

# Comprehensive Guide to Setting Up AWS Disaster Recovery Service

## Table of Contents

### 1. Introduction

- Overview of AWS Disaster Recovery
- Benefits of Disaster Recovery
- Importance of Disaster Recovery Planning

### 2. Pre-Requisites

- AWS Account Setup
- IAM User Creation
- Understanding AWS Services Involved

### 3. Setting Up the Replication Server

- Step 1: Configure the Replication Server
- Step 2: Volume and Security Groups
- Step 3: Additional Settings
- Step 4: Review Page

### 4. Installing the DRS Agent on the Source Server

- Connecting to the Source Server
- Pre-Installation Checks
- Agent Installation Steps

### 5. Monitoring and Verification

- DRS Console Overview
- Checking Replication Progress
- Troubleshooting Common Issues

### 6. Initiating Recovery

- Starting the Drill
- Creating Recovery Instances
- Verifying Recovery Instances

## 7. Failover and Failback Process

- Overview of Failover and Failback
- Detailed Steps for Failover
- Detailed Steps for Failback

## 8. Disconnection and Cleanup

- Disconnection from AWS
- Deleting Recovery Instances
- Final Cleanup of Resources

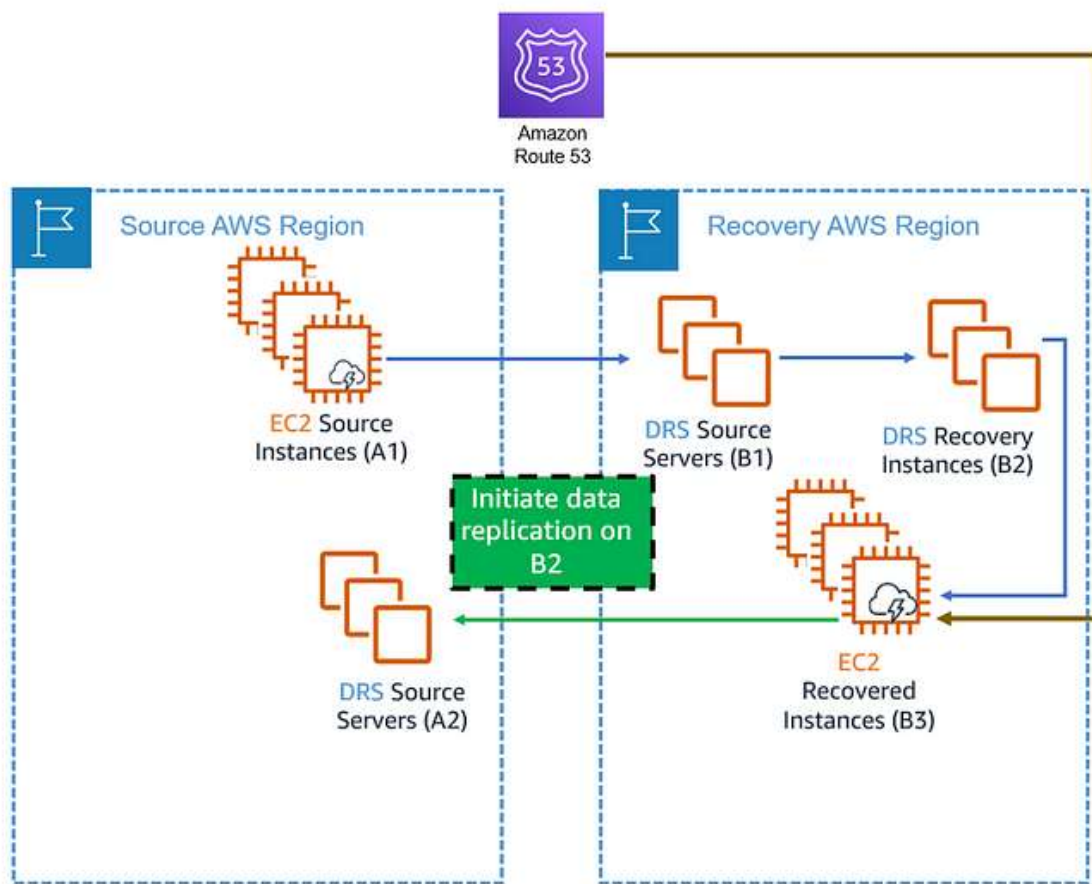
## 9. Best Practices for Disaster Recovery

