



AWS + DEVOPS

• SPECIALIZATION •

FOUNDATION PROGRAM



contact@regexsoftware.com



www.regexsoftware.com

ABOUT THE PROGRAM

At Regex Software, we offer a comprehensive 18–24 Month AWS + DevOps Specialization Foundation Program designed especially for B.Tech, BCA, B.Sc, BBA, B.Com and other college students who want to build a strong career in cloud computing and DevOps.

This program is ideal for 1st and 2nd year students who want to start learning cloud and DevOps technologies early and gain a competitive edge in the industry. The course is designed to build strong fundamentals first and then gradually move toward advanced AWS services and DevOps practices.

At Regex Software, we believe in practical learning rather than only theoretical knowledge. Our training approach combines conceptual understanding with hands-on implementation through real-world projects, live assignments, and industry case studies. Throughout the 18–24 month journey, students continuously work on practical tasks that help them understand how cloud and DevOps tools are used in real companies and business environments.

The curriculum covers AWS Core Services, DevOps tools and pipelines, Infrastructure as Code, CI/CD, Monitoring, Security, and Cost Optimization. Students also learn Git & GitHub, Docker, Kubernetes, Terraform, Jenkins, Ansible, and other modern tools. The entire learning path is designed according to current industry requirements so students become job-ready with practical cloud and DevOps skills.



MODE

Physical
(Jaipur & Ahmedabad)
or Online
(Google Meet)



DURATION

18–24 months
+ 6 months support



PARTICIPANTS

18 - 20 Per
Batch

WHAT YOU WILL LEARN ?



AWS Core Services



Amazon S3



Amazon EC2



Amazon RDS



Amazon VPC



Amazon CloudWatch



AWS IAM



AWS Cost Optimization



AWS CloudFormation



AWS Lambda



AWS CodePipeline



AWS CodeBuild



Amazon EKS



Docker



Kubernetes



Terraform



Jenkins



Ansible



CI / CD



Git



GitHub



Security Best Practices



Monitoring & Logging



DevOps Culture

AWS + DEVOPS

with

AI Specialization

FOUNDATION PROGRAM



STUDY MATERIAL

- E-Notes
- Assignments & PDF test
- Live Video Lectures
- Access of Recordings & Study Material
- Mentorship Support
- Work on multiple Projects & Use Cases
- Work on Live Projects
- Interview Preparation Guidance

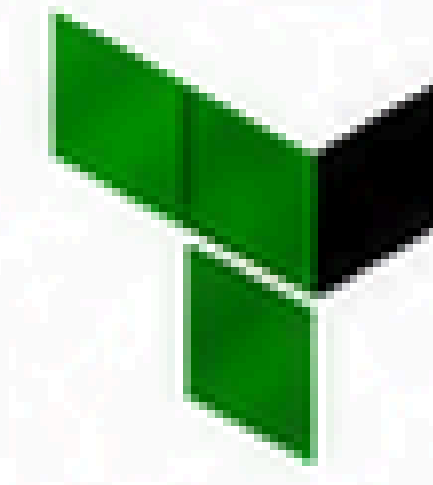


OUTPUT

- Build production-ready cloud infrastructure on AWS
- Become job-ready DevOps Engineer with real-world projects
- Understand CI/CD, Automation & Infrastructure as Code in depth
- Build projects on multiple domains
- Work on more than 25 Use CASES & Projects
- Learn to monitor, secure & scale applications on AWS
- Learn DevOps tools & practices used in industry



AWS
Cloud



Terraform



Docker



Kubernetes



Jenkins



Ansible

PACKAGE OFFERED SO FAR

IT CANDIDATES



Minimum Package
4 LPA



Average Package
4 – 7 LPA

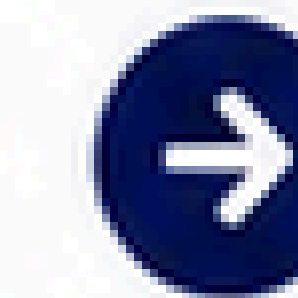


Overall Highest Package
36 LPA

NON-IT CANDIDATES



Minimum Package
3 LPA



Average Package
3.5 – 5.5 LPA



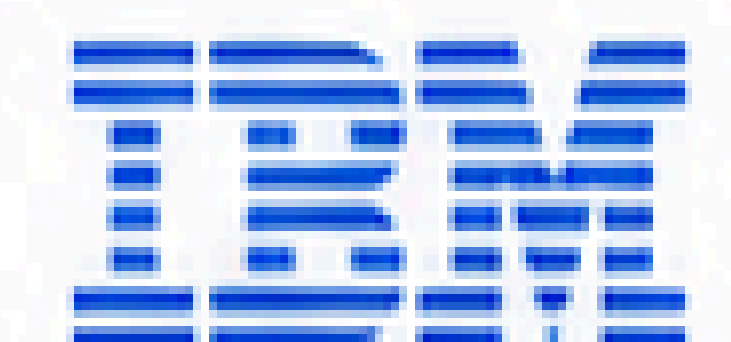
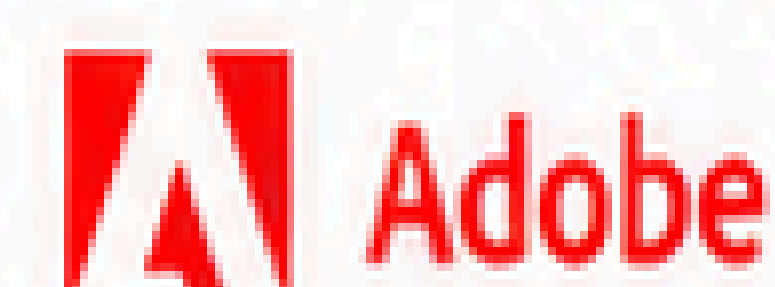
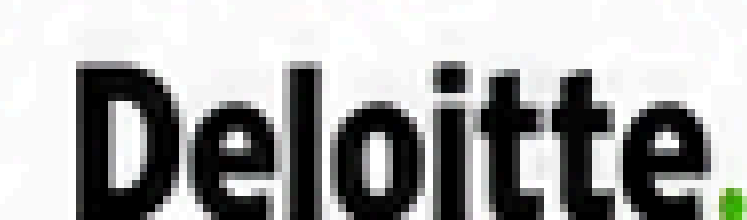
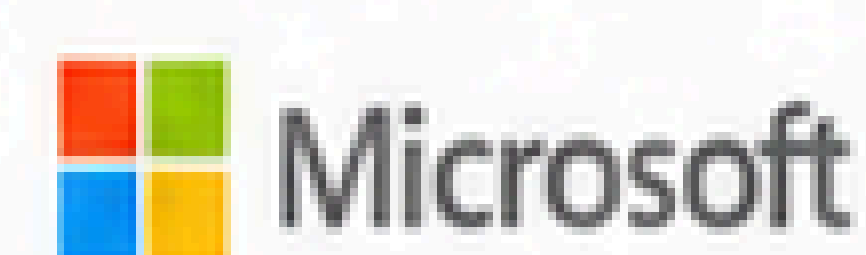
Overall Highest Package
14.5 LPA



EXTRA SESSIONS:

