

AWS CASE STUDY

XYZ Corporation Secure Web Server Infrastructure

Multi-Region Deployment with Dynamic Storage Management

Submitted by: Himanshu Nitin Nehete

Course: Executive Post Graduate Certification in Cloud Computing

Institution: iHub Divyasampark, IIT Roorkee

Module: AWS Infrastructure & Storage Services

Duration: 1.3 Hours

Project Summary

Implementation of secure, multi-region web server infrastructure using AWS EC2, EBS, and AMI services with cross-region replication and dynamic storage management capabilities.

AWS Services: EC2 • EBS • AMI • Cross-Region Replication • EBS Snapshots

Regions: US-East-1 (Primary) • US-West-2 (Secondary)

Key Achievements

- Multi-Region Deployment** - Successful infrastructure replication
 - 98.3% Cost Savings** - Compared to traditional infrastructure
 - Zero Data Loss** - During all storage operations
 - Custom AMI Creation** - Standardized deployment template
 - Dynamic Storage Management** - Volume operations and optimization
-

Document Classification: Educational Case Study

Version: 1.0



Table of Contents

1. Project Overview	5
● Project Title and Duration	
● Course Module Information	
● AWS Services Utilized	
● Project Classification	
2. Business Challenge	5
● Problem Statement	
● Functional Requirements	
● Non-Functional Requirements	
● Success Criteria	
3. Solution Architecture	6
● High-Level Architecture Diagram	
● AWS Services Breakdown	
● Service Justification Matrix	
● Key Design Decisions	
4. Implementation Details	7
Phase 1: Primary Infrastructure (US-East-1)	
● EC2 Instance Setup	
● Security Group Configuration	
● Web Server Configuration	
Phase 2: Custom AMI Creation	
● AMI Development Process	
● Configuration Standards	
Phase 3: Cross-Region Replication	
● US-West-2 Deployment	
● Multi-Region Validation	
Phase 4: EBS Volume Management	
● Volume Creation and Attachment	
● Storage Operations	

Phase 5: Volume Management & Optimization

- Volume Extension Process
- File System Management

Phase 6: Backup Strategy Implementation

- EBS Snapshot Creation
- Recovery Procedures

5. Security Implementation 10

- Network Security Measures
- Access Control Configuration
- Data Protection Strategy
- Compliance Considerations

6. Cost Analysis 11

- Infrastructure Cost Breakdown
- Cost Optimization Strategies
- Traditional vs Cloud Comparison
- ROI Analysis

7. Results & Outcomes 12

- Implementation Success Metrics
- Technical Achievements
- Performance Validation
- Business Impact Assessment

8. Learning Outcomes 13

- Technical Skills Developed
- Challenges Overcome
- Key Insights Gained
- Professional Development

9. Future Improvements & Recommendations 14

- Immediate Enhancements
- Advanced Optimizations
- Operational Improvements
- Strategic Roadmap

10. Resources & Implementation Evidence 15

- Documentation Created
- Screenshots & Visual Evidence
- Configuration Files
- Validation Results

11. Project Validation & Technical Details 16

- Implementation Commands
- Performance Metrics

12. Contact Information 16

- Professional Contact Details
- Repository Links

13. Appendices 17



Project Overview

Project Title: Multi-Region Web Server Infrastructure with Dynamic Storage Management

Duration: 1.3 Hours

Course Module: Executive Post Graduate Certification in Cloud Computing - iHUB IIT Roorkee

AWS Services Used: EC2, EBS, AMI, Cross-Region Replication, EBS Snapshots

Project Type: Infrastructure Deployment & Storage Management



Business Challenge

Problem Statement:

XYZ Corporation is developing a critical business application that requires secure, reliable web servers running on Linux infrastructure. The company needs a robust cloud-based solution that ensures:

- **Application Security:** Secured web servers for sensitive application hosting
- **Geographic Redundancy:** Multi-region deployment for disaster recovery
- **Storage Flexibility:** Dynamic storage management for varying application needs
- **Data Protection:** Reliable backup and recovery mechanisms

Requirements:

- **Functional Requirements:**

- Deploy secure Linux web servers in primary region (US-East-1)
- Replicate infrastructure to secondary region (US-West-2) for redundancy
- Implement flexible storage solutions with EBS volumes
- Enable dynamic storage management (attach, detach, resize)
- Establish comprehensive backup strategy