- Regular Testing of Disaster Recovery Plan
- Documentation and Training
- Continuous Monitoring and Updates

## 10. Conclusion

- Summary of Steps
- Final Thoughts on Disaster Recovery

---



## 1. Introduction

### Overview of AWS Disaster Recovery

AWS Disaster Recovery (DRS) is a vital service offered by Amazon Web Services that enables businesses to protect their data and applications from unexpected failures and disasters. By replicating workloads across different geographic regions, organizations can ensure that they can quickly recover operations with minimal downtime.

Disaster recovery is essential for maintaining business continuity, particularly in an era where data breaches and system outages are becoming increasingly common. AWS DRS allows organizations to implement a comprehensive strategy that minimizes risks and enhances resilience.

### Benefits of Disaster Recovery

#### - Reduced Downtime:

With a well-implemented disaster recovery plan, businesses can significantly reduce the amount of downtime experienced during an outage.

#### - Cost-Effective Solutions:

AWS offers a pay-as-you-go pricing model, allowing organizations to manage costs effectively by only paying for resources used during a recovery operation.

#### - Scalability:

AWS services are designed to scale with your business needs. Whether you are a small start-up or a large enterprise, AWS DRS can grow alongside your organization.

#### - Flexibility:

Organizations can choose from a variety of recovery options based on their specific needs, including point-in-time recovery and continuous data protection.

#### - Compliance:

Many industries have regulatory requirements for data protection and disaster recovery. AWS DRS can help organizations meet these compliance mandates.

### Importance of Disaster Recovery Planning

Disaster recovery planning is not just about technology; it's about ensuring business continuity and protecting critical assets. A well-structured disaster recovery plan can save organizations from financial losses, reputational damage, and legal repercussions. Regular updates and testing of the plan ensure its effectiveness in real-world scenarios.

---

## 2. Pre-Requisites

### AWS Account Setup

Before implementing AWS Disaster Recovery, ensure you have an AWS account set up with sufficient permissions. If you do not have an account, follow these steps:

#### 1. Sign Up for an AWS Account:

- Go to the [AWS homepage](<https://aws.amazon.com/>).
- Click on “Create an AWS Account” and follow the instructions.
- Enter your email, password, and account name.

#### 2. Select a Support Plan:

- Choose a support plan that fits your organization's needs. Basic support is free

#### 3. Set Up Billing Information:

- Enter your payment details. AWS uses a pay-as-you-go model, so you will only be charged for the resources you use.

### IAM User Creation

Creating an IAM (Identity and Access Management) user with appropriate permissions is essential for managing AWS resources securely.

### 1. Navigate to the IAM Console:

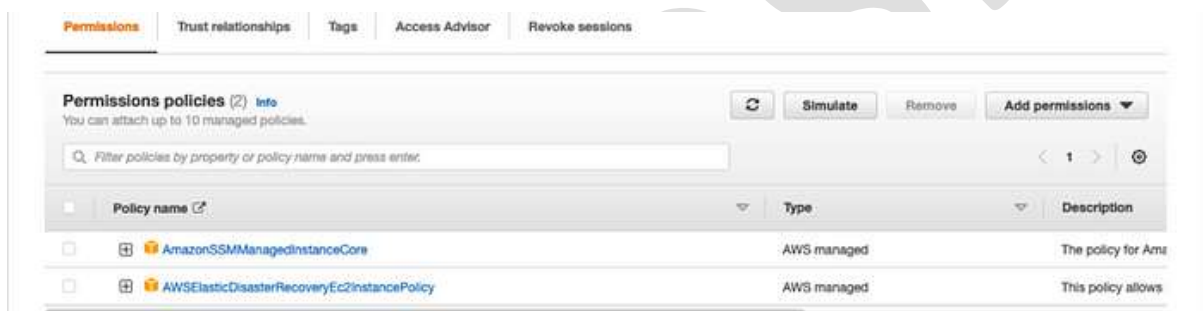
- Log in to your AWS Management Console and search for IAM.

