DEVOPS

BEGINNER TO PRO ROADMAP

A complete, self-paced study plan with videos, hands-on labs, and notes space

Video courses: Learn With Mithran & Greens Technologys (YouTube)
https://www.youtube.com/@LearnWithMithran/videos
https://www.youtube.com/@greenstechnologys3238/videos
Reference roadmap: roadmap.sh/devops | Practice: KodeKloud · Killercoda · iximiuz Labs

How to Use This Roadmap

This document is a self-paced, phase-by-phase DevOps study plan built for going from complete beginner to job-ready. Each phase includes: what to learn, matching video lessons, hands-on labs to practice on, and a blank notes area for you to fill in as you study.

Video Channels Used

Two connected YouTube channels are referenced throughout — Mithran is Head Trainer at Greens Technologys, so both channels cover the same syllabus from slightly different angles:

- **Learn With Mithran (personal channel — English + Tamil, cloud/DevOps/Linux/Python/AI):** https://www.youtube.com/@LearnWithMithran/videos
- **Greens Technologys (official training institute channel — Tamil, structured numbered DevOps + AWS series):** https://www.youtube.com/@greenstechnologys3238/videos

Every video listed in this document names which of the two channels it's on. Both catalogs are larger than what's captured here (they add new parts regularly) — use the channel links above or the RSS feeds in Appendix A to browse the full, current libraries.

Suggested Pace

- Spend 1–2 weeks per phase if studying part-time (evenings/weekends).
- Don't skip the hands-on labs — DevOps is a practical discipline; watching alone will not build muscle memory.
- Re-do each phase's practice labs after a break to check retention before moving on.
- Keep a running log of commands and errors you hit — that log becomes your personal cheat sheet.

Roadmap Reference

This plan follows the same overall shape as the community-maintained roadmap at roadmap.sh, adapted into a linear, teachable order:

- **Official DevOps Roadmap:** https://roadmap.sh/devops
- **DevOps Roadmap Projects (build these once you finish a phase):** https://roadmap.sh/devops/projects

roadmap.sh describes DevOps as a cultural and collaborative mindset — not a single tool — that blends development and operations through automation, CI/CD, infrastructure as code, monitoring and cloud platforms. The path below turns that philosophy into a concrete, orderable curriculum.

PHASE 0

Set Up Your Machine (Windows + WSL + Docker)

Before writing a single command, get a real Linux environment running locally. On Windows, WSL (Windows Subsystem for Linux) gives you a genuine Linux kernel and command line without dual-booting or a full VM.

Step 1 — Install WSL (Windows Subsystem for Linux)

Open PowerShell as Administrator (right-click → "Run as administrator"), then run:

```
wsl --install
```

This single command enables the required Windows features and installs Ubuntu by default. Restart your PC when prompted, then launch "Ubuntu" from the Start Menu and create your Linux username/password on first launch.

Useful WSL commands

```
wsl --list --online          # see all installable distros
wsl --install -d <Distro>    # install a specific distro, e.g. Debian, Kali
wsl --list --verbose         # check installed distros & WSL version (1 or 2)
wsl --set-version <Distro> 2 # upgrade a distro to WSL 2
wsl --update                 # update the WSL kernel
```

- **Official Microsoft WSL install docs:** https://learn.microsoft.com/en-us/windows/wsl/install
- **Best practices for a WSL dev environment:** https://learn.microsoft.com/en-us/windows/wsl/setup/environment

Step 2 — Practice Linux Commands Inside WSL

Once inside your Ubuntu WSL terminal, you are running real Linux. Everything in Phase 1 (Linux fundamentals) below can be practiced directly here — no cloud account or VM needed.

```
sudo apt update && sudo apt upgrade -y   # always start here on a fresh install
sudo apt install git curl wget vim -y    # install core CLI tools
```

Step 3 — Install Docker Engine (inside WSL / Ubuntu)

Use the official apt-based install so Docker stays updatable via apt like any other package:

```
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin

sudo usermod -aG docker $USER   # run docker without typing sudo every time
sudo docker run hello-world     # verify it works
```

Alternative: Docker Desktop for Windows also has a native WSL 2 integration toggle in Settings → Resources → WSL Integration, which lets you use the same Docker Engine from inside any WSL distro without installing Docker separately in Linux.

