PRJ-SAP-019: Multi-Strategy Disaster Recovery Plan

Certification: AWS Certified Solutions Architect – Professional (SAP-C02)**Domain:** Business Continuity & Disaster Recovery**Author:** Mo | CloudGuard Portfolio

---

1. Business Context

Every production system will eventually face a failure event — whether a regional outage, data corruption, accidental deletion, or a ransomware incident. The question is not *if* it will happen, but *how fast you can recover* and *how much data you can afford to lose*. These two dimensions are formalized as **Recovery Time Objective (RTO)** and **Recovery Point Objective (RPO)**.

This project demonstrates three of the four AWS DR strategies across a real multi-region architecture. It is designed to give Solutions Architects hands-on experience with the trade-offs between cost, complexity, and recovery speed — a core competency tested in the AWS SAP-C02 exam and required in enterprise architecture roles.

---

2. GRC Mapping

Framework	Control	How This Project Satisfies It
NIST CSF	RC.RP-1	Recovery plan executed and maintained
NIST CSF	RC.CO-3	Recovery activities communicated to stakeholders
ISO 27001	A.17.1.1	Planning information security continuity
ISO 27001	A.17.1.2	Implementing information security continuity
ISO 27001	A.17.1.3	Verify, review, and evaluate continuity
SOC 2	A1.2	Environmental protections, software, data backup, and recovery
SOC 2	A1.3	Recovery plan testing
AWS Well-Architected	REL 9	How do you plan for disaster recovery?
AWS Well-Architected	REL 10	How do you use fault isolation to protect your workload?

---

3. DR Strategy Comparison

Strategy	RTO	RPO	Cost	Use Case
Backup & Restore	Hours	Hours	$	Dev/test, non-critical workloads
Pilot Light	10–30 min	Minutes	$$	Internal apps, moderate criticality
Warm Standby	Minutes	Seconds	$$$	Customer-facing apps, high criticality
Multi-Site Active-Active	Near zero	Near zero		Mission-critical (covered in PRJ-SAP-013)

---

4. Architecture Overview

The architecture spans two AWS regions: **us-east-1 (Primary)** and **us-west-2 (DR)**.

Primary Region Components:

- Application Load Balancer → EC2 Auto Scaling Group (web tier)
- RDS MySQL Multi-AZ (database tier)
- AWS Backup Vault with cross-region copy rules (configured via CloudFormation)
- Route 53 Failover routing with health checks
- SSM Parameter Store for primary configuration

DR Region Components (Pilot Light):

- VPC + subnets (always deployed via CloudFormation)
- RDS Read Replica (always running, promoted on failover)
- Auto Scaling Group with `DesiredCapacity: 0` (dormant until failover)
- ALB pre-configured (target group empty until ASG scales up)
- SSM Parameter Store holding DR database endpoint
- Backup Vault to receive cross-region copies

---

5. Prerequisites

Before starting, ensure the following are in place:

- AWS account with administrative IAM permissions in both `us-east-1` and `us-west-2`
- AWS CLI configured: `aws configure`
- A registered domain name managed in Route 53 (or a hosted zone already created)

---

6. Step-by-Step Implementation

Step 6.1 — Deploy the Primary Region Stack

Deploy the base application stack in `us-east-1`. This stack provisions the VPC, ALB, ASG, RDS instance, SSM parameters, and the AWS Backup plan and vault.

```
aws cloudformation create-stack \
  --stack-name PRJ-SAP-019-Primary \
  --template-body file://prj-sap-019-primary.yml \
  --capabilities CAPABILITY_NAMED_IAM \
  --region us-east-1
```

Wait for `CREATE_COMPLETE`, then note the following from the Outputs tab:

- `PrimaryALBDNS` — the load balancer DNS name
- `PrimaryRDSEndpoint` — the RDS instance endpoint

> ***Note:*** *The AWS Backup plan for the Backup & Restore strategy is automatically created by this stack. It includes a daily backup rule with cross-region copy to* `us-west-2`*.*

