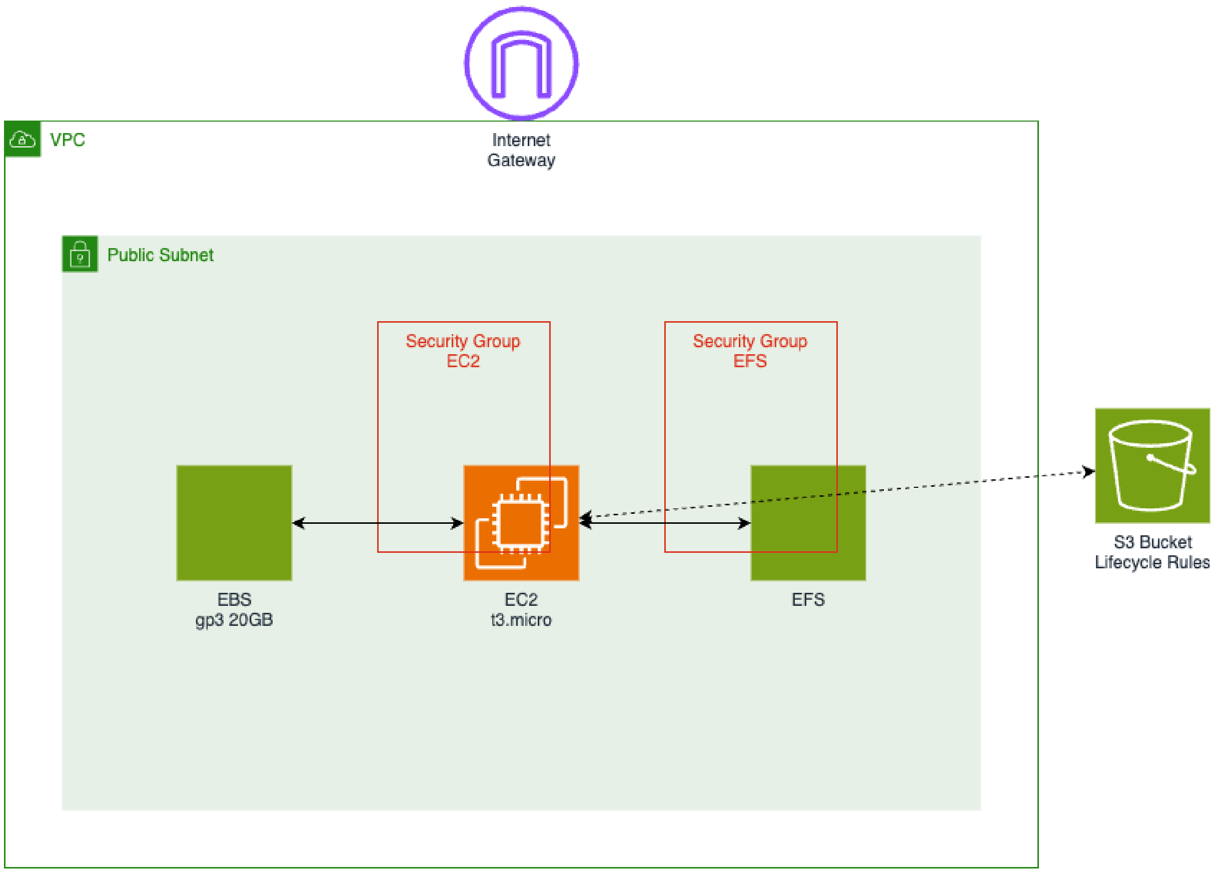
Day5 Real Flow AWS Storage Solutions

Project Overview



```
graph TB
    subgraph "VPC 10.0.0.0/16"
        subgraph "Public Subnet 10.0.1.0/24"
            EC2[EC2 t3.micro<br>with IAM Role] --> |Mount NFS| EFS
            EC2 --> |Attached| EBS[EBS gp3 20GB]
        end
        EFS[EFS Filesystem<br>IA after 30 days]
        SG1[EC2 SG<br>Port 22] --> EC2
        SG2[EFS SG<br>Port 2049] --> EFS
    end
    subgraph "S3 Lifecycle"
        S3[S3 Standard] --> |30 days| S3IA[S3-IA]
        S3IA --> |90 days| Glacier[Glacier]
    end

    IG[Internet Gateway] --> VPC
```

