AWS Implementation Solution Guide - ENTA & EPPV

Table of Contents

Initial Configuration

Key Pairs

Região US-EAST-1

1. Aceder à consola AWS e mudar para a região us-east-1
2. Navegar até EC2 > Network & Security > Key Pairs
3. Clicar em "Create Key Pair"
4. Nome: pdl-keypair
5. Formato: .pem (para Linux) ou .ppk (para PuTTY)
6. Clicar em "Create" e guardar o ficheiro com segurança

Região US-WEST-2

1. Mudar para região us-west-2
2. Repetir o processo acima
3. Nome: angra-keypair
4. Guardar o segundo ficheiro de chave

Verificação de AMIs

Processo a fazer para todas as regiões

1. Vá para EC2 > AMIs

2. Verifique disponibilidade:
    - Amazon Linux 2023 AMI
    - Ubuntu Server 22.04 LTS
    - Windows Server 2022
3. Note os IDs das AMIs para uso posterior

Preparação do Ambiente Local

MySQL Workbench

1. Download: https://dev.mysql.com/downloads/workbench/
2. Instalar com opções padrão
3. Testar conexão local

FileZilla

1. Download: https://filezilla-project.org/
2. Instalar versão cliente
3. Configurar para usar chaves SSH

Browsers

1. Instalar/Atualizar:
    - Chrome
    - Firefox
    - Edge
2. Instalar extensões úteis:
    - JSON Formatter
    - HTTPS Everywhere

Ferramentas de Monitorização

1. Ativar CloudWatch:
    - Acessar IAM
    - Criar role com permissões CloudWatchAgentServerPolicy
    - Preparar para anexar às instâncias EC2

VPC Configuration

PDL-VPC (US-EAST-1) Configuration

Criar VPC PDL-VPC

```
aws ec2 create-vpc \
  --cidr-block 10.0.0.0/20 \
  --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=pdl-vpc}]' \
  --region us-east-1
```

Criar Subnets PDL-VPC

A primeira subnet é publica e as ultimas duas são privadas

```
aws ec2 create-subnet \
  --vpc-id vpc-XXXXX \
  --cidr-block 10.0.0.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=pdl-public}]'
```

```
aws ec2 create-subnet \
  --vpc-id vpc-XXXXX \
  --cidr-block 10.0.1.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=pdl-private-1}]'
```

```
aws ec2 create-subnet \
  --vpc-id vpc-XXXXX \
  --cidr-block 10.0.2.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=pdl-private-2}]'
```

WEB-VPC (US-EAST-1) Configuration

Criar VPC WEB-VPC

```
aws ec2 create-vpc \
  --cidr-block 10.0.16.0/20 \
  --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=web-vpc}]' \
  --region us-east-1
```

Criar Subnets WEB-VPC

```
aws ec2 create-subnet \
  --vpc-id vpc-XXXXX \
  --cidr-block 10.0.16.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=web-public-1a}]'
```

```
aws ec2 create-subnet \
  --vpc-id vpc-XXXXX \
  --cidr-block 10.0.17.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=web-public-1b}]'
```

```
aws ec2 create-subnet \
  --vpc-id vpc-XXXXX \
  --cidr-block 10.0.18.0/24 \
  --availability-zone us-east-1c \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=web-public-1c}]'
```

ANGRA-VPC (US-WEST-2) Configuration

Criar VPC ANGRA-VPC

```
aws ec2 create-vpc \
  --cidr-block 172.16.0.0/16 \
  --amazon-provided-ipv6-cidr-block \
  --tag-specifications 'ResourceType=vpc,Tags=[{Key=Name,Value=angra-vpc}]' \
  --region us-west-2
```

Criar Subnets ANGRA-VPC

```
aws ec2 create-subnet \
  --vpc-id vpc-XXXXX \
  --cidr-block 172.16.0.0/24 \
  --ipv6-cidr-block XXXX:XXXX:XXXX:XXXX::/64 \
  --availability-zone us-west-2a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=angra-public}]'
```

```
aws ec2 create-subnet \
  --vpc-id vpc-XXXXX \
  --cidr-block 172.16.1.0/24 \
  --ipv6-cidr-block XXXX:XXXX:XXXX:XXXX::/64 \
  --availability-zone us-west-2a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=angra-private-1}]'
```

