For more information, see [Mounting File Systems](#) in the *Amazon Elastic File System User Guide*.

To launch two instances and mount an EFS file system

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance**.
3. For **Step 1: Choose an Amazon Machine Image (AMI)**, select an Amazon Linux AMI.
4. For **Step 2: Choose an Instance Type**, keep the default instance type, `t2.micro`, and choose **Next: Configure Instance Details**.
5. For **Step 3: Configure Instance Details**, do the following:
 - . For **Number of instances**, enter **2**.
 - a. [Default VPC] If you have a default VPC, it is the default value for **Network**. Keep the default VPC and the default value for **Subnet** to use the default subnet in the Availability Zone that Amazon EC2 chooses for your instances.

[Nondefault VPC] Select your VPC for **Network**, and a public subnet from **Subnet**.
 - b. [Nondefault VPC] For **Auto-assign Public IP**, choose **Enable**. Otherwise, your instances do not get public IP addresses or public DNS names.
 - c. For **File systems**, choose **Add file system**. Ensure that the value matches the file system ID that you created in [Step 1: Create an EFS file system](#). The path shown next to the file system ID is the mount point that the instance will use, which you can change. Under **Advanced Details**, the **User data** is automatically generated, and includes the commands needed to mount the file system.
 - d. Advance to Step 6 of the wizard.
6. On the **Configure Security Group** page, choose **Select an existing security group** and select the security group that you created in [Prerequisites](#). Then choose **Review and Launch**.
7. On the **Review Instance Launch** page, choose **Launch**.

8. In the **Select an existing key pair or create a new key pair** dialog box, select **Choose an existing key pair** and choose your key pair. Select the acknowledgment check box, and choose **Launch Instances**.
9. In the navigation pane, choose **Instances** to see the status of your instances. Initially, their status is pending. After the status changes to running, your instances are ready for use.

Your instance is now configured to mount the Amazon EFS file system at launch and whenever it's rebooted.

Step 3: Test the file system

You can connect to your instances and verify that the file system is mounted to the directory that you specified (for example, /mnt/efs).

To verify that the file system is mounted

1. Connect to your instances. For more information, see [Connect to your Linux instance](#).
2. From the terminal window for each instance, run the **df -T** command to verify that the EFS file system is mounted.

3. \$ **df -T**

4. Filesystem	Type	1K-blocks	Used
Available	Use%	Mounted on	
5. /dev/xvda1	ext4	8123812	1949800
6073764	25%	/	
6. devtmpfs	devtmpfs	4078468	56
4078412	1%	/dev	
7. tmpfs	tmpfs	4089312	0
4089312	0%	/dev/shm	

<i>efs-dns</i>	nfs4	9007199254740992	0
9007199254740992	0%	<i>/mnt/efs</i>	

Note that the name of the file system, shown in the example output as *efs-dns*, has the following form.

```
file-system-id.efs.aws-region.amazonaws.com: /
```

8. (Optional) Create a file in the file system from one instance, and then verify that you can view the file from the other instance.

- From the first instance, run the following command to create the file.

```
$ sudo touch /mnt/efs/test-file.txt
```

- a. From the second instance, run the following command to view the file.

```
b. $ ls /mnt/efs
```

```
test-file.txt
```

Step 4: Clean up

When you are finished with this tutorial, you can terminate the instances and delete the file system.

To terminate the instances

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Instances**.
3. Select the instances to terminate.
4. Choose **Instance state, Terminate instance**.
5. Choose **Terminate** when prompted for confirmation.

To delete the file system

1. Open the Amazon Elastic File System console at <https://console.aws.amazon.com/efs/>.
2. Select the file system to delete.
3. Choose **Actions, Delete file system**.
4. When prompted for confirmation, enter the file system ID and choose **Delete file system**.

CHAPTER 4: RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

Creating an efs:

Introducing EFS Replication
Keep an up-to-date copy of your file system in a region or availability zone of your choice.
[What's new](#) | [Documentation](#) | [AWS Storage Blog](#)

File systems (4)

Name	File system ID	Encryption	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA	Provisioned Throughput (MiB/s)	File system state	Creation time
myefs	fs-0360bc5d706c9cd9a	Unencrypted	18.00 KiB	18.00 KiB	0 Bytes	-	Available	Mon, 2022-13:11 GMT
efsdfs	fs-01cb9ef5c8e136e59	Encrypted	6.00 KiB	6.00 KiB	0 Bytes	-	Available	Mon, 2022-09:31 GMT

Creating an instances:

