SRPCE

2021-2025

NAME - R.Kaviya

DEPARTMENT-CSE

ROLL-NO - 14

SUBJECT - cloud Application Development

REG-NO - 422021104014

SEMESTER- 5

IBM Cloud Disaster Recovery

Write a providers of disaster recovery with IBM cloud virtual servers with sample program

I can provide you with a general outline of how to set up disaster recovery using IBM Cloud virtual servers and a sample program for reference. Please note that specific configurations may vary based on your needs and the services provided by IBM at the time of your setup. Always refer to the latest IBM documentation for the most accurate information.

1. Set Up Disaster Recovery in IBM Cloud:

- Create two virtual server instances in different data centers or regions. One will serve as the primary server, and the other as the backup or failover server.
- Configure data replication or synchronization between the primary and backup servers. IBM Cloud offers various solutions for this, such as block storage replication or database synchronization, depending on your application's requirements.
- Implement a load balancer or traffic manager to route traffic to the primary server by default, but switch to the backup server in case of a disaster.
- Monitor the health of the primary server and automate the failover process when needed.

2. Sample Program:

Below is a simplified Python script to illustrate how you can automate the failover process in the event of a disaster. This is a basic example and should be adapted to your specific setup and requirements.

python Copy code

```python
import requests

# Define the URLs of the primary and backup servers
primary_server_url = 'http://primary-server-ip'
backup_server_url = 'http://backup-server-ip'
```

```python
# Function to check the health of the primary server
def is_primary_server_healthy():
    try:
        response = requests.get(primary_server_url, timeout=5)
        return response.status_code == 200
    except requests.exceptions.RequestException:
        return False

# Function to perform failover to the backup server
def perform_failover():
    print("Performing failover to the backup server...")
    # Redirect traffic to the backup server using your load balan

# Main program
if __name__ == '__main__':
    if is_primary_server_healthy():
        print("Primary server is healthy.")
    else:
        print("Primary server is down. Initiating failover...")
        perform_failover()
```

This script checks the health of the primary server and, if it's not responsive, triggers a failover to the backup server. In practice, you would integrate this with your monitoring and automation tools and configure the failover mechanism based on your specific setup. Please note that this is a simple example, and a complete disaster recovery solution involves more advanced configurations and considerations, such as data synchronization, failback procedures, and detailed monitoring.