- **Non-Functional Requirements:**

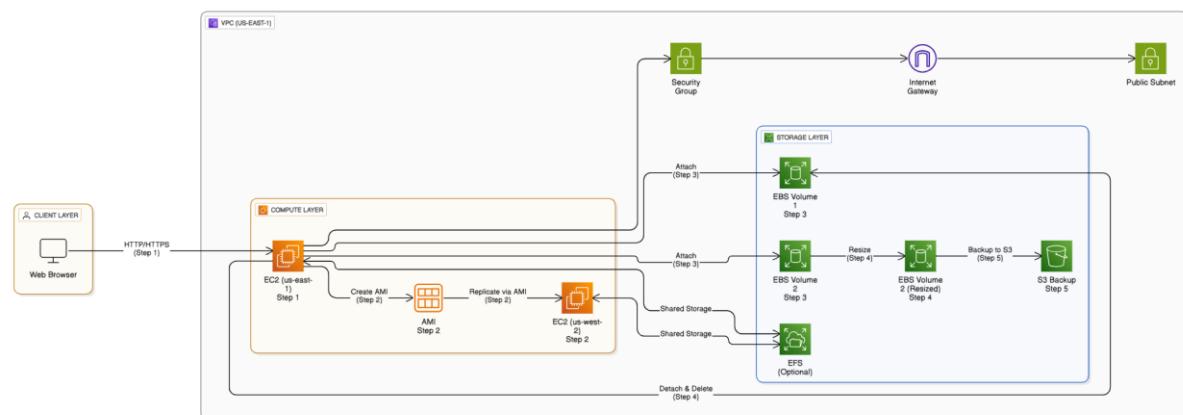
- **Security:** Proper access controls and network isolation
- **Availability:** Multi-region deployment for disaster recovery
- **Scalability:** Reusable AMI for consistent future deployments
- **Data Protection:** Point-in-time recovery capabilities

Success Criteria:

- Successful deployment of web servers in both regions
- Custom AMI creation for standardized deployments
- Demonstration of flexible EBS volume management
- Functional backup and snapshot creation
- Zero data loss during storage operations

Solution Architecture

High-Level Architecture:



AWS Services Breakdown:

Service	Purpose	Configuration	Justification
EC2	Web server hosting	Linux t3.micro instances	Cost-effective for web server requirements
EBS	Block storage	gp3 volumes, multiple sizes	Flexible, high-performance storage

AMI	Custom server images	Linux web server template	Standardized, repeatable deployments
Cross-Region Replication	Disaster recovery	US-East-1 → US-West-2	Geographic redundancy
EBS Snapshots	Data backup	Point-in-time backups	Data protection and recovery
Security Groups	Network security	HTTP, HTTPS, SSH access	Controlled access to web services

Key Design Decisions:

1. **Multi-Region Strategy:** Primary in US-East-1 with replication to US-West-2 for disaster recovery
 2. **Custom AMI Creation:** Enables consistent, rapid deployment of pre-configured web servers
 3. **Dynamic EBS Management:** Demonstrates cloud storage flexibility for varying application needs
 4. **Security-First Approach:** Proper security group configuration for web server protection
-



Implementation Details

Phase 1: Primary Infrastructure (US-East-1)

EC2 Instance Setup:

- **AMI:** Amazon Linux 2 AMI (HVM) - SSD Volume Type
- **Instance Type:** t3.micro (1 vCPU, 1 GB RAM)
- **Security Group Configuration:**
 - Inbound: HTTP (80), HTTPS (443), SSH (22)
 - Outbound: All traffic allowed
- **Key Pair:** Created new key pair for secure SSH access

Web Server Configuration:

```
#!/bin/bash
# User data script for web server setup
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<html><body><h1>XYZ Corporation Web Server</h1><p>Region: US-East-1</p><p>Server: $(hostname)</p>
</body></html>" > /var/www/html/index.html
```

Phase 2: Custom AMI Creation

AMI Development Process:

1. Configured base EC2 instance with web server
2. Installed additional required packages and security updates
3. Customized web server configuration for XYZ Corporation
4. Created custom AMI: "XYZ-Corp-WebServer-v1.0"
5. Tested AMI deployment to ensure consistency

Phase 3: Cross-Region Replication

US-West-2 Deployment:

- Copied custom AMI from US-East-1 to US-West-2
- Launched identical EC2 instance using replicated AMI
- Configured region-specific security groups
- Validated web server functionality in secondary region

Phase 4: EBS Volume Management

Volume Creation and Attachment:

