IT Management & Audits

Practical Lab Manual

Lab Setup & IT Infrastructure Walkthrough

Practical P01

Learning Domain

Cloud Computing & Infrastructure as a Service (IaaS)

Course Learning Outcomes

CLO01: Understand FinTech IT infrastructure fundamentals

Unit

Unit I: FinTech IT Infrastructure Fundamentals

Time Allocation: 3 hours

Learning Mode: Hands-on (80%) + Theory (20%)

Difficulty Level: Intermediate

Lab Setup & IT Infrastructure

Practical P01

Quick Reference

Practical Code	P01
Practical Name	Lab Setup & IT Infrastructure Walkthrough
Slot	T/P-1
Duration	3 hours
CLO Mapping	CLO01
Unit	Unit I: FinTech IT Infrastructure Fundamentals
Delivery Mode	Hands-on Lab
Target Audience	Intermediate Level
India Integration	MEDIUM
Screenshot Count	5 Required

Prerequisites

- Basic understanding of cloud computing concepts
- Familiarity with command-line interface (CLI)
- Internet connection and valid email for account creation
- Computer with browser support (Chrome, Firefox, Safari, Edge)
- Basic networking knowledge (IP addresses, ports, domains)

Tools Required

Tool	Version	Free Tier	Notes
AWS Free Tier Account	-	✓	Check current availability
Azure Free Account	-	✓	Alternative option
Web Browser	Latest	✓	Chrome/Firefox recommended
Terminal/CLI	-	✓	bash (Linux/Mac) or PowerShell (Windows)
Text Editor	-	✓	VS Code recommended

Learning Objectives

- ✓ Create and configure a cloud platform account (AWS Free Tier)
- ✓ Deploy a virtual machine (EC2 instance) on cloud infrastructure
- ✓ Configure network security settings and security groups
- ✓ Install and configure a web server (Apache or Nginx)
- ✓ Enable remote access and verify connectivity
- ✓ Monitor cloud resources and understand billing implications
- ✓ Apply cloud security best practices for India-specific compliance (RBI guidelines)

What You Will Learn

By the end of this practical, you will:

1. Understand Infrastructure as a Service (IaaS) concepts and deployment models
2. Navigate AWS/Azure cloud console and understand available services
3. Deploy compute resources (virtual machines) on public cloud
4. Configure and manage network security (Security Groups, Firewalls)
5. Install and configure a web application server
6. Monitor cloud resources and understand cost optimization
7. Apply industry best practices for cloud infrastructure in India

Real-World Application

Cloud infrastructure is the backbone of modern FinTech applications in India. Companies like **Razorpay**, **PhonePe**, and **ICICI Bank's digital services** rely on AWS or Azure for hosting payment processing systems, APIs, and web applications. By completing this practical, you'll understand the fundamental concepts that power India's FinTech ecosystem.

Hands-On Procedure

Part A: Account Setup

Step 1: Create AWS Free Tier Account

Objective: Set up a free AWS account with necessary permissions and security settings.
Instructions:

1. Navigate to `https://aws.amazon.com/free`
2. Click **Create a Free Account**
3. Enter your email address and set a password
4. Verify your email address by clicking the link in the confirmation email
5. Complete account information (Name, Address, Phone)
6. Add a payment method (required for identity verification, no charges for free tier)
7. Verify phone number via SMS or automated call
8. Choose **Basic Support** plan (free)
9. Complete account creation

Code/Command: None for this step (UI-based)

Expected Output

AWS Console Dashboard displays: EC2, RDS, Lambda, S3, and other services available in free tier.

The AWS Free Tier includes 750 hours of EC2 (t2.micro instance) per month for 12 months. This is sufficient for running small applications continuously.