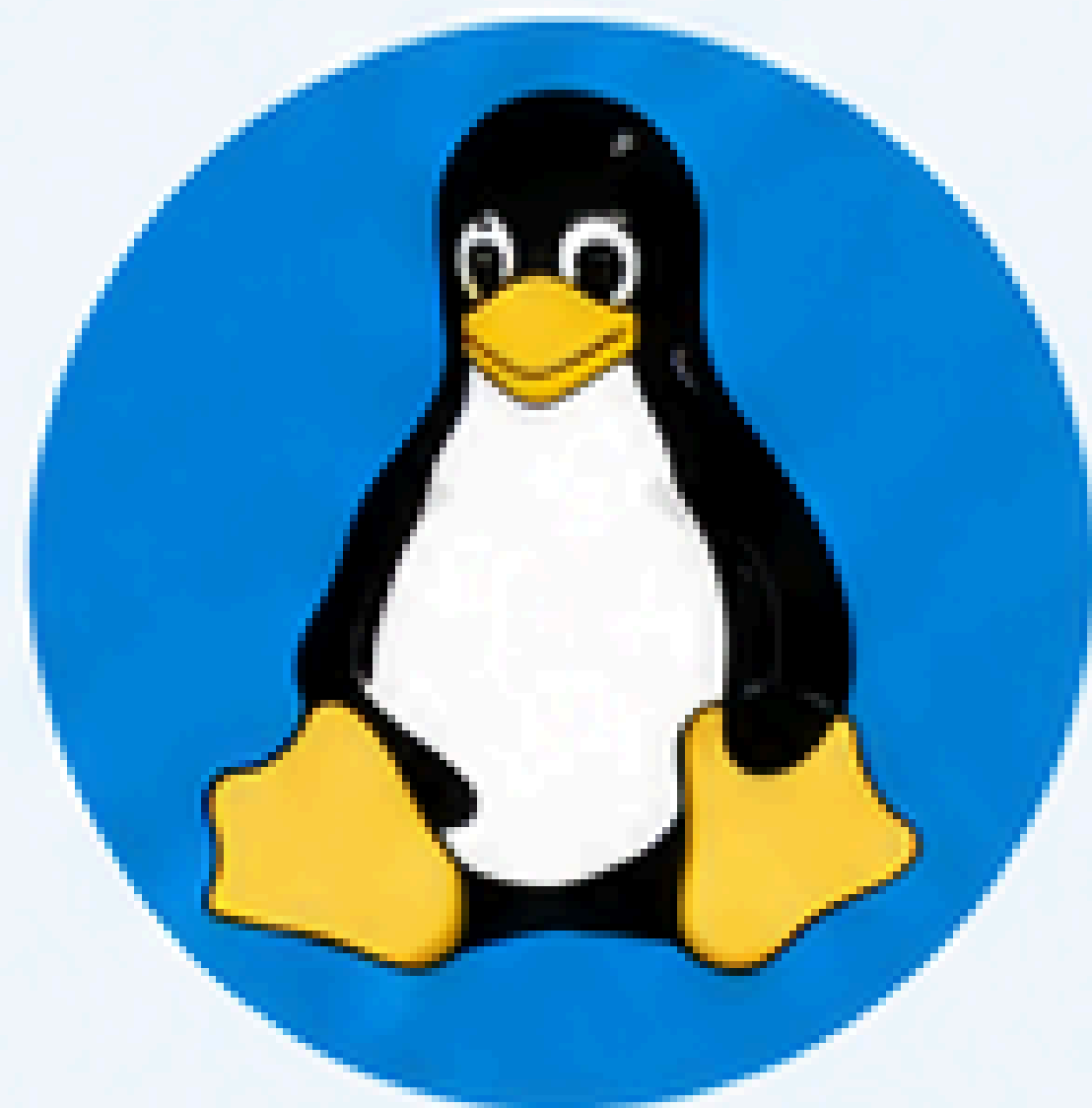
Additional Session on AWS Services, DevOps Tools, Cloud Security, Interview Preparation & Soft Skills and many more for all students.

OUR STUDENTS PLACED // PARTNERSHIP



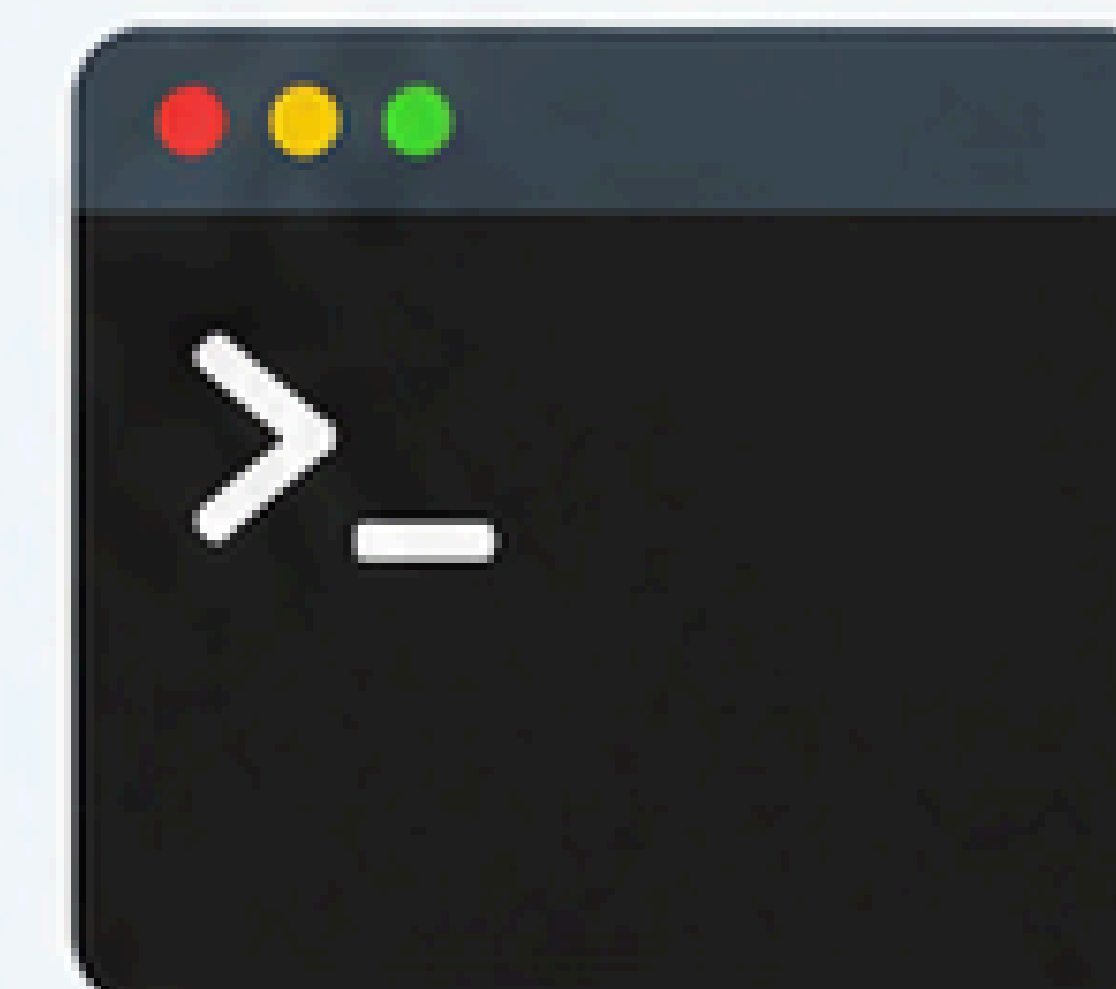
COURSE CONTENT:

PHASE 1



1. LINUX BASICS

- Introduction to Linux
- Basic Commands of Linux OS
- Vi Editor
- Tar Archive
- User Management and Permission