Implementation Steps

Prepare storage json (parameters)and yaml(template) files.

```
[
    {
      "ParameterKey": "Environment",
      "ParameterValue": "prod"
    },
    {
      "ParameterKey": "BucketPrefix",
      "ParameterValue": "day5-storage-demo"
    }
  ]
```

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'Day 5 - Storage Infrastructure'

Parameters:
  Environment:
    Type: String
    Default: prod
    AllowedValues: [dev, prod]

  BucketPrefix:
    Type: String
    Default: my-storage-demo

Resources:
```

```
# IAM Role for EC2
EC2Role:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ec2.amazonaws.com
          Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
      - arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
      - arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
    Policies:
      - PolicyName: EC2CustomPolicy
        PolicyDocument:
          Version: '2012-10-17'
          Statement:
            - Effect: Allow
              Action:
                - cloudformation:DescribeStacks
                - efs:DescribeFileSystems
              Resource: '*'

EC2InstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: /
    Roles:
      - !Ref EC2Role

# S3 Bucket with Lifecycle Rules
StorageBucket:
  Type: AWS::S3::Bucket
  Properties:
    BucketName: !Sub ${BucketPrefix}-${AWS::AccountId}
```

```
      VersioningConfiguration:
        Status: Enabled
      LifecycleConfiguration:
        Rules:
          - Id: TransitionToIA
            Status: Enabled
            Transitions:
              - StorageClass: STANDARD_IA
                TransitionInDays: 30
          - Id: TransitionToGlacier
            Status: Enabled
            Transitions:
              - StorageClass: GLACIER
                TransitionInDays: 90

  # VPC and Network Configuration
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
      Tags:
        - Key: Name
          Value: !Sub ${Environment}-storage-vpc

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Sub ${Environment}-igw

  AttachGateway:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref VPC
      InternetGatewayId: !Ref InternetGateway
```

```
PublicSubnet:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone: !Select [0, !GetAZs '']
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${Environment}-public-subnet

PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${Environment}-public-rt

PublicRoute:
  Type: AWS::EC2::Route
  DependsOn: AttachGateway
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

PublicSubnetRouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref PublicSubnet
    RouteTableId: !Ref PublicRouteTable

# Security Groups
EC2SecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for EC2 instance
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
    Tags:
      - Key: Name
        Value: !Sub ${Environment}-ec2-sg

EFSSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for EFS
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 2049
        ToPort: 2049
        SourceSecurityGroupId: !Ref EC2SecurityGroup
    Tags:
      - Key: Name
        Value: !Sub ${Environment}-efs-sg

# EFS File System
FileSystem:
  Type: AWS::EFS::FileSystem
  Properties:
    Encrypted: true
    FileSystemTags:
      - Key: Name
        Value: !Sub ${Environment}-efs
    LifecyclePolicies:
      - TransitionToIA: AFTER_30_DAYS
    PerformanceMode: generalPurpose
    ThroughputMode: bursting
MountTarget:
  Type: AWS::EFS::MountTarget
  Properties:
    FileSystemId: !Ref FileSystem
    SubnetId: !Ref PublicSubnet
    SecurityGroups:
      - !Ref EFSSecurityGroup

# EC2 Instance
EC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    InstanceType: t3.micro
    ImageId: ami-0d3bbfd074edd7acb
    KeyName: storage-demo-key
    SubnetId: !Ref PublicSubnet
    SecurityGroupIds:
      - !Ref EC2SecurityGroup
    IamInstanceProfile: !Ref EC2InstanceProfile
    BlockDeviceMappings:
      - DeviceName: /dev/xvda
        Ebs:
          VolumeSize: 20
          VolumeType: gp3
          DeleteOnTermination: true
    Tags:
      - Key: Name
        Value: !Sub ${Environment}-storage-demo
      - Key: StackName
        Value: !Ref AWS::StackName

Outputs:
BucketName:
  Description: S3 Bucket Name
  Value: !Ref StorageBucket

FileSystemId:
  Description: EFS File System ID
  Value: !Ref FileSystem

EC2InstanceId:
  Description: EC2 Instance ID
  Value: !Ref EC2Instance

EC2PublicIP:
  Description: EC2 Instance Public IP
  Value: !GetAtt EC2Instance.PublicIp

VpcId:
  Description: VPC ID
  Value: !Ref VPC

EFSMountTarget:
  Description: EFS Mount Target DNS Name
  Value: !Join
    - ''
    - - !Ref FileSystem
      - '.efs.'
      - !Ref 'AWS::Region'
      - '.amazonaws.com'
```

Implementation Steps

1. Create Key Pair