### 2. Create a New User:

- Click on “Users” and then “Add user.”
- Provide a username and select “Programmatic access” for access type.

### 3. Attach Policy:

- Attach the policy named `Elastic Disaster Recovery Agent Installation Policy` to grant the necessary permissions.



### 4. Complete User Creation:

- Review your settings and create the user. Make sure to save the access key ID and secret access key as you will need them later.

## Understanding AWS Services Involved

Familiarize yourself with the AWS services that are integral to disaster recovery:

- Amazon EC2 (Elastic Compute Cloud): Provides resizable compute capacity in the cloud.
- Amazon EBS (Elastic Block Store): Offers persistent block storage for EC2 instances.
- Amazon S3 (Simple Storage Service): Ideal for storing and retrieving any amount of data at any time.
- AWS Cloud Formation: Automates the deployment of AWS resources in a systematic way.

- AWS IAM: Manages user access and permissions securely.

---

### 3. Setting Up the Replication Server

#### Step 1: Configure the Replication Server

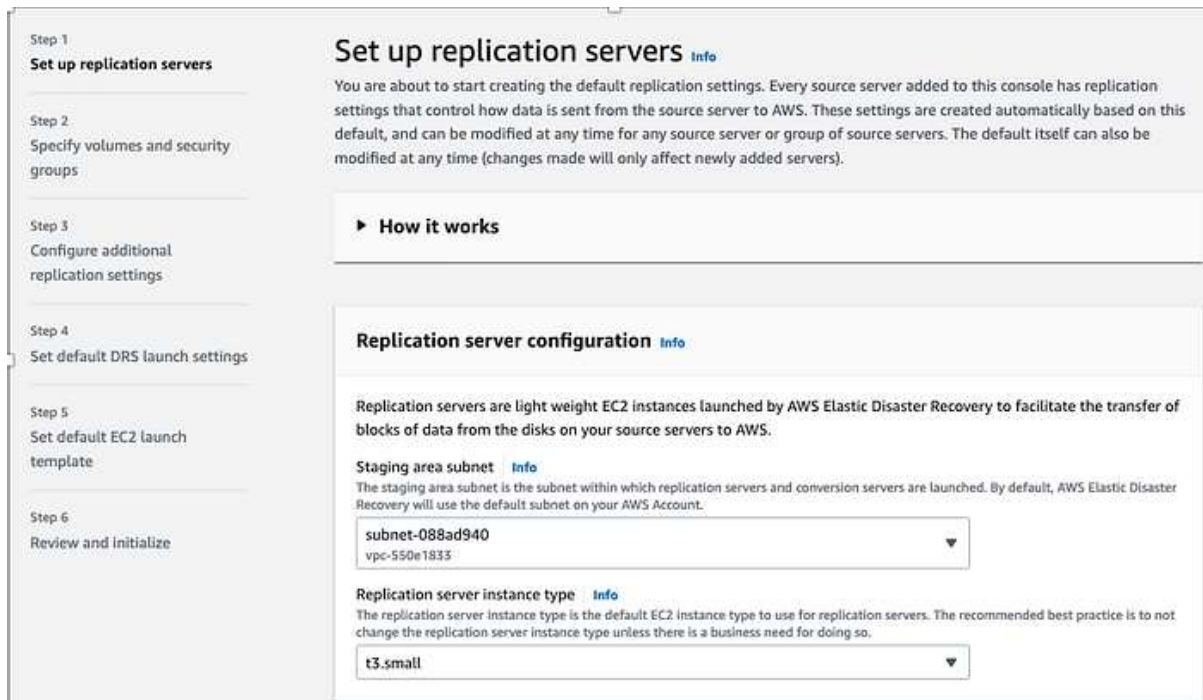
##### 1. Access Disaster Recovery Service:

- Log into the AWS Management Console and navigate to the Disaster Recovery Service.



##### 2. Setup Replication Server:

- Click on the option to set up a new replication server.
- Select the desired subnet in the staging area where the replicated data will be stored.



