ARUNAI ENGINEERING COLLEGE

CODE:5104

DEPARTMENT OF INFORMATION TECHNOLOGY

Disaster Recovery With IBM Cloud Virtual Server

Presented by: D.Kiruthiga
Guided by: K.Hari Priya

Agenda

- Introduction
- Data replication in disaster recovery
- Source code
- On-premises
- Conclusion

Introduction

Disaster recovery with cloud virtual servers is an approach that leverages cloud computing resources to ensure business continuity and data resilience in the event of a disaster or unexpected downtime. By using cloud virtual servers, organizations can replicate and store critical data and applications in remote cloud environments, allowing for rapid recovery and minimal data loss.

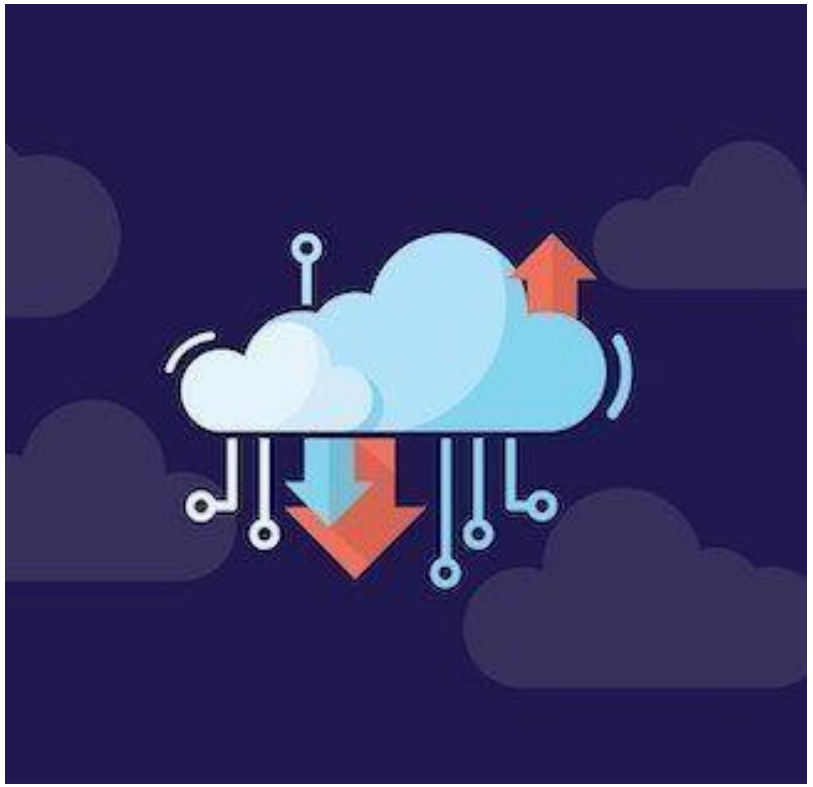
Data replication in disaster recovery

It is a critical component of ensuring data availability and business continuity in the event of a disaster. Data replication involves creating and maintaining copies of data at one or more locations to ensure data redundancy and accessibility. Below, I'll outline a basic process to implement data replication in your disaster recovery strategy:

➢ Assessment and Planning

Before you begin implementing data replication, you need to assess your organization's needs and create a comprehensive disaster recovery plan. This plan should include:

- Identification of critical data and applications.
- Recovery time objectives and recovery point objectives for each system.
- Budget allocation for DR implementation.
- Selection of a remote site for data replication.



➢ Choose Replication Methods

Select the appropriate replication methods based on your needs:

Synchronous Replication: Data is mirrored in real-time to a secondary site. It ensures zero data loss but may impact performance due to the latency introduced by synchronous communication.

Asynchronous Replication: Data is copied to a secondary site with a time delay, which minimizes performance impact but may result in some data loss in the event of a disaster.

Near-Synchronous Replication: A compromise between synchronous and asynchronous replication, where data is replicated at short, defined intervals.