```
# Create new key pair
aws ec2 create-key-pair \\
    --key-name storage-demo-key \\
    --query 'KeyMaterial' \\
    --output text > storage-demo-key.pem

# Set permissions
chmod 400 storage-demo-key.pem
```

2. Deploy CloudFormation Stack

```
# Create stack
aws cloudformation create-stack \\
    --stack-name storage-stack \\
    --template-body file://templates/storage.yaml \\
    --parameters file://parameters/storage.json \\
    --capabilities CAPABILITY_IAM
```

3. Monitor Stack Creation

```
# Check stack status
aws cloudformation describe-stacks \\
    --stack-name storage-stack \\
    --query 'Stacks[0].StackStatus'
```

4. Connect to EC2 Instance

```
# Get EC2 public IP
EC2_IP=$(aws cloudformation describe-stacks \\
    --stack-name storage-stack \\
    --query 'Stacks[0].Outputs[?OutputKey==`EC2PublicIP`].OutputValue' \\
    --output text)

# SSH into instance
ssh -i storage-demo-key.pem ec2-user@$EC2_IP
```

5. Mount EFS on EC2

```
# Install EFS utilities
sudo yum install -y amazon-efs-utils



STACK_NAME="storage-stack"

# Get EFS DNS name using instance role
EFS_DNS=$(aws cloudformation describe-stacks \\
    --stack-name $STACK_NAME \\
    --query 'Stacks[0].Outputs[?OutputKey==`EFSMountTarget`].OutputValu
e' \\
    --output text \\
    --region ap-northeast-1)

# Create mount point
sudo mkdir -p /efs

# Mount EFS
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,tim
eo=600,retrans=2,noresvport $EFS_DNS:/ /efs

# Add to fstab
echo "$EFS_DNS:/ /efs nfs4 nfsvers=4.1,rsize=1048576,wsize=1048576,ha
rd,timeo=600,retrans=2,noresvport 0 0" | sudo tee -a /etc/fstab
```

6. Test Storage Components

Test EFS

```
# Create test file
cd /efs
sudo touch test.txt
sudo chmod 777 test.txt
echo "EFS test content" > test.txt

# Verify
cat test.txt
```

Test EBS Volume

```
# Check EBS volume
df -h /dev/xvda1

# Write test file
echo "EBS test content" > ~/ebs-test.txt
```

Test S3 Lifecycle

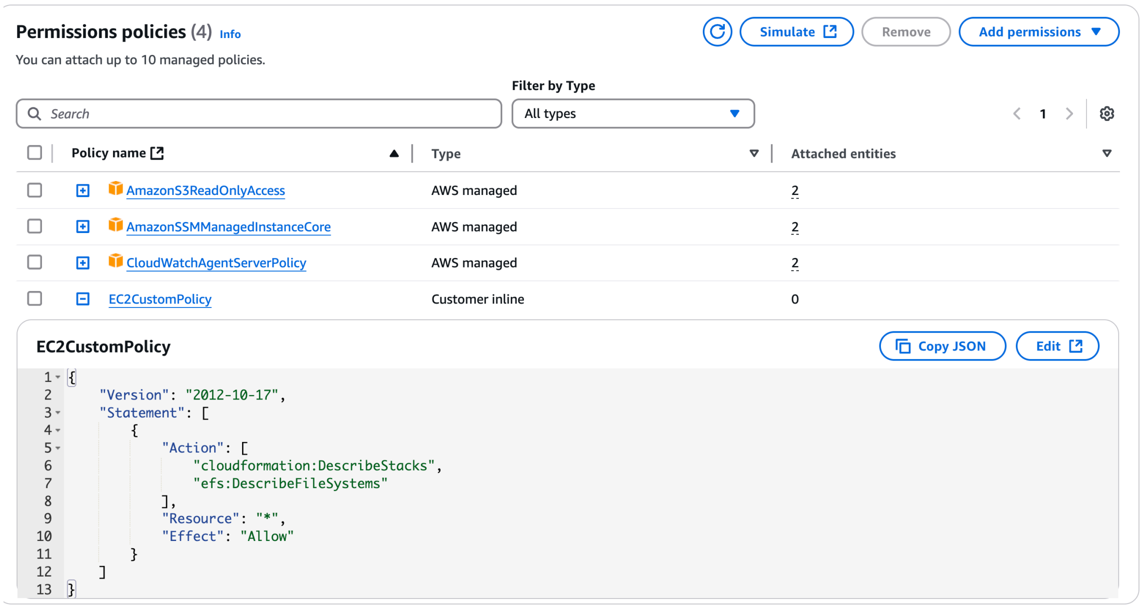
```
# Get bucket name
BUCKET_NAME=$(aws cloudformation describe-stacks \\
    --stack-name $STACK_NAME \\
    --query 'Stacks[0].Outputs[?OutputKey==`BucketName`].OutputValue' \\
    --output text)