Common Configuration for All VPCs

Enable DNS hostnames and resolution

```
aws ec2 modify-vpc-attribute \
  --vpc-id vpc-XXXXX \
  --enable-dns-hostnames
aws ec2 modify-vpc-attribute \
  --vpc-id vpc-XXXXX \
  --enable-dns-support
```

Create S3 Gateway Endpoint

```
aws ec2 create-vpc-endpoint \
  --vpc-id vpc-XXXXX \
  --service-name com.amazonaws.region.s3 \
  --route-table-ids rtb-XXXXX
```

VPC Interconnection

Transit Gateway Configuration (US-EAST-1)

Create Transit Gateway

```
aws ec2 create-transit-gateway \
  --description "Main Transit Gateway" \
  --tag-specifications 'ResourceType=transit-gateway,Tags=[{Key=Name,Value=main-tgw}]'
\
  --region us-east-1
```

Attach PDL-VPC

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-XXXXX \
  --vpc-id vpc-XXXXX \
  --subnet-ids subnet-XXXXX subnet-YYYYY \
  --tag-specifications 'ResourceType=transit-gateway-attachment,Tags=
[{Key=Name,Value=pdl-vpc-attachment}]'
```

Attach WEB-VPC

```
aws ec2 create-transit-gateway-vpc-attachment \
  --transit-gateway-id tgw-XXXXX \
  --vpc-id vpc-YYYYY \
  --subnet-ids subnet-AAAAA subnet-BBBBB \
  --tag-specifications 'ResourceType=transit-gateway-attachment,Tags=
[{Key=Name,Value=web-vpc-attachment}]'
```

Transit Gateway Peering (Cross-Region)

Create Peering Attachment

```
aws ec2 create-transit-gateway-peering-attachment \
  --transit-gateway-id tgw-XXXXX \
  --peer-transit-gateway-id tgw-YYYYY \
  --peer-region us-west-2 \
  --tag-specifications 'ResourceType=transit-gateway-attachment,Tags=
[{Key=Name,Value=us-east-1-to-us-west-2}]'
```

Accept Peering (in us-west-2)

```
aws ec2 accept-transit-gateway-peering-attachment \
  --transit-gateway-attachment-id tgw-attach-XXXXX \
  --region us-west-2
```

Route Tables Configuration

PDL-VPC Route Table

```
aws ec2 create-route \
  --route-table-id rtb-XXXXX \
  --destination-cidr-block 172.16.0.0/16 \
  --transit-gateway-id tgw-XXXXX
```

WEB-VPC Route Table

```
aws ec2 create-route \
  --route-table-id rtb-YYYYY \
  --destination-cidr-block 172.16.0.0/16 \
  --transit-gateway-id tgw-XXXXX
```

ANGRA-VPC Route Table (us-west-2)

```
aws ec2 create-route \
  --route-table-id rtb-ZZZZZ \
  --destination-cidr-block 10.0.0.0/16 \
  --transit-gateway-id tgw-YYYYY \
  --region us-west-2
```

Security Group Updates

Allow traffic between VPCs

```
aws ec2 authorize-security-group-ingress \
  --group-id sg-XXXXX \
  --protocol all \
  --source-group sg-YYYYY
```

Verify Connectivity

```
# Test connectivity using EC2 instances in each VPC
aws ec2 describe-transit-gateway-attachments
aws ec2 describe-transit-gateway-route-tables
```

S3 Configuration

Create Buckets

Create PDL bucket in us-east-1

```
aws s3api create-bucket \
  --bucket pdl-data-bucket \
  --region us-east-1
```

Create ANGRA bucket in us-west-2

```
aws s3api create-bucket \
  --bucket angra-data-bucket \
  --region us-west-2 \
  --create-bucket-configuration LocationConstraint=us-west-2
```

Enable Versioning

```
# Enable for both buckets
aws s3api put-bucket-versioning \
  --bucket pdl-data-bucket \
  --versioning-configuration Status=Enabled

aws s3api put-bucket-versioning \
  --bucket angra-data-bucket \
  --versioning-configuration Status=Enabled
```

Configure Lifecycle Rules

Create lifecycle policy for PDL bucket

```
aws s3api put-bucket-lifecycle-configuration \
  --bucket pdl-data-bucket \
  --lifecycle-configuration file://lifecycle-policy.json
```