Step 2: Access AWS Console and Navigate Services

Objective: Familiarize yourself with AWS console structure and locate EC2 service.
Instructions:

1. Log in to AWS Console at `https://console.aws.amazon.com`
2. Review the main dashboard showing recent services and quick links
3. In the **Find Services** search box, type `EC2`
4. Click on **Elastic Compute Cloud (EC2)** from results

5. Observe the EC2 Dashboard showing instances, volumes, security groups, and key pairs
6. Note the current region (top-right corner) - ensure it's set to your preferred region
7. For India-based applications, consider using `Asia Pacific (Mumbai) ap-south-1` region

Key Menu Items:

```
Dashboard > Instances
Dashboard > Images (AMIs)
Dashboard > Security Groups
Dashboard > Key Pairs
Network & Security > Security Groups
Instances > Instances (Launch new instances
   )
```

AWS EC2 Dashboard Navigation

Expected Output

EC2 Dashboard loads showing 0 running instances, available resources, and service status.

Select the correct region to reduce latency for Indian users. Mumbai (ap-south-1) is recommended for FinTech applications serving India.

Step 3: Create Security Group

Objective: Configure firewall rules to allow HTTP/HTTPS and SSH access.
Instructions:

1. In EC2 Dashboard, navigate to **Security Groups** (under **Network & Security**)
2. Click **Create Security Group**
3. Set name: `FinTech-WebServer-SG`
4. Set description: `Security group for FinTech lab web server`
5. Select your VPC (default VPC is fine)
6. Add inbound rules:
   - Type: SSH, Protocol: TCP, Port: 22, Source: Your IP (or 0.0.0.0/0 for testing)
   - Type: HTTP, Protocol: TCP, Port: 80, Source: 0.0.0.0/0
   - Type: HTTPS, Protocol: TCP, Port: 443, Source: 0.0.0.0/0
7. Outbound rules: Allow all (default)

8. Click **Create Security Group**

Code/Command:

```
# Create security group
aws ec2 create-security-group \
--group-name FinTech-WebServer-SG \
--description "Security␣group␣for␣FinTech␣lab␣web␣server" \
--region ap-south-1

# Add inbound rules
aws ec2 authorize-security-group-ingress \
--group-id sg-xxxxxxxx \
--region ap-south-1 \
--protocol tcp --port 22 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress \
--group-id sg-xxxxxxxx \
--region ap-south-1 \
--protocol tcp --port 80 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress \
--group-id sg-xxxxxxxx \
--region ap-south-1 \
--protocol tcp --port 443 --cidr 0.0.0.0/0
```

AWS CLI: Create Security Group

Expected Output

Security group created with ID: sg-xxxxxxxx
Inbound rules configured for SSH (port 22), HTTP (port 80), and HTTPS (port 443)

If you cannot access the instance, check that your security group allows your IP address. Use `https://www.whatismyip.com` to find your IP, then update the security group rules.

Screenshot 1

What to paste: AWS EC2 Security Group console showing created security group with inbound rules (SSH, HTTP, HTTPS).

Paste your screenshot here

Step 4: Launch EC2 Instance and Configure

Objective: Create a virtual machine instance and configure it for web server deployment.
Instructions:

1. Navigate to **Instances** in EC2 Dashboard
2. Click **Launch Instances**
3. **Step 1 - Choose AMI:** Select `Ubuntu Server 22.04 LTS (HVM)` (free tier eligible)
4. **Step 2 - Choose Instance Type:** Select `t2.micro` (free tier eligible, 1 vCPU, 1 GB RAM)
5. **Step 3 - Configure Instance:**
   - Number of instances: 1
   - Network: Default VPC
   - Auto-assign IPv4: Enable
   - Monitoring: Enable detailed CloudWatch monitoring (optional)
6. **Step 4 - Storage:** Keep default 30 GB gp2 EBS volume
7. **Step 5 - Tags:** Add tag `Name = FinTech-WebServer`
8. **Step 6 - Security Group:** Select `FinTech-WebServer-SG`
9. **Step 7 - Review:** Review settings and click **Launch**
10. Create or select existing key pair for SSH access
11. Download key pair file (.pem) and save securely
12. Click **Launch Instances**

