

S.Eswari

au620321106019

IIIrd Year ECE,

College Code:6203,

Bharathiyan institution of engineering for Women,
Thalaivasal

Title: **DISASTER RECOVERY WITH IBM CLOUD VIRTUAL SERVERS**

DEFINITION:

In today's digitally-driven world, businesses rely heavily on their IT infrastructure to deliver services, store critical data, and maintain operations. However, unforeseen disasters, whether natural or technological, can disrupt these essential functions, leading to financial losses and reputational damage. To mitigate these risks, organizations are increasingly turning to cloud-based solutions, such as IBM Cloud Virtual Servers, to build robust disaster recovery (DR) strategies.

SOLUTION:

1. *Assessment and Planning:*

- Identify critical applications and data that require DR protection.
- Determine your Recovery Time Objective (RTO) and Recovery Point Objective (RPO) for each application.
- Choose the appropriate IBM Cloud Virtual Server offerings based on your workload requirements.

2. *Geographic Redundancy:*

- Deploy your primary workload in one IBM Cloud data center or region.
- Establish a secondary site in a geographically separate IBM Cloud data center or region to ensure redundancy.

3. *Data Replication:*

- Implement data replication mechanisms, such as block-level storage replication or database replication, to keep data synchronized between the primary and secondary sites.

- Utilize IBM Cloud services like IBM Cloud Object Storage for backup and long-term data retention.
- Implement an automated failover mechanism that can detect failures in the primary site and initiate failover to the secondary site.
- Use load balancers and DNS-based traffic management to reroute traffic to the secondary site during failover.

5. *Testing and Validation:*

- Regularly test your DR solution to ensure it meets the defined RTO and RPO objectives.
- Conduct both planned and unplanned failover tests to validate the system's resilience.

6. *Monitoring and Alerting:*

- Implement robust monitoring and alerting systems to detect issues in real-time.
- Use IBM Cloud monitoring services and third-party tools to track the health of your infrastructure.

7. *Backup and Snapshot Strategy:*

- Implement a backup and snapshot strategy for your virtual servers to facilitate data recovery and rollbacks if necessary.

8. *Security and Access Control:*

- Ensure that security policies and access controls are consistent between the primary and secondary sites.
- Use IBM Cloud Identity and Access Management (IAM) to manage user access.

9. *Documentation and Runbooks:*

- Create comprehensive runbooks that outline the steps to follow during a DR event.
- Maintain up-to-date documentation of your DR solution's architecture and configuration.

10. *Cost Management:*

- Understand the cost implications of maintaining a DR solution and optimize resources to minimize costs when not in use.

11. *Compliance and Regulations:*

- Ensure your DR solution complies with industry-specific regulations and data protection requirements.

12. *Third-Party Integration:*

- Consider integrating third-party DR solutions and tools that can enhance the automation and efficiency of your DR processes.