**Step 1**  
**Set up replication servers**

**Step 2**  
Specify volumes and security groups

**Step 3**  
Configure additional replication settings

**Step 4**  
Set default DRS launch settings

**Step 5**  
Set default EC2 launch template

**Step 6**  
Review and initialize

## Set up replication servers [Info](#)

You are about to start creating the default replication settings. Every source server added to this console has replication settings that control how data is sent from the source server to AWS. These settings are created automatically based on this default, and can be modified at any time for any source server or group of source servers. The default itself can also be modified at any time (changes made will only affect newly added servers).

► **How it works**

### Replication server configuration [Info](#)

Replication servers are light weight EC2 instances launched by AWS Elastic Disaster Recovery to facilitate the transfer of blocks of data from the disks on your source servers to AWS.

**Staging area subnet** [Info](#)  
The staging area subnet is the subnet within which replication servers and conversion servers are launched. By default, AWS Elastic Disaster Recovery will use the default subnet on your AWS Account.

subnet-088ad940  
vpc-550e1833

**Replication server instance type** [Info](#)  
The replication server instance type is the default EC2 instance type to use for replication servers. The recommended best practice is to not change the replication server instance type unless there is a business need for doing so.

t3.small

### 3. Specify Region and Availability Zone:

- Choose the region and availability zone based on your organization's needs. Consider proximity to your source server for latency optimization.

### Step 2: Volume and Security Groups

#### 1. Select Volume:

- Choose the Elastic Block Store (EBS) volumes that you want to replicate. This includes the root volume and any additional data volumes.

#### 2. EBS Encryption:

- Ensure that EBS encryption is enabled for security. This can be set as the default for all new volumes.

#### 3. Configure Security Groups:

- Mark the Security Group checkbox. This will automatically create a security group tailored to your replication needs.

### Step 3: Additional Settings



## 1. Data Routing Throttling:

- Configure data routing throttling to manage bandwidth usage during replication, ensuring that other applications maintain performance.

AWS Elastic Disaster Recovery > Set up AWS Elastic Disaster Recovery

Step 1  
Set up replication servers

Step 2  
Specify volumes and security groups

Step 3  
**Configure additional replication settings**

Step 4  
Set default DRS launch settings

Step 5  
Set default EC2 launch template

Step 6  
Review and initialize

### Configure additional replication settings

#### Data routing and throttling [Info](#)

This setting controls how data flows from the external server to the replication servers. If you choose not to use a private IP, your replication servers will be automatically assigned a public IP and data will flow over the public internet.

☐ Use private IP for data replication (VPN, DirectConnect, VPC peering)

☒ Create public IP

☐ Throttle network bandwidth (per server - in Mbps)

#### Point in time (PIT) policy [Info](#)

Point in Time (PIT) is a disaster recovery feature which allows launching an instance from a snapshot captured at a specific point in time. As source servers are replicated, snapshots are taken over time. This section allows to configure a retention policy that will determine which snapshots are not required after a defined duration.

Snapshot retention (in days)

7

Valid values: (1 - 365)

#### MAP program tagging [Info](#)

## 2. Point-in-Time Policy:

- Define a point-in-time recovery policy to establish how frequently you want to create snapshots of your data.

## 3. Snapshot Retention Policies:

- Set retention policies for snapshots to control how long backups are kept. This can help manage storage costs.

Step 1

Set up replication servers

Step 2

Specify volumes and security groups

Step 3

Configure additional replication settings

Step 4

Set default DRS launch settings

Step 5

Set default EC2 launch template

Step 6

Review and initialize

## Set default EC2 launch template [Info](#)

Every source server added to DRS has an EC2 launch template that affects how instances are launched into AWS. You can modify the default settings at any time, but changes will only affect new servers.

### Basic settings

If you do not include a setting, the default value will be used.

Subnet

Associate the subnet with the launched instance.

subnet-abcde0123  
vpc-01234567890abcd us-east-1b

Security groups

Associate the security groups with the launched instance.

Select security groups

Instance type

Use the instance type for the launched instance.

Using instance type right-sizing

EBS volume type

Use the EBS volume type for all volumes of the launched instance.