Step 6.2 — Create the RDS Read Replica (Pilot Light & Warm Standby)

Strategies 2 & 3 — Pilot Light and Warm Standby

1. Go to **RDS Console → Databases** in `us-east-1`.
2. Select your production RDS instance (`PRJ-SAP-019-primary-db`) → **Actions → Create read replica**.
3. **Destination region:** `us-west-2`
4. **DB instance identifier:** `prj-sap-019-dr-replica`
5. **DB instance class:** `db.t3.micro` (smaller for Pilot Light cost savings)
6. **Multi-AZ deployment:** No (DR replica does not need Multi-AZ)

7. Click **Create read replica**.

> *The replica will take 5–15 minutes to become available. It continuously replicates from the primary using asynchronous replication — RPO is typically under 1 minute for low-traffic databases.*

Step 6.3 — Copy the Application AMI to the DR Region

1. Go to **EC2 Console → Instances** in `us-east-1`.
2. Select one of your running application servers → **Actions → Image and templates → Create image**.
3. **Image name:** `PRJ-SAP-019-AppServer-AMI`
4. Click **Create image** and wait for status `available`.
5. Go to **EC2 Console → AMIs**, select the new AMI → **Actions → Copy AMI**.
6. **Destination region:** `us-west-2`
7. Click **Copy AMI** and note the new AMI ID in `us-west-2`.

Step 6.4 — Deploy the Pilot Light DR Stack

Before deploying, confirm the read replica from Step 6.2 is fully available and retrieve its endpoint. Run the following in CloudShell — the command will return nothing until the replica status is `available`:

```
# Wait until this returns 'available' before proceeding
aws rds describe-db-instances \
  --db-instance-identifier prj-sap-019-dr-replica \
  --query "DBInstances[0].DBInstanceStatus" \
  --output text \
  --region us-west-2

# Once available, retrieve the endpoint
REPLICA_ENDPOINT=$(aws rds describe-db-instances \
  --db-instance-identifier prj-sap-019-dr-replica \
  --query "DBInstances[0].Endpoint.Address" \
  --output text \
  --region us-west-2)

echo "Replica endpoint: $REPLICA_ENDPOINT"
```

Alternatively, find it in the console: **RDS → Databases → `prj-sap-019-dr-replica` → Connectivity & security tab → Endpoint**.

Deploy the dormant DR infrastructure in `us-west-2`. Replace `<AMI_ID_US_WEST_2>` with the AMI ID from Step 6.3:

```
aws cloudformation create-stack \
  --stack-name PRJ-SAP-019-PilotLight \
  --template-body file://prj-sap-019-pilot-light.yml \
  --parameters ParameterKey=AmiId,ParameterValue=<AMI_ID_US_WEST_2> \

ParameterKey=RDSReplicaEndpoint,ParameterValue=$REPLICA_ENDPOINT \
  --capabilities CAPABILITY_NAMED_IAM \
  --region us-west-2
```

What this deploys:

- VPC (`10.20.0.0/16`) with public and private subnets across 2 AZs
- Application Load Balancer (pre-configured, no targets yet)
- Auto Scaling Group with `DesiredCapacity: 0` — no EC2 instances running
- SSM Parameter Store entry `/prj-sap-019/dr/db-endpoint` with the replica endpoint

- IAM instance profile for SSM access
- AWS Backup Vault to receive cross-region copies

