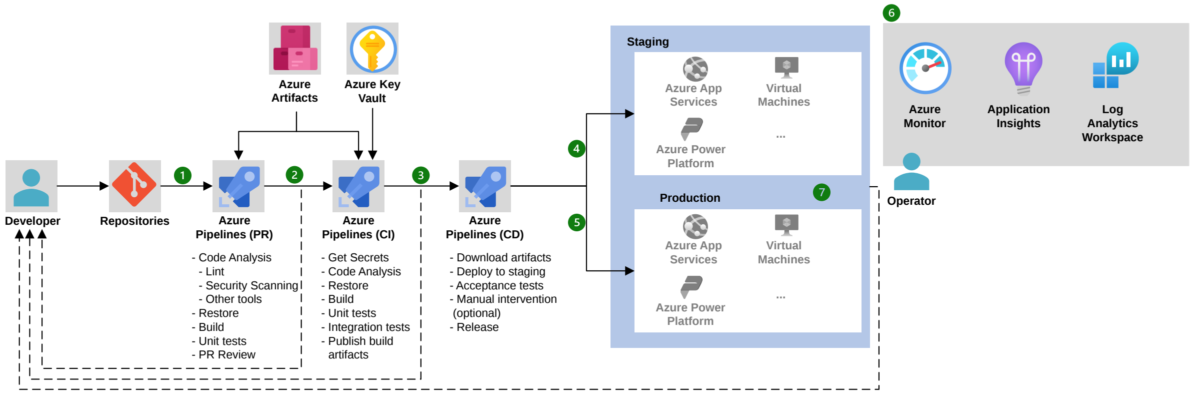
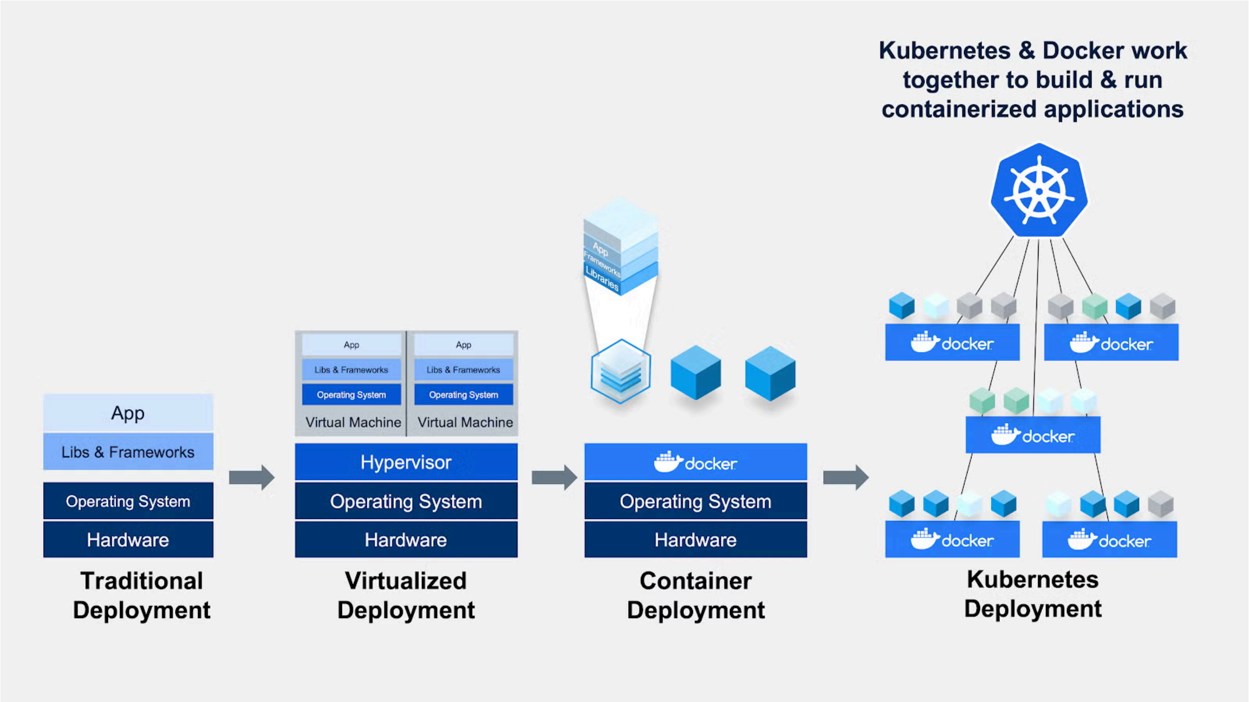
REALITY CHECK (IMPORTANT)

