



Best Strategy

- 👉 For the next **8-10 months**, master **Java Backend + System Design**.
- 👉 After **10 months**, start **DevOps** learning and certification while working.
- 👉 By the time you complete **2 years**, you'll have **Java + DevOps skills**,



Phase

📅 Phase 1 (0-10 months): Master Java Backend & System Design

- Deep dive into **Spring Boot, Hibernate, Microservices**.
- Improve **Data Structures & Algorithms (DSA)**.
- Learn **System Design (Scalability, Caching, Load Balancing)**.
- Work on **real-world projects & problem-solving**.

📅 Phase 2 (10-15 months): Start Learning DevOps

- Learn **Linux & Shell Scripting**.
- Learn **Git, GitHub Actions, Jenkins (CI/CD)**.
- Master **Docker & Kubernetes**.
- Learn **Cloud (AWS, Azure, or GCP)**.
- Work on **infrastructure automation using Terraform**.

📅 Phase 3 (15-24 months): Advanced DevOps & Certification

- Get **AWS DevOps Certification**.
- Learn **Logging & Monitoring (Prometheus, Grafana, ELK Stack)**.
- Work on **High Availability & Security in Cloud**.
- Apply for **DevOps roles in top companies**.

I. Prerequisites for System Design

Before diving into system design, you need to have a strong foundation in the following:

1. Basics of Distributed Systems

- **Concepts to Learn:**
 - Client-server architecture
 - Distributed systems principles (CAP theorem, eventual consistency)
 - Load balancing and replication
 - High availability and fault tolerance
- **Resources:**
 - *"Designing Data-Intensive Applications"* by Martin Kleppmann
 - GFG or Medium blogs on distributed systems

2. Networking Basics

- **Concepts to Learn:**
 - HTTP/HTTPS, REST, and WebSockets
 - DNS and CDNs
 - TCP/IP and UDP
 - Firewalls and NATs
- **Resources:**
 - *"Computer Networking: A Top-Down Approach"* by Kurose and Ross
 - YouTube: Networking basics by Tech Primers

3. Databases

- **Concepts to Learn:**
 - SQL vs NoSQL databases
 - ACID and BASE properties
 - Indexing, partitioning, and sharding
 - Database replication and consistency models
- **Tools to Explore:**
 - MySQL, PostgreSQL, MongoDB, Cassandra, Redis
- **Resources:**
 - *"Seven Databases in Seven Weeks"* by Eric Redmond

- TutorialsPoint or GFG for database fundamentals

Learning Timeline for System Design

- ◆ Beginner (1–2 months) → Learn Fundamentals
 - ◆ Intermediate (3–4 months) → Learn Scalability & Components
 - ◆ Advanced (5–6 months) → Work on Real-world Design Case Studies

You can learn System Design in 6 months with consistent learning and practice.

System Design Syllabus (Beginner to Advanced)

Basics of System Design (1st Month)

What is System Design?

-  Understanding Requirements (Functional & Non-Functional)
-  Types of Architectures

- Monolithic vs. Microservices
- 3-Tier Architecture (Client, Server, Database)
-  Networking Basics
- Load Balancers
- Reverse Proxy
- CDN (Content Delivery Network)

Best Resources:

- [System Design Primer \(GitHub\)](#)
 - YouTube: *Gaurav Sen - System Design Playlist*
-

Scalability & Performance Optimization (2nd Month)

Horizontal & Vertical Scaling

-  Caching Strategies

- Redis, Memcached
- LRU Cache
-  Database Scaling Techniques
- Sharding, Replication, Partitioning
- SQL vs NoSQL

Best Resources:

- Book: *Designing Data-Intensive Applications* (Martin Kleppmann)

- YouTube: *System Design by Tech Dummies*
-

3 High Availability & Distributed Systems (3rd Month)

- ✓ Message Queues (Kafka, RabbitMQ, SQS)
 - ✓ Event-Driven Architecture
 - ✓ CAP Theorem & Consistency Models
 - ✓ Microservices Communication (REST, gRPC, WebSockets)
 - ✓ Rate Limiting, Circuit Breakers

Best Resources:

- Book: *Building Microservices* by Sam Newman
 - YouTube: *CodeKarle - System Design Case Studies*
-

4 Security & Reliability (4th Month)

- ✓ Authentication & Authorization (OAuth, JWT, OpenID)
 - ✓ Logging & Monitoring (Prometheus, ELK Stack)
 - ✓ Fault Tolerance & Disaster Recovery
 - ✓ Data Consistency & Transactions

Best Resources:

- Udemy: *System Design Interviews*
 - YouTube: *ByteByteGo System Design*
-

5 Solving Real-world System Design Problems (5th-6th Month)

- ✓ Design a URL Shortener (like Bitly)
 - ✓ Design a Social Media Platform (like Twitter, Instagram)
 - ✓ Design an E-Commerce System (like Amazon, Flipkart)
 - ✓ Design a Video Streaming Service (like YouTube, Netflix)

How to Practice?