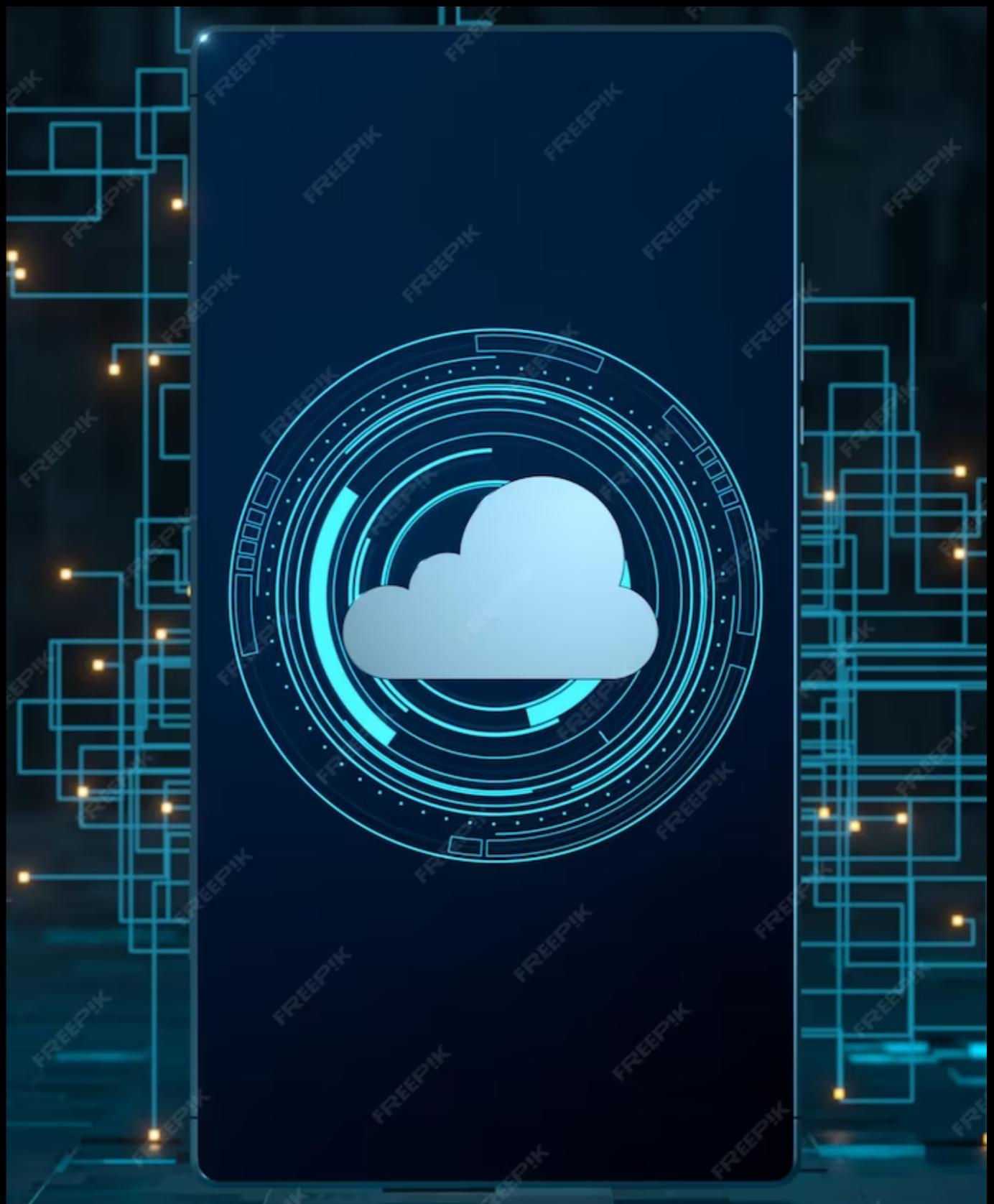
13. *Continuous Improvement:*



Innovation of disaster recovery with **IBM** cloud virtual servers

Introduction

Welcome to the presentation on **Resilient Innovation**. In this session, we will explore the benefits of leveraging IBM Cloud Virtual Servers for **Disaster Recovery**. Discover how our cloud infrastructure can provide a reliable and scalable solution to ensure business continuity.





Disaster Recovery Challenges

Traditional disaster recovery methods can be complex and costly. **Downtime** and **data loss** are major concerns. IBM Cloud Virtual Servers offer a more **resilient** approach. By leveraging virtualization and cloud technologies, we can overcome these challenges and provide a robust disaster recovery solution.