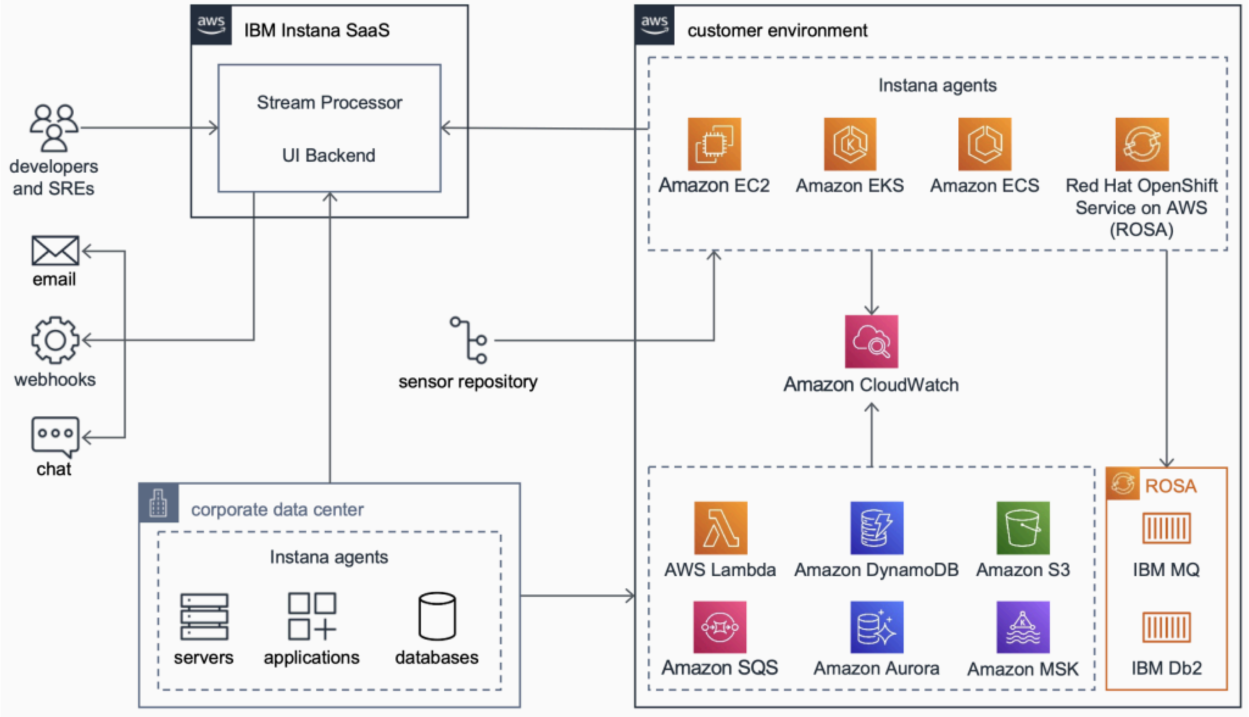
- **DevOps fresher roles exist**, but hiring depends on:
    - how strong your **projects** are
    - how well you **explain them**
- You are **not aiming for "Senior DevOps"**
- You are aiming for:
    - **Junior DevOps**
    - **Cloud / DevOps Support**
    - **Ops Engineer / Platform Support**

That's realistic in 28 days.

28-DAY "DO OR DIE" PLAN (8:00–11:00 DAILY)







🔥 WEEK 1 (Linux + Shell = Foundation)

Goal: Sound like a Linux ops engineer

Day 1–2

- Linux filesystem
- permissions
- systemctl
- logs

Day 3–4

- Users & groups
- networking basics
- dnf/yum

Day 5–7

- Shell scripting
- loops, conditions
- automation scripts

📌 Output:

- 3 shell scripts
- README explaining usage

🔥 WEEK 2 (Docker + AWS = Deployment Skill)

Goal: Deploy something real

Day 8–9

- Dockerfile
- images
- containers

Day 10–11

- AWS EC2
- SSH

- security groups

Day 12–14

- Dockerized Node app on EC2

📌 Output:

- App running on EC2
- GitHub repo + screenshots

🔥 WEEK 3 (CI/CD + Kubernetes)

Goal: Look production-ready

Day 15–16

- Git + GitHub workflows
- Jenkins pipeline

Day 17–18

- Jenkins → Docker → Deploy

Day 19–21

- Kubernetes basics
- Deploy app on minikube

📌 Output:

- Jenkinsfile
- K8s manifests

🔥 WEEK 4 (Interview + Applying)

Goal: Get calls

Day 22–24

- Resume (DevOps-focused)
- Project explanations

Day 25–26

- Interview Q&A practice

Day 27–28

- Apply **50–70 jobs**
-

⏱️ DAILY ROUTINE (NON-NEGOTIABLE)

8:00 – 9:00 → Learn (concepts + notes)
9:00 – 10:00 → Hands-on (commands / labs)
10:00 – 11:00 → Practice + revise + small task
Every day ends with:
• Commands written in a notes file
• One Git commit (even small)

📅 WEEK 1–2: LINUX + SHELL (FOUNDATION)

Week 1 (Linux Core)