Step 6.5 — Configure Route 53 Failover Routing

1. Go to **Route 53 → Health checks → Create health check**.
2. **Name:** `PRJ-SAP-019-Primary-Health`
3. **What to monitor:** Endpoint
4. **Protocol:** HTTP
5. **Domain name:** Paste your `PrimaryALBDNS` value
6. **Path:** Leave this field empty. (Note: The path field only appears if you select 'IP address' instead of 'Domain name'. When using Domain name, Route 53 automatically probes the root path `/`, which is correct for this project.)
7. Click **Create health check**.
8. Go to **Route 53 → Hosted zones → your domain → Create record**.
9. **Record name:** `app` (creates `app.yourdomain.com`)
10. **Record type:** A
11. **Alias:** Yes → **Alias to Application and Classic Load Balancer** → `us-east-1` → select your primary ALB
12. **Routing policy:** Failover
13. **Failover record type:** Primary
14. **Health check:** Select `PRJ-SAP-019-Primary-Health`
15. Click **Add another record** with the same name:
    - **Alias:** Yes → `us-west-2` → select the DR ALB
    - **Routing policy:** Failover
    - **Failover record type:** Secondary

- **Health check: None — leave this blank.** This is critical. If you attach a health check to the Secondary record, Route 53 will evaluate both records independently. If the same health check is applied to both, or if the secondary check is also unhealthy, Route 53 will have no valid record to fail over to and will continue serving the Primary regardless of its health state. The Secondary record must always be unconditional.

16. Save both records.

---

7. The Failover Drill (Pilot Light)

Run this drill to validate the DR plan. Time each step — your total should be under 30 minutes for Pilot Light.

Step 7.1 — Declare the Disaster (Simulate Failure)

To trigger the failover, you must simulate a primary region failure. The cleanest and most recommended way to do this for a demo is to force the Route 53 health check to fail by blocking HTTP traffic at the primary ALB security group. This triggers the exact same automated DNS failover path that a real outage would, without destroying any infrastructure.

Run this in CloudShell to block inbound HTTP:

```
# Get the primary ALB security group
PRIMARY_ALB_SG=$(aws elbv2 describe-load-balancers \
  --names PRJ-SAP-019-Primary-ALB \
  --query 'LoadBalancers[0].SecurityGroups[0]' \
  --output text --region us-east-1)

# Revoke inbound HTTP — simulates the region becoming unreachable
aws ec2 revoke-security-group-ingress \
  --group-id $PRIMARY_ALB_SG \
  --protocol tcp --port 80 --cidr 0.0.0.0/0 \
  --region us-east-1
```

Watch the Route 53 health check status flip from **Healthy** to **Unhealthy**. Start a timer.

(Alternative simulations: You could also stop all primary EC2 instances to simulate an application-layer failure, or scale the primary ASG to 0 and stop the primary RDS instance to simulate a complete regional outage.)

Step 7.2 — Scale Up the Application Tier

```
# Scale the DR Auto Scaling Group from 0 to 2 instances
aws autoscaling update-auto-scaling-group \
  --auto-scaling-group-name PRJ-SAP-019-DR-ASG \
  --min-size 2 \
  --desired-capacity 2 \
  --region us-west-2
```

Monitor instance launch in the EC2 console. Instances should be `running` within 3–5 minutes.

Step 7.3 — Promote the RDS Read Replica

```
# Promote the read replica to a standalone master database
aws rds promote-read-replica \
  --db-instance-identifier prj-sap-019-dr-replica \
  --region us-west-2
```

> ***Important:*** *Promotion takes 5–15 minutes. The replica will be briefly unavailable during promotion. Monitor status with:“ `bash aws rds describe-db-instances –db-instance-identifier prj-sap-019-dr-replica –query “DBInstances[0].DBInstanceStatus” –region us-west-2*

```
### Step 7.4 — Update the Database Endpoint in Parameter Store

After promotion, the replica gets a new endpoint. Update Parameter Store so
the application servers pick it up:

```bash
# Get the new promoted DB endpoint
NEW_ENDPOINT=$(aws rds describe-db-instances \
  --db-instance-identifier prj-sap-019-dr-replica \
  --query "DBInstances[0].Endpoint.Address" \
  --output text \
  --region us-west-2)

echo "New DB endpoint: $NEW_ENDPOINT"

