Light Disaster Recovery (DR) Solution Plan in Azure Using Terraform and Scripting

# 1. Assessment and Planning

## 1.1 Identify Critical Components

Determine the critical servers, databases, and applications that need to be included in the DR plan. Document their current configurations and dependencies.

## 1.2 Define Recovery Objectives

RPO (Recovery Point Objective): Establish the maximum acceptable amount of data loss (e.g., 1 hour).  
RTO (Recovery Time Objective): Establish the maximum acceptable downtime (e.g., 2 hours).

## 1.3 Select Azure Regions

Choose a primary Azure region where your current data center operates (e.g., West US). Select a secondary Azure region for the DR site (e.g., East US).

# 2. Define Infrastructure in Terraform

## 2.1 Resource Groups

Create resource groups in both the primary and secondary regions to organize resources.

## 2.2 Virtual Networks and Subnets

Define virtual networks and subnets in both regions to ensure network isolation and organization.

## 2.3 Network Interfaces and Virtual Machines

Set up network interfaces and virtual machines in the primary region. Plan the configuration for VMs, including OS details, sizing, and network settings.

# 3. Set Up Azure Resources

## 3.1 Initialize Terraform Project

Create and organize Terraform configuration files (main.tf, variables.tf, outputs.tf). Define variables for resource names, locations, and other configurations.

## 3.2 Deploy Resources

Use Terraform to deploy the infrastructure in the primary region. Validate the deployment to ensure all resources are correctly provisioned.

# 4. Configure Backup and Replication

## 4.1 Azure Backup Configuration

Create a Recovery Services vault in the primary region. Define backup policies (e.g., daily backups with 7-day retention). Apply backup policies to critical VMs.

## 4.2 Azure Site Recovery (ASR)

Configure ASR to replicate VMs from the primary region to the secondary region. Set up ASR replication policies and recovery plans.

# 5. Testing and Validation

## 5.1 Failover Testing

Conduct a test failover to ensure the DR setup functions correctly. Validate that applications and services operate as expected in the secondary region.

## 5.2 Data Integrity Check

Verify data integrity post-failover. Ensure that no critical data is lost and that data consistency is maintained.

# 6. Automation with Scripting

## 6.1 Failover and Failback Scripts

Develop scripts to automate the failover process using Azure CLI or PowerShell. Create scripts to automate the failback process to revert services to the primary region.

## 6.2 Scheduled Automation

Set up automation scripts to run on a schedule or trigger based on specific conditions.

# 7. Documentation and Maintenance

## 7.1 Document the DR Plan

Create detailed documentation for the DR setup, including:  
- Steps for failover and failback  
- Backup and replication configurations  
- Contact information for key personnel

## 7.2 Regular Testing

Schedule regular DR tests (e.g., quarterly) to ensure the setup remains effective. Document the results of each test and make necessary adjustments.

## 7.3 Monitoring and Updating

Continuously monitor the DR setup for any issues or changes in requirements. Update Terraform scripts and DR documentation as needed to reflect changes in the infrastructure or business requirements.