Day 1: Linux intro, RHEL, filesystem, basic commands
`ls, cd, pwd, cp, mv, rm, cat, less`
Day 2: Files, permissions
`chmod, chown, umask`
Day 3: Users & groups
`useradd, userdel, groupadd, passwd`
Day 4: Package management
`yum, dnf, repo concept`
Day 5: systemd & services
`systemctl start/stop/status/enable`
Day 6: Networking basics
`ip, ss, ping, curl, nmcli`
Day 7: Revision + mock interview Qs (Linux)

Week 2 (Shell Scripting)

Day 8: Shell basics, variables, input/output
Day 9: Conditions & loops
`if, for, while, case`
Day 10: Exit codes & error handling
Day 11: Functions & arguments
Day 12: File automation scripts
Day 13: Monitoring script (CPU/memory/disk)

Day 14: **PROJECT** – User & group automation with logs

📌 **Deliverable**: 3 shell scripts + README on GitHub

📅 **WEEK 3: AWS CORE (HIGH PRIORITY FOR JOBS)**

Week 3

Day 15: AWS global infra + shared responsibility

Day 16: EC2 basics + launch instance

Day 17: EC2 hardening, security groups, SSH

Day 18: EBS, snapshots, AMIs

Day 19: IAM basics (users, roles, policies)

Day 20: S3 (buckets, lifecycle, permissions)

Day 21: **PROJECT** – Secure EC2 + S3 setup

📌 **Deliverable**: EC2 + S3 architecture diagram

📅 **WEEK 4: GIT + DOCKER (MANDATORY)**

Week 4

Day 22: Git basics, stages, commits

Day 23: Branching, merging, conflicts

Day 24: GitHub PR workflow

Day 25: Docker intro + architecture

Day 26: Dockerfile + images

Day 27: Docker networking & volumes

Day 28: **PROJECT** – Dockerize FastAPI app

📌 **Deliverable**: Dockerized app repo

📅 **WEEK 5: CI/CD + QUALITY**

Week 5

Day 29: Jenkins intro + install

Day 30: Jenkinsfile (pipeline syntax)

Day 31: GitHub webhook integration

Day 32: Artifacts & versioning

Day 33: SonarQube + quality gates

Day 34: Jenkins + Sonar integration

Day 35: **PROJECT** – CI pipeline (build → test → scan)

📌 **Deliverable**: Jenkins pipeline screenshot + repo

📅 **WEEK 6: KUBERNETES (JOB DECIDER)**

Week 6

Day 36: K8s architecture, kubectl

Day 37: Pods, Deployments, Services

Day 38: ConfigMaps, Secrets
Day 39: Rollouts, probes, HPA
Day 40: Volumes (PV/PVC)
Day 41: Helm basics
Day 42: **PROJECT** – Deploy app on K8s
📌 **Deliverable**: K8s manifests repo

📅 WEEK 7: TERRAFORM + ANSIBLE

Week 7

Day 43: Terraform basics & state
Day 44: VPC + subnets
Day 45: EC2 via Terraform
Day 46: Ansible intro & inventory
Day 47: Playbooks, roles
Day 48: Ansible Vault
Day 49: **PROJECT** – Infra via Terraform + config via Ansible

📅 WEEK 8: INTERVIEWS + APPLYING (CRITICAL)

Week 8

Day 50–52: Resume (DevOps focused)
Day 53–55: Interview Q&A (Linux, AWS, Docker, K8s)
Day 56–58: Mock interviews + revise projects
Day 59: Apply to **50+ jobs**
Day 60: HR + technical prep

📚 STUDY MATERIALS (ONLY THESE)

Linux & Shell

- Red Hat docs
- TLDR pages
- Practice: `explainshell.com`

AWS

- AWS Free Tier
- AWS EC2 + IAM docs

Docker & K8s

- Docker official docs
- Kubernetes.io docs

CI/CD

- Jenkins official docs
- SonarQube docs

Terraform & Ansible

- HashiCorp Learn
- Ansible docs

🎯 **RULES IF YOU WANT THE JOB**

- No skipping days
- No watching random YouTube
- Build **projects > certificates**
- Push code **daily**
- Apply aggressively (LinkedIn, Naukri, Indeed)

If you want, next I can:

- Build a **DevOps resume**
- Give **interview questions with answers**
- Create **exact project repo structures**

Straight answer first: **there is no magic shortcut**.

But there *is* a **fast lane**. If you execute this properly, you can realistically land a **DevOps fresher / junior / support DevOps role** in ~2 months.

You're already on **Fedora** and you know **basic Linux + MERN** → good starting advantage.

THE ONLY DEVOPS ROADMAP THAT WORKS FAST

🔑 **CORE RULE (READ THIS TWICE)**

Companies don't hire "DevOps learners".

They hire people who can:

- manage Linux
- deploy apps
- automate infra
- explain pipelines confidently

So we optimize for **INTERVIEW + PRACTICAL SKILLS**, not theory.