- 👉 Write your own design document for each system.
- 👉 Compare your design with expert solutions.

Best Resources:

- Book: *Grokking the System Design Interview* ([Educative.io](#))
- YouTube: *System Design by Alex Xu* (*ByteByteGo*)

4. Algorithms and Data Structures

- **Focus Areas:**
 - Hashing, trees, and graphs (for caches, indexes, and routing)
 - Sorting and searching (for query optimization)
 - Dynamic programming (for backend processing)
- **Resources:**
 - LeetCode and GFG for problem-solving

5. Operating Systems

- **Concepts to Learn:**
 - Threads, processes, and concurrency
 - Scheduling, file systems, and memory management
 - Virtualization and containers (Docker)
 - **Resources:**
 - *"Operating Systems: Three Easy Pieces"* by Remzi H. Arpaci-Dusseau
-

II. Roadmap for System Design Mastery

Step 1: Learn the Basics

Start with small components and understand how they work:

1. **APIs and Microservices:**
 - Learn to design APIs (REST/GraphQL).
 - Understand service discovery, load balancing, and API gateways.
 - Tools: Swagger, Postman.
2. **Caching:**
 - Concepts: Write-through, write-around, and write-back caching.
 - Tools: Redis, Memcached.
3. **Load Balancers:**
 - Learn algorithms like Round Robin, Least Connections.
 - Tools: HAProxy, NGINX.

Step 2: Understand Key Components

Learn how various building blocks of a system interact:

1. Message Queues:

- Tools: RabbitMQ, Apache Kafka.
- Use cases: Task queues, log processing.

2. Content Delivery Networks (CDNs):

- Concepts: Edge caching, latency reduction.
- Providers: Cloudflare, Akamai.

3. Database Scaling:

- Vertical vs horizontal scaling.
- Learn database sharding and replication.

Step 3: Explore Architectures

1. Monolithic vs Microservices:

- Pros and cons of each.
- Transition strategies.

2. Event-Driven Architecture:

- Tools: Kafka, AWS SNS/SQS.

3. Serverless Architecture:

- Providers: AWS Lambda, Google Cloud Functions.

Step 4: Practice Designing Systems

Solve design problems and learn to create high-level designs:

1. High-Level Design (HLD):

- Focus on how components interact.
- Tools: Lucidchart, Draw.io.

2. Low-Level Design (LLD):

- Dive deeper into classes, APIs, and data models.
- Use UML diagrams for clarity.

Step 5: Learn Scalability Patterns

1. Horizontal Scaling:

- Add more servers to handle increased load.

2. Partitioning and Sharding:

- Split databases and services to reduce load.

3. Rate Limiting:

- Limit API usage per user or app.

- Tools: NGINX, Kong API Gateway.

Step 6: Explore Advanced Topics

1. Search Systems:

- Elasticsearch, Solr.
- Concepts: Inverted indexes, stemming.

2. Big Data and Analytics:

- Tools: Hadoop, Apache Spark, Google BigQuery.

3. Monitoring and Logging:

- Tools: Grafana, Prometheus, ELK stack.
-

III. Practice System Design Problems

Common System Design Problems:

1. Design a URL Shortener (e.g., Bitly)
2. Design an Online Chat System (e.g., WhatsApp, Messenger)
3. Design a Video Streaming Service (e.g., YouTube, Netflix)
4. Design a Search Engine (e.g., Google)
5. Design a Ride-Sharing App (e.g., Uber, Lyft)
6. Design a Social Media Platform (e.g., Twitter, Instagram)
7. Design a File Storage System (e.g., Dropbox, Google Drive)

Platforms for Practice:

- **Grokking the System Design Interview:** Covers a wide range of design problems.
 - **Educative.io:** Offers system design and low-level design courses.
 - **LeetCode Discussions:** Provides system design questions discussed by the community.
-

IV. Tools to Use

1. **Visualization:** Lucidchart, Draw.io, Figma
2. **APIs:** Postman, Swagger
3. **Databases:** MySQL, MongoDB, Redis