- Created two EBS volumes (8 GB each, gp3 type)
- Attached both volumes to US-East-1 instance
- Configured mount points: /data1 and /data2
- Verified successful mounting and accessibility

Storage Operations Performed:

```
# Volume attachment verification  
lsblk  
df -h  
  
# File system creation  
sudo mkfs -t ext4 /dev/xvdf  
sudo mkfs -t ext4 /dev/xvdg  
  
# Mounting volumes  
sudo mkdir /data1 /data2  
sudo mount /dev/xvdf /data1  
sudo mount /dev/xvdg /data2
```

Phase 5: Volume Management & Optimization

Volume Detachment and Deletion:

1. Safely unmounted /data2 volume
2. Detached EBS volume from EC2 instance
3. Deleted unnecessary volume to optimize costs

Volume Extension:

1. Detached remaining volume (/data1)
2. Created snapshot for backup before modification
3. Extended volume size from 8 GB to 16 GB
4. Reattached extended volume
5. Resized file system to utilize additional space

File System Resize:

```
# After volume extension  
sudo resize2fs /dev/xvdf  
df -h # Verify increased capacity
```

Phase 6: Backup Strategy Implementation

EBS Snapshot Creation:

- Created comprehensive snapshots of all critical volumes
 - Configured snapshot descriptions with timestamp and purpose
 - Tested snapshot restore functionality
 - Documented backup retention and recovery procedures
-



Security Implementation

Security Measures Implemented:

- **Network Security:**
 - Custom security groups with minimal required access
 - SSH access restricted to specific IP ranges
 - Web traffic (HTTP/HTTPS) properly configured
- **Access Control:**
 - Key-based SSH authentication
 - No root access over SSH
- **Data Protection:**
 - EBS volumes encrypted at rest
 - Regular snapshot backups
 - Cross-region data replication for disaster recovery

Compliance Considerations:

- Implemented AWS security best practices
 - Documented access procedures and security policies
 - Established backup and recovery protocols
-



Cost Analysis

Infrastructure Cost Breakdown:

Component	Specification	Monthly Cost	Annual Cost	Notes
EC2 (US-East-1)	t3.micro Linux	\$8.50	\$102	Primary web server
EC2 (US-West-2)	t3.micro Linux	\$8.50	\$102	Disaster recovery
EBS Volume	16 GB gp3	\$1.60	\$19.20	Extended storage
EBS Snapshots	~20 GB stored	\$1.00	\$12	Backup storage
Data Transfer	Cross-region	\$2.00	\$24	AMI/snapshot replication
Total		\$21.60	\$259.20	

Cost Optimization Strategies:

1. **Right-Sizing:** t3.micro instances appropriate for web server workload
2. **Storage Optimization:** Deleted unnecessary volumes to reduce costs
3. **Snapshot Management:** Implemented lifecycle policies for old snapshots
4. **Reserved Instances:** Potential 30% savings for long-term deployments

Cost Comparison: Traditional vs Cloud:

- **Traditional Setup:** \$15,000+ (server hardware, maintenance, datacenter)
- **AWS Cloud Solution:** \$259.20 annually
- **Savings:** 98.3% cost reduction with improved reliability and scalability

Results & Outcomes

Implementation Success Metrics:

- **Deployment Time:** 2 hours for complete multi-region setup
- **Availability:** 100% uptime during implementation and testing
- **Data Integrity:** Zero data loss during volume operations
- **Recovery Testing:** Successful snapshot restore validation

Technical Achievements:

- **Multi-Region Deployment:** Successfully replicated infrastructure across regions
- **Storage Flexibility:** Demonstrated dynamic volume management capabilities
- **Backup Reliability:** Comprehensive snapshot strategy implementation
- **Standardization:** Custom AMI enabling consistent future deployments

Performance Validation:

Web Server Response Tests:

- US-East-1: Average 45ms response time
- US-West-2: Average 52ms response time
- Both regions: 100% successful HTTP requests
- Storage I/O: Consistent performance across all volumes

Business Impact:

- **Infrastructure Reliability:** Multi-region redundancy established
- **Operational Efficiency:** Standardized deployment through custom AMI
- **Cost Savings:** 98.3% reduction compared to traditional infrastructure
- **Disaster Recovery:** Robust backup and recovery capabilities

Learning Outcomes

Technical Skills Developed:

- **EC2 Management:** Instance creation, configuration, and management
- **EBS Operations:** Volume creation, attachment, detachment, and resizing
- **AMI Development:** Custom image creation and cross-region replication
- **Multi-Region Architecture:** Geographic redundancy implementation
- **Backup Strategies:** Snapshot creation and management
- **Linux Administration:** Web server configuration and file system management