WHAT YOU SHOULD COMPLETELY IGNORE ❌

- Deep MERN interviews
- Advanced DSA
- Over-learning cloud services
- Certificates without projects

WHAT YOU MUST MASTER (NON-NEGOTIABLE) ✅

1️⃣ **Linux (You already started — good)**

Focus ONLY on:
- File permissions
- systemd
- networking basics
- logs
- package management

👉 Fedora is perfect. No need to switch to RHEL now.

Daily commands you must be fluent in:

```
ls, find, grep, awk, sed
chmod, chown
systemctl
journalctl
ip, ss, curl
dnf
```

If you can explain these confidently → Linux round cleared.

2️⃣ Shell Scripting (This is your FIRST shortcut)

Most freshers skip this. Don't.

You must build:
- user & group automation script
- disk / CPU monitoring script
- log cleanup script

💡 Interviewers LOVE shell scripts because:
- low competition
- real ops skill
- easy to verify

If asked:**"What automation have you done?"**

You already win.

3️⃣ AWS (Only what gets asked in interviews)

You do **NOT** need full AWS.

Learn ONLY:
- EC2
- IAM
- Security Groups
- S3
- AMI & EBS basics

If you can:

- launch EC2
- secure it
- deploy an app
- explain IAM roles

👉 You're hire-ready for junior roles.

4️⃣ Docker (Your BIGGEST ADVANTAGE)

You know MERN → use it.

Shortcut strategy:

- Take a simple Node.js app
- Dockerize it
- Run it on EC2

That's it.

Most candidates:**"I know Docker theory"**

You:**"I containerized and deployed a Node app on EC2"**

Game over.

5️⃣ CI/CD (This gets you shortlisted)

Learn:

- Git
- Jenkins pipeline
- GitHub webhook

One pipeline is enough:

`GitHub → Jenkins → Docker build → Deploy`

Even **one working pipeline** puts you ahead of 80% applicants.

6️⃣ Kubernetes (Learn just enough)

You don't need advanced K8s.

Focus on:

- Pods
- Deployments
- Services
- ConfigMaps
- Secrets

Run locally using **minikube**.

Interview shortcut:**"I deployed my containerized app on Kubernetes using**

Deployment and Service objects."

That sentence alone matters.

THE REAL SHORTCUT (THIS IS IMPORTANT)

🔥 **BUILD 3 PROJECTS ONLY**

No more. No less.

Project 1: Linux Automation

- user/group creation
- logs
- monitoring
- shell scripts

Project 2: Docker + CI/CD

- Node app
- Dockerfile
- Jenkins pipeline
- GitHub repo

Project 3: Kubernetes Deployment

- same app
- deployed on minikube or EKS (optional)

These 3 projects = **resume + interview answers**.

DAILY TIMETABLE (8:00 – 11:00)

8:00 – 9:00 → Learn one concept

9:00 – 10:00 → Hands-on (terminal only)

10:00 – 11:00 → Build / improve project

NO YouTube binge.

NO passive learning.

HOW PEOPLE ACTUALLY GET DEVOPS JOBS FAST

Truth:

- Many are hired as **DevOps Support / Cloud Ops / SRE Intern**
- Then promoted in 6–12 months

So apply for:

- DevOps Engineer (Junior)
- Cloud Support Engineer
- Site Reliability Engineer (Fresher)
- Platform Engineer (Junior)

Don't filter yourself out.

FINAL REALITY CHECK (HONEST)

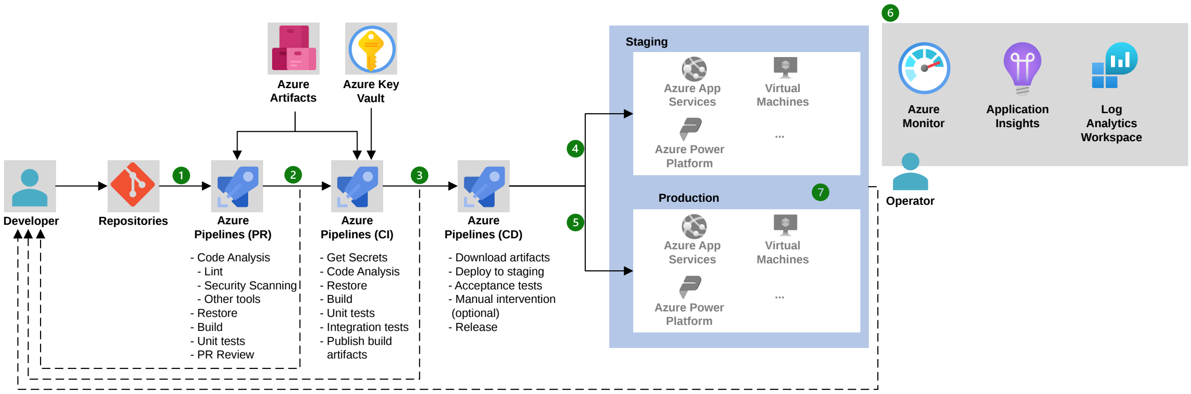
2 months is aggressive but **possible IF**:
• You practice daily
• You push code daily
• You apply aggressively (30–50 applications/week)
No excuses. No overthinking.
NEXT STEPS (Tell me ONE)
I can:
1️⃣ Make a **2-month day-by-day plan**
2️⃣ Build a **DevOps resume using your MERN background**
3️⃣ Give **real interview Q&A with answers**
4️⃣ Design **exact GitHub project structure**
Pick one.Is this conversation helpful so far?ChatGPT can make mistakes. Check important info. See Cookie Preferences.

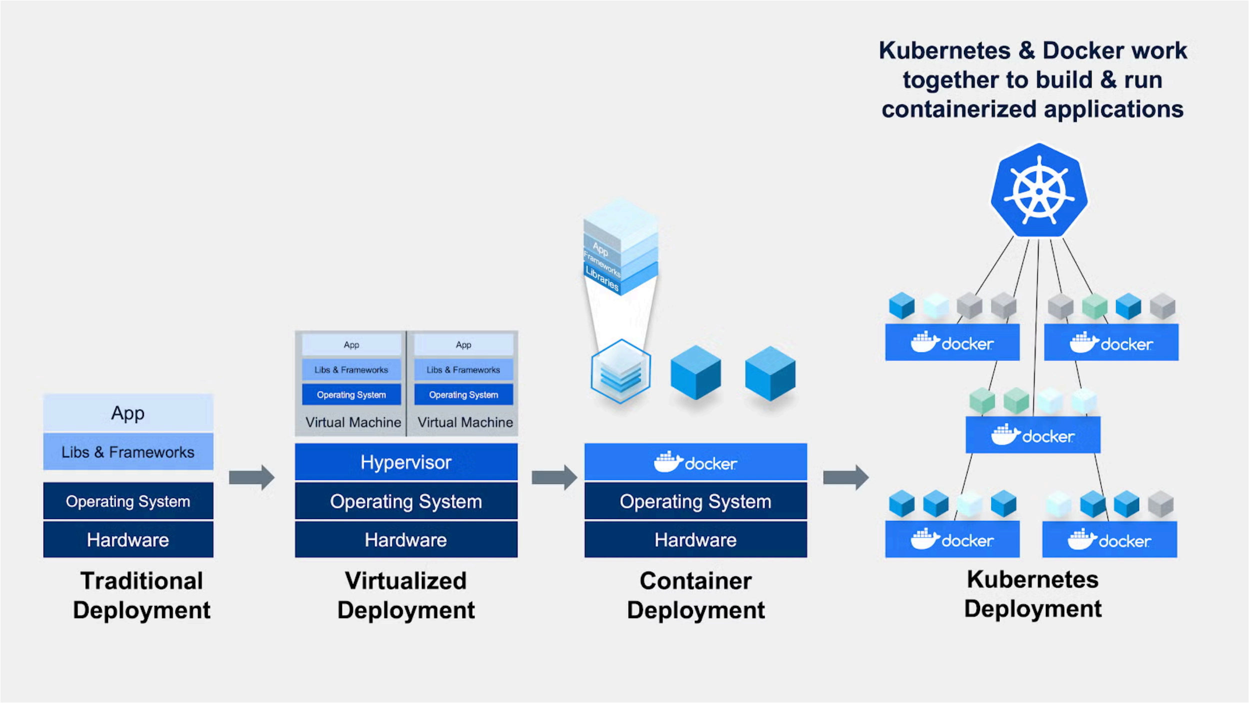
REALITY CHECK (IMPORTANT)