# Update SSM Parameter Store
aws ssm put-parameter \
  --name "/prj-sap-019/dr/db-endpoint" \
  --value "$NEW_ENDPOINT" \
  --type "SecureString" \
  --overwrite \
  --region us-west-2
```

Step 7.5 — Verify DNS Failover

Route 53 will automatically detect the primary health check failure and start routing traffic to the secondary record. Verify:

```
# Check DNS resolution (run from your local machine or CloudShell)
nslookup app.yourdomain.com 8.8.8.8

# Should now resolve to the DR ALB DNS name
curl -I http://app.yourdomain.com
```

Record your total RTO — the time from Step 7.1 to confirmed traffic flowing through the DR region.

```

8. Warm Standby Upgrade

To upgrade from Pilot Light to Warm Standby, set the ASG minimum capacity to 1 so at least one instance is always running in the DR region:

```
aws autoscaling update-auto-scaling-group \
  --auto-scaling-group-name PRJ-SAP-019-DR-ASG \
  --min-size 1 \
  --desired-capacity 1 \
  --region us-west-2
```

This reduces RTO from ~30 minutes to ~5 minutes because the application tier is already warm. The trade-off is a continuous EC2 cost in the DR region.

---

9. Troubleshooting

Symptom	Likely Cause	Fix
ASG instances launch but fail health checks	Application cannot connect to DB	Verify SSM Parameter Store has the correct endpoint; check security group allows DB port from app subnet
RDS promotion stuck in `modifying`	Replication lag too high at time of promotion	Wait for `ReplicaLag` CloudWatch metric to drop below 5 seconds before promoting
Route 53 not failing over	Health check threshold not met	Default is 3 consecutive failures over 30 seconds — wait 90 seconds or lower the threshold
DNS still resolving to primary after failover	TTL not expired	Default TTL is 60 seconds; wait or flush local DNS cache with `sudo dscacheutil -flushcache`
EC2 instances in DR region cannot reach SSM	Missing VPC endpoint or IAM role	Ensure the instance profile has `AmazonSSMManagedInstanceCore` policy attached

---

10. Cleanup

Run in this order to avoid dependency errors:

```
# 1. Reset ASG to dormant (Pilot Light ) or delete (full cleanup)
aws autoscaling update-auto-scaling-group \
  --auto-scaling-group-name PRJ-SAP-019-DR-ASG \
  --min-size 0 --desired-capacity 0 \
  --region us-west-2

# 2. Delete the DR CloudFormation stack
aws cloudformation delete-stack \
  --stack-name PRJ-SAP-019-PilotLight \
  --region us-west-2

# 3. Delete the RDS read replica (or promoted instance)
aws rds delete-db-instance \
  --db-instance-identifier prj-sap-019-dr-replica \
  --skip-final-snapshot \
  --region us-west-2

# 4. Delete the primary stack
aws cloudformation delete-stack \
  --stack-name PRJ-SAP-019-Primary \
  --region us-east-1

# 5. Delete the copied AMI and snapshots in us-west-2
# Console: EC2 → AMIs → Deregister
# Console: EC2 → Snapshots → Delete

# 6. Remove Route 53 health check and failover records
# Console: Route 53 → Health checks → Delete PRJ-SAP-019-Primary-Health
# Console: Route 53 → Hosted zones → Delete failover A records
```

Note: The AWS Backup vaults created by the CloudFormation stacks will be deleted automatically when the stacks

are deleted, provided they are empty. If they contain recovery points, you must manually delete the recovery points first before deleting the stacks.

Part of the CloudGuard 100-Project Portfolio — PRJ-SAP-019 of 100

---

11. Optional: Automating the Failover with AWS Lambda

While Route 53 handles the DNS cutover automatically, the steps in Section 7 (scaling the ASG, promoting the database, updating SSM) require manual execution. In a mature enterprise environment, these steps are automated using an Event-Driven Architecture.

You can configure an EventBridge (CloudWatch Events) rule to trigger a Lambda function the moment the Route 53 health check state changes to `ALARM`.

