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.

```python
# Disaster Recovery Program using IBM Cloud Virtual Servers

# Import necessary libraries
import ibm_cloud
import replication_service
import backup_service
import monitoring_service

# Define critical workloads and data
critical_workloads = ["WebServer", "DatabaseServer"]
critical_data = ["CustomerData", "TransactionLogs"]

backup_service.setup_backup(critical_data)

# Develop a failover plan
```

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

```
                replication_service
                    .test_replication
                    (workload)
                monitoring_service
                    .check_health
                    (workload)
                failover_plan()

# Main program
if __name__ == "__main__":
    while True:
        ibm_cloud
            .monitor_for_disasters
            ()
        test_disaster_recove
```

Run