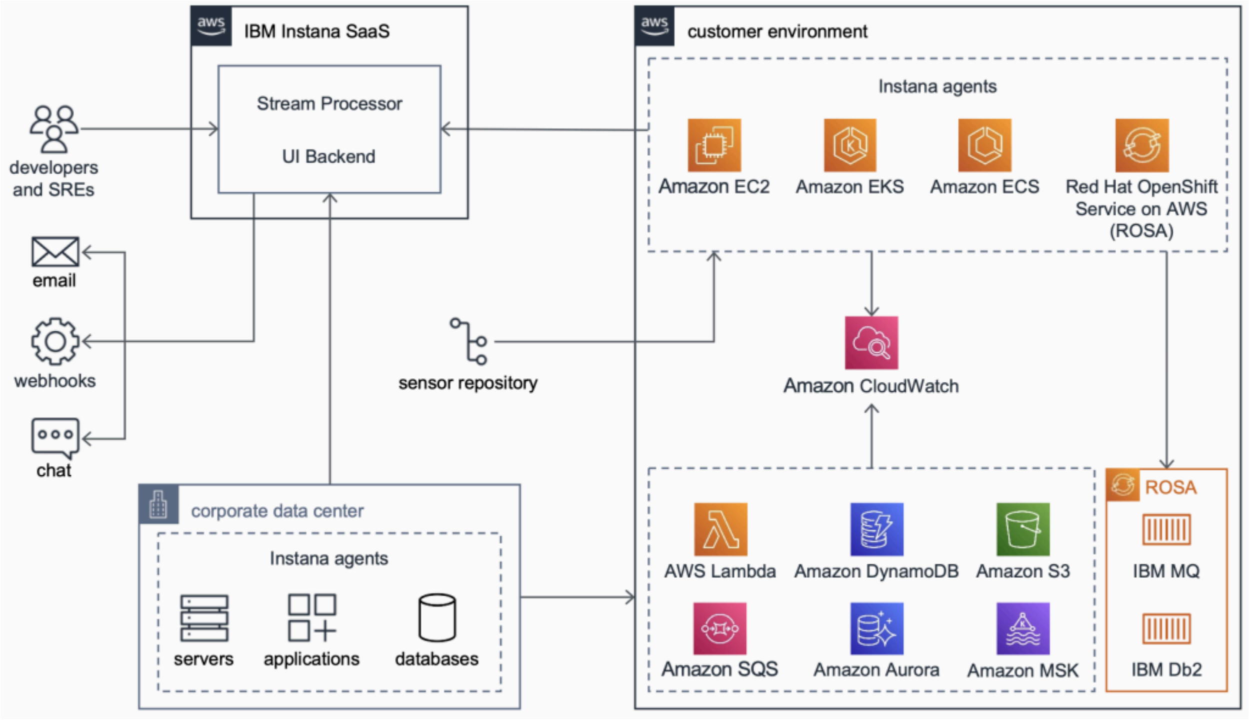
- **DevOps fresher roles exist**, but hiring depends on:
    - how strong your **projects** are
    - how well you **explain them**
- You are **not aiming for "Senior DevOps"**
- You are aiming for:
    - **Junior DevOps**
    - **Cloud / DevOps Support**
    - **Ops Engineer / Platform Support**

That's realistic in 28 days.

28-DAY "DO OR DIE" PLAN (8:00–11:00 DAILY)


Azure Artifacts
Azure Key Vault
Developer
Repositories
1
Azure Pipelines (PR)
- Code Analysis
- Lint
- Security Scanning
- Other tools
- Restore
- Build
- Unit tests
- PR Review
2
Azure Pipelines (CI)
- Get Secrets
- Code Analysis
- Restore
- Build
- Unit tests
- Integration tests
- Publish build artifacts
3
Azure Pipelines (CD)
- Download artifacts
- Deploy to staging
- Acceptance tests
- Manual intervention (optional)
- Release
4
5
Staging
Azure App Services
Virtual Machines
Azure Power Platform
...
Production
7
Azure App Services
Virtual Machines
Azure Power Platform
...
6
Azure Monitor
Application Insights
Log Analytics Workspace
Operator


Kubernetes & Docker work together to build & run containerized applications
App
Libs & Frameworks
Operating System
Hardware
Traditional Deployment
App
Libs & Frameworks
Operating System
Virtual Machine
App
Libs & Frameworks
Operating System
Virtual Machine
Hypervisor
Operating System
Hardware
Virtualized Deployment
App
Frameworks
Libraries
docker
Operating System
Hardware
Container Deployment
docker
docker
docker
docker
docker
Kubernetes Deployment



🔥 WEEK 1 (Linux + Shell = Foundation)

Goal: Sound like a Linux ops engineer

Day 1–2

- Linux filesystem
- permissions
- systemctl
- logs

Day 3–4

- Users & groups
- networking basics
- dnf/yum

Day 5–7

- Shell scripting
- loops, conditions

- automation scripts

📌 Output:

- 3 shell scripts
- README explaining usage

🔥 WEEK 2 (Docker + AWS = Deployment Skill)

Goal: Deploy something real

Day 8–9

- Dockerfile
- images
- containers

Day 10–11

- AWS EC2
- SSH
- security groups

Day 12–14

- Dockerized Node app on EC2

📌 Output:

- App running on EC2
- GitHub repo + screenshots

🔥 WEEK 3 (CI/CD + Kubernetes)

Goal: Look production-ready

Day 15–16

- Git + GitHub workflows
- Jenkins pipeline

Day 17–18

- Jenkins → Docker → Deploy

Day 19–21

- Kubernetes basics
- Deploy app on minikube

📌 Output:

- Jenkinsfile
- K8s manifests

🔥 WEEK 4 (Interview + Applying)

Goal: Get calls

Day 22–24

- Resume (DevOps-focused)
- Project explanations

Day 25–26

- Interview Q&A practice

Day 27–28

- Apply **50–70 jobs**
-
- -
  Commands written in a notes file
- -
  One Git commit (even small)
- -
  Red Hat docs
- -