Cold HDD (sc1)

▼ Advanced settings

Additional fields that add optional capabilities, including IAM instance profile, tenancy, user data, and reservation configuration. If you do not include a setting, the specific capability will be excluded.

IAM instance profile

Automatically assign an instance profile to the launched instance.

## Step 4: Review Page

- Review all configurations before proceeding to ensure that they meet your organization's disaster recovery strategy.

Step 1

Set up replication servers

Step 2

Specify volumes and security groups

Step 3

Configure additional replication settings

Step 4

Set default DRS launch settings

Step 5

Set default EC2 launch template

Step 6

Review and initialize

Review and initialize

Step 1: Replication servers

Replication server configuration

Subnet

subnet-b1dba347 (us-east-1c)

Replication server instance type

t3.small

Step 2: Volumes and security groups

Volumes

EBS volume type (for replicating disks over 125 GiB)

Auto volume type selection

EBS encryption

Default

Security groups

Always use Elastic Disaster Recovery security group

Yes

Additional security groups

None

Step 3: Additional replication settings

Data routing and throttling

---

## 4. Installing the DRS Agent on the Source Server

### Connecting to the Source Server

#### 1. Access the Source Server:

- Use SSH or Remote Desktop Protocol (RDP) to connect to your source server.

#### 2. Check Existing Files:

- Verify that you are in the correct environment by checking for existing files or configurations.

11

## Pre-Installation Checks

### 1. Operating System Compatibility:

- Ensure that your operating system is compatible with the DRS agent. AWS DRS supports multiple Linux distributions and Windows Server.

### 2. Python Installation:

- Ensure that Python 3 is installed on the source server, as it is required for running the DRS agent installation script.

## Agent Installation Steps

### 1. Download the DRS Documentation:

- Visit the AWS DRS documentation site and find the section on agent installation instructions for source servers, specifically for your operating system.

Use `wget` or `curl` to download the installation script directly to your server. For example:

```
bash
Copy code
wget https://link-to-agent-installation-script
```

### 2. Modify Installation Instructions:

- Replace any placeholders in the installation instructions with your specific region and availability zone.

### 3. Run the Installation Command:

- Open a terminal or command prompt and execute the following command:

```
```bash
sudo python3 aws-replication-installer-init.py
```
```

#### 4. Provide Required Information:

- Input the source region (e.g., `us-east-1`) when prompted.
- Enter the access key and secret key for the IAM user created earlier.

#### 5. Select Disk for Replication:

- Choose whether to replicate all data or specific drives based on your organization's needs.

#### 6. Wait for Installation:

- The system will proceed to download and install the AWS replication agent. This may take a few minutes.

#### 7. Confirmation of Installation:

- Once the installation is complete, note the replication ID provided in the output. This confirms that the agent was successfully installed.

---

#### 5. Monitoring and Verification

##### DRS Console Overview

##### 1. Check DRS Console:

- Open the DRS console to verify that a new server instance appears, indicating that the replication process has begun.

##### 2. Hostname Verification:

- Check the hostname to ensure it matches the private IP of the source server. This confirms that the setup is correct.

##### Checking Replication Progress

##### 1. Monitor Replication Status:

- In the DRS console, you can monitor the status of replication. Look for indicators that show whether the data is currently being replicated successfully.

## 2. Troubleshooting Common Issues:

- If there are issues with replication, review the logs provided in the DRS console for error messages. Common issues may include network connectivity problems or misconfigured security groups.

## 3. Verification of EBS Snapshots:

- Go to the EC2 console and check

the EBS snapshots to ensure that snapshots are being created as per your defined policies.