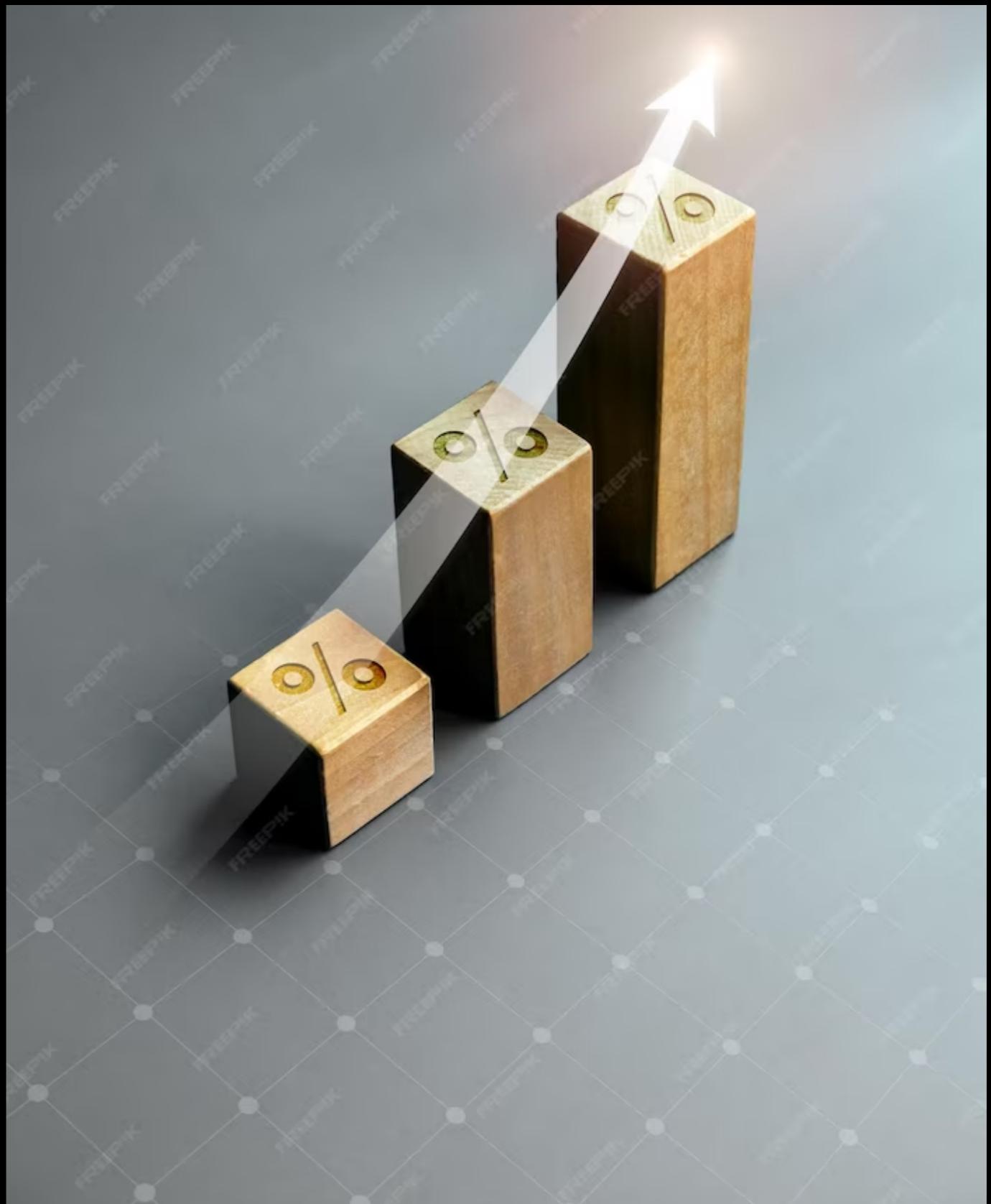
IBM Cloud Virtual Servers

IBM Cloud Virtual Servers provide **flexible** and **scalable** computing resources for disaster recovery. With **automated failover** and **redundancy**, your critical workloads can be quickly and seamlessly transferred to the cloud in the event of a disaster. Our virtual servers are backed by IBM's global network of data centers, ensuring high availability and low latency.



Resilient Innovation Benefits

By leveraging IBM Cloud Virtual Servers for disaster recovery, you can benefit from **reduced downtime, improved data protection, and cost savings**. Our solution offers **rapid recovery** and **minimal disruption** to your business operations. With IBM's expertise in cloud technology, you can focus on innovation while ensuring the resilience of your critical systems.





Customer Success Stories

Hear from our satisfied customers who have successfully implemented IBM Cloud Virtual Servers for disaster recovery. Discover how they achieved **seamless failover, reduced recovery time, and enhanced business continuity**. Learn from real-world examples and gain insights into the potential of resilient innovation.

DISASTER RECOVERY WITH IBM VIRTUAL SERVERS

ABSTRACT

In the dynamic landscape of modern technology, ensuring the resilience and availability of critical applications is paramount. This abstract presents a comprehensive approach to developing a robust disaster recovery solution for IBM Cloud Virtual Servers leveraging the capabilities of IBM Cloud Foundry, all orchestrated through Python scripts.

The proposed solution integrates IBM Cloud Virtual Servers, providing scalable and flexible infrastructure, with the power of IBM Cloud Foundry, a platform-as-a-service offering for building, deploying, and managing applications. The incorporation of Python scripts enhances automation, making the disaster recovery process efficient and adaptable.