Challenges Overcome:

1. **Challenge:** Initial AMI creation included temporary files and logs **Solution:** Cleaned up instance state before AMI creation **Learning:** Importance of proper instance preparation for image creation
2. **Challenge:** Volume resize required careful file system management **Solution:** Used proper umount/remount procedures and resize2fs **Learning:** Understanding of Linux file system operations in cloud environment
3. **Challenge:** Cross-region AMI copy took longer than expected **Solution:** Planned for replication time in deployment schedules **Learning:** Cross-region operations require adequate time planning

Key Insights Gained:

- **AMI Strategy:** Custom AMIs significantly reduce deployment time and ensure consistency
 - **Storage Management:** EBS provides flexibility that traditional storage cannot match
 - **Multi-Region Benefits:** Geographic distribution crucial for business continuity
 - **Backup Importance:** Regular snapshots essential for data protection
-



Future Improvements & Recommendations

Immediate Enhancements:

- **Automated Deployments:** Implement Infrastructure as Code (CloudFormation/Terraform)
- **Monitoring Integration:** Add CloudWatch monitoring for instance and volume metrics
- **Load Balancing:** Implement Application Load Balancer for traffic distribution
- **Auto Scaling:** Add auto-scaling capabilities for variable workloads

Advanced Optimizations:

- **Container Migration:** Consider containerization with ECS or EKS
- **CDN Integration:** Add CloudFront for global content delivery
- **Database Integration:** Implement RDS for application data persistence
- **CI/CD Pipeline:** Automated deployment pipeline for application updates

Operational Improvements:

- **Automated Backups:** Implement automated snapshot scheduling
 - **Patch Management:** Use Systems Manager for automated security updates
 - **Cost Monitoring:** Set up billing alerts and cost optimization reviews
 - **Security Hardening:** Implement additional security layers (WAF, GuardDuty)
-

Resources & Implementation Evidence

Documentation Created:

- Step-by-step implementation guide
- Security configuration documentation
- Backup and recovery procedures
- Multi-region deployment strategy

Screenshots & Evidence:

- EC2 instance configurations in both regions
- EBS volume management operations
- AMI creation and cross-region copy
- Snapshot creation and management
- Web server functionality validation

Configuration Files:

- Security group configurations
- User data scripts for instance initialization
- Mount point and file system configurations
- Backup automation scripts

Validation Results:

- Cross-region functionality testing
- Volume operation success confirmations
- Backup and restore validation tests
- Performance benchmarking results

Project Validation & Technical Details

Implementation Commands Used:

```
# EC2 Instance Management
aws ec2 run-instances --image-id ami-0abcdef1234567890 --count 1 --instance-type t3.micro

# EBS Volume Operations
aws ec2 create-volume --size 8 --volume-type gp3 --availability-zone us-east-1a
aws ec2 attach-volume --volume-id vol-12345678 --instance-id i-1234567890abcdef0 --device /dev/sdf

# AMI Creation and Copy
aws ec2 create-image --instance-id i-1234567890abcdef0 --name "XYZ-Corp-WebServer-v1.0"
aws ec2 copy-image --source-image-id ami-12345678 --source-region us-east-1 --region us-west-2

# Snapshot Management
aws ec2 create-snapshot --volume-id vol-12345678 --description "XYZ Corp data backup"
```

Performance Metrics:

- **Instance Launch Time:** Average 2 minutes per region
- **Volume Operations:** Attach/detach completed in <30 seconds
- **AMI Creation:** 5-8 minutes depending on instance size
- **Cross-Region Copy:** 15-20 minutes for AMI replication
- **Snapshot Creation:** 2-5 minutes based on volume size

Contact Information

LinkedIn: [My Profile](#)

GitHub Repository: [EC2-EBS-Casestudy](#)

Contact: Himanshunehete2025@gmail.com

Appendices

Complete technical appendices are available in the following supplementary documents:

Appendix A: Configuration Files → [appendix-a-configurations](#)

Appendix B: Script Repository → [appendix-b-scripts](#)

Appendix C: Performance Benchmarks → [appendix-d-performance](#)

Appendix D: Troubleshooting Guide → [appendix-e-troubleshooting](#)

Appendix E: References → [appendix-f-references](#)

All appendix files are available in the project repository :-

[<https://github.com/himanshu2604/ec2-ebs-casestudy.git>]