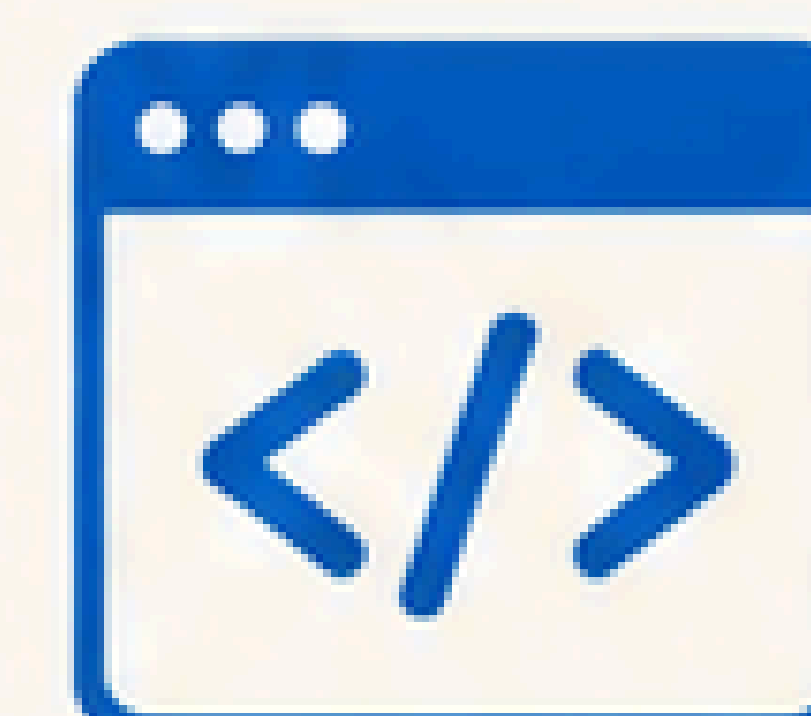
2. GITHUB

- Git vs GitHub
- Git Commands
- Git Branches
- Branching and Merging
- Git Push vs Pull Commands



3. C PROGRAMMING – CORE CONCEPTS

- Data Types
- Variables
- Variable Scope (Local, Global)
- Constants
- Operators
- Decision Making Statements (if Statement, if..else, Switch)
- Loops (While Loop, Do-While Loop, for-loop)



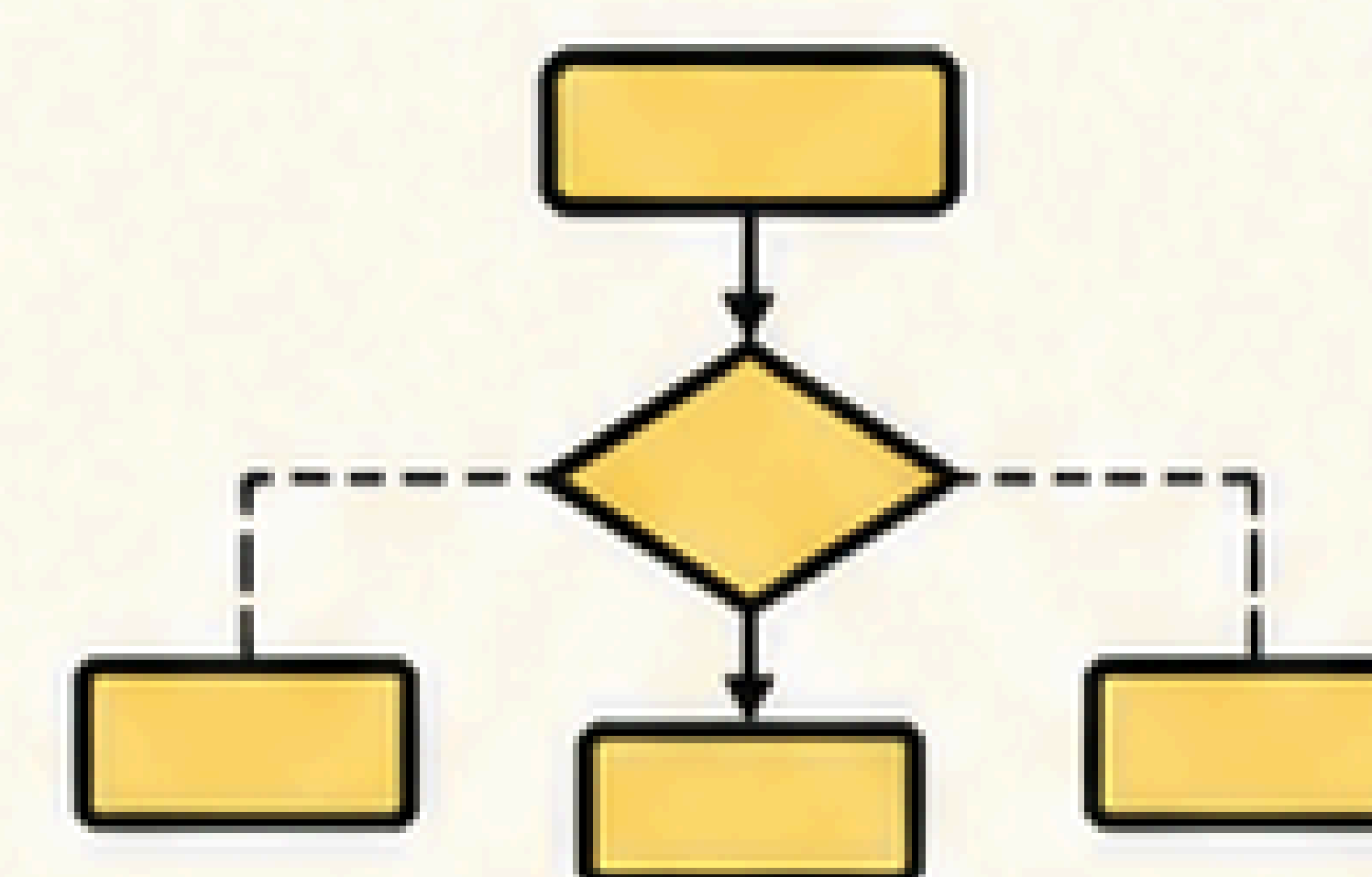
4. C PROGRAMMING – ADVANCED BASICS

- While Loop
- Do-While Loop
- for-loop
- Basic I/O Functions



5. LOGIC BUILDING

- Logical Thinking and Approach
- Flowcharts and Pseudocode
- Pattern Recognition
- Basic Problem-Solving



6. BASICS PROBLEM-SOLVING

- Understanding Problem Statements
- Breaking Problems into Steps
- Test Cases and Debugging
- Optimizing Solutions (Basics)