Instances (7)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 address
-	i-0443adcf15123af18	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-54-81-101
-	i-04ec4753eb01d5264	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-18-212-16
-	i-01c027c2114b186d7	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-18-206-22
-	i-09246c76c269a4154	Running	t2.micro	2/2 checks passed	No alarms	us-east-1b	ec2-3-93-239
-	i-070509e5518f5d4e8	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-35-172-22

Mounting efs with instance1:

```
root@ip-172-31-16-7:/home/ec2-user/efs -----Install 1 Package (+1 Dependent package)

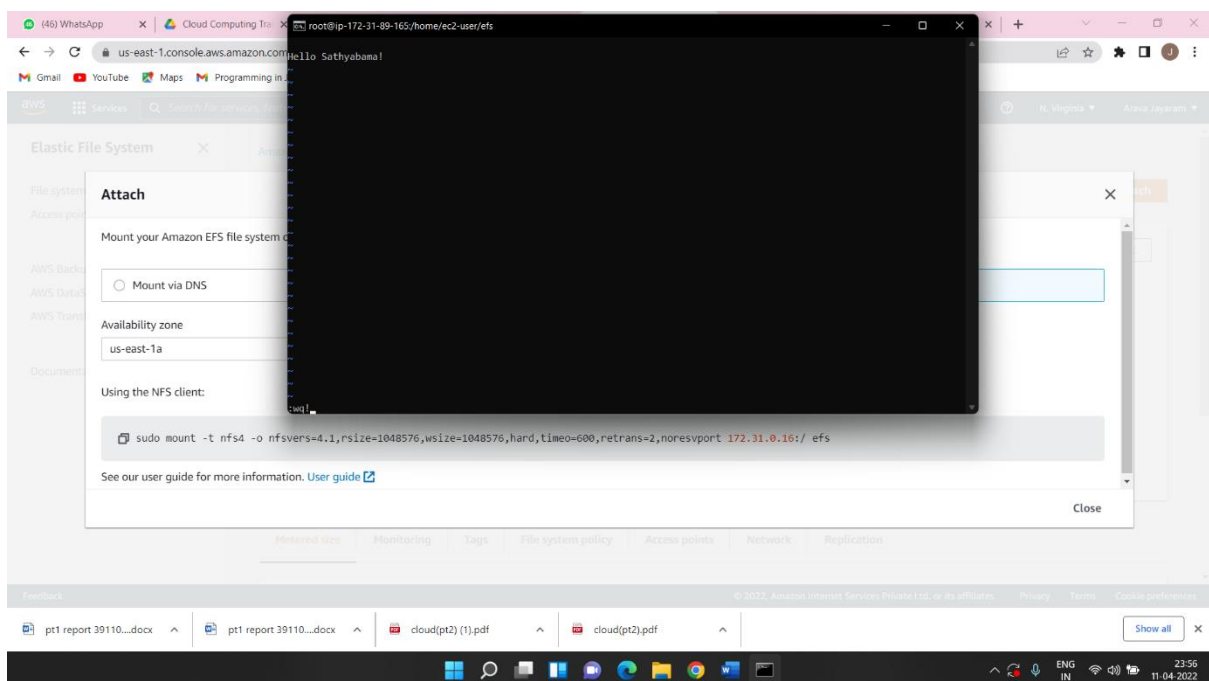
total download size: 197 k
Installed size: 490 k
Downloading packages:
(1/2): amazon-efs-utils-1.31.3-1.amzn2.noarch.rpm | 48 kB 00:00:00
(2/2): stunnel-4.56-6.amzn2.0.3.x86_64.rpm | 149 kB 00:00:00
-----Total
1.4
KB/s | 197 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : stunnel-4.56-6.amzn2.0.3.x86_64 1/2
Installing : amazon-efs-utils-1.31.3-1.amzn2.noarch 2/2
Verifying : stunnel-4.56-6.amzn2.0.3.x86_64 1/2
Verifying : amazon-efs-utils-1.31.3-1.amzn2.noarch 2/2

Installed:
amazon-efs-utils.noarch 0:1.31.3-1.amzn2

Dependency Installed:
stunnel.x86_64 0:4.56-6.amzn2.0.3

Complete!
[ec2-user@ip-172-31-16-7 ~]$ sudo su
[root@ip-172-31-16-7 ec2-user]# mkdir efs
[root@ip-172-31-16-7 ec2-user]# ls
efs
[root@ip-172-31-16-7 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport 172.31.0.16:/ efs
mount.nfs4: an incorrect mount option was specified
[root@ip-172-31-16-7 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsiz=1048576,hard,timeo=600,retrans=2,noresvport 172.31.0.16:/ efs
[root@ip-172-31-16-7 ec2-user]# cd efs
[root@ip-172-31-16-7 efs]# ll
total 12
-rw-r--r-- 1 root root 79 Apr 11 14:06 abc.txt
-rw-r--r-- 1 root root 19 Apr 11 18:26 create.txt
drwx-r-xr-x 2 root root 6144 Apr 11 14:15 hello
[root@ip-172-31-16-7 efs]# cat create.txt
Hello Sathyabama!
[root@ip-172-31-16-7 efs]#
```