- **Official Docker Engine install docs (all OS options):** https://docs.docker.com/engine/install/
- **Docker Engine install — Ubuntu (detailed):** https://docs.docker.com/engine/install/ubuntu/

My Setup Notes

PHASE 1

Linux & Command Line Fundamentals

Be fully comfortable navigating, managing and troubleshooting a Linux server from the terminal — this is the foundation every other DevOps tool sits on top of.

What to Learn

- Linux file system hierarchy (/etc, /var, /home, /usr, /opt)
- Navigation & file operations: ls, cd, pwd, cp, mv, rm, find, locate
- File permissions & ownership: chmod, chown, umask
- Users & groups: useradd, usermod, passwd, sudo, /etc/passwd
- Process management: ps, top, htop, kill, systemctl, journalctl
- Package managers: apt (Debian/Ubuntu), yum/dnf (RHEL/CentOS)
- Text processing: grep, sed, awk, cat, less, cut, sort, uniq
- Networking basics on Linux: ip a, netstat/ss, curl, wget, ping, traceroute
- Disk & storage: df, du, mount, fdisk, lsblk
- SSH & remote access: ssh, scp, rsync, ssh-keygen
- Environment variables, PATH, and shell profiles (.bashrc, .bash_profile)
- Cron jobs & task scheduling

Video Resource(s)

- Intro to Cloud & DevOps in Tamil | Feat. Greens Technologies *— Learn With Mithran*

Note: Both channels present DevOps as a numbered series — check each channel's Playlists tab for a dedicated Linux Masterclass and Shell scripting episodes, since exact part numbers change as new videos are added.

Practice On

- **KodeKloud Linux Playground (free, browser-based):** https://kodekloud.com/playgrounds/#LinuxPlaygrounds
- **KodeKloud Linux Learning Path:** https://kodekloud.com/learning-path/linux
- **iximiuz Labs — Linux challenges:** https://labs.iximiuz.com/challenges?category=linux
- **Killercoda — Linux scenarios:** https://killercoda.com/

My Notes

PHASE 2

Networking Fundamentals

DevOps engineers debug connectivity constantly — you need a working mental model of how packets move from a browser to a server and back.

What to Learn

- OSI model & TCP/IP model — the 7 (or 4) layers and what lives where
- IP addressing: IPv4, IPv6, public vs. private IPs, CIDR notation
- Subnetting — splitting networks, calculating subnet masks and host ranges
- DNS — how domain resolution works, A/CNAME/MX/TXT records
- TCP vs UDP, the three-way handshake, ports & sockets
- HTTP/HTTPS, TLS/SSL certificates, how a browser request actually works
- Routing, NAT, firewalls, and load balancers
- VPCs, subnets (public/private), security groups, and route tables (cloud networking)
- Common troubleshooting tools: ping, traceroute/tracert, nslookup/dig, curl, netstat

Video Resource(s)

- VPC Prerequisites Made Easy: Public & Private IP, CIDR Range (Part 6) *— Learn With Mithran*
- AWS - VPC in Tamil (Part 1) *— Greens Technologys*

Note: Supplement with the dedicated resources below for OSI/TCP fundamentals and subnetting drills — neither channel has a video purely on OSI/TCP theory yet.

Practice On

- **iximiuz Labs — Computer Networking Fundamentals (course):** https://labs.iximiuz.com/courses/computer-networking-fundamentals
- **iximiuz Labs — Networking challenges:** https://labs.iximiuz.com/challenges?category=networking
- **Cisco Networking Academy — Introduction to Networks (free):** https://www.netacad.com
- **A-to-Z of Networking for DevOps (GitHub notes):** https://github.com/iam-veeramalla/a-to-z-of-networking
- **Subnetting practice drills:** https://subnettingpractice.com

My Notes

PHASE 3

Git & Version Control

Every pipeline starts with a commit. Master branching, merging and collaboration workflows before touching automation.

What to Learn

- Git fundamentals: init, clone, add, commit, status, log
- Branching & merging: branch, checkout/switch, merge, rebase
- Remote workflows: push, pull, fetch, remotes
- Undoing changes: reset, revert, stash, checkout -- file
- Collaboration: pull requests, code review, forking workflow
- GitHub/GitLab basics: issues, actions, protected branches
- Git strategies: Gitflow, trunk-based development

Video Resource(s)

- Master Git fetch, pull, clone, reset, revert & stash (Part 3) *— Learn With Mithran*
- DevOps - GitHub (Part 1) in Tamil *— Greens Technologys*

Practice On

- **KodeKloud Git labs:** https://kodekloud.com/studio/labs/git
- **Learn Git Branching (interactive, free):** https://learngitbranching.js.org

My Notes

PHASE 4

Scripting — Bash & Python

Automation is the whole point of DevOps. You need to comfortably write scripts that glue tools together.