4. Load Testing: Apache JMeter, Locust

V. Weekly Study Plan

Week	Focus Area	Activities
1	Distributed Systems Networking Basics	Learn CAP theorem, HTTP/HTTPS, DNS. Solve small design problems like load balancers.
2	Databases Caching	Understand indexing, sharding, and caching. Practice designing database-heavy systems.
3	Core Components (APIs, Queues)	Design APIs and integrate queues like Kafka. Practice problems like rate limiting and task queues.
4	Scalability and Fault Tolerance	Study replication, partitioning, and high availability. Implement designs with redundancy.
5	Architecture Practice	Design systems like URL shorteners or chat apps. Focus on HLD and LLD.
6	Advanced Topics	Explore search systems, analytics, and monitoring tools. Build a basic prototype for a real problem.

VI. Tips for Success

1. Understand Trade-Offs:

- Prioritize between scalability, consistency, and performance based on requirements.

2. Start Small, Then Scale:

- Focus on basic functionality first, then add optimizations.

3. Be Consistent:

- Practice system design problems regularly and document learnings.

4. Collaborate:

- Discuss system design problems with peers or in forums like LeetCode discussions.

5. Read Case Studies:

- Learn how large-scale systems (like Google and Facebook) solve common challenges.

Master Plan: Cloud + DevOps Engineer in 2 Years

 Goal: After 2 years, you're certified, hands-on, and job-ready for roles like **Cloud Engineer**, **DevOps Engineer**, **Site Reliability Engineer**, or **Cloud Consultant**.

Index / Curriculum Plan (with Timeline)

#	Phase	Topics / Tools	Outcome / Skills	Duration
1	Foundation	Linux, Git, Networking, Bash	Strong base for DevOps tools	2 Months
2	Version Control	Git + GitHub	CI/CD basics, branching, teamwork	1 Month
3	Cloud Basics	AWS or Azure Fundamentals	Understand core cloud services	1.5 Months
4	Containers	Docker, Docker Compose	Package & run apps as containers	1 Month
5	Kubernetes	K8s basics, minikube, EKS/GKE	Manage, scale, and deploy containers	1.5 Months
6	CI/CD Tools	Jenkins, GitHub Actions, GitLab CI	Build deployment pipelines	1 Month
7	IaC (Infrastructure as Code)	Terraform, CloudFormation	Automate infra using code	1.5 Months
8	Monitoring & Logging	Prometheus, Grafana, ELK Stack	Site reliability and alerts	1 Month
9	Cloud Architecting	AWS Architect / Azure Architect skills	Scale, secure, and design cloud apps	2 Months
10	Security & DevSecOps	IAM, KMS, Security Groups, SAST/DAST	Secure pipelines & infra	1 Month
11	Project + Certification Prep	Real projects + Mock tests	Build portfolio, get certified	3 Months

Certifications Roadmap (Progressively)

Level	Certification	Provider	When to Take
● Beginner	AWS Cloud Practitioner / Azure AZ-900	AWS / Microsoft	Month 4

Level	Certification	Provider	When to Take
Intermediate	Docker & Kubernetes	Udemy + CKA/CKAD	Month 7–10
Intermediate	Terraform Associate	HashiCorp	Month 10
Advanced	AWS DevOps Engineer / Azure DevOps Expert	AWS / Microsoft	Month 16–20
Bonus	Linux Foundation DevOps Bootcamp	LF	Optional

Hands-On Project Ideas (Do 1-2 per Phase)

1. Deploy Node/React app using Docker
 2. Create a CI/CD pipeline in Jenkins for Java App
 3. Deploy containerized app on Kubernetes
 4. Automate infrastructure using Terraform (AWS EC2, RDS)
 5. Use GitHub Actions to build & deploy
 6. Monitor app with Prometheus & Grafana
 7. Build a real-world project: E-commerce / Booking System end-to-end in AWS with full CI/CD
-

Tools & Technologies to Learn

- **Cloud:** AWS (EC2, S3, IAM, VPC, Lambda, RDS, EKS), Azure (VMs, AKS)
- **CI/CD:** Jenkins, GitHub Actions, GitLab CI
- **Containers:** Docker, Docker Compose
- **Orchestration:** Kubernetes, Helm
- **IaC:** Terraform, Ansible (optional)
- **Monitoring:** Prometheus, Grafana, ELK Stack
- **Scripting:** Bash, Python (basic automation)
- **Version Control:** Git, GitHub