PHASE 2



1. C++ PROGRAMMING (OOP CONCEPTS)

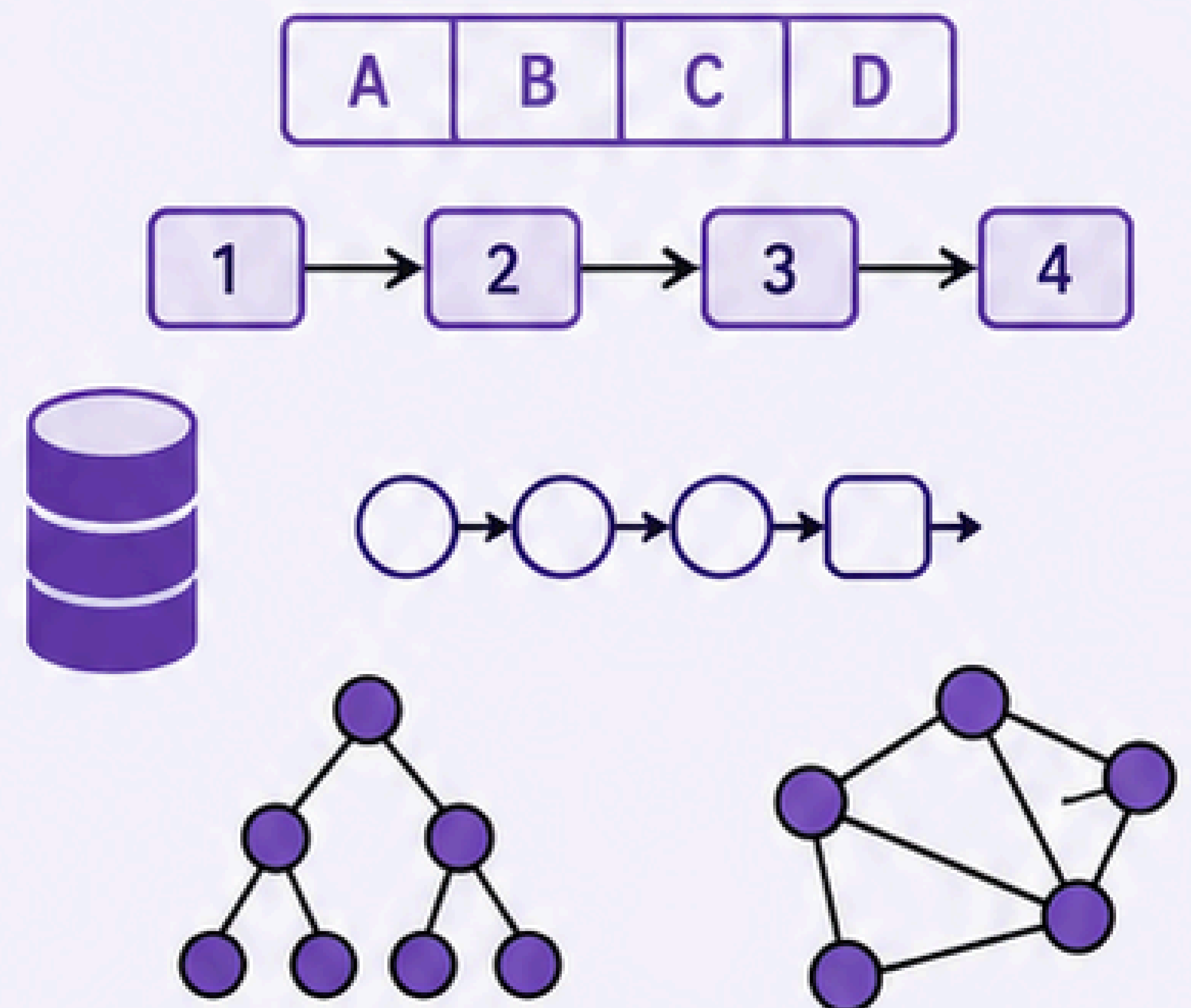
- Classes and Objects
- Constructors and Destructors
- this Pointer
- Access Specifiers: public, private, protected
- Getter and Setter Functions

```
class Student {  
public:  
    void set(int x);  
    int get();  
};
```



2. DATA STRUCTURES (ARRAY, LINKED LIST, STACK, QUEUE, TREES, GRAPHS)

- Arrays
- Linked List
- Stack
- Queue
- Trees
- Graphs



3. ALGORITHMS (SORTING, SEARCHING, RECURSION)

- Sorting Algorithms
- Searching Algorithms
- Recursion and Backtracking
- Time and Space Complexity

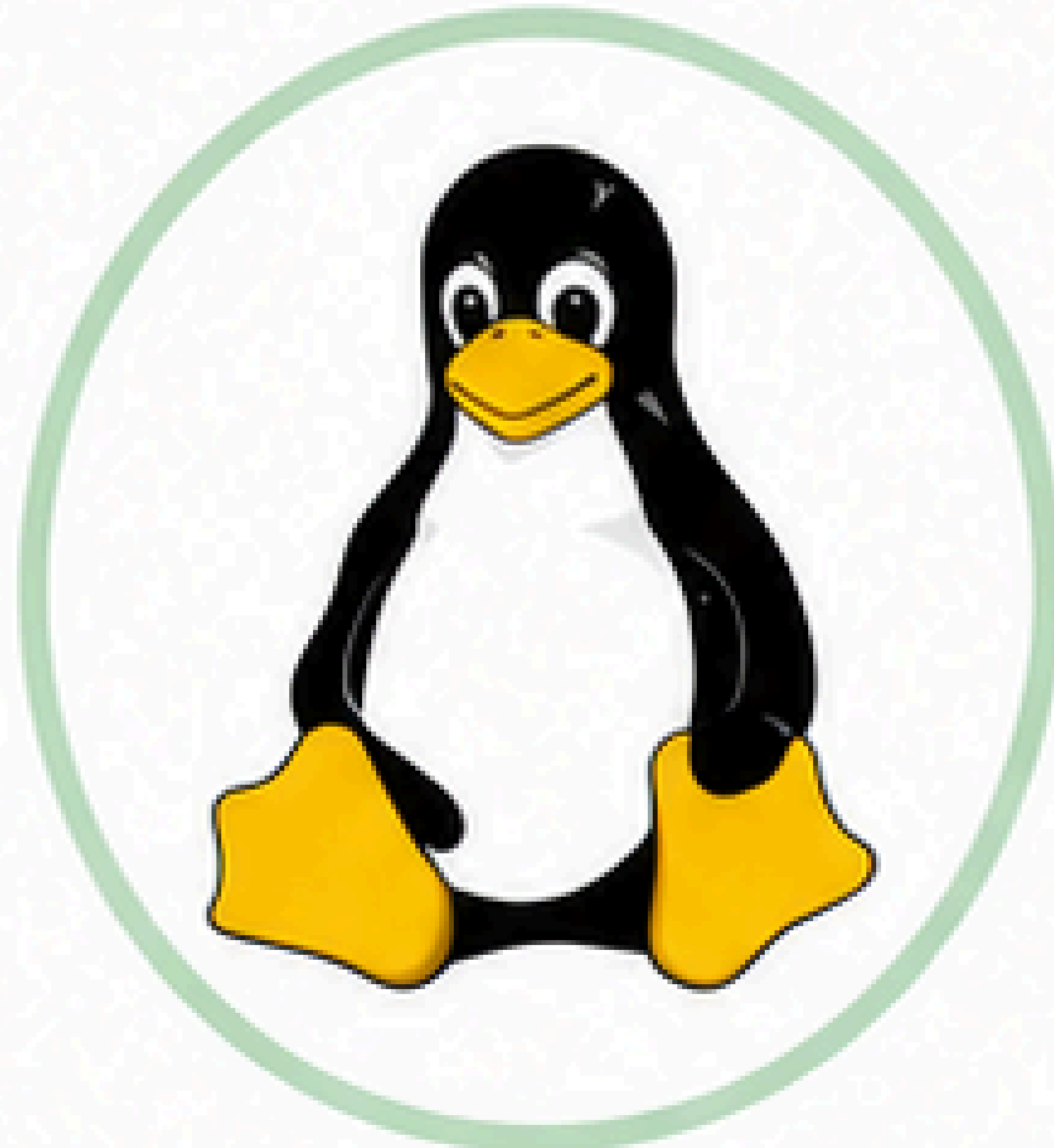


4. PROBLEM SOLVING (LEETCODE / CODEFORCES LEVEL)

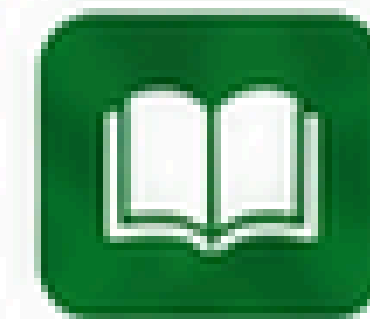
- Problem Solving Techniques
- Pattern Recognition
- Practice on LeetCode
- Practice on Codeforces



