Disaster Recovery Project Documentation

Project Overview

Objective

The primary objective of this project is to establish a robust disaster recovery plan using IBM Cloud Virtual Servers. This plan aims to ensure the continuity of critical business operations in the event of unforeseen disasters or disruptions.

Design Thinking Process

The design thinking process followed in this project includes comprehensive steps:

1. **Problem Definition and Design Thinking:** Understanding the project requirements, defining the disaster recovery strategy, backup configuration, replication setup, and recovery testing procedures.
2. **Transformation:** Implementing the designed plan, configuring replication, conducting recovery tests, and aligning the plan with business continuity strategies.

Development Phases

The project progresses through the following development phases:

1. **Phase 1: Problem Definition and Design Thinking:** Defining the disaster recovery strategy, backup configuration, replication setup, and recovery testing procedures.
2. **Phase 2: Transformation:** Implementing the plan, testing, aligning with business continuity, and ensuring maintenance and updates.

Disaster Recovery Strategy

The disaster recovery strategy includes:

- **Defined RTO and RPO:** Recovery Time Objectives (RTO) and Recovery Point Objectives (RPO) to determine maximum acceptable downtime and data loss.
- **Backup Configuration:** Regular backups of critical data and configurations.
- **Replication Setup:** Synchronization of data and virtual machine images from on-premises systems to IBM Cloud Virtual Servers.
- **Recovery Testing Procedures:** Comprehensive testing of recovery procedures to validate the effectiveness of the plan.

Business Continuity Assurance

The disaster recovery plan guarantees business continuity by:

- Minimizing downtime through well-defined RTO and RPO objectives.
- Ensuring regular backups and replication to secure critical data and configurations.

- Conducting rigorous recovery tests to validate the effectiveness of the plan in various disaster scenarios.

Setup and Deployment Instructions

To set up and deploy the disaster recovery plan using IBM Cloud Virtual Servers:

1. Clone the repository from the provided link.
2. Follow the instructions in the README file for step-by-step deployment guidelines.
3. Ensure that all dependencies are installed and configured as specified.

README File

The README file in the repository contains:

- Detailed navigation guidelines for the disaster recovery plan.
- Instructions on updating content and configurations.
- Dependency details and setup guidelines.

```
# Example: Connecting to IBM Cloud Virtual Server using Python SDK

from ibm_cloud import VirtualServer


# Connect to IBM Cloud

ibm_cloud = VirtualServer(api_key='YOUR_API_KEY', region='us-south')

ibm_cloud.connect()


# Backup a specific dataset to IBM Cloud

backup_dataset(dataset_name='important_data', destination='ibm_cloud_virtual_server')


# Replication setup

replicate_data(source='on_premises_server', destination='ibm_cloud_virtual_server')


# Test recovery procedure

simulate_disaster_scenario('network_failure')

recovery_procedure.execute()


# Align disaster recovery plan with business objectives

align_with_bc_strategy()
```

```
generate_report()




from ibm_cloud import IBMCloudClient

# Connect to IBM Cloud Virtual Servers
ibm_cloud_virtual_server = IBMCloudClient(api_key='YOUR_API_KEY', service='virtual_server')

# Function to configure replication to IBM Cloud Virtual Servers
def configure_replication(source_server, target_virtual_server):
    # Here, 'source_server' refers to your on-premises server or another existing server
    # 'target_virtual_server' refers to the IBM Cloud Virtual Server you want to replicate to

    replication_config = {
        'source_server': source_server,
        'target_virtual_server': target_virtual_server,
        # Add other necessary parameters like replication method, frequency, etc.
    }

    # Call the IBM Cloud SDK function to set up the replication
    ibm_cloud_virtual_server.configure_replication(replication_config)

# Execute the function with required data
configure_replication(source_server='on_premises_server',
target_virtual_server='ibm_cloud_virtual_server')
```

```
from ibm_cloud import IBMCloudClient

# Connect to IBM Cloud Virtual Servers
ibm_cloud_virtual_server = IBMCloudClient(api_key='YOUR_API_KEY', service='virtual_server')

# Simulate a disaster scenario and perform recovery testing
def simulate_and_test_recovery(disaster_scenario):
    # Simulate disaster scenarios (e.g., network failure, server crash)
    if disaster_scenario == 'network_failure':
        # Code to simulate a network failure
        # Trigger recovery procedures
        recovery_process.execute()

    # You can add more conditions for various disaster scenarios and recovery testing

# Execute recovery testing for a specific disaster scenario
simulate_and_test_recovery('network_failure')
```