KGiSL INSTITUTE OF TECHNOLOGY

Coimbatore – 641035

Institution code :7117

Disaster recovery with IBM cloud virtual servers

MENTOR:

MRS.INDU POORNIMA.R

TEAM MEMBERS:

DHAKSHATA.R

ANANTHI.C

ARUNA.S

HARINI.V

Disaster recovery with IBM cloud virtual servers

Problem Definition:

The objective of this project is to establish a comprehensive disaster recovery plan utilizing IBM Cloud Virtual Servers. The primary aim is to ensure the continuity of business operations in the face of unforeseen events that may disrupt our on-premises virtual machine infrastructure. This disaster recovery plan encompasses multiple phases, including defining the disaster recovery strategy, configuring backup and replication processes, validating recovery procedures, and ultimately ensuring seamless business continuity.

Disaster Recovery Strategy:

Disaster recovery strategy involves establishing key parameters and prioritizing your virtual machines to ensure business continuity in the event of a disaster.

Recovery Time Objective (RTO):

RTO is the maximum allowable downtime for your systems, applications, and virtual machines in the event of a disaster. It defines the time within which these resources must be restored to normal operation to minimize the impact on your organization. To set RTO:

- **Identify Critical Applications**: Start by identifying the critical applications and services that your organization relies on for day-to-day operations. These are the systems that need the shortest RTO.
- **Assess Tolerable Downtime**: For each critical application, determine the maximum tolerable downtime. This can vary based on the application's criticality. For example, a customer-facing e-commerce platform may have a very short RTO, while a less critical internal application might have a longer RTO.
- **Document RTO Values**: Document the RTO values for each application or system.

Recovery Point Objective (RPO):

RPO is the maximum acceptable data loss in the event of a disaster. It defines the point in time to which data must be restored. To set RPO:

- **Determine Data Criticality**: Evaluate the importance of data associated with each application. For critical applications, data loss should be minimized.
- **Backup Frequency**: Define how frequently backups or snapshots should be taken for each application. This frequency should align with the application's data sensitivity and business requirements. For critical applications, you may need near real-time backups.
- **Document RPO Values**: Document the RPO values for each application.

Priority of Virtual Machines:

Determining the priority of virtual machines helps in planning the sequence of recovery during a disaster. Virtual machines are categorized into tiers based on their criticality:

- **Tier 1 (Critical):** These virtual machines host the most critical applications and services with the shortest RTO and RPO. Failover for Tier 1 VMs should be a top priority.
- **Tier 2 (Important):** VMs in this category host important but less critical applications. Their RTO and RPO are more lenient than Tier 1 but still significant.
- **Tier 3 (Less Critical):** These VMs are associated with non-critical or secondary services. Their RTO and RPO can be relaxed.
- **Tier N (Non-Essential):** Virtual machines in this category can have a longer RTO and RPO, as they are not critical for immediate recovery.

Code for developing a basic Disaster Recovery on IBM Cloud Virtual Server:

This PowerShell script connects to a Veeam Backup & Replication server, defines virtual machines to be backed up, creates a backup job, and schedules the backups.

```python
import requests

# Veeam Backup & Replication server details
veeam_server = "https://your-veeam-server:9398"
username = "your-username"
password = "your-password"

# Define VMs to be backed up
vms_to_backup = ["VirtualMachine1", "VirtualMachine2"]

# Create a backup job
backup_job_name = "BackupJob"

# Authentication and session setup
session = requests.Session()
session.auth = (username, password)
session.verify = False  # You may want to set this to True in a production environment with a valid SSL certificate.

# Create a backup job
job_creation_url = f"{veeam_server}/api/jobs"
job_data = {
    "Name": backup_job_name,
    "Objects": [{"Name": vm} for vm in vms_to_backup],
    # Additional job settings can be added here
}

response = session.post(job_creation_url, json=job_data)

if response.status_code == 201:
    print(f"Backup job '{backup_job_name}' created successfully.")
else:
    print(f"Failed to create backup job. Status code: {response.status_code}")
    print(response.text)
```

The below code is a simplified Python code snippet that demonstrates how you can use the IBM Cloud Python SDK to work with IBM Cloud Virtual Servers.

Step:1

You can use the IBM Cloud Python SDK to manage virtual server instances, create snapshots, and automate failover.

Step:2

Remember to install the required SDK if you haven't already.

Step:3

pip install ibm-cos-sdk

```python
import ibm_boto3
from ibm_botocore.client import Config

# Set your IBM Cloud credentials and region
api_key = 'YOUR_API_KEY'
instance_id = 'YOUR_INSTANCE_ID'
region = 'us-south'  # Change this to your desired region

# Initialize the IBM Cloud Virtual Servers client
cos = ibm_boto3.resource("s3",
    ibm_api_key_id=api_key,
    ibm_service_instance_id=instance_id,
    config=Config(signature_version="oauth"),
    endpoint_url=f"https://s3.{region}.cloud-object-storage.appdomain.cloud"
)

# List your virtual server instances
for instance in cos.buckets.all():
    print(f"Virtual Server Name: {instance.name}")

# You can use the IBM Cloud Python SDK to manage virtual server instances, create snapshots, and automate failover.

# Remember to install the required SDK if you haven't already:
# pip install ibm-cos-sdk
```

This code demonstrates how to initialize the IBM Cloud Python SDK and list virtual server instances. It's a starting point for interacting with IBM Cloud resources for a disaster recovery plan.

The below diagram is the Overall architecture of our Project: