Day 3: AWS CloudFormation – Advanced Practical Labs

This guide provides hands-on CloudFormation labs to build a complete secure AWS environment using infrastructure-as-code (IaC). You'll create a Bastion Host (Jump Server), Private EC2, NAT Gateway, and full VPC setup — all managed via CloudFormation.

Lab Objective

- Build an AWS VPC with Public and Private Subnets - Deploy a Bastion Host in Public Subnet - Deploy a Private EC2 Instance accessible only via Bastion - Configure NAT Gateway for Private EC2 outbound access - Automate everything with CloudFormation template

■ CloudFormation Template (bastion-private-ec2-cloudformation.yml)

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Secure VPC with Bastion Host & Private EC2
Parameters:
  KeyPairName:
    Type: AWS::EC2::KeyPair::KeyName
    Description: Name of an existing EC2 KeyPair

Resources:
  MyVPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsSupport: true
      EnableDnsHostnames: true

  InternetGateway:
    Type: AWS::EC2::InternetGateway

  AttachIGW:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref MyVPC
      InternetGatewayId: !Ref InternetGateway

  PublicSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: 10.0.1.0/24
      MapPublicIpOnLaunch: true

  PrivateSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref MyVPC
      CidrBlock: 10.0.2.0/24
      MapPublicIpOnLaunch: false
  BastionInstance:
```

```
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t2.micro
      KeyName: !Ref KeyPairName
      SubnetId: !Ref PublicSubnet
      ImageId: ami-0c02fb55956c7d316

  PrivateInstance:
    Type: AWS::EC2::Instance
    Properties:
      InstanceType: t2.micro
      KeyName: !Ref KeyPairName
      SubnetId: !Ref PrivateSubnet
      ImageId: ami-0c02fb55956c7d316

Outputs:
  BastionPublicIP:
    Value: !GetAtt BastionInstance.PublicIp
  PrivateInstanceID:
    Value: !Ref PrivateInstance
```

■ Deployment Steps

1. Open AWS Console → CloudFormation → Create Stack. 2. Upload `bastion-private-ec2-cloudformation.yml` file. 3. Specify your KeyPair and stack name. 4. Acknowledge IAM resource creation. 5. Click “Create Stack” and wait 5–8 minutes.

Using AWS CLI

```
aws cloudformation create-stack --stack-name BastionPrivateVPC --template-body file://bastion-pr
```

■ Connecting to Private EC2

1. SSH into Bastion Host: ssh -i my-key.pem ec2-user@ 2. From Bastion, connect to Private EC2: ssh -i my-key.pem ec2-user@10.0.2.x 3. Verify connectivity and NAT access: ping google.com

■ Cleanup

```
aws cloudformation delete-stack --stack-name BastionPrivateVPC
```

■ Conclusion

You successfully automated secure AWS infrastructure creation using CloudFormation. This setup is production-ready, scalable, and aligned with CloudOps best practices.