11.1 The Lambda Execution Role

The Lambda function needs permission to interact with Auto Scaling, RDS, and SSM in the `us-west-2` region.

1. Go to **IAM → Roles → Create role**.
2. **Trusted entity type:** AWS service → **Use case:** Lambda.
3. Attach the `AWSLambdaBasicExecutionRole` policy.
4. Create an inline policy with the following permissions:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:UpdateAutoScalingGroup",
                "rds:PromoteReadReplica",
                "rds:DescribeDBInstances",
                "ssm:PutParameter"
            ],
            "Resource": "*"
        }
    ]
}
```

1. Name the role `PRJ-SAP-019-DR-Automation-Role`.

11.2 The Lambda Function

1. Go to **Lambda → Create function** in `us-west-2`.
2. **Name:** `Automate-DR-Failover`
3. **Runtime:** Python 3.12
4. **Execution role:** Use an existing role → select `PRJ-SAP-019-DR-Automation-Role`.
5. Under **Configuration → General configuration**, increase the **Timeout** to 15 minutes (RDS promotion can take time).
6. Paste the following Python code into the editor and click **Deploy**:

```python
import boto3
import time
import os

def lambda_handler(event, context):
    region = os.environ.get('AWS_REGION', 'us-west-2')
    asg_name = 'PRJ-SAP-019-DR-ASG'
    db_id = 'prj-sap-019-dr-replica'
    param_name = '/prj-sap-019/dr/db-endpoint'

    asg_client = boto3.client('autoscaling', region_name=region)
    rds_client = boto3.client('rds', region_name=region)
    ssm_client = boto3.client('ssm', region_name=region)

    print(f"Starting automated DR failover in {region}...")

    # 1. Scale up the ASG
    print(f"Scaling up ASG {asg_name} to 2 instances...")
    asg_client.update_auto_scaling_group(
        AutoScalingGroupName=asg_name,
        MinSize=2,
        DesiredCapacity=2
    )

    # 2. Promote the RDS Read Replica
    print(f"Promoting RDS Read Replica {db_id}...")
    try:
        rds_client.promote_read_replica(DBInstanceIdentifier=db_id)
    except rds_client.exceptions.InvalidDBInstanceStateFault as e:
        print(f"Note: {e} - it may already be promoted or in modifying
state.")

    # Wait for the DB to become available and get the new endpoint
    print("Waiting for RDS instance to become available...")
    waiter = rds_client.get_waiter('db_instance_available')
    waiter.wait(
        DBInstanceIdentifiers=[db_id],
        WaiterConfig={'Delay': 30, 'MaxAttempts': 40}
    )

    # Get the new endpoint
    response = rds_client.describe_db_instances(DBInstanceIdentifier=db_id)
    new_endpoint = response['DBInstances'][0]['Endpoint']['Address']
    print(f"RDS is available. New endpoint: {new_endpoint}")
```

```
    # 3. Update SSM Parameter Store
    print(f"Updating SSM Parameter {param_name}...")
    ssm_client.put_parameter(
        Name=param_name,
        Value=new_endpoint,
        Type='SecureString',
        Overwrite=True
    )

    print("Automated DR failover complete.")
    return {
        'statusCode': 200,
        'body': f'Failover successful. New DB endpoint: {new_endpoint}'
    }
```

11.3 The EventBridge Trigger

To wire it up automatically:

1. Create a CloudWatch Alarm in `us-east-1` based on the Route 53 Health Check metric `HealthCheckStatus` (Alarm when < 1).
2. Create an EventBridge Rule in `us-west-2` that listens for that specific CloudWatch Alarm state change.
3. Set the target of the EventBridge Rule to the `Automate-DR-Failover` Lambda function.

Now, when you run the disaster simulation in Step 7.1, the entire recovery sequence will execute without human intervention, reducing your RTO to the absolute minimum time required by the AWS control plane.