**Knowledge Transfer( Automating EFS Mount By FSTAB )**

**Dated : 17/08/2023**

**\*\*Challenge :\*\***

Hey Geeks, Recently while working on EFS I have come to know that the mount of the EFS is removed once the instance is rebooted, So Today we are going to automate the process of mounting an Amazon Elastic File System (EFS) onto Amazon Elastic Compute Cloud (EC2) instances which will automatically mount it whenever the instance is rebooted.

==============================================================================

**\*\*Task :\*\***

Ticket: Create a Automation for EC2 Instance

==============================================================================

**\*\*Solution :\*\***

WOW !!! I have completed the challenge using the FSTAB file which ihav open in Vi in which I have given some instruction of NFS mount and change it with the directory

==============================================================================

**\*\*Pre-requisites:\*\***

• Have a good knowledge about EFS

• Knowing the knowledge about EC2

• Have knowledge about NFS

==============================================================================

**\*\*Objective:\*\***

The objective is to mounting an Amazon Elastic File System (EFS) onto multiple Amazon Elastic Compute Cloud (EC2) instances

==============================================================================

**\*\*Description:\*\***

In the "Automate Mounting Amazon EFS on EC2 Instances" challenge, participants will embark on a journey to streamline the process of mounting an Amazon Elastic File System (EFS) onto multiple Amazon Elastic Compute Cloud (EC2) instances. This challenge addresses the need for efficient and consistent shared file storage solutions across EC2 instances, fostering collaboration and data integrity.

Participants will start by creating an EFS filesystem using the AWS Management Console, tailoring the configuration to meet their requirements for storage capacity and throughput. With a set of EC2 instances established in the same AWS region, equipped with the necessary permissions and security groups for EFS access, the focus will shift to the heart of the challenge: automation.

==============================================================================

**\*\*Steps:\*\***

# In this we are going to write a script to automate the mount toEC2 whenever it is rebooted…

# In this whole procedure is same as of connecting EFS to EC2 Instance so do that first and then follow tee following steps in EC2 Instance…

# To automate we have to edit the pre generate file named /etc/fstab and we have to backup it so that any mistake happen it should not affect the original file…

# Before making any changes, it's a good practice to create a backup of the original /etc/fstab file in case you need to revert any changes. You can create a backup using the following command…

* sudo cp /etc/fstab /etc/fstab.bak

# Now open the /etc/fstab file for editing using a text editor like nano or vi by…

* sudo nano /etc/fstab

# Next add a new line to the /etc/fstab file that specifies the EFS mount. The format of the line should be as follows:

* <EFS-DNS-Name>:/ <mount-directory> nfs4 nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,\_netdev 0 0

# Replace <EFS-DNS-Name> with the DNS name of your EFS file system and <mount-directory> with the path where you want to mount the EFS file system.

# You will EFS DNS NAME in SNS Topic and mount directory is the path where you want to mount it e.g /var/www/html…

* Now save the editor and exit it

# Next test the mount by typing…

* sudo mount –a

# IF no error shown the mount is done if any error is shown then see whether any spaces or anything is improper…

# Now finally reboot the server by…

* sudo reboot

# Now after reboot again check the instance whether it still mounted or not by typing…

* df –h

# It will show the EFS id at the botton with the directory attached with it.

**#### So Here EFS AutoMounting come to conclusion ####**

==============================================================================

**\*\*Explanation:\*\***

Automating the mounting process of an Amazon Elastic File System (Amazon EFS) onto Amazon Elastic Compute Cloud (Amazon EC2) instances involves developing a systematic approach to simplify and expedite the setup of shared file storage across multiple instances. This automation eliminates the manual effort required to configure each instance individually and ensures consistent access to shared data. Typically achieved through scripting or infrastructure-as-code techniques, the process entails retrieving crucial information like the EFS DNS name and desired mount directory programmatically. With this data, entries are generated for the /etc/fstab file on each EC2 instance, specifying the details needed for mounting the EFS filesystem, such as mount options and timeout settings.

A robust automation solution accounts for changes in the environment, dynamically adapting to different instance counts and potential modifications to EFS configurations. The automation is executed during instance boot-up, seamlessly setting up the EFS mount based on the predefined parameters. Documentation and thorough testing are vital components of the process, ensuring the solution's reliability and functionality across various scenarios. Ultimately, automating the mounting process offers efficiency gains, enhances collaboration, and contributes to a more streamlined and scalable infrastructure in the AWS ecosystem.sudo mount command with specific options, including the EFS ID, availability zone, and mount target. Optionally, automating the mounting process upon instance reboot can be achieved by adding an entry to the /etc/fstab file. Testing the setup involves navigating to the mount directory and creating or copying files to confirm the EFS mount's proper functionality, enabling synchronized access and modifications across multiple EC2 instances.

Through these steps, one can effectively establish a shared file storage system by mounting an Amazon EFS filesystem to an EC2 instance. This approach streamlines collaborative workflows, capitalizes on EFS's scalability and robustness, and maintains data consistency among participating instances. It's essential to adhere to security best practices and configure access permissions appropriately to ensure data integrity and secure usage.