***7145- UNITED INSTITUTE OF TECHNOLOGY***

***DISASTER RECOERY WITH IBM CLOUD VIRTUAL SERVER***

***Team Members***

***1****.Arthi S - 714521104302*

***2****.Babeetha shalini S - 714521104008*

***3.****Bharath Kumar S - 714521104010*

***4****.Dhina Bharani L - 714521104303*

***5.****Sairam V - 714521104043*

***6****.Thalir P - 714521104052*

***Mentor***

***Nishanthi M -*** *nishanthi@uit.ac.in*

**INTRODUCTION:**

Disaster recovery with IBM Cloud Virtual Server is a critical aspect of safeguarding your business operations in the face of unforeseen disruptions. IBM Cloud offers a robust platform for hosting your virtual servers, and an effective disaster recovery plan is essential to ensure the resilience and availability of your applications and data. In this discussion, we will explore the key components and strategies for implementing disaster recovery using IBM Cloud Virtual Server, helping you minimize downtime, protect your data, and maintain business continuity in the event of a disaster.

**1. \*Data Encryption\*:**

Ensure that your data is encrypted both at rest and in transit. IBM Cloud offers encryption services and tools to help safeguard your data.

**2. \*Data Retention and Versioning\*:**

Define policies for data retention and versioning to manage how long backups are kept and how many versions are retained.

**3. \*Compliance and Regulations\*:**

Understand the regulatory requirements that may apply to your data and disaster recovery processes, especially if you're in a highly regulated industry.

**4. \*Third-Party Solutions\*:**

Evaluate third-party disaster recovery solutions that are compatible with IBM Cloud services. Some third-party tools offer advanced features for failover, replication, and recovery.

**5. \*Cost Management\*:**

Manage the costs associated with disaster recovery. IBM Cloud provides cost management tools and features to help control expenses.

**6. \*Network Considerations\*:**

Ensure that your network configurations, such as VPNs and connectivity options, are part of your disaster recovery plan.

**7. \*Documentation and Runbooks\*:**

Create detailed runbooks that guide your IT and operations teams through the steps of disaster recovery. Keep these documents up-to-date.

**8. \*Service-Level Agreements (SLAs)\*:**

Understand the SLAs provided by IBM Cloud for your virtual services. Ensure they align with your recovery time and recovery point objectives.

**9. \*Hybrid and Multi-Cloud Solutions\*:**

Consider hybrid and multi-cloud approaches to further enhance resilience. You can replicate data and services to multiple cloud providers or on-premises locations for added redundancy.

**10. \*Employee Training\*:**

Train your personnel on disaster recovery procedures. Make sure your team is familiar with the tools and processes necessary for recovery.

**11. \*Regular Audits\*:**

Periodically audit your disaster recovery plan to ensure it remains effective and compliant with changing business needs.

**12. \*Managed Services\*:**

Consider using managed disaster recovery services offered by IBM or third-party providers. These services can offload the operational burden of maintaining a disaster recovery solution.

**13. \*Scalability\*:**

Ensure that your disaster recovery solution can scale as your business grows. IBM Cloud allows you to adjust your resources as needed.

**\*Python Code:\***

python

import time

# Simulate primary and backup servers

primary\_server = {

"name": "Primary Server",

"healthy": True

}

backup\_server = {

"name": "Backup Server",

"healthy": True

}

# Function to initiate failover

def initiate\_failover():

primary\_server["healthy"] = False

print("Initiating failover to the backup server...")

time.sleep(2)

print("Failover complete.")

primary\_server["healthy"] = True

# Main function to monitor and trigger failover

def main():

while True:

if not primary\_server["healthy"]:

initiate\_failover()

# Simulate a health check every 5 seconds

time.sleep(5)

if \_\_name\_\_ == "\_\_main\_\_":

main()

\***Expected Output:\***

This Python script simulates a scenario where the primary server becomes unhealthy, triggering a failover to the backup server. Here's what you might expect in the output:

**Initiating failover to the backup server...**

**Failover complete.**

**CONCLUSION:**

Again, this is a very simplified example for demonstration purposes. In a real-world situation, a disaster recovery system would be much more complex and involve cloud services, extensive configurations, and the consideration of various possible failures.

It's important to consult with IT professionals and potentially work with experts to design and implement a robust disaster recovery plan specific to your environment.