Code/Command:

```
aws ec2 run-instances \
--image-id ami-0c55b159cbfafe1f0 \
--instance-type t2.micro \
--key-name FinTech-Lab-Key \
--security-groups FinTech-WebServer-SG \
--region ap-south-1 \
--tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=FinTech-WebServer}]'
```

AWS CLI: Launch EC2 Instance

Expected Output

Instance launched successfully with:
- Instance ID: i-xxxxxxxxxxxxxxxxx
- Public IPv4 Address: aaa.bbb.ccc.ddd
- Instance State: Running (after 1-2 minutes)
- Network: Connected to default VPC

Save the key pair file (.pem) securely. This is required for SSH access to your instance. AWS cannot recover this file.

Step 5: Connect to Instance via SSH

Objective: Establish secure remote connection to the EC2 instance.
Instructions:
On Linux/Mac:

1. Open terminal
2. Change permissions on key pair: `chmod 400 your-key.pem`
3. Connect via SSH: `ssh -i your-key.pem ubuntu@<public-ip-address>`
4. Accept the connection (type `yes` when prompted)

On Windows (PowerShell):

1. Open PowerShell
2. Connect via SSH: `ssh -i your-key.pem ubuntu@<public-ip-address>`
3. Or use PuTTY GUI tool by converting .pem to .ppk format

Code/Command:

```
# Set correct permissions on key file
chmod 400 FinTech-Lab-Key.pem

# Connect to instance (replace with your public IP)
ssh -i FinTech-Lab-Key.pem ubuntu@ec2-xxx-xxx-xxx-xxx.ap-south-1.compute.amazonaws.com

# Or using IP address directly
ssh -i FinTech-Lab-Key.pem ubuntu@52.xx.xx.xx

# Once connected, you should see:
# ubuntu@ip-xxx-xxx-xxx-xxx:~$
```

SSH Connection to EC2 Instance

Expected Output

SSH connection established. Terminal displays:
`ubuntu@ip-xxx-xxx-xxx-xxx:~$`

This confirms successful remote access to your cloud instance.

If `ssh: connect to host` times out: (1) Verify instance is in Running state, (2) Check security group allows port 22 from your IP, (3) Wait 1-2 minutes for instance to fully initialize, (4) Verify public IP address is correct.

Screenshot 2

What to paste: Terminal window showing successful SSH connection to EC2 instance (showing the `ubuntu@ip-xxx-xxx: $` prompt).

Paste your screenshot here

Step 6: Install and Configure Web Server

Objective: Deploy Nginx web server and configure it to serve a simple web page.
Instructions (via SSH):

1. Update system packages: `sudo apt-get update`
2. Install Nginx: `sudo apt-get install -y nginx`
3. Start Nginx service: `sudo systemctl start nginx`
4. Enable auto-start on reboot: `sudo systemctl enable nginx`
5. Verify Nginx is running: `sudo systemctl status nginx`
6. Create a simple HTML file: Edit `/var/www/html/index.html`
7. Add custom content showing FinTech lab identification
8. Save and verify accessibility

Code/Command:

```
# Update system
sudo apt-get update
sudo apt-get upgrade -y

# Install Nginx
sudo apt-get install -y nginx

# Start and enable service
sudo systemctl start nginx
sudo systemctl enable nginx

# Verify status
sudo systemctl status nginx

# Create backup of default page
sudo cp /var/www/html/index.html /var/www/
    html/index.html.bak

# Edit the HTML file
sudo nano /var/www/html/index.html

# Add this content:
# <html>
#   <head><title>FinTech Lab Server</title
    ></head>
#   <body>
#     <h1>FinTech Lab Infrastructure - AWS
    EC2</h1>
#     <p>This server is running on AWS Free
     Tier</p>
#     <p>Server configuration: t2.micro
    instance</p>
#     <p>Region: ap-south-1 (Mumbai)</p>
#   </body>
# </html>
```

Install Nginx and Deploy Web Server

Expected Output

Nginx installed and running:

```
• nginx.service - A high performance web server and reverse proxy
server
Loaded:  loaded
Active:  active (running)
```

Web server accessible at: `http://<your-public-ip>`

Screenshot 3

What to paste: Web browser showing the FinTech Lab web page accessed via the instance's public IP address.