What to Learn

- Bash scripting: variables, conditionals, loops, functions, exit codes
- Working with arguments, reading input, and writing reusable scripts
- Python basics: data types, control flow, functions, modules
- Python for automation: os, sys, subprocess, requests, argparse
- Working with APIs and JSON/YAML in scripts
- Writing scripts that call cloud CLIs (aws-cli, az-cli) and Docker

Video Resource(s)

- DevOps - Python (Part 1) in Tamil *— Greens Technologys*

Note: No confirmed Bash-specific upload was found on either channel during this pass — check each channel's Playlists tab, since new episodes are added regularly.

Practice On

- **KodeKloud Python Learning Path:** https://kodekloud.com/learning-path/aws
- **iximiuz Labs — Linux/shell challenges:** https://labs.iximiuz.com/challenges?category=linux

My Notes

PHASE 5

Docker & Containers

Package an app once, run it anywhere. This is the single most important practical skill in modern DevOps.

What to Learn

- Containers vs. virtual machines — why containers are lightweight
- Docker architecture: daemon, client, images, containers, registries
- Writing a Dockerfile: FROM, RUN, COPY, CMD, ENTRYPOINT, multi-stage builds
- Image management: build, tag, push, pull (Docker Hub / private registries)
- Container lifecycle: run, exec, logs, stop, rm, ps
- Networking in Docker: bridge, host, overlay networks, port mapping
- Persisting data: volumes vs. bind mounts
- Docker Compose for multi-container apps
- Image optimization & security basics (small base images, non-root users, scanning)

Install Docker Engine on Ubuntu / WSL (official apt method)

```
# 1. Remove any old versions
sudo apt-get remove docker docker-engine docker.io containerd runc

# 2. Set up Docker's apt repository
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

# 3. Install Docker Engine, CLI, containerd and plugins
sudo apt-get update
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-
plugin

# 4. Run Docker without sudo (log out/in after this)
sudo usermod -aG docker $USER

# 5. Verify installation
sudo docker run hello-world
```

Quick alternative — Docker's convenience script (any Debian/Ubuntu-based Linux)