PHASE 3



1. LINUX FUNDAMENTALS

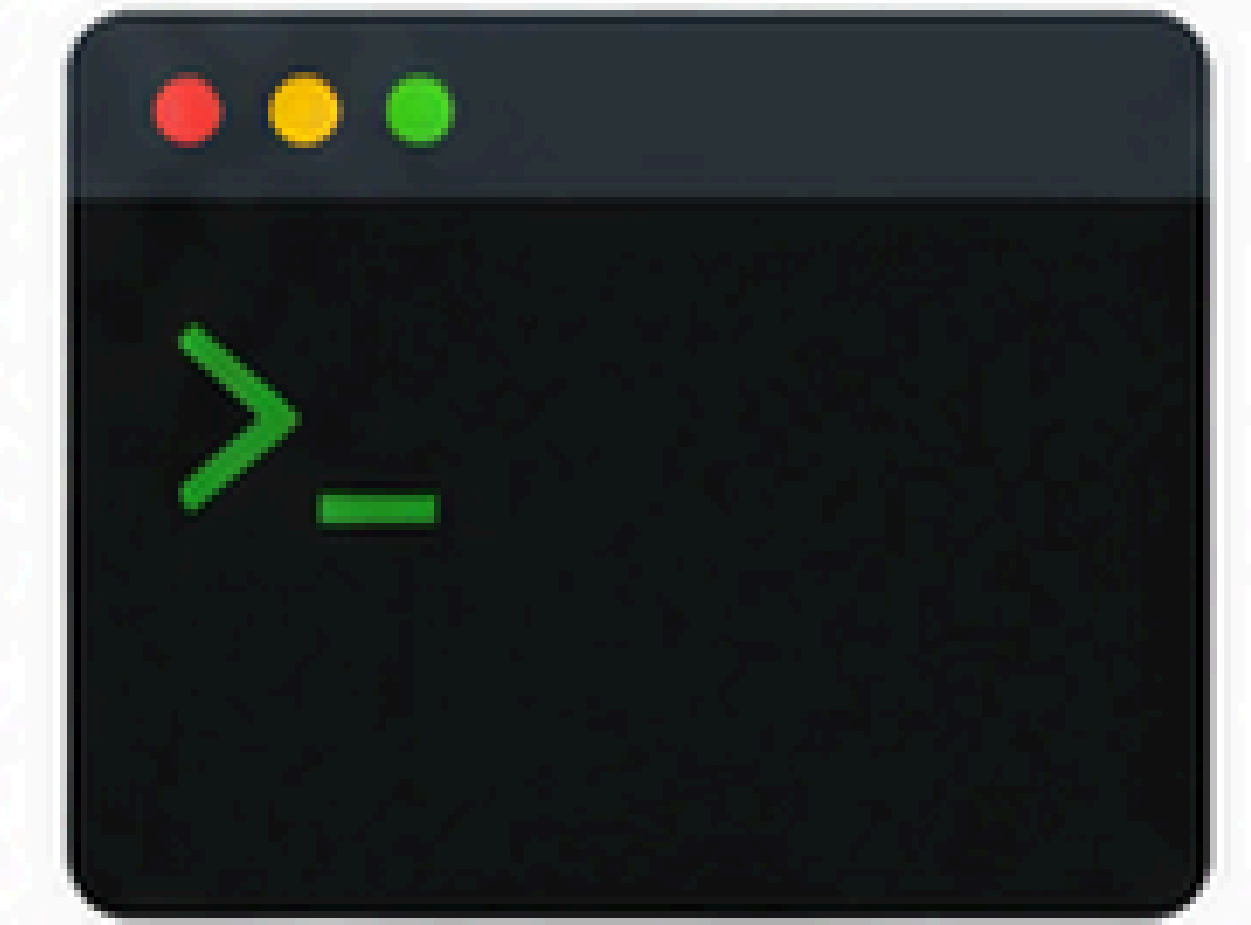


WHAT YOU'LL LEARN

- Introduction to Linux & Open Source
- Linux System Architecture
- Basic Commands & File Operations
- Directory Structure
- File Permissions (rwx) & Ownership
- User & Group Management
- Process Management
- Package Management (apt/yum)
- Shell Scripting Basics
- Text Editors (vi/nano)
- Cron Jobs (Task Scheduling)

COMMON COMMANDS

ls	List files and directories
cd	Change directory
pwd	Print working directory
cp	Copy files/directories
mv	Move/Rename files
rm	Remove files/directories
chmod	Change file permissions
ps	Check running processes
top	Monitor system processes
tar	Archive files
grep	Search text in files



KEY CONCEPTS

- File System
- Permissions
- Processes
- Shell & CLI



2. NETWORK BASICS



WHAT YOU'LL LEARN

- Introduction to Computer Networks
- OSI Model & TCP/IP Model
- IP Addressing (IPv4)
- Subnetting Basics
- DNS (Domain Name System)
- DHCP (Dynamic Host Configuration Protocol)
- HTTP/HTTPS
- Ports & Protocols (TCP, UDP, ICMP)
- Networking Tools (ping, traceroute, nslookup, netstat)
- Firewalls & Network Security Basics

TCP/IP MODEL

Application Layer
(HTTP, DNS, FTP)

Transport Layer
(TCP, UDP)

Internet Layer
(IP, ICMP)

Network Access Layer
(MAC, ARP)

KEY CONCEPTS

- IP Addressing
- Subnets
- Protocols
- Ports
- Network Tools



3. PYTHON SCRIPTING



WHAT YOU'LL LEARN

- Python Basics (Syntax, Indentation, Comments)
- Variables & Data Types
- Operators
- Input & Output
- Conditional Statements
- Loops (for, while)
- Functions
- Lists, Tuples, Dictionaries, Sets
- File Handling
- Exception Handling
- Simple Automation Scripts

SAMPLE PYTHON CODE

```
# Simple Python Script
name = input("Enter your name: ")
age = int(input("Enter your age: "))

if age >= 18:
    print(f"Hello {name}! You are an adult.")
else:
    print(f"Hello {name}! You are a minor.")
```

KEY CONCEPTS

- Syntax
- Data Structures
- Control Flow
- Functions
- File Handling

PHASE 4



1. AWS CORE SERVICES

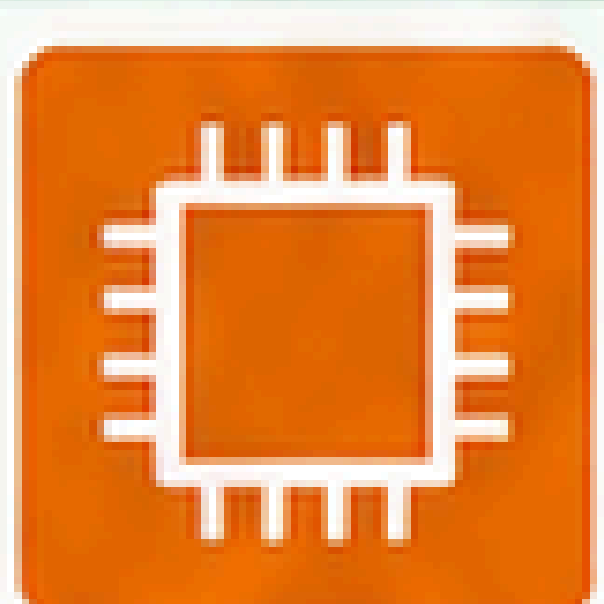
WHAT YOU'LL LEARN

- Introduction to AWS Cloud
- AWS Global Infrastructure
- Regions, AZs & Edge Locations
- AWS Management Console
- IAM (Identity & Access Management)
- AWS Billing & Pricing Basics
- AWS Support Plans
- AWS Well-Architected Framework
- Core AWS Services Overview

CORE SERVICE CATEGORIES

 Compute	 Storage	 Database	 Networking & Content Delivery
 Security, Identity & Compliance	 Management & Governance	 Developer Tools	 Analytics

2. EC2, S3, IAM, VPC



EC2



S3



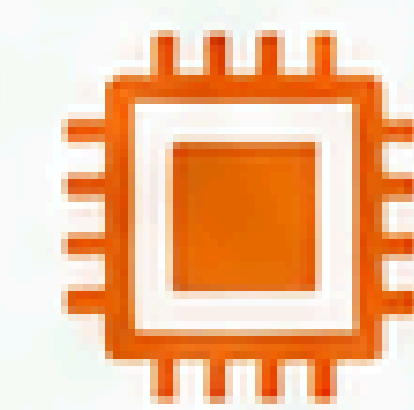
IAM



VPC

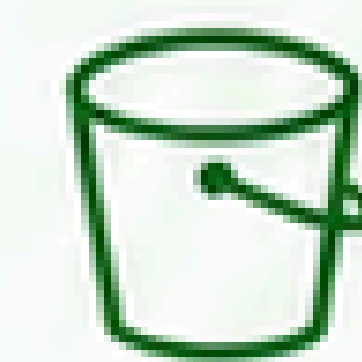
WHAT YOU'LL LEARN

- Amazon EC2 – Launch & Manage Instances
- Amazon S3 – Storage & Management
- IAM – Users, Groups, Roles, Policies
- VPC – Virtual Private Cloud Basics
- Subnets, Route Tables, Internet Gateway
- Security Groups & Network ACLs
- Elastic IP & NAT Gateway
- Key Pairs & SSH Access
- Best Practices for Security