Weekly Plan Example (First 2 Months)

Week	Focus
1	Linux Basics (Shell, Permissions, Files)
2	Bash Scripting (Loops, Variables, Scripts)
3	Git & GitHub (clone, commit, merge, PR)
4	Networking Basics (Ports, IP, DNS)
5	AWS Cloud Practitioner – EC2, S3, IAM
6	IAM, VPC, CloudShell practice
7	Build Docker containers
8	Push Docker images to Docker Hub

Udemy Courses to Follow (Start With These)

1. [Linux Command Line Basics – Udemy]
 2. [Git & GitHub Masterclass]
 3. [AWS Cloud Practitioner Essentials (Beginner)]
 4. [Docker Mastery: Kubernetes + Swarm From a Docker Captain]
 5. [Kubernetes for Beginners: Learn K8s from Scratch]
 6. [CI/CD with Jenkins: Beginner to Pro]
 7. [Terraform for Beginners]
 8. [Prometheus and Grafana Monitoring]
-

Final Tips

- Start **hands-on early**: Use AWS Free Tier & Minikube
- Keep **1 project running in parallel** to apply what you learn
- Create a **GitHub portfolio** to show your DevOps + Cloud work
- After 1 year: Try **freelancing / open source** or internal DevOps migration
- **Network on LinkedIn** and **update your resume every 6 months**

/*******Certificate*******/

1)RED HAT CERTIFICATION

Red Hat Certified System Administrator

2)RED HAT CERTIFICATION

Red Hat Certified Engineer

3)RED HAT CERTIFICATION

Red Hat Certified System Administrator

4)Associate

Oracle Cloud Infrastructure

Certified Architect

Associate

5)Professional

Administration

Oracle Database

Administration 2019

Certified Professional

6)Microsoft CERTIFIED EXPERT

Microsoft Certified: Azure Fundamentals

AZURE FUNDAMENTAL

7)CERTIFICATIONS

Microsoft Certified: Azure Solutions Architect Expert

DevOps Roadmap 2025

The Only Roadmap You Need to become a DevOps Engineer from Zero to hero.
 By - TrainWithShubham (Shubham Londhe)
 Live Classes: <https://bit.ly/devops-live>
 Roadmap: https://miro.com/app/board/uXjVKq3-e8l=/?share_link_id=666035483059

Subject	Topic	Video/Playlist	Status	Time	Start Date
Operating System	Linux For DevOps	https://youtube.com/playlist?list=PLIfy9GnSVerSpMjLEJKoFPTnDK8xPYtmb&si=fKWI5d496xda1ejs	In progress	4 hrs	10/3/2024
Scripting	Shell Scripting and BASH	https://youtube.com/playlist?list=PLIfy9GnSVerQu-uL0Adl3O5VzUmlv5pPf&si=1Y7EY529NYxiHy	In progress	4 hrs	10/5/2024
Scripting	Python Basics and Data Structures	https://youtube.com/playlist?list=PLIfy9GnSVerS_L5zOCOAf7rsbgWmjXTOM	Not started	10 Hrs	12/12/2024
Scripting	Python For DevOps	https://www.youtube.com/live/9ErAlY2lfw0?si=spNxxPuPeKexf3-r	Done	4 hrs	12/12/2024
Scripting	Python Project for Your Resume	https://youtube.com/playlist?list=PLIfy9GnSVerSzFmQ8jqP9v0XHHOAeWbj0	Not started	2 hrs	12/13/2024
Computer Networking	Computer Networking for DevOps	https://www.youtube.com/live/mNTs-shuFno?si=vZkzNRFlbZvmkWGp	Not started	4 hrs	12/13/2024
Version Control	Git and GitHub in one Video	https://youtu.be/AT1uxOLsCdk	Not started	2 hrs	12/14/2024
Version Control	Git & GitHub For DevOps	https://www.youtube.com/live/DyqAdz96mon?si=4wpL27SQL_zXKt7E	Not started	4 hrs	12/14/2024
Docker	Docker Explained for Beginners with Project	https://youtu.be/Tevxhn6Odc8	Not started	2 hrs	12/15/2024
Docker	Docker Revision Part 1	https://www.youtube.com/live/Bn7V8llfs3g?si=SjhK8mund-QD3eYp	Not started	2 hrs	12/15/2024
Docker	Docker Revision Part 2	https://www.youtube.com/live/-U-uQer9pXo?si=EcjZYmFAw1VjaLEj	Not started	2 hrs	12/15/2024
Cloud Service Providers	AWS Setup	https://youtu.be/gz3dr6o5gxI	Not started	2 hrs	12/16/2024
Cloud Service Providers	AWS Important services	https://youtube.com/playlist?list=PLIfy9GnSVerRj2fl0_P9_YFxhDGSFChU7	Not started	20 hrs	12/17/2024
Cloud Service Providers	AWS Lambda project	https://youtube.com/playlist?list=PLIfy9GnSVerTB0twnC5eaGD-oiHprpnW-	Not started	2 hrs	12/18/2024
CI/CD	Jenkins in One Shot	https://youtu.be/XaSdKR2fOU4?si=gE79kH8cxSp0ocuo	Not started	7 hrs	12/18/2024
CI/CD	Jenkins Setup and Installation	https://youtu.be/OkVtBKqMt7I	Not started	10 hrs	12/19/2024
CI/CD	Jenkins CI/CD with Project	https://youtu.be/wwNWgG5htxs	Not started	12 hrs	12/20/2024
CI/CD	GitLab For DevOps	https://youtube.com/playlist?list=PLIfy9GnSVerTAj-KOrYMHBCm36fuzl1Pv&si=TOZNhfzHHsAIWaVb	Not started	4 hrs	12/20/2024
Infrastructure as Code	YAML For Beginners	https://youtu.be/LFUij5Gr0cE	Not started	2 hrs	12/21/2024