# Create and upload test file
echo "S3 test content" > test.txt
aws s3 cp test.txt s3://$BUCKET_NAME/
```

Cost Estimation

Service	Configuration	Monthly Cost (Est.)	Optimization
EC2	t3.micro	~$8.50	Use Spot for dev
EBS	20GB gp3	~$2.40	Delete when not needed
EFS	Standard + IA	~$0.30/GB	Use IA for old data
S3	Lifecycle enabled	~$0.023/GB	Use lifecycle rules

Security Features

1. IAM Role with least privilege



Permissions policies (4)

You can attach up to 10 managed policies.

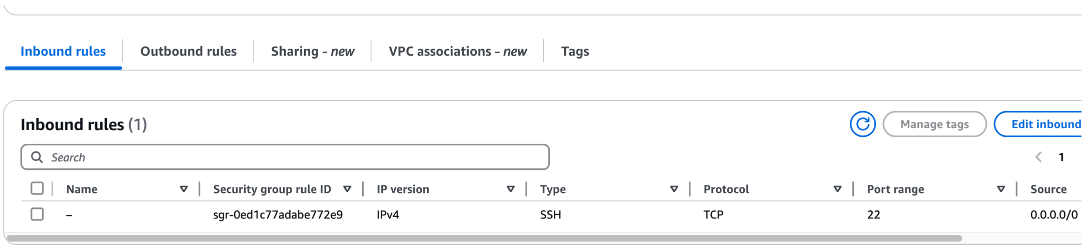
Filter by Type: All types

Policy name	Type	Attached entities
AmazonS3ReadOnlyAccess	AWS managed	2
AmazonSSMManagedInstanceCore	AWS managed	2
CloudWatchAgentServerPolicy	AWS managed	2
EC2CustomPolicy	Customer inline	0

EC2CustomPolicy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "cloudformation:DescribeStacks",
                "efs:DescribeFileSystems"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

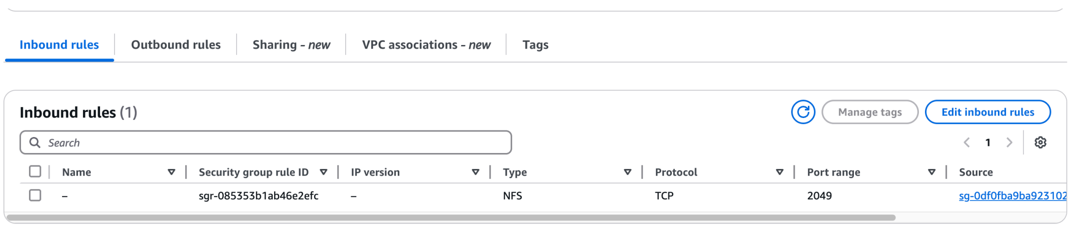
1. Security Groups:
    - EC2: Port 22 only



Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
–	sgr-0ed1c77adabe772e9	IPv4	SSH	TCP	22	0.0.0.0/0

    - EFS: Port 2049 from EC2 only



Inbound rules (1)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
–	sgr-085353b1ab46e2efc	–	NFS	TCP	2049	sg-0df0fba9ba923102

2. Encrypted EFS
3. S3 versioning enabled

Cleanup Process

```
# 1. Empty S3 bucket
aws s3 rm s3://$BUCKET_NAME/ --recursive

# 2. Unmount EFS
sudo umount /efs

# 3. Delete CloudFormation stack
aws cloudformation delete-stack --stack-name storage-stack

# 4. Delete key pair
```

```
aws ec2 delete-key-pair --key-name storage-demo-key
rm storage-demo-key.pem
```

Monitoring Points

1. CloudWatch Metrics:
   - EBS: VolumeReadOps, VolumeWriteOps
   - EFS: BurstCreditBalance, PercentIOLimit
   - S3: BucketSizeBytes, NumberOfObjects
2. Cost Alerts:
   - S3 storage tiers transition
   - EFS IA storage usage
   - EBS volume utilization