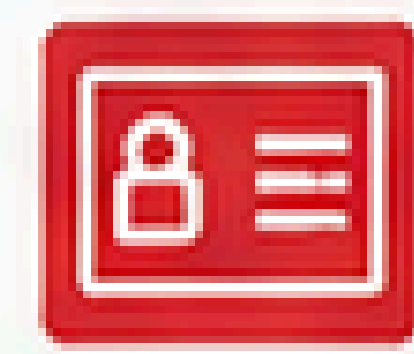
EC2 (Elastic Compute Cloud)

- Launch instances
- Instance types
- Security groups
- Key pairs
- User data



S3 (Simple Storage Service)

- Buckets & objects
- Storage classes
- Permissions
- Versioning
- Lifecycle policies



IAM (Identity & Access Management)

- Users, Groups, Roles
- Policies & Permissions
- IAM Best Practices



VPC (Virtual Private Cloud)

- VPC components
- Subnets
- Route tables
- Internet Gateway
- Security groups

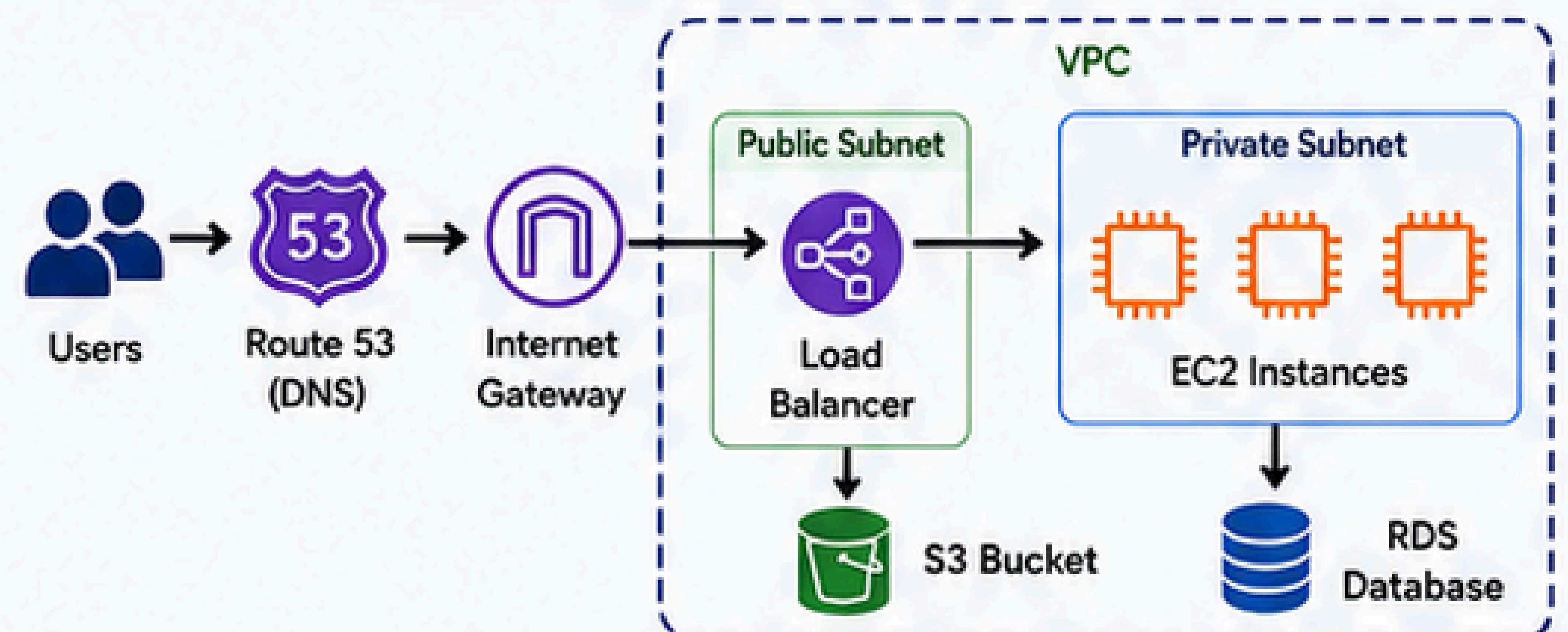


3. CLOUD ARCHITECTURE

WHAT YOU'LL LEARN

- Cloud Architecture Fundamentals
- Design Principles (Scalability, Reliability, High Availability, Security)
- Well-Architected Framework
- High Availability & Fault Tolerance
- Load Balancing Concepts
- Auto Scaling Basics
- Multi-Tier Architecture
- Cost Optimization Strategies
- Real-World Architecture Examples

ARCHITECTURE CONCEPTS



KEY TAKEAWAYS



Secure by Design



Scalable & Elastic



Reliable & Highly Available



Cost Optimized

PHASE 5



1. DOCKER



WHAT YOU'LL LEARN

- Introduction to Docker
- Docker Architecture
- Docker Images
- Docker Containers
- Dockerfile – Build Images
- Docker Commands (run, build, ps, logs, exec)
- Docker Volumes & Bind Mounts
- Docker Networking
- Docker Compose
- Best Practices for Docker

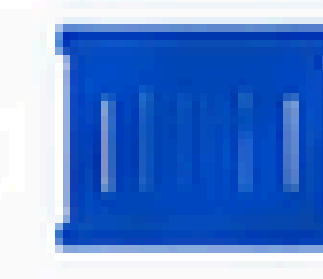
COMMON DOCKER COMMANDS

<code>docker build</code>	Build an image from Dockerfile
<code>docker run</code>	Run a container
<code>docker ps</code>	List running containers
<code>docker images</code>	List all images
<code>docker stop</code>	Stop a container
<code>docker rm</code>	Remove a container
<code>docker rmi</code>	Remove an image
<code>docker logs</code>	View container logs
<code>docker exec</code>	Execute command in container

DOCKER COMPONENTS



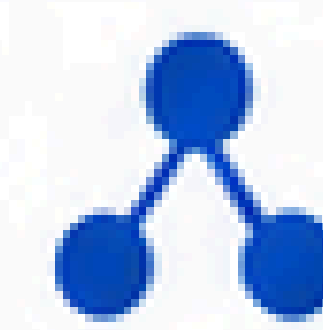
Images
Read-only templates



Containers
Running instances



Volumes
Persistent data



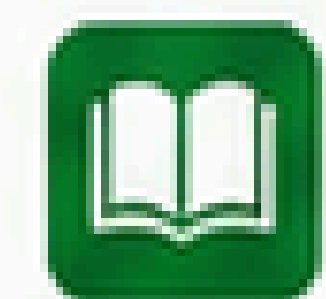
Networks
Container communication



Dockerfile
Instructions to build image



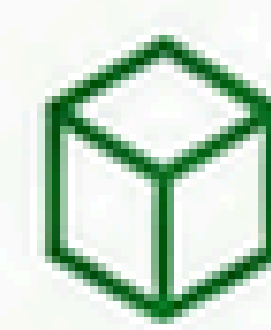
2. KUBERNETES



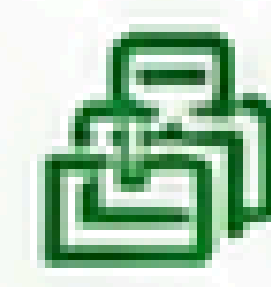
WHAT YOU'LL LEARN

- Introduction to Kubernetes
- Kubernetes Architecture
- Nodes, Pods & Containers
- Deployments
- ReplicaSets
- Services & Networking
- ConfigMaps & Secrets
- Persistent Volumes (PV) & Claims (PVC)
- Namespaces & Resource Quotas
- Helm – Package Manager
- Kubernetes CLI (kubectl) Basics
- Monitoring & Logging Overview