Example file:



Mounting efs with instance2:

```

C:\root@ip-172-31-16-7/home/ec2-user/efs
Microsoft Windows [Version 10.0.22000.318]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Jayaram Arava>cd downloads

C:\Users\Jayaram Arava\Downloads>ssh -i efs.pem ec2-user@35.172.227.130
The authenticity of host '35.172.227.130 (35.172.227.130)' can't be established.
ECDSA key fingerprint is SHA256:1VlpruyRyHfghjTzCMz2BHRlHS+TAqM+0BF1070pno.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.172.227.130' (ECDSA) to the list of known hosts.

  ____      )
 /_  \_    ) Amazon Linux 2 AMI
/_  \_

https://aws.amazon.com/amazon-linux-2/
14 package(s) needed for security, out of 17 available
Run "sudo yum update" to apply all updates.
ec2-user@ip-172-31-16-7 ~$ sudo yum install -y amazon-efs-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                     | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package amazon-efs-utils.noarch 0:1.31.3-1.amzn2 will be installed
--> Processing Dependency: stunnel >= 4.56 for package: amazon-efs-utils-1.31.3-1.amzn2.noarch
--> Running transaction check
--> Package stunnel.x86_64 0:4.56-6.amzn2.0.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

===== Package Arch Version Rep
Repository Size
Installing:
amazon-efs-utils noarch 1.31.3-1.amzn2 amzn2-core 48 k
Installing for dependencies:
stunnel x86_64 4.56-6.amzn2.0.3 amzn2-core 149 k

Transaction Summary
=====Install 1 Package (+1 Dependent package)

Total download size: 197 k
Installed size: 490 k
Downloading packages:
(1/2): amazon-efs-utils-1.31.3-1.amzn2.noarch.rpm | 48 kB 00:00:00
(2/2): stunnel-4.56-6.amzn2.0.3.x86_64.rpm | 149 kB 00:00:00
-----Total
MB/s | 197 kB 00:00:00
Running transaction check

```

```

root@ip-172-31-16-7/home/ec2-user/efs
=====Install 1 Package (+1 Dependent package)=====
Total download size: 197 k
Installed size: 490 k
Downloading packages:
(1/2): amazon-efs-utils-1.31.3-1.amzn2.noarch.rpm | 48 kB 00:00:00
(2/2): stunnel-4.56-6.amzn2.0.3.x86_64.rpm | 149 kB 00:00:00
-----Total-----
1.
MB/s | 197 kB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : stunnel-4.56-6.amzn2.0.3.x86_64 1/2
Installing : amazon-efs-utils-1.31.3-1.amzn2.noarch 2/2
Verifying : stunnel-4.56-6.amzn2.0.3.x86_64 1/2
Verifying : amazon-efs-utils-1.31.3-1.amzn2.noarch 2/2
Installed:
amazon-efs-utils.noarch 0:1.31.3-1.amzn2
Dependency Installed:
stunnel.x86_64 0:4.56-6.amzn2.0.3
Complete!
ec2-user@ip-172-31-16-7 ~]$ sudo su
[root@ip-172-31-16-7 ec2-user]# mkdir efs
[root@ip-172-31-16-7 ec2-user]# ls
efs
[root@ip-172-31-16-7 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retr=2,noresvport 172.31.0.16:/ efs
mount.nfs4: an incorrect mount option was specified
[root@ip-172-31-16-7 ec2-user]# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retr=2,noresvport 172.31.0.16:/ efs
[root@ip-172-31-16-7 ec2-user]# cd efs
[root@ip-172-31-16-7 efs]# ll
total 12
-rw-r--r-- 1 root root 79 Apr 11 14:06 abc.txt
-rw-r--r-- 1 root root 19 Apr 11 18:26 create.txt
drwxr-xr-x 2 root root 64M Apr 11 14:15 helloworld
[root@ip-172-31-16-7 efs]# cat create.txt
Hello Sathyabama!
[root@ip-172-31-16-7 efs]#

```

CHAPTER 5:Summary And Conclusion:

In this project exploration of a lot of AWS services had been done. created a EFS in where we located our create.text file which was created in instance 1 can be viewed as well as edited by instance2. These are done with the help of AWS services such as EFS and Ec2. This was the speciality of EFS which can be operated by multiple systems at the same time.

THE END

Presented By :

Arava Jayaram