```
{
  "Rules": [
    {
      "ID": "Move to IA and Glacier",
      "Status": "Enabled",
      "Transitions": [
        {
          "Days": 30,
          "StorageClass": "STANDARD_IA"
        },
        {
          "Days": 90,
          "StorageClass": "GLACIER"
        }
      ]
    }
  ]
}
```

Configure Bucket Policies

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VPCEndpointAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::pdl-data-bucket",
        "arn:aws:s3:::pdl-data-bucket/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceVpc": ["vpc-XXXXX", "vpc-YYYYY"]
        }
      }
    }
  ]
}
```

Configure Cross-Region Replication

Enable replication from PDL to ANGRA bucket

```
aws s3api put-bucket-replication \
  --bucket pdl-data-bucket \
  --replication-configuration file://replication.json
```

```
{
  "Role": "arn:aws:iam::ACCOUNT-ID:role/s3-replication-role",
  "Rules": [
    {
      "Status": "Enabled",
      "Priority": 1,
      "DeleteMarkerReplication": { "Status": "Enabled" },
      "Destination": {
        "Bucket": "arn:aws:s3:::angra-data-bucket",
        "StorageClass": "STANDARD"
      }
    }
  ]
}
```

RDS MySQL Configuration

Create Subnet Groups

Create subnet group for PDL-VPC

```
aws rds create-db-subnet-group \
  --db-subnet-group-name pdl-db-subnet \
  --db-subnet-group-description "Subnet group for PDL RDS" \
  --subnet-ids subnet-XXXXX subnet-YYYYY \
  --region us-east-1
```

Create subnet group for ANGRA-VPC

```
aws rds create-db-subnet-group \
  --db-subnet-group-name angra-db-subnet \
  --db-subnet-group-description "Subnet group for ANGRA RDS" \
  --subnet-ids subnet-AAAAA subnet-BBBBB \
  --region us-west-2
```

Create Security Groups

PDL Database Security Group

```
aws ec2 create-security-group \
  --group-name pdl-db-sg \
  --description "Security group for PDL RDS" \
  --vpc-id vpc-XXXXX
```

```
aws ec2 authorize-security-group-ingress \
  --group-id sg-XXXXX \
  --protocol tcp \
  --port 3306 \
  --source-group sg-YYYYY
```

ANGRA Database Security Group

```
aws ec2 create-security-group \
  --group-name angra-db-sg \
  --description "Security group for ANGRA RDS" \
  --vpc-id vpc-YYYYY \
  --region us-west-2
```

Create Primary RDS Instance (PDL)

```
aws rds create-db-instance \
  --db-instance-identifier pdl-primary \
  --db-instance-class db.t3.medium \
  --engine mysql \
  --engine-version 8.0.28 \
  --master-username admin \
  --master-user-password YOUR_PASSWORD \
  --allocated-storage 20 \
  --db-subnet-group-name pdl-db-subnet \
  --vpc-security-group-ids sg-XXXXX \
  --backup-retention-period 7 \
  --multi-az true \
  --region us-east-1
```

Create Read Replica (ANGRA)

```
aws rds create-db-instance-read-replica \
  --db-instance-identifier angra-replica \
  --source-db-instance-identifier arn:aws:rds:us-east-1:ACCOUNT-ID:db:pdl-primary \
  --db-instance-class db.t3.medium \
  --availability-zone us-west-2a \
  --db-subnet-group-name angra-db-subnet \
  --vpc-security-group-ids sg-YYYYY \
  --region us-west-2
```

Configure Monitoring

```
# Enable Enhanced Monitoring
aws rds modify-db-instance \
  --db-instance-identifier pdl-primary \
  --monitoring-interval 60 \
  --monitoring-role-arn arn:aws:iam::ACCOUNT-ID:role/rds-monitoring-role

# Create CloudWatch Alarms
aws cloudwatch put-metric-alarm \
  --alarm-name PDL-DB-CPUUtilization \
  --metric-name CPUUtilization \
  --namespace AWS/RDS \
  --statistic Average \
  --period 300 \
  --threshold 80 \
  --comparison-operator GreaterThanThreshold \
  --evaluation-periods 2 \
  --alarm-actions arn:aws:sns:us-east-1:ACCOUNT-ID:notifications
```