TLDR pages

- -

Practice: `explainshell.com`

- -

AWS Free Tier

- -

AWS EC2 + IAM docs

- -

Docker official docs

- -

Kubernetes.io docs

- -

Jenkins official docs

- -

SonarQube docs

- -

HashiCorp Learn

- -

Ansible docs

- -

No skipping days

- -

No watching random YouTube

- -

Build **projects > certificates**

- -

Push code **daily**

• •

Apply aggressively (LinkedIn, Naukri, Indeed)

• •

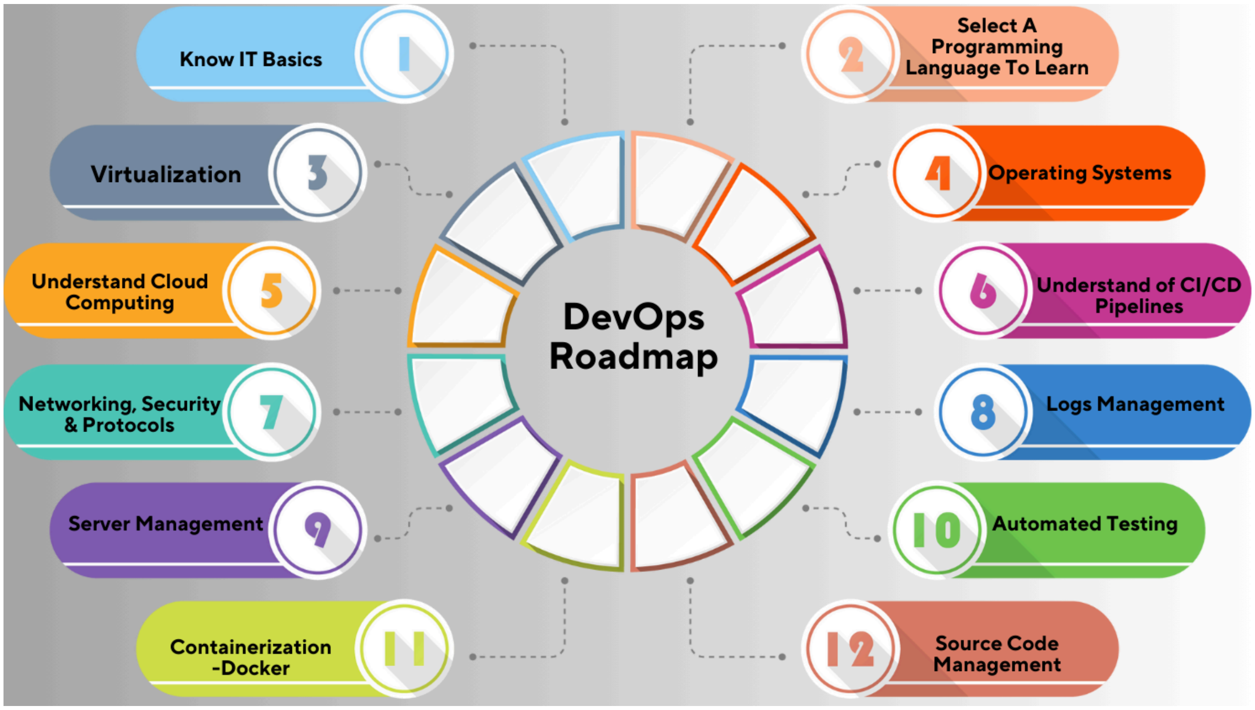
Build a **DevOps resume**

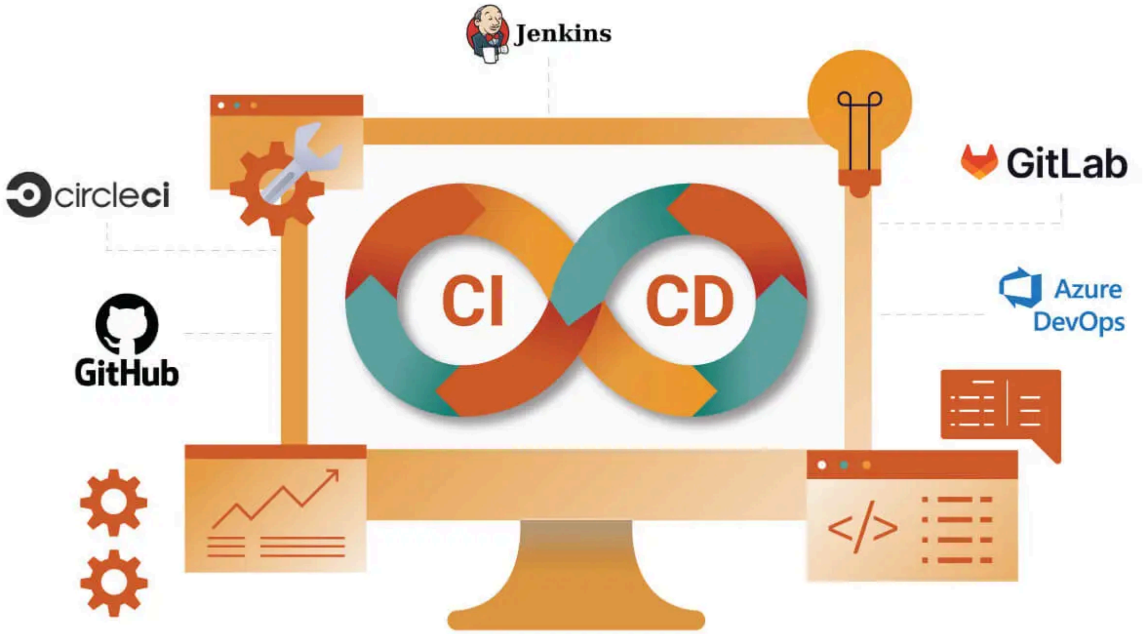
• •

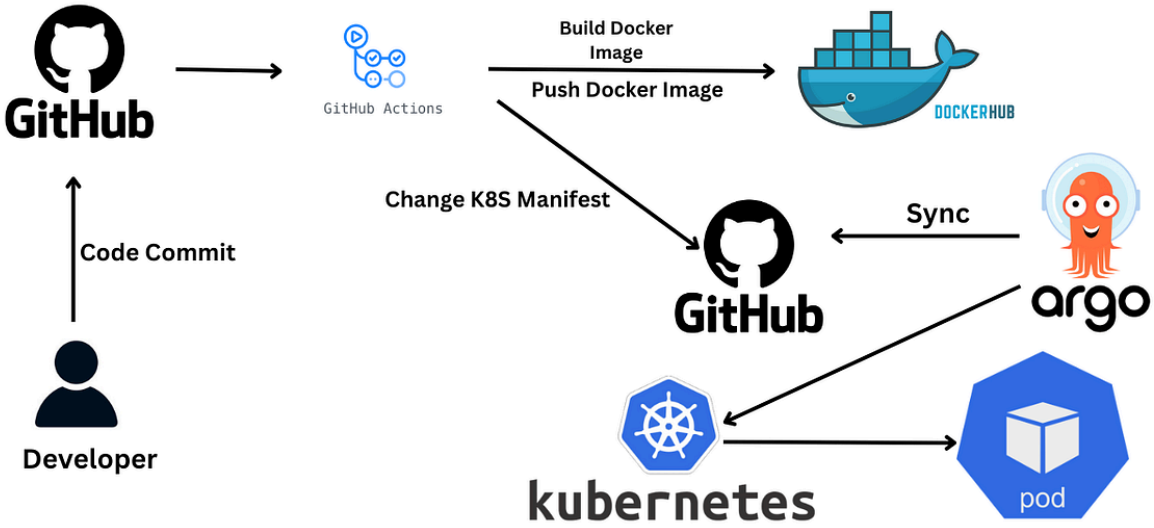
Give **interview questions with answers**

• •

Create **exact project repo structures**


DevOps Roadmap
1 Know IT Basics
2 Select A Programming Language To Learn
3 Virtualization
4 Operating Systems
5 Understand Cloud Computing
6 Understand of CI/CD Pipelines
7 Networking, Security & Protocols
8 Logs Management
9 Server Management
10 Automated Testing
11 Containerization -Docker
12 Source Code Management





- -

  manage Linux

- -

deploy apps

- •

automate infra

- •

explain pipelines confidently

- •

Deep MERN interviews

- •

Advanced DSA

- •

Over-learning cloud services

- •

Certificates without projects

