Using Amazon EFS (Elastic File Storage) with EC2 Instances

Context

Students are currently using **capacity block EC2 instances** for their work. Occasionally, these instances are terminated or users lose access, which results in **loss of all unsaved work**. To prevent data loss and provide persistent storage across EC2 instance restarts or replacements, we are sharing instruction on how can we leverage **Amazon EFS** to persist the work being done.

Amazon Elastic File System (EFS) is a fully managed, elastic, shared file system for Linux workloads. It automatically scales storage capacity up or down as files are added or removed, without any need for manual provisioning or resizing. To learn more about EFS, please refer here.

EFS will allow the students to store their data on a **network file system** that can be mounted to their respective EC2 instances, ensuring that their work persists independently of the instance lifecycle.
This document explains how to:

1. Create an Amazon EFS using a CloudFormation template
2. Attach EFS to -
    a. an **existing EC2 instance** manually
    b. a **fresh EC2 instance** via user data script
3. Verify EFS has been successfully mounted

Prerequisites -

1. You have `aws-cli` installed on your local machine
2. You have access to shared AWS account for Stanford.

Step 1 - Creating EFS

A. Login to your AWS account using AWS CLI with temporary credentials

```
export ISENGARD_PRODUCTION_ACCOUNT=false
export AWS_ACCESS_KEY_ID=<YOUR_AWS_ACCESS_KEY_ID>
export AWS_SECRET_ACCESS_KEY=<YOUR_AWS_SECRET_ACCESS_KEY>
export AWS_SESSION_TOKEN=<YOUR_AWS_SESSION_TOKEN>
```

B. Create and save `efs-template.yaml`

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Create an Amazon EFS file system and mount targets
in a VPC

Parameters:
  VpcId:
    Type: AWS::EC2::VPC::Id
    Description: VPC ID where the EFS will be created

  SubnetId:
    Type: AWS::EC2::Subnet::Id
    Description: Subnet ID for the first mount target

  EFSName:
    Type: String
    Description: "Custom name for the EFS file system"

Resources:
  EfsSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: Allow NFS traffic to EFS
      VpcId: !Ref VpcId
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 2049
          ToPort: 2049
          CidrIp: 0.0.0.0/0

  MyEFSFileSystem:
    Type: AWS::EFS::FileSystem
    Properties:
      Encrypted: true
      PerformanceMode: generalPurpose
      ThroughputMode: bursting
      LifecyclePolicies:
        - TransitionToIA: AFTER_30_DAYS
      BackupPolicy:
        Status: ENABLED
      FileSystemTags:
        - Key: Name
          Value: !Ref EFSName

  MountTarget:
    Type: AWS::EFS::MountTarget
    Properties:
      FileSystemId: !Ref MyEFSFileSystem
```

```
      SubnetId: !Ref SubnetId
      SecurityGroups:
        - !Ref EfsSecurityGroup

Outputs:
  FileSystemId:
    Description: The ID of the created EFS
    Value: !Ref MyEFSFileSystem

  EfsSecurityGroupId:
    Description: Security group ID associated with EFS
    Value: !Ref EfsSecurityGroup
```

C. Run the cloud formation template

```
aws cloudformation create-stack \
  --stack-name my-efs-stack \
  --template-body file://efs-template.yaml \
  --parameters ParameterKey=VpcId,ParameterValue=<YOUR_VPC> \
               ParameterKey=SubnetId,ParameterValue=<YOUR_SUBNET
_ID> \
               ParameterKey=EFSName,ParameterValue=<YOUR_EFS_NAM
E> \
  --capabilities CAPABILITY_IAM
```

Note: Make sure to use the VPC and Subnet where your existing instance resides, or where you plan to launch a new instance.

D. Check the status of CloudFormation stack

```
aws cloudformation describe-stacks \
    --stack-name my-efs-stack \
    --query "Stacks[0].StackStatus" \
    --output text
```

E. Verify the EFS is created

```
aws efs describe-file-systems --query
"FileSystems[?Name=='<your_efs_name>']"
```

In the following section, you will learn how to attach the newly created EFS to an EC2 instance. There are two options: attaching the EFS to an existing instance or attaching it to a fresh instance at launch.

Step 2.1 - Attaching EFS to existing instance

A. ssh into your instance

```
ssh -i "<your_key.pem>"
ubuntu@your_instance.compute.amazonaws.com
```

B. Install `efs-utils`

```
# Update packages and install prerequisites for building efs-
utils on Ubuntu
# For Amazon Linux, adjust the commands accordingly to match its
package manager and package names.
apt-get update -y
apt-get install -y git binutils make rustc cargo pkg-
config libssl-dev stunnel4

# Clone and build efs-utils from source for Ubuntu
git clone https://github.com/aws/efs-utils.git
cd efs-utils
./build-deb.sh
apt-get install -y ./build/amazon-efs-utils*deb
```

C. **Create a mount point and mount the EFS**

```
sudo mkdir -p /mnt/efs
sudo chown -R ubuntu:ubuntu /mnt/efs
sudo mount -t efs -o tls fs-xxxxxxxxxxxxxxxxx:/ /mnt/efs
```

Step 2.2 - Attaching EFS to a fresh instance

When you launching a fresh instance using AWS Console or AWS CLI, just pass the following User Data script -

```
#!/bin/bash
exec > >(tee /var/log/user-data.log|logger -t user-data -
s 2>/dev/console) 2>&1

# Update packages and install prerequisites for building efs-
utils on Ubuntu
# For Amazon Linux, adjust the commands accordingly to match its
package manager and package names.
apt-get update -y
apt-get install -y git binutils make rustc cargo pkg-
config libssl-dev stunnel4

# Clone and build efs-utils from source for Ubuntu
git clone https://github.com/aws/efs-utils.git
cd efs-utils
./build-deb.sh
```

```
apt-get install -y ./build/amazon-efs-utils*deb

# Create the mount point directory
sudo mkdir -p /mnt/efs

# Change the ownership of the mount directory
sudo chown -R ubuntu:ubuntu /mnt/efs

# Mounting the EFS
sudo mount -t efs -o tls fs-xxxxxxxxxxxxxxxxx:/ /mnt/efs
```

Step 3 - Verify EFS has been attached

A. ssh into your instance, and run the following commands -

```
df -h | grep efs
```

You should be able to see the attached EFS

```
127.0.0.1:/      8.0E     0  8.0E   0% /mnt/efs
```

Now, navigate to `/mnt/efs` and use this directory as your workspace. Any data saved here will persist independently of your EC2 instance's lifecycle. If you lose access to your current instance, you can still recover your data by attaching this EFS to your instance by following the Step#2.