Auto Scaling Group Configuration

Create Launch Template for Web Servers

```
# Create Launch Template in US-EAST-1
aws ec2 create-launch-template \
  --launch-template-name web-launch-template \
  --version-description v1 \
  --launch-template-data '{
    "ImageId": "ami-XXXXX",
    "InstanceType": "t3.micro",
    "KeyName": "pdl-keypair",
    "SecurityGroupIds": ["sg-XXXXX"],
    "UserData":
"IyEvYmluL2Jhc2gKYXB0IHVwZGF0ZSAteSAmJiBhcHQgaW5zdGFsbCAteSBuZ2lueAo=",
    "IamInstanceProfile": {
        "Name": "WebServerRole"
    },
    "BlockDeviceMappings": [
        {
            "DeviceName": "/dev/xvda",
            "Ebs": {
                "VolumeSize": 20,
                "VolumeType": "gp3"
            }
        }
    ]
}'
```

Create Auto Scaling Group

```
# Create ASG for web servers
aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name web-asg \
  --launch-template "LaunchTemplateName=web-launch-template,Version=1" \
  --min-size 2 \
  --max-size 6 \
  --desired-capacity 2 \
  --vpc-zone-identifier "subnet-XXXXX,subnet-YYYYY,subnet-ZZZZZ" \
  --target-group-arns "arn:aws:elasticloadbalancing:us-east-1:ACCOUNT-
ID:targetgroup/web-tg/XXXXX" \
  --health-check-type ELB \
  --health-check-grace-period 300
```

Configure Scaling Policies

```
# Create CPU based scaling policy
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name web-asg \
  --policy-name cpu-scale-out \
  --policy-type TargetTrackingScaling \
  --target-tracking-configuration '{
    "TargetValue": 70.0,
    "PredefinedMetricSpecification": {
        "PredefinedMetricType": "ASGAverageCPUUtilization"
    }
}'

# Create request count based scaling policy
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name web-asg \
  --policy-name request-scale-out \
  --policy-type TargetTrackingScaling \
  --target-tracking-configuration '{
    "TargetValue": 1000.0,
    "PredefinedMetricSpecification": {
        "PredefinedMetricType": "ALBRequestCountPerTarget"
    }
}'
```

Configure Load Balancer

```
# Create Application Load Balancer
aws elbv2 create-load-balancer \
  --name web-alb \
  --subnets subnet-XXXXX subnet-YYYYY subnet-ZZZZZ \
  --security-groups sg-XXXXX \
  --scheme internet-facing \
  --tags Key=Environment,Value=Production

# Create Target Group
aws elbv2 create-target-group \
  --name web-tg \
  --protocol HTTP \
  --port 80 \
  --vpc-id vpc-XXXXX \
  --health-check-path /health \
  --target-type instance

# Create Listener
aws elbv2 create-listener \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-east-1:ACCOUNT-
ID:loadbalancer/app/web-alb/XXXXX \
  --protocol HTTPS \
  --port 443 \
  --certificates CertificateArn=arn:aws:acm:us-east-1:ACCOUNT-ID:certificate/XXXXX \
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-east-
1:ACCOUNT-ID:targetgroup/web-tg/XXXXX
```

Instance Configurations in US-EAST-1

srv.pdl.local Configuration

```
# Create EC2 Instance
aws ec2 run-instances \
  --image-id ami-XXXXX \
  --instance-type t3.small \
  --key-name pdl-keypair \
  --subnet-id subnet-XXXXX \
  --security-group-ids sg-XXXXX \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=srv-pdl-local}]' \
  --user-data file://srv-user-data.sh \
  --iam-instance-profile Name=SrvInstanceProfile

# srv-user-data.sh content:
#!/bin/bash
yum update -y
yum install -y httpd mysql php
systemctl enable httpd
systemctl start httpd
```

cli.pdl.local Configuration

```
aws ec2 run-instances \
  --image-id ami-XXXXX \
  --instance-type t3.micro \
  --key-name pdl-keypair \
  --subnet-id subnet-YYYYY \
  --security-group-ids sg-YYYYY \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=cli-pdl-local}]' \
  --user-data file://cli-user-data.sh

# cli-user-data.sh content:
#!/bin/bash
yum update -y
yum install -y mysql-client
```

DMZ Instances Configuration

```
# dmzwin.pdl.local (Windows)
aws ec2 run-instances \
  --image-id ami-windows-XXXXX \
  --instance-type t3.small \
  --key-name pdl-keypair \
  --subnet-id subnet-ZZZZZ \
  --security-group-ids sg-ZZZZZ \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=dmzwin-pdl-
local}]'

# dmzlux.pdl.local (Linux)
aws ec2 run-instances \
  --image-id ami-YYYYY \
  --instance-type t3.micro \
  --key-name pdl-keypair \
  --subnet-id subnet-ZZZZZ \
  --security-group-ids sg-ZZZZZ \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=dmzlux-pdl-
local}]' \
  --user-data file://dmz-user-data.sh
```