- •

File permissions

- •

systemd

- •

networking basics

- •

logs

- •

package management

- •

user & group automation script

- •

disk / CPU monitoring script

- •

log cleanup script

- •

low competition

- •

real ops skill

- •

easy to verify

- •

EC2

- •

IAM

- •

Security Groups

- •

S3

- •

AMI & EBS basics

- •

launch EC2

- •

secure it

- •

deploy an app

- •

explain IAM roles

- •

Take a simple Node.js app

- •

Dockerize it

- •

Run it on EC2

- •

Git

- •

Jenkins pipeline

- •

GitHub webhook

- •

Pods

- •

Deployments

- •

Services

- •

ConfigMaps

- •

Secrets

- •

user/group creation

- •

logs

- •

monitoring

- •

shell scripts

- •

Node app

- •

Dockerfile

- •

Jenkins pipeline

- •

GitHub repo

- •

same app

- •

deployed on minikube or EKS (optional)

- •

Many are hired as **DevOps Support / Cloud Ops / SRE Intern**

- •

Then promoted in 6–12 months

- •

DevOps Engineer (Junior)

- •

Cloud Support Engineer

- •

Site Reliability Engineer (Fresher)

- •

  Platform Engineer (Junior)

- •

  You practice daily

- •

  You push code daily

- •

  You apply aggressively (30–50 applications/week)