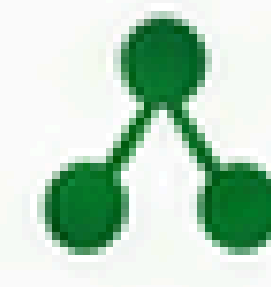
KUBERNETES KEY CONCEPTS



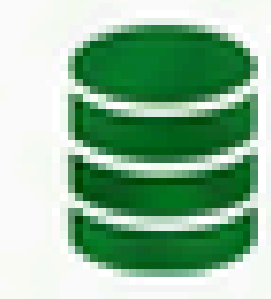
Pod
Smallest deployable unit (one or more containers)



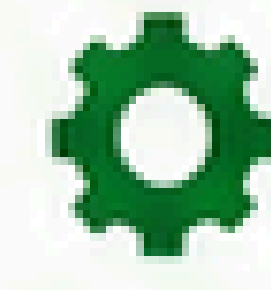
Deployment
Manages replica sets and ensures desired state



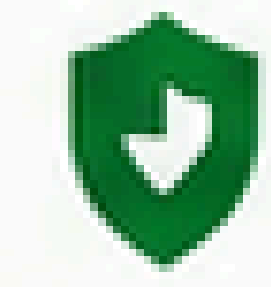
Service
Stable endpoint to access pods



Volume
Persistent storage for pods



Namespace
Logical isolation within a cluster



ConfigMap & Secret
Manage configuration & sensitive data

KUBECTL BASIC COMMANDS

<code>kubectl get pods</code>	List pods
<code>kubectl get nodes</code>	List nodes
<code>kubectl get svc</code>	List services
<code>kubectl apply -f file.yaml</code>	Apply config
<code>kubectl describe pod <name></code>	Describe pod
<code>kubectl logs <pod></code>	View logs
<code>kubectl delete -f file.yaml</code>	Delete resources



3. CONTAINERIZATION



WHAT YOU'LL LEARN

- What is Containerization?
- Benefits of Containerization
- Containers vs VM
- Use Cases & Real-World Examples
- Container Orchestration Overview
- CI/CD with Containers (Overview)
- Best Practices & Security
- Considerations

CONTAINERIZATION BENEFITS

 Lightweight	 Consistent Environment	 Portability	 Scalability
 Faster Deployment	 Resource Efficiency	 Isolation	 Cost Effective

USE CASES

- ✓ Microservices Architecture
- ✓ Dev & Test Environments
- ✓ CI/CD Pipelines
- ✓ Application Portability
- ✓ Cloud-Native Applications

PHASE 6



1. JENKINS



WHAT YOU'LL LEARN

- Introduction to Jenkins
- Jenkins Architecture
- Installation & Setup
- Jobs in Jenkins (Freestyle & Pipeline)
- Build Triggers
- Jenkins Plugins
- Managing Credentials
- Shared Libraries
- Blue Ocean Overview
- Best Practices for Jenkins

KEY CONCEPTS



Automation Server
Automate build, test & deployment processes



Extensible
Thousands of plugins for integrations



Distributed Builds
Execute jobs on multiple nodes/agents

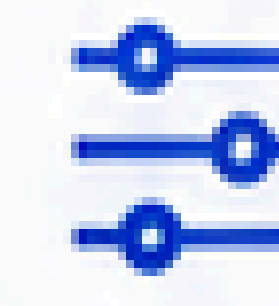


Security
Manage users, roles and access

COMMON JENKINS FEATURES



Pipeline as Code



Parameterized Builds



Build History



Email Notifications



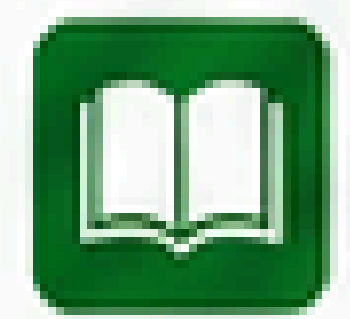
Artifact Management



Role-Based Access Control



2. CI/CD PIPELINES



WHAT YOU'LL LEARN

- Introduction to CI/CD
- CI vs CD vs CD
- CI/CD Pipeline Concepts
- Pipeline Lifecycle
- Source Code Management (Git)
- Automated Build & Test
- Artifact Management
- Automated Deployment
- Environment Management
- Pipeline as Code (Jenkinsfile)
- Best Practices & Strategies

PIPELINE STAGES



Plan



Code



Build



Test



Deploy



Monitor

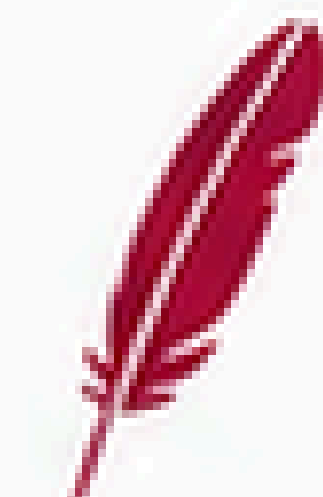
TOOLS IN CI/CD ECOSYSTEM



Git



Jenkins



Maven



SonarQube



Docker



Nexus



WHAT YOU'LL LEARN

- Introduction to IaC
- Why Terraform?
- Terraform Architecture
- Terraform Installation
- Providers in Terraform
- Resources, Data Sources
- Variables & Outputs
- State Management
- Modules
- Workspaces
- Terraform CLI Commands
- Best Practices for IaC

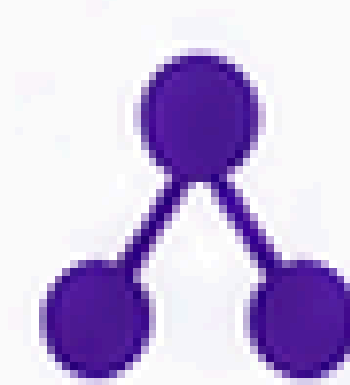
TERRAFORM CORE CONCEPTS



Declarative
Define desired state, Terraform does the rest



Infrastructure as Code
Versionable, Reusable & Consistent



Execution Plan
Preview changes before applying

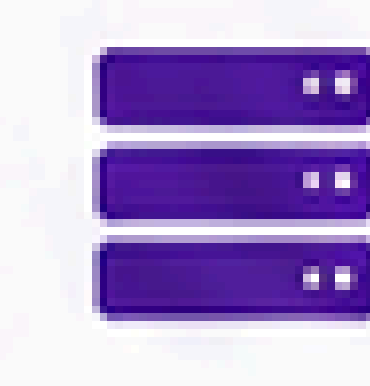


State Management
Track infrastructure state securely

COMMON USE CASES



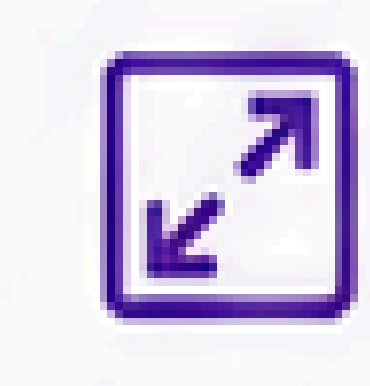
Provision Infrastructure (AWS, Azure, GCP, etc.)