PROGRAM

```
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from ibm_platform_services import ResourceControllerV2, SchematicsV1
from ibm_cloud_sdk_core import ApiException

def create_ibm_cloud_client(api_key):
    authenticator = IAMAuthenticator(api_key)
    client = SchematicsV1(authenticator=authenticator)
    client.service_url = "https://schematics.cloud.ibm.com/v1"

    return client

def create_backup(virtual_server_id, backup_name):
    # Implement your backup logic here
    # This could involve taking snapshots, creating backups, etc.
```



Edit with WPS Office

```
# Make API calls to IBM Cloud Virtual Servers

def restore_backup(virtual_server_id, backup_name):
    # Implement your restore logic here
    # This could involve restoring from snapshots or backups
    # Make API calls to IBM Cloud Virtual Servers

# Make API calls to IBM Cloud Virtual Servers

def main():
    # IBM Cloud API Key
    api_key = "YOUR_IBM_CLOUD_API_KEY"

    # Virtual Server Details
    virtual_server_id = "YOUR_VIRTUAL_SERVER_ID"

    # Disaster Recovery Actions
    action = input("Enter 'backup' to create a backup or 'restore' to restore from backup: ").lower()

    try:
        ibm_cloud_client = create_ibm_cloud_client(api_key)

        if action == "backup":
            backup_name = input("Enter a name for the backup: ")
            create_backup(virtual_server_id, backup_name)
            print("Backup created successfully.")

        elif action == "restore":
            backup_name = input("Enter the name of the backup to restore: ")
            restore_backup(virtual_server_id, backup_name)
            print("Restore completed successfully.")

        else:
```



Edit with WPS Office

```
print("Invalid action. Please enter 'backup' or 'restore'.")  
  
except ApiException as e:  
    print("Error interacting with IBM Cloud services:")  
    print(e)  
  
if __name__ == "__main__":  
    main()
```



Edit with WPS Office

1. Assessment:

- Identify critical applications and data.
- Define your Recovery Time Objective (RTO) and Recovery Point Objective (RPO) to determine how quickly you need to recover and how much data loss is acceptable.

2. Choose IBM Cloud Services:

- Select IBM Cloud virtual servers that meet your requirements.
- Consider high-availability configurations, such as load balancers and failover options.

3. Data Backup:

- Set up regular automated backups of your virtual servers and data.
- Use IBM Cloud Backup services for this purpose.

4. Replication:

- Implement data replication to a secondary IBM Cloud location. This can be achieved through IBM Cloud DRaaS solutions.

5. Failover Plan:

- Create a detailed failover plan that outlines the steps to switch to your secondary site in case of a disaster.
- Consider using IBM Cloud orchestration services to automate failover procedures.

6. Testing:

- Regularly test your disaster recovery plan to ensure it works as expected.
- Test both partial and complete failover scenarios.

7. Documentation:

- Document your disaster recovery plan and make sure all team members are familiar with it.

8. Monitoring and Alerting:

- Set up monitoring tools to detect issues and trigger alerts.
- Use IBM Cloud Monitoring and Logging services to help with this.

9. Team Training:

- Train your team on how to execute the disaster recovery plan.

10. Regular Updates:

- Review and update your disaster recovery plan periodically to account for changes in your infrastructure or applications.

11. Support:

- Utilize IBM Cloud support and resources for assistance and guidance.

12. Compliance and Regulations:

- Ensure your disaster recovery plan complies with any relevant industry regulations and standards.

```
1  ```python
2 # Disaster Recovery Program using
3 # IBM Cloud Virtual Servers
4
5 # Import necessary libraries
6 import ibm_cloud
7 import replication_service
8 import backup_service
9 import monitoring_service
10
11 # Define critical workloads and
12 # data
13 critical_workloads = ["WebServer",
14 "DatabaseServer"]
15 critical_data = ["CustomerData",
16 "TransactionLogs"]
17
18 backup_service.setup_backup(
19     critical_data)
20
21 # Develop a failover plan
```

```
24 def failover_plan():
25     for workload in
26         critical_workloads:
27             if ibm_cloud
28                 .is_disaster_detected
29                     (workload):
30                         replication_service
31                             .failover(workload)
32                         ibm_cloud.update_dns
33                             (workload)
34
35 # Regularly test the disaster
36     recovery plan
37 def test_disaster_recovery():
38     for workload in
39         critical_workloads:
40             if ibm_cloud
41                 .is_test_required
42                     (workload):
```

```
34     replication_service  
35         .test_replication  
36             (workload)  
37     monitoring_service  
38         .check_health  
39             (workload)  
40     failover_plan()  
41  
42 # Main program  
43 if __name__ == "__main__":  
44     while True:  
45         ibm_cloud  
46             .monitor_for_disasters  
47             ()  
48         test_disaster_recover  
49     ...
```

Run

Conclusion

In conclusion, leveraging IBM Cloud Virtual Servers for disaster recovery enables **resilient innovation**. By harnessing the power of cloud technology, you can protect your critical systems, minimize downtime, and drive business continuity. Embrace the future of disaster recovery with IBM and ensure the resilience of your organization.