Paste your screenshot here

Step 7: Monitor Cloud Resources and Billing

Objective: Understand cloud resource monitoring and cost implications.
Instructions:

1. Return to AWS Console
2. Navigate to **CloudWatch** for monitoring
3. View the EC2 instance metrics:
   - CPU utilization
   - Network in/out traffic
   - Disk read/write operations
4. Set up billing alerts for free tier usage
5. Check the **Billing Dashboard**
6. Review free tier usage and remaining benefits
7. Understand cost implications for:
   - Exceeding free tier limits (data transfer, storage)
   - Different instance types
   - Cross-region data transfer

Code/Command:

```
# View instance metrics via AWS CLI
aws cloudwatch get-metric-statistics \
--namespace AWS/EC2 \
--metric-name CPUUtilization \
--dimensions Name=InstanceId,Value=i-xxxxxxxx \
--start-time 2026-02-19T00:00:00Z \
--end-time 2026-02-21T23:59:59Z \
--period 3600 \
--statistics Average

# Check billing information
aws ce get-cost-and-usage \
--time-period Start=2026-02-01,End=2026-02-21 \
--granularity MONTHLY \
--metrics UnblendedCost \
--group-by Type=DIMENSION,Key=SERVICE
```

CloudWatch Monitoring Commands

Expected Output

CloudWatch Dashboard displays:
- CPU Utilization: 5-15% (idle instance)
- Network Traffic: Minimal (unless generating load)
- Disk Operations: Minimal
- Billing: $0.00 (within free tier limits)

The AWS Free Tier includes:
- 750 hours of t2.micro instance per month
- 15 GB of data transfer out per month
- 5 GB of S3 storage

Monitor these limits to avoid unexpected charges.

Screenshot 4

What to paste: AWS CloudWatch monitoring dashboard showing EC2 instance metrics (CPU utilization, network traffic).

Paste your screenshot here

Alternative: Azure Deployment

For students preferring Azure over AWS, follow similar steps:

1. Create free Azure account at `https://azure.microsoft.com/en-in/free`
2. Navigate to Virtual Machines
3. Create new VM with:
   - Image: Ubuntu Server 20.04 LTS
   - Size: B1s (free tier eligible)
   - Region: Southeast Asia or South India
4. Configure Network Security Group (NSG) with HTTP/HTTPS/SSH rules
5. Connect via SSH using the public IP
6. Install Nginx following the same commands as AWS

Screenshot 5

What to paste: Azure Portal Virtual Machines dashboard or alternative cloud platform of your choice.

Paste your screenshot here

Conceptual Background

Cloud Computing Fundamentals

Cloud computing is the delivery of computing resources (servers, storage, databases, software, analytics) over the internet (*"the cloud"*) on a pay-as-you-go basis. Instead of owning and maintaining physical infrastructure, organizations rent resources from cloud providers.

Key Benefits:

- **Cost Efficiency:** Pay only for what you use, no capital expenditure
- **Scalability:** Easily increase or decrease resources based on demand
- **Flexibility:** Access from anywhere with internet connection
- **Reliability:** Cloud providers maintain uptime SLAs (99.99%)
- **Security:** Providers employ enterprise-grade security practices

Cloud Deployment Models

1. **Public Cloud:** Resources shared among multiple organizations (AWS, Azure, GCP)
2. **Private Cloud:** Dedicated resources for single organization
3. **Hybrid Cloud:** Mix of public and private cloud resources
4. **Community Cloud:** Shared infrastructure for specific community (e.g., government)

Cloud Service Models

1. Infrastructure as a Service (IaaS)

Definition: Computing resources delivered over the internet - servers, storage, networking.