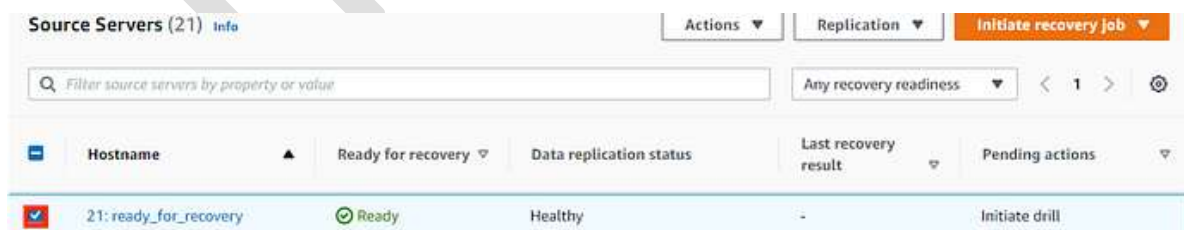
---

## 6. Initiating Recovery

### Starting the Drill

#### 1. Initiate a Recovery Drill:

- In the DRS console, select the option to initiate a recovery drill. This allows you to test the recovery process without impacting your production environment.



| Source Servers (21) <a href="#">Info</a>                                |                    | Actions                      | Replication          | Initiate recovery job |
|---|--------------------|------------------------------|----------------------|-----------------------|
| <input type="text" value="Filter source servers by property or value"/> |                    | Any recovery readiness < 1 > |                      |                       |
| Hostname  | Ready for recovery | Data replication status      | Last recovery result | Pending actions       |
| 21: ready_for_recovery  | Ready              | Healthy                      | -                    | Initiate drill        |



## 2. Region Selection:

- Confirm the region where the recovery instance will be created. This should be a different region from your source server.

## Creating Recovery Instances

### 1. Select Snapshot for Recovery:

- You will be prompted to select a snapshot to use for recovery. Choose the most recent snapshot for minimal data loss.

### 2. Monitor EC2 Console:

- Open the EC2 console to verify the creation of the new recovery instance. This instance will be labeled accordingly, indicating it is a disaster recovery instance.

### 3. Instance Initialization:

- Wait for the recovery instance to initialize. This process may take several minutes. Ensure that the instance is running before proceeding.

## Verifying Recovery Instances

### 1. Access the Recovery Instance:

- Use SSH or RDP to access the newly created recovery instance and verify that the application and data are intact.

| Recovery instances (1) <a href="#">Info</a>  |                                     |                                 |                         |                 |  |                  |                |
|--|-------------------------------------|---------------------------------|-------------------------|-----------------|--|------------------|----------------|
| <input type="text" value="Filter recovery instances by property or value"/> <span>&lt; 1 &gt; ⚙</span> |                                     |                                 |                         |                 |  |                  |                |
| <input type="checkbox"/>   | Instance ID                         | Reversed direction launch state | Data replication status | Pending actions | Replicating to source server                   | Replicating from | Replicating to |
| <input type="checkbox"/>   | <a href="#">i-05ae4eacc1f7847b7</a> | ⏸ Not started                   | Stalled                 | -               | <a href="#">us-east-1; s-3c9fc68c27b52961f</a> | us-west-2d       | us-east-1f     |

## 2. Test Application Functionality:

- Run tests to ensure that the applications hosted on the recovery instance are functioning as expected.

## 3. Logging and Monitoring:

- Check logs in the recovery instance to verify that there were no errors during the recovery process.

| Job log <a href="#">Info</a>   |   |  |  |
|--|---|--|--|
| <input type="text" value="Filter job log by property or value"/> <span>&lt; 1 2</span> |   |  |  |
| Time   | Event                                   | Additional data  |  |
| 05/01/2024, 18:22:16   | Job ended                               |  |  |
| 05/01/2024, 18:22:15   | Successfully launched recovery instance | Source server: <a href="#">EC2AMAZ-NETGR8F (s-5752922b8715e944b)</a><br>Recovery instance ID: <a href="#">i-05ae4eacc1f7847b7</a>          |  |
| 05/01/2024, 18:17:10   | Started launching recovery instance     | Source server: <a href="#">EC2AMAZ-NETGR8F (s-5752922b8715e944b)</a>   |  |
| 05/01/2024, 18:17:09   | Conversion ended                        | Source server: <a href="#">EC2AMAZ-NETGR8F (s-5752922b8715e944b)</a><br>Conversion server instance ID: <a href="#">i-03ad64e4d34b4dca9</a> |  |
| 05/01/2024, 18:08:31   | Conversion started                      | Source server: <a href="#">EC2AMAZ-NETGR8F (s-5752922b8715e944b)</a>   |  |
| 05/01/2024, 18:07:51   | Finished taking snapshot                | Source server: <a href="#">EC2AMAZ-NETGR8F (s-5752922b8715e944b)</a>   |  |
| 05/01/2024, 18:07:50   | Started taking snapshot                 | Source server: <a href="#">EC2AMAZ-NETGR8F (s-5752922b8715e944b)</a>   |  |

---

## 7. Failover and Failback Process

### Overview of Failover and Failback



Failover is the process of switching operations from the primary site to a backup site when a failure occurs.