Environment Management (Dev, Staging, Prod)



Networking & Security Automation



Scalable & Repeatable Deployments



Cost Optimization & Governance



3. TERRAFORM & IaC

PHASE 7



1. MONITORING & SECURITY



WHAT YOU'LL LEARN

- Importance of Monitoring in DevOps
- Infrastructure & Application Monitoring
- Centralized Logging
- Alerting & Notifications
- Security Best Practices
- Secrets Management
- Vulnerability Scanning
- Identity & Access Management
- Compliance & Governance
- Incident Response Basics

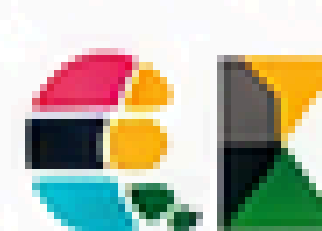
MONITORING TOOLS



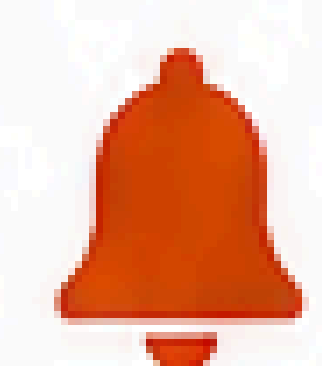
Prometheus
Metrics collection & alerting



Grafana
Visualization & dashboards

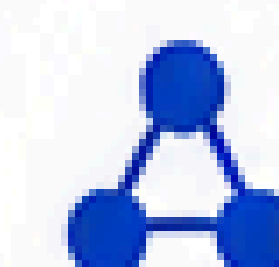


ELK Stack (Elasticsearch, Logstash, Kibana)
Centralized logging & analytics



Alertmanager
Alert routing & management

SECURITY FOCUS AREAS



Network Security



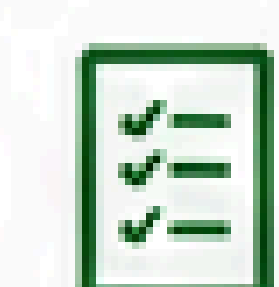
Secrets Management (HashiCorp Vault)



IAM & Least Privilege



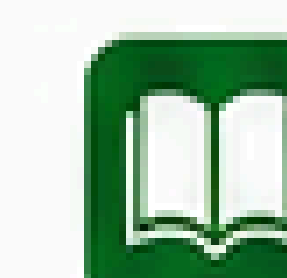
Image Scanning (Trivy, Clair)



Compliance & Auditing



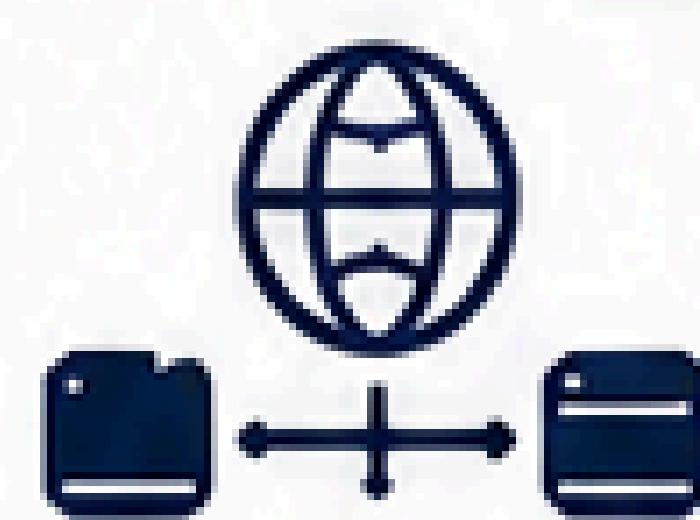
2. REAL-WORLD DEVOPS PROJECTS



WHAT YOU'LL LEARN

- End-to-End DevOps Project Lifecycle
- Requirement Analysis & Planning
- Infrastructure Provisioning (IaC)
- Application Deployment with CI/CD
- Monitoring, Logging & Alerting Setup
- Security Implementation
- Backup & Disaster Recovery
- Scaling & Performance Optimization
- Documentation & Handover

SAMPLE PROJECT IDEAS



Deploy a Web Application (End-to-End)



CI/CD Pipeline for Microservices Application



Multi-Tier App Deployment on AWS



Containerized App Deployment on Kubernetes



Infrastructure as Code with Terraform on AWS



Secure DevOps Pipeline with Monitoring



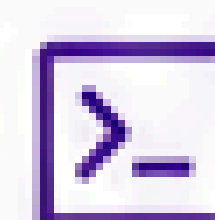
3. RESUME + INTERVIEW PREPARATION



WHAT YOU'LL LEARN

- Crafting a Strong DevOps Resume
- Highlighting Projects & Achievements
- GitHub Profile Optimization
- Technical Interview Preparation
- DevOps Tools & Concepts Q&A
- Scenario-Based Questions
- System Design Basics for DevOps
- Mock Interviews
- Communication & Soft Skills Tips

INTERVIEW TOPICS COVERED



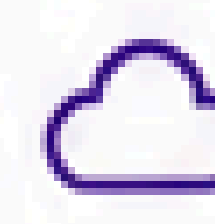
Linux, Networking, and Scripting Basics



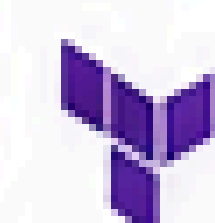
Docker & Kubernetes



CI/CD & Jenkins



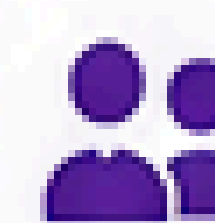
AWS & Cloud Concepts



Terraform & IaC

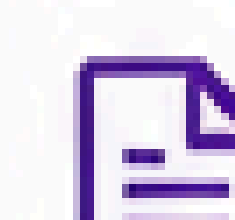


Monitoring & Security

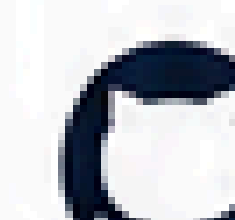


Real-world Scenarios

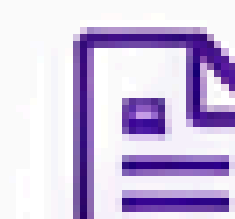
DELIVERABLES



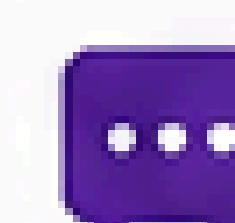
Industry-Ready Resume (DevOps Focused)



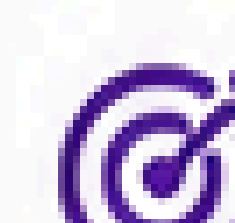
GitHub Portfolio with Projects



Interview Cheat Sheet (PDF)



Mock Interview Feedback



Placement Guidance & Resources

JOIN OUR COMMUNITY:



**Telegram
Channel**

<https://t.me/REGexSoftware>



**YouTube
Channel**

[https://www.youtube.com/
@REGexSoftware](https://www.youtube.com/@REGexSoftware)



**Instagram
Profile**

[https://www.instagram.com/
regexsoftware/?hl=en](https://www.instagram.com/regexsoftware/?hl=en)

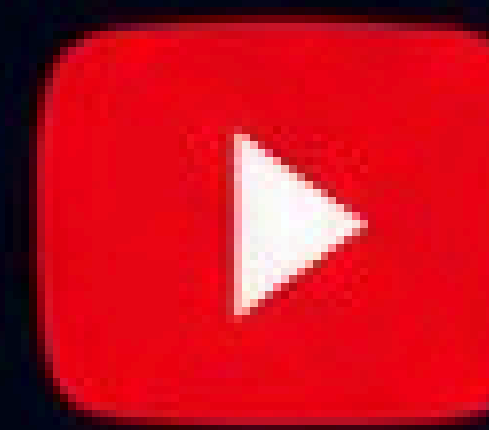


**LinkedIn
Company Page**

[https://www.linkedin.com/
company/regexsoftware/
posts/?feedView=all](https://www.linkedin.com/company/regexsoftware/posts/?feedView=all)



JOIN TELEGRAM



JOIN YOUTUBE



JOIN INSTAGRAM



JOIN LINKEDIN