Examples: AWS EC2, Microsoft Azure VM, Google Compute Engine

You manage: Operating system, middleware, applications

Provider manages: Virtualization, servers, storage, networking

FinTech Use Cases:

- Hosting payment processing APIs

- Deploying trading engines
- Running data analytics pipelines
- Testing blockchain networks

2. Platform as a Service (PaaS)

Definition: Managed platform for building and deploying applications.

Examples: AWS Lambda, Heroku, Google App Engine

Provider manages: Infrastructure, OS, middleware

3. Software as a Service (SaaS)

Definition: Ready-to-use applications accessed via browser (no installation).

Examples: Salesforce, Microsoft 365, Slack

Security in Cloud Infrastructure

Shared Responsibility Model:

Component	Cloud Provider	Customer
Physical Data Center	AWS/Azure	-
Network Infrastructure	AWS/Azure	-
Virtualization Layer	AWS/Azure	-
Operating System	-	Customer
Applications	-	Customer
Security Groups/Firewalls	-	Customer
Access Control	-	Customer
Encryption	Shared	Customer

India-Specific Cloud Compliance

RBI Guidelines on Cloud Computing

The Reserve Bank of India (RBI) has issued guidelines for banks and financial institutions using cloud services:

1. **Data Residency:** Customer data must reside within India (no cross-border data transfer without explicit approval)

2. **Cloud Service Classification:** Banks can only use public cloud for non-critical systems
3. **Audit Requirements:** Regular third-party audits mandatory
4. **Incident Reporting:** Cloud-related incidents must be reported to RBI
5. **Encryption:** Data must be encrypted in transit and at rest

AWS India Region - Benefits

AWS Mumbai Region (ap-south-1) offers:

- ✓ Low latency for Indian users (< 20 ms)
- ✓ Compliance with RBI data residency requirements
- ✓ Data residency in Indian data centers
- ✓ Reduced data transfer costs
- ✓ Support for Indian payment systems integration

Real-World FinTech Example: Razorpay Infrastructure

Company: Razorpay (India's largest payments platform)

Infrastructure Strategy:

- ▷ Primary infrastructure on AWS
- ▷ Multiple EC2 instances running payment APIs
- ▷ RDS for transaction database
- ▷ S3 for logs and backups
- ▷ CloudFront for CDN acceleration
- ▷ All servers in Mumbai and Bangalore regions
- ▷ Multiple availability zones for high availability (99.99% uptime)

Security Approach:

- End-to-end encryption for transactions
- PCI DSS Level 1 compliance
- Regular security audits
- DDoS protection via AWS Shield
- Web Application Firewall (WAF)

Cost Optimization in Cloud

Strategies for Reducing Cloud Costs:

1. **Reserved Instances:** Commit to 1-3 year terms for 30-40% discount
2. **Spot Instances:** Use unused capacity for 70% discount (volatile)
3. **Auto-Scaling:** Scale instances based on demand
4. **Resource Tagging:** Track costs by project/department
5. **Data Transfer Optimization:** Use VPCs to minimize cross-region costs
6. **Storage Optimization:** Archive old data to Glacier (cheaper long-term storage)

AWS Free Tier Limits (12 months):

Service	Limit	Notes
EC2 (t2.micro)	750 hours/month	Only for 12 months
EBS Storage	30 GB (General Purpose)	Per month
Data Transfer	15 GB outbound/month	Includes all AWS services
CloudWatch	Free tier included	Limited to basic monitoring
RDS	750 hours of db.t2.micro	12 months only

Assessment & Deliverables

Deliverables Checklist

Item	Description	Type	Status
Screenshot 1	Security Group Configuration	Paste	□
Screenshot 2	SSH Connection to Instance	Paste	□
Screenshot 3	Web Server Access (Browser)	Paste	□
Screenshot 4	CloudWatch Monitoring Dashboard	Paste	□
Screenshot 5	Alternative Platform (Azure/Other)	Paste	□
Instance Info	Document Instance ID, Public IP	Text	□
Security Config	List security group rules	Text	□
Web Server Status	Show Nginx service status	Paste	□
Cost Analysis	Estimated/Actual usage charges	Text	□