DevOps Roadmap 2025

The Only Roadmap You Need to become a DevOps Engineer from Zero to hero.
 By - TrainWithShubham (Shubham Londhe)
 Live Classes: <https://bit.ly/devops-live>
 Roadmap: https://miro.com/app/board/uXjVKq3-e8l=/?share_link_id=666035483059

Subject	Topic	Video/Playlist	Status	Time	Start Date
Infrastructure as Code	Ansible Tutorial	https://youtu.be/SGB7EdiP39E	Not started	2 hrs	12/22/2024
Kubernetes	Kubernetes Tutorial	https://youtu.be/FqfoDUhzyDo	Not started	6 hrs	12/23/2024
Kubernetes	Kubernetes Project	https://youtu.be/ONrbWFIXLLk	Not started	6 hrs	12/24/2024
ArgoCD	ArgoCD Project	https://youtu.be/Kbvch_swZWA?si=MuBWhxMyH3JpUnZl https://www.youtube.com/live/yyU2UiNI08M?si=9cTbroGPX6KX7YaZ	Not started	2 hrs	12/24/2024
Infrastructure as Code	Ansible With Project	https://bit.ly/tws-terraform	Not started	5 hrs	12/25/2024
Infrastructure as Code	Terraform for Beginners	https://youtu.be/QwGm5m4AxNA?si=EX8QCqw6TdTnQC89	Not started	20 hrs	12/26/2024
Monitoring	Grafana	https://youtu.be/DXZUunEeHqM?si=6bYLe djRqlDLb0hl	Not started	-	12/27/2024
Monitoring	Prometheus	https://youtube.com/playlist?list=PLIfy9Gn SVeSJ0Fe-arr96QNCGzoL6O6Z	Not started	2 hrs	12/27/2024
Resume and LinkedIn	LinkedIn Best practices	https://youtu.be/QPSAlmWidb0	Not started	2 hrs	12/28/2024
Resume and LinkedIn	Resume for DevOps	https://bit.ly/tws-junoon	Not started	2 hrs	12/29/2024
DevOps Course	DevOps - Zero To Hero (Hindi)	https://bit.ly/devops-live	Not started	100 hrs	12/30/2024
DevOps Course	DevOps - Zero To Hero (English)	https://bit.ly/aws-zero-to-hero	Not started	40 hrs	12/31/2024
AWS	AWS - Zero To Hero	https://dockerschools.collabnix.com/docker/dc.html	Not started	20 hrs	12/31/2024
Certification	Docker Certified Associate (DCA)	https://training.linuxfoundation.org/certification/certified-kubernetes-administrator-cka/	Not started	20 hrs	1/1/2025
Certification	Certified Kubernetes Administrator (CKA)	https://explore.skillbuilder.aws/learn	Not started	35 hrs	1/2/2025
Certification	AWS Free Training and Certifications		Not started	50+ hrs	1/3/2025

