**Phase 5: Deployment**

**Objective**

To deploy the DR solution into a real-world cloud environment (IBM Cloud or AWS) ensuring all components are operational, automated, and monitored.

**Deployment Steps**

**1. Environment Preparation**

* Finalize VPC, IAM Roles, EC2, RDS, S3 buckets, Route 53 (or IBM equivalent).
* Ensure infrastructure code is ready (cloud\_dr\_resources.yaml).

**2. Deploy Infrastructure (CloudFormation or IBM Terraform)**

aws cloudformation deploy \

--template-file cloud\_dr\_resources.yaml \

--stack-name dr-solution \

--capabilities CAPABILITY\_IAM

**3. Automate Backups**

* Upload backup\_to\_s3.sh to an EC2 instance.
* Schedule with cron:

crontab -e

# Daily at 2 AM

0 2 \* \* \* /home/ec2-user/backup\_to\_s3.sh >> /var/log/backup.log 2>&1

**4. Lambda Deployment**

* Deploy lambda\_start\_ec2.py.
* Set trigger: CloudWatch alarm for instance failure.

**5. Health Checks and Routing**

* Configure Route 53 failover using create\_route53\_health\_check.sh.

**6. Monitoring and Alerts**

* Set up CloudWatch (or IBM Cloud Monitoring).
* Enable notifications via SNS or email.

**Post-Deployment Validation**

* Test service accessibility (Web app, DB).
* Manually fail components to verify auto-recovery.
* Check log files and monitoring dashboards.
* Simulate a DR event to verify system-wide recovery.

**Artifacts to Include in Repository**

deployment/

├── deployment\_steps.md

├── validation\_checklist.md

├── cloudformation\_logs.txt

└── final\_report.md

**Suggested Final Repo Structure**

cloud-dr-solution-enterprises/

├── backup\_to\_s3.sh

├── lambda\_start\_ec2.py

├── create\_route53\_health\_check.sh

├── cloud\_dr\_resources.yaml

├── testing/

│ └── (test plans, reports)

├── deployment/

│ └── (deployment steps, validations)

├── README.md

└── LICENSE