Verification Checklist

Complete all items below before submitting:

- □ AWS/Azure Free Tier account created successfully
- □ EC2 instance running in correct region (ap-south-1 recommended)
- □ Security group configured with SSH (22), HTTP (80), HTTPS (443)
- □ SSH connection established and confirmed working
- □ Nginx web server installed and running
- □ Web server accessible from public internet
- □ Custom index.html page deployed and visible
- □ CloudWatch monitoring enabled and metrics visible
- □ All 5 required screenshots captured and ready to submit
- □ No unexpected charges (within free tier limits)
- □ Instance can be stopped/terminated without data loss

Grading Rubric

Criteria	Description	Points	Score
Account Setup	Free tier account created & verified	10	___/10
Infrastructure	EC2 instance launched in correct region	15	___/15
Security Config	Security groups configured correctly	15	___/15
SSH Access	Successfully connected via SSH	15	___/15
Web Server	Nginx installed & serving web page	20	___/20
Monitoring	CloudWatch metrics observable	10	___/10
Screenshots	All 5 screenshots submitted clearly	10	___/10
Documentation	Answers & explanations complete	5	___/5
	TOTAL	**100**	___/100

Assessment Questions

Answer the following questions in your submission:

Q1. What is the difference between IaaS, PaaS, and SaaS? Provide examples for each.

Q2. Why is the Mumbai region (ap-south-1) important for FinTech applications in India?

Q3. Explain the shared responsibility model in cloud computing. What are your responsibilities?

Q4. What are the advantages of using cloud infrastructure for payment processing systems?

Q5. How do security groups work? What ports did you open and why?

Q6. Calculate the estimated monthly cost if your instance runs 24/7 for a full year (after free tier expires).

Q7. What are the compliance requirements for Indian banks using cloud services (RBI guidelines)?

Q8. Describe how you would set up auto-scaling for your web server if traffic increased 10x.

Appendix A: AWS CLI Commands Reference

EC2 Instance Management

```
# List all instances
aws ec2 describe-instances --region ap-south-1

# Get specific instance details
aws ec2 describe-instances --instance-ids i-xxxxxxx --region ap-south-1

# Stop instance
aws ec2 stop-instances --instance-ids i-xxxxxxx --region ap-south-1

# Start instance
aws ec2 start-instances --instance-ids i-xxxxxxx --region ap-south-1

# Terminate instance (WARNING: Deletes instance)
aws ec2 terminate-instances --instance-ids i-xxxxxxx --region ap-south-1

# Get instance status
aws ec2 describe-instance-status --instance-ids i-xxxxxxx --region ap-south-1

# Modify instance type (must stop instance first)
aws ec2 modify-instance-attribute --instance-id i-xxxxxxx \
--instance-type "{\"Value\":␣\"t2.small\"}" --region ap-south-1
```

Useful AWS CLI Commands

Appendix B: Nginx Web Server Configuration

Common Nginx Configuration

```
# Main Nginx config file
sudo nano /etc/nginx/nginx.conf

# Site-specific config
sudo nano /etc/nginx/sites-available/default

# Restart Nginx after changes
sudo systemctl restart nginx

# Check syntax for errors
sudo nginx -t

```

```
# View Nginx logs
sudo tail -f /var/log/nginx/access.log
sudo tail -f /var/log/nginx/error.log

# Change file permissions
sudo chown -R www-data:www-data /var/www/html
sudo chmod -R 755 /var/www/html
```