DevOps Roadmap 2025

The Only Roadmap You Need to become a DevOps Engineer from Zero to hero.
Part 1: What Should I Learn (Shoulder to shoulder)

By - TrainWithShubham (Shubham Londhe)

Live Classes: <https://bit.ly/devops-live>

Roadmap: https://miro.com/app/board/uXjVKq3-e8l/?share_link_id=666035483059

Subject	Topic	Video/Playlist	Status	Time	Start Date

DevOps Zero To Hero (Live)

Crack DevOps and Cloud Jobs in 2025

<https://bit.ly/devops-batch-9>

Subject	Topic	Sub-Topics	Status	Hours	Start Date
Introduction	Introduction to DevOps and Cloud	Introduction to DevOps Culture Introduction to Course Curriculum Meet and Greet with Batchmates and Alumni	Not started	4	1/25/2025
Linux	Linux For DevOps	Understanding Linux And Linux Architecture Basic Linux Commands File Permissions and Users Management Advanced Linux Commands Shell Scripting	Not started	4	1/26/2025
Scripting	Linux For DevOps			3	2/1/2025
Version Control	Git and GitHub	Git Fundamentals Git Clone, Add, Commit, Push Pull Git Branching, Merge, Rebase, conflicts Git and GitHub Scenarios	Not started	3	2/8/2025
Python	Python For DevOps	Python Programming Basics Python Loops, Conditions, Functions Python Modules and Boto Python Project for Your Resume	Not started	10	10/26/2024
Docker	Docker with Project	Docker Architecture Docker Commands, Images, Containers Docker Advanced With Project Docker Compose, DockerHub, Docker Init, Docker Scout	Not started	4	2/15/2025
AWS	AWS - Zero To Hero	Introduction To AWS and Cloud Computing Important AWS Services EC2, S3, RDS, IAM Advanced AWS Services Lambda, APIGateway, ECS, ECR	Not started	12	11/3/2024

Serverless Framework						
Category	Tool / Service	Description	Status	Progress (%)	Last Update	Created
CI/CD	Jenkins	Jenkins Architecture, Installation, Concept of CI/CD Jenkins Declarative CI/CD Pipeline, Jenkins Agents SonarQube, Trivy, OWASP, DevSecOps, Docker Scout GitLab CI/CD with Project	Not started	4	2/22/2025	
	DevSecOps GitLab CI/CD			3	2/23/2025	4
Kubernetes	Kubernetes	Kubernetes Architecture, Installation of Minikube and Kubeadm, AWS EKS Kubernetes Pods, Deployment, Service, Secrets, ConfigMaps Kubernetes Persistent Volume, Ingress HELM Charts ArgoCD	Not started	3	3/2/2025	
Infrastructure as Code	Terraform	Terraform Installation, HCL Syntax, Scripting, Providers Terraform with Docker, AWS Terraform Remote Backends and State Management	Not started	4	3/22/2025	
	Ansible	Ansible Architecture, Setup Ansible Inventory, ad-hoc commands Ansible Playbooks, Roles, Templates	Not started	4	3/23/2025	3/29/2025
Monitoring	Grafana	Observability, Alerting, Logging, Monitoring, Visualisation Grafana Installation and Integration with Loki, Promtail	Not started	4	3/30/2025	
	Prometheus	Prometheus Setup and Integration with Grafana Prometheus with Docker / Kubernetes on Advisor		3	4/5/2025	
Job Assistance	LinkedIn and Resume Building	LinkedIn and Resume Optimisation DevOps Resume Templates and Cover Letter	Not started	3	4/6/2025	

Mock Interviews	Effective Communication for DevOps Interviews Mock Interviews everyday	3 2	4/12/2025 4/13/2025
-----------------	---	--------	------------------------

TOTAL 103

Projects	Minor Projects
	Backup and Restore on Linux Server by Shell Script DevSecOps CI/CD Pipeline using Jenkins, AWS, Docker, Kubernetes, SonarQube CI/CD with AWS Code Pipeline Web App Deployments through Ansible and Terraform Serverless Deployment for 3 tier application (Production Level) Continuous Delivery via ArgoCD for Kubernetes
Major Project	End-To-End DevOps Tools for a 3 Tier Application

Topics

Learn these To become a DevOps Engineer

Operating System

Scripting

Version Control

Docker

CI/CD

Cloud Service Providers

Kubernetes

Monitoring

Infrastructure as Code

Resume and LinkedIn

DevOps Course

Certification

Introduction

Linux

Python

AWS

Job Assistance

Projects

ArgoCD