Failback refers to the process of restoring operations back to the primary site once it is operational again.

### Detailed Steps for Failover

#### 1. Initiate Failover:

- In the DRS console, select the option to initiate failover. Confirm your choice to switch operations to the recovery instance.

#### 2. Update DNS Records:

- If necessary, update DNS records to point to the new recovery instance. This step is crucial to redirect traffic to the new server.

#### 3. Monitor the Failover Process:

- Keep an eye on the DRS console to monitor the status of the failover. Ensure that all services are running correctly on the recovery instance.

#### 4. Notify Stakeholders:

- Inform relevant stakeholders of the failover to keep them updated on the situation.

### Detailed Steps for Failback

#### 1. Prepare the Primary Site:

- Ensure that the primary site is operational again. This may involve restoring data from backups and fixing underlying issues that caused the failover.

#### 2. Reinstall DRS Agent (if needed):

- If the DRS agent was removed from the primary site during the failover process, reinstall it following the installation steps previously outlined.

#### 3. Initiate Failback:

- In the DRS console, select the option to initiate failback. Confirm your choice to switch operations back to the primary site.

#### 4. Monitor the Failback Process:

- Keep an eye on the progress of the failback to ensure a smooth transition back to the primary site.

#### 5. Validate Data Integrity:

- Once the failback is complete, validate that all data has been restored correctly and that applications are functioning as expected.

---

#### 8. Disconnection and Cleanup

##### Disconnection from AWS

##### 1. Disconnect Recovery Instance:

- In the DRS console, select the recovery instance and click on "Actions" > "Disconnect from AWS." This step ensures that the instance is no longer managed by DRS.

##### 2. Delete Recovery Instance:

- Confirm the deletion of the recovery instance. This action helps in managing costs and resource allocation.

##### Final Cleanup of Resources

##### 1. Disconnect from AWS on Source Server:

- If no longer needed, disconnect the source server from AWS. This may involve removing the DRS agent or disabling replication.

##### 2. Delete the Source Instance:

- If the source instance is no longer required, delete it from the EC2 console.

### 3. Clean Up Additional Resources:

- Go to the EC2 console and delete any remaining servers, snapshots, or EBS volumes associated with the disaster recovery setup.

### 4. Review Billing Dashboard:

- Check the AWS billing dashboard to ensure that you are not being charged for any leftover resources.

---

## 9. Best Practices for Disaster Recovery

### Regular Testing of Disaster Recovery Plan

Regularly test your disaster recovery plan to ensure that it remains effective and up to date. Schedule drills at least once or twice a year and incorporate learnings into your strategy.

### Documentation and Training

Document all aspects of your disaster recovery plan, including procedures for failover and failback. Train your staff on these procedures to ensure that everyone knows their roles in the event of a disaster.

### Continuous Monitoring and Updates

Continuously monitor your AWS environment for changes that could impact your disaster recovery plan. Update your plan regularly to account for new applications, data, and infrastructure changes.

---

## 10. Conclusion

## Summary of Steps

This comprehensive guide has provided a detailed step-by-step process for setting up AWS Disaster Recovery, including the installation of necessary agents, monitoring replication progress, and initiating recovery. Following these steps will help you implement an effective disaster recovery strategy to protect your organization's data and applications.

## Final Thoughts on Disaster Recovery

Disaster recovery planning is an ongoing process that requires continuous evaluation and adaptation. By leveraging AWS DRS, organizations can build a robust disaster recovery strategy that minimizes downtime and ensures business continuity. Regular testing, documentation, and staff training are essential to maintaining an effective disaster recovery plan.