Basic Nginx Configuration

Sample Nginx Virtual Host Configuration

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        server_name _;

        root /var/www/html;
        index index.html index.htm index.nginx-debian.html;

        location / {
                try_files $uri $uri/ =404;
        }

        # Enable compression
        gzip on;
        gzip_types text/plain text/css text/javascript;

        # Block access to sensitive files
        location ~ /\. {
                deny all;
        }
}
```

Advanced Virtual Host Configuration

Appendix C: Troubleshooting Guide

Common Issues and Solutions

Problem: `ssh: connect to host ... port 22: Connection refused`
Solutions:

1. Verify instance is in "Running" state
2. Check security group allows port 22 from your IP
3. Wait 1-2 minutes for instance to fully boot
4. Verify you're using correct key file
5. Run: `chmod 400 your-key.pem`

Problem: Cannot access `http://public-ip` in browser
Solutions:

1. Verify Nginx is running: `sudo systemctl status nginx`
2. Check security group allows port 80 and 443
3. Verify web server is listening: `sudo netstat -tlnp | grep :80`
4. Check firewall rules: `sudo ufw status`
5. Restart Nginx: `sudo systemctl restart nginx`

Problem: Receiving AWS bills beyond free tier
Solutions:

1. Check free tier usage: AWS Console > Billing Dashboard
2. Stop unused instances: `aws ec2 stop-instances ...`
3. Delete unused volumes and snapshots
4. Check data transfer charges (expensive for cross-region)
5. Terminate instances if no longer needed
6. Set up billing alerts

Appendix D: Security Best Practices

Cloud Infrastructure Security Checklist

- ☐ Security groups restrict traffic to minimum required ports
- ☐ SSH access limited to your IP address (not 0.0.0.0/0)
- ☐ Regular OS and application updates applied
- ☐ Access logs enabled and monitored
- ☐ Instances backed up regularly
- ☐ Encryption enabled in transit (HTTPS) and at rest
- ☐ IAM users configured with least privilege principle
- ☐ Multi-factor authentication (MFA) enabled on root account
- ☐ CloudTrail enabled for audit logging
- ☐ DDoS protection enabled (AWS Shield)

India-Specific Security and Compliance

RBI Compliance Checklist

- ✓ All data stored within India region (ap-south-1)
- ✓ Encryption of sensitive financial data
- ✓ Regular third-party security audits
- ✓ Incident response plan documented
- ✓ Disaster recovery plan with backup
- ✓ Access control and audit logging enabled
- ✓ Compliance with SEBI cybersecurity requirements

NPCI Guidelines for Payment Systems

Payment systems integrated with UPI or other NPCI networks must comply with:

- PCI DSS Level 1 certification
- ISO 27001 information security management

- End-to-end encryption for transactions
- Tokenization for sensitive data
- Regular penetration testing
- Fraud detection and prevention systems

Appendix E: Additional Resources

Official Documentation

→ AWS EC2 Documentation: `https://docs.aws.amazon.com/ec2`

→ AWS Free Tier: `https://aws.amazon.com/free`

→ RBI Guidelines on Cloud: `https://www.rbi.org.in`

→ SEBI Cybersecurity Guidelines: `https://www.sebi.gov.in`

→ Nginx Documentation: `https://nginx.org/en/docs`

→ Ubuntu Server Guide: `https://ubuntu.com/server/docs`

Learning Resources

- "Cloud Computing Fundamentals" - AWS Training
- "Linux Command Line Basics" - Linux Academy
- "Web Server Security" - SANS Institute
- "FinTech Infrastructure Design" - Case studies from Razorpay, PhonePe

Tools Used in This Practical

Tool	**Purpose**	**Cost**
AWS EC2	Virtual machine hosting	Free (12 months)
AWS Security Groups	Firewall configuration	Free
AWS CloudWatch	Performance monitoring	Free (basic)
Nginx	Web server software	Free (open source)
ssh/PuTTY	Remote access	Free
curl/wget	Command-line HTTP client	Free (included)

Appendix F: Important Contact Information

Support Resources

- ⋆ **AWS Support:** https://console.aws.amazon.com/support
- ⋆ **AWS Training:** https://www.aws.training/
- ⋆ **Institution IT Services:** [Contact from institution]
- ⋆ **RBI Helpdesk:** complaints@rbi.org.in
- ⋆ **SEBI Helpdesk:** grievance@sebi.gov.in

—END OF LAB MANUAL—

Document Version: 1.0

IT Management & Audits – Practical Lab Series