- **Select Storage Technology**

  Choose the appropriate storage technology for replication. This could be a combination of:

  - Network-attached storage (NAS)
  - Storage Area Network (SAN)
  - Cloud-based storage solutions.

- **Implement Data Replication**

  Set up data replication tools and software that match your chosen replication method and storage technology. This could include database replication software, file-level replication tools, or cloud services.

- **Test and Validate**

  Regularly test the data replication process to ensure it works as expected. Simulate disaster scenarios and validate that data can be recovered from the secondary location.

- **Automation**

  Automate the data replication process as much as possible. This includes automating failover procedures and regularly updating configurations.

- **Security and Compliance:**

  Ensure that data replication complies with your organization's security policies and regulatory.

- **Data Encryption and Security**

Ensure that data is encrypted both during transmission and at rest in the secondary location. Security measures should be in place to protect against unauthorized access.

- **Training and Awareness**

Ensure that your IT and operations teams are trained on disaster recovery procedures, including the use of data replication tools.

- **Review and Improve:**

Periodically review your disaster recovery plan and make improvements as needed. Technology and business needs evolve, so your plan should adapt accordingly.

- **Incident Response Plan**

Develop an incident response plan to address any issues that may arise during data replication or failover.

Source code

```
import shutil
import os
import time
def replicate_data(source_dir, destination_dir):
    try:
        # Check if the source directory exists
        if not os.path.exists(source_dir):
            print("Source directory does not exist.")
            return
        # Create the destination directory if it doesn't exist
        if not os.path.exists(destination_dir):
            os.makedirs(destination_dir)
        # List all files in the source directory
        files = os.listdir(source_dir)
        for file in files:
            source_file =  os.path.join(source_dir, file)
            destination_file= os.path.join(destination_dir, file)
            # Copy the file from the source to the destination
            shutil.copy2(source_file, destination_file)
```

```
        print(f"Copied {source_file} to {destination_file}")
        print("Data replication completed.")
    except Exception as e:
        print(f"An error occurred: {str(e)}")
if __name__ == "__main__":
    source_directory= "/path/to/source_data"
    destination_directory = "/path/to/secondary_location"
    while True:
        replicate_data(source_directory, destination_directory)
        time.sleep(3600)  # Replicate data every hour (adjust as needed)
```

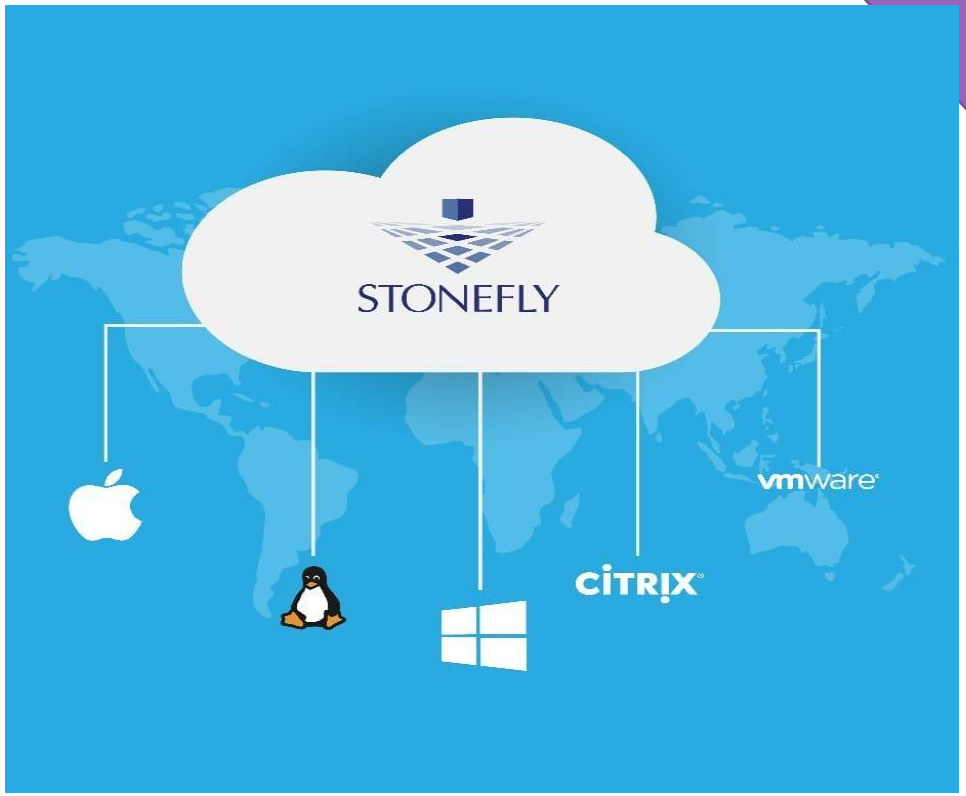
Output:

```
Copied /path/to/source_data/file1.txt to /path/to/secondary_location/file1.txt
Copied /path/to/source_data/file2.txt to /path/to/secondary_location/file2.txt
Copied /path/to/source_data/file3.txt to /path/to/secondary_location/file3.txt
Data replication completed.
Copied /path/to/source_data/file1.txt to /path/to/secondary_location/file1.txt
Copied /path/to/source_data/file2.txt to /path/to/secondary_location/file2.txt
Copied /path/to/source_data/file3.txt to /path/to/secondary_location/file3.txt
Data replication completed.
... (Repeats every hour)
```