Security Groups Configuration US-EAST-1

```
# Internal Network SG
aws ec2 create-security-group \
  --group-name pdl-internal-sg \
  --description "Internal network security group" \
  --vpc-id vpc-XXXXX

aws ec2 authorize-security-group-ingress \
  --group-id sg-XXXXX \
  --protocol tcp \
  --port 3389 \
  --cidr 10.0.0.0/20

# DMZ SG
aws ec2 create-security-group \
  --group-name pdl-dmz-sg \
  --description "DMZ security group" \
  --vpc-id vpc-XXXXX

aws ec2 authorize-security-group-ingress \
  --group-id sg-YYYYY \
  --protocol tcp \
  --port 80 \
  --cidr 0.0.0.0/0
```

Instance Configurations in US-WEST-2 (ANGRA)

srv.angra.local Configuration

```
# Create EC2 Instance
aws ec2 run-instances \
  --region us-west-2 \
  --image-id ami-XXXXX \
  --instance-type t3.small \
  --key-name angra-keypair \
  --subnet-id subnet-XXXXX \
  --security-group-ids sg-XXXXX \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=srv-angra-local}]'
\
  --user-data file://srv-angra-user-data.sh \
  --iam-instance-profile Name=SrvAngraInstanceProfile

# srv-angra-user-data.sh content:
#!/bin/bash
yum update -y
yum install -y httpd mysql php awscli
systemctl enable httpd
systemctl start httpd
```

cli.angra.local Configuration

```
aws ec2 run-instances \
  --region us-west-2 \
  --image-id ami-XXXXX \
  --instance-type t3.micro \
  --key-name angra-keypair \
  --subnet-id subnet-YYYYY \
  --security-group-ids sg-YYYYY \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=cli-angra-local}]'
\
  --user-data file://cli-angra-user-data.sh

# cli-angra-user-data.sh content:
#!/bin/bash
yum update -y
yum install -y mysql-client awscli
```

Intranet Instances Configuration

```
# intrawin.angra.local (Windows)
aws ec2 run-instances \
  --region us-west-2 \
  --image-id ami-windows-XXXXX \
  --instance-type t3.small \
  --key-name angra-keypair \
  --subnet-id subnet-ZZZZZ \
  --security-group-ids sg-ZZZZZ \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=intrawin-angra-
local}]'

# intralux.angra.local (Linux)
aws ec2 run-instances \
  --region us-west-2 \
  --image-id ami-YYYYY \
  --instance-type t3.micro \
  --key-name angra-keypair \
  --subnet-id subnet-ZZZZZ \
  --security-group-ids sg-ZZZZZ \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=intralux-angra-
local}]' \
  --user-data file://intra-user-data.sh
```

Security Groups Configuration US-WEST-2

```
# Internal Network SG
aws ec2 create-security-group \
  --region us-west-2 \
  --group-name angra-internal-sg \
  --description "Internal network security group" \
  --vpc-id vpc-XXXXX

aws ec2 authorize-security-group-ingress \
  --region us-west-2 \
  --group-id sg-XXXXX \
  --protocol -1 \
  --source-group sg-YYYYY

# Intranet SG
aws ec2 create-security-group \
  --region us-west-2 \
  --group-name angra-intranet-sg \
  --description "Intranet security group" \
  --vpc-id vpc-XXXXX

aws ec2 authorize-security-group-ingress \
  --region us-west-2 \
  --group-id sg-YYYYY \
  --protocol tcp \
  --port 80 \
  --cidr 172.16.0.0/16
```

DNS and Certificate Configuration

Route 53 Setup

```
# Create Private Hosted Zone for PDL
aws route53 create-hosted-zone \
  --name pdl.local \
  --vpc VPCRegion=us-east-1,VPCId=vpc-XXXXX \
  --caller-reference $(date +%s) \
  --hosted-zone-config Comment="PDL Private Zone"

# Create Private Hosted Zone for ANGRA
aws route53 create-hosted-zone \
  --name angra.local \
  --vpc VPCRegion=us-west-2,VPCId=vpc-YYYYY \
  --caller-reference $(date +%s) \
  --hosted-zone-config Comment="ANGRA Private Zone"
```