On-premises

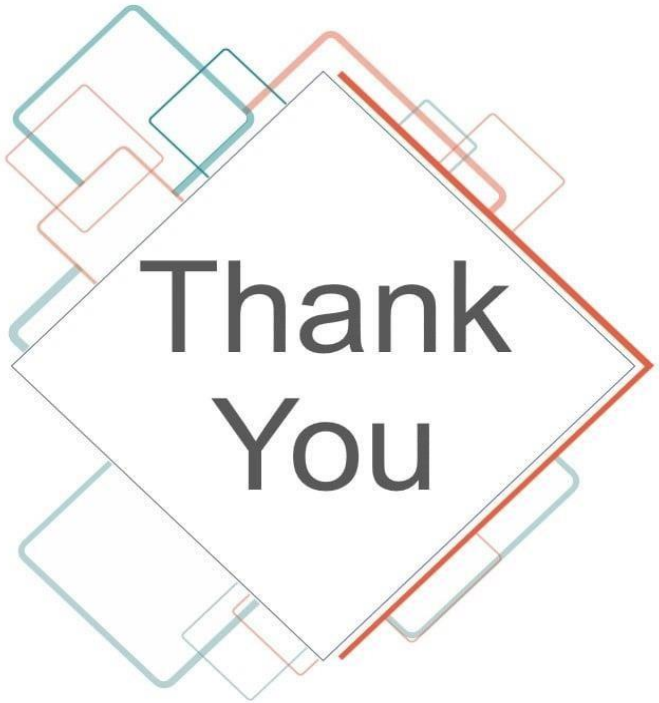
In some cases, keeping certain backup or disaster recovery processes on-premises can help you retrieve data and recover IT services rapidly. Retaining some sensitive data on premises might also seem appealing if you need to comply with strict data privacy or data sovereignty regulations.

For disaster recovery, a plan that relies wholly on an on-premises environment would be challenging. If a natural disaster or power outage strikes, your entire data center—with both primary and secondary systems—would be affected. That's why most disaster recovery strategies employ a secondary site that is some distance away from the primary data center. You might locate that other site across town, across the country or across the globe depending on how you decide to balance factors such as performance, regulatory compliance and physical accessibility to the secondary site.



Conclusion

In conclusion, disaster recovery through data replication is a critical aspect of modern business continuity and risk management. Whether implemented on-premises or in the cloud, data replication plays a pivotal role in safeguarding an organization's critical data and applications.


Thank
You