Create DNS Records

```
# PDL DNS Records
aws route53 change-resource-record-sets \
  --hosted-zone-id ZXXXXX \
  --change-batch '{
    "Changes": [
      {
        "Action": "CREATE",
        "ResourceRecordSet": {
          "Name": "srv.pdl.local",
          "Type": "A",
          "TTL": 300,
          "ResourceRecords": [{"Value": "10.0.1.10"}]
        }
      }
    ]
  }'

# ANGRA DNS Records
aws route53 change-resource-record-sets \
  --hosted-zone-id ZYYYYY \
  --change-batch '{
    "Changes": [
      {
        "Action": "CREATE",
        "ResourceRecordSet": {
          "Name": "srv.angra.local",
          "Type": "A",
          "TTL": 300,
          "ResourceRecords": [{"Value": "172.16.1.10"}]
        }
      }
    ]
  }'
```

ACM Certificate Configuration

```
# Request Certificate
aws acm request-certificate \
  --domain-name "*.pdl.local" \
  --validation-method DNS \
  --subject-alternative-names "*.angra.local" \
  --region us-east-1

# Validate Certificate
aws acm describe-certificate \
  --certificate-arn arn:aws:acm:us-east-1:ACCOUNT-ID:certificate/XXXXX

# Add Validation Records to Route 53
aws route53 change-resource-record-sets \
  --hosted-zone-id ZXXXXX \
  --change-batch file://validation-records.json
```

Final Verification Points DNS

```
# Check VPC DNS Settings
aws ec2 describe-vpc-attribute \
  --vpc-id vpc-XXXXX \
  --attribute enableDnsHostnames

aws ec2 describe-vpc-attribute \
  --vpc-id vpc-XXXXX \
  --attribute enableDnsSupport

# Verify DNS Resolution
aws ec2 describe-instances \
  --filters "Name=tag:Name,Values=srv-pdl-local" \
  --query 'Reservations[].Instances[].PrivateDnsName'

# Test Certificate
aws acm list-certificates \
  --region us-east-1

# Verify Route 53 Health Checks
aws route53 list-health-checks
```

Final Verification Points

Network Connectivity Testing

```
# Test VPC Peering
aws ec2 describe-vpc-peering-connections \
  --filters "Name=status-code,Values=active"

# Test Transit Gateway
aws ec2 describe-transit-gateway-attachments \
  --filters "Name=state,Values=available"
```

DNS Resolution Verification

```
# Test Private DNS Resolution
aws route53 test-dns-answer \
  --hosted-zone-id ZXXXXX \
  --record-name srv.pdl.local \
  --record-type A

# Verify Cross-Region DNS
aws route53 test-dns-answer \
  --hosted-zone-id ZYYYYY \
  --record-name srv.angra.local \
  --record-type A \
  --region us-west-2
```

Security Verification

```
# Check Security Groups
aws ec2 describe-security-groups \
  --filters "Name=vpc-id,Values=vpc-XXXXX"

# Verify NACL Settings
aws ec2 describe-network-acls \
  --filters "Name=vpc-id,Values=vpc-XXXXX"
```

Service Health Checks

```
# Check RDS Status
aws rds describe-db-instances \
  --query 'DBInstances[*].[DBInstanceIdentifier,DBInstanceStatus]'

# Verify Auto Scaling
aws autoscaling describe-auto-scaling-groups \
  --auto-scaling-group-names web-asg

# Check Load Balancer Health
aws elbv2 describe-target-health \
  --target-group-arn arn:aws:elasticloadbalancing:us-east-1:ACCOUNT-
ID:targetgroup/web-tg/XXXXX
```

Resource Monitoring

```
# CloudWatch Metrics
aws cloudwatch get-metric-statistics \
  --namespace AWS/EC2 \
  --metric-name CPUUtilization \
  --dimensions Name=AutoScalingGroupName,Value=web-asg \
  --start-time $(date -v-1H -u +%Y-%m-%dT%H:%M:%SZ) \
  --end-time $(date -u +%Y-%m-%dT%H:%M:%SZ) \
  --period 300 \
  --statistics Average

# Check CloudWatch Alarms
aws cloudwatch describe-alarms \
  --state-value ALARM
```

Backup Verification

```
# Verify RDS Backups
aws rds describe-db-snapshots \
  --db-instance-identifier pdl-primary

# Check S3 Replication
aws s3api get-bucket-replication \
  --bucket pdl-data-bucket
```