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
sudo usermod -aG docker $USER
```

Video Resource(s)

- DevOps – Docker (Part 1) in Tamil *— Greens Technologys*
- DevOps – Docker (Part 2) in Tamil *— Greens Technologys*

Practice On

- **KodeKloud Docker labs:** https://kodekloud.com/studio/labs/docker
- **iximiuz Labs — Docker: Zero to Hero roadmap:** https://labs.iximiuz.com/roadmaps/docker
- **iximiuz Labs — Docker playground:** https://labs.iximiuz.com/playgrounds/docker
- **Killercoda — Docker scenarios:** https://killercoda.com/

My Notes

PHASE 6

CI/CD with Jenkins

Turn 'build → test → deploy' into an automatic pipeline instead of a manual checklist.

What to Learn

- CI/CD concepts: continuous integration, delivery, deployment
- Jenkins architecture: master/controller, agents/nodes, plugins
- Freestyle jobs vs. Declarative Pipelines (Jenkinsfile)
- Integrating Jenkins with Git/GitHub (webhooks, credentials, SCM polling)
- Building, testing and packaging in a pipeline stage
- Integrating Docker builds and pushes into a Jenkins pipeline
- Deploying to servers or Kubernetes from a pipeline
- Alternatives to know: GitHub Actions, GitLab CI, Argo CD (GitOps)

Video Resource(s)

- Complete CI/CD + GitOps Pipeline Tutorial — Jenkins, Argo CD, Docker, SonarQube, Trivy, Kubernetes *— Learn With Mithran*
- Learn ArgoCD & GitOps from Scratch for Continuous Deployment *— Learn With Mithran*
- AWS - Codedeploy & Codepipeline in Tamil *— Greens Technologys*

Practice On

- **KodeKloud CI/CD playground:** https://kodekloud.com/playgrounds/playground-ci-cd
- **KodeKloud Certified Jenkins Engineer course:** https://kodekloud.com/courses/certified-jenkins-engineer

My Notes

PHASE 7

Configuration Management — Ansible

Configure and enforce consistent state across dozens or thousands of servers without logging into each one by hand.

What to Learn

- Agentless architecture: control node, managed nodes, SSH-based
- Inventory files (static & dynamic) and host groups
- Ad-hoc commands vs. Playbooks (YAML)
- Modules, tasks, handlers, variables, and facts
- Roles — organizing reusable automation
- Templates with Jinja2, loops and conditionals
- Ansible Vault for secrets management
- Aware of alternatives: SaltStack (event-driven config management, less common than Ansible)

Video Resource(s)

- Introduction to Ansible (Part 1) *— Learn With Mithran*
- DevOps - Ansible (Part -1) in Tamil *— Greens Technologys*
- DevOps - Ansible (Part -2) in Tamil *— Greens Technologys*
- DevOps - SaltStack (Part -1) in Tamil (alternative tool, optional) *— Greens Technologys*

Practice On

- **KodeKloud Ansible course:** https://kodekloud.com/courses
- **Killercoda — Ansible scenarios:** https://killercoda.com/

My Notes

PHASE 8

Infrastructure as Code — Terraform

Define your entire cloud infrastructure in version-controlled code instead of clicking around a console.

What to Learn

- IaC concepts: declarative vs. imperative, desired state
- Terraform core workflow: init, plan, apply, destroy
- Providers, resources, and data sources
- Variables, outputs, and locals
- State files — local vs. remote state (S3 + DynamoDB locking)
- Modules — reusing and sharing infrastructure code
- Workspaces and managing multiple environments (dev/stage/prod)

Video Resource(s)

- Learn Terraform with AWS from scratch (Part 1) *— Learn With Mithran*
- DevOps - Terraform in Tamil *— Greens Technologys*

Practice On

- **KodeKloud Terraform labs:** https://kodekloud.com/studio/labs/terraform
- **KodeKloud IaC Learning Path:** https://kodekloud.com/learning-path/iac

My Notes

PHASE 9

Kubernetes

Orchestrate containers at scale — deployment, scaling, self-healing and networking, all managed declaratively.

What to Learn

- Kubernetes architecture: control plane, nodes, kubelet, etcd, scheduler
- Core objects: Pods, ReplicaSets, Deployments, Services, Namespaces
- kubectl fundamentals and YAML manifests
- ConfigMaps & Secrets
- Networking: ClusterIP, NodePort, LoadBalancer, Ingress
- Storage: Volumes, PersistentVolumes, PersistentVolumeClaims
- Scaling & self-healing: HPA, liveness/readiness probes
- Helm — packaging and templating Kubernetes apps
- RBAC, IAM integration (e.g., AWS IRSA), and cluster security basics
- Managed Kubernetes: EKS (AWS), AKS (Azure), GKE (GCP)

Video Resource(s)

- Kubernetes from scratch — Introduction & Installation (Part 1) *— Learn With Mithran*
- Learn Kubernetes Deployments & Services from scratch *— Learn With Mithran*
- Create EKS Cluster and Manage IAM Users with RBAC & IRSA (Part 4) *— Learn With Mithran*

Practice On

- **KodeKloud Kubernetes labs:** https://kodekloud.com/studio/labs/kubernetes
- **KodeKloud CKA preparation path:** https://kodekloud.com/learning-path/cka
- **Killercoda — Kubernetes scenarios (used by many CKA/CKAD candidates):** https://killercoda.com/
- **iximiuz Labs — Kubernetes playground:** https://labs.iximiuz.com/playgrounds/k8s-omni

My Notes

PHASE 10

Cloud Platforms — AWS (+ Azure/GCP awareness)

Learn one cloud provider deeply (AWS is the most in-demand for DevOps roles), and understand the equivalents on Azure/GCP. Note: AWS launched in 2006 and is not a new or static platform — it now offers 200+ services and ships thousands of updates a year, so treat any single course or video as a snapshot in time and always cross-check against the current official docs below.

What to Learn

- Core compute: EC2 instances, AMIs, key pairs, security groups
- Networking: VPC, subnets, route tables, Internet/NAT gateways
- Storage: S3, EBS, EFS
- IAM: users, groups, roles, policies, least privilege
- Containers on AWS: ECS, ECR, Fargate, EKS
- Serverless & API layer: Lambda, API Gateway
- Databases: RDS, DynamoDB basics
- Cost awareness & the AWS Well-Architected Framework
- Azure equivalents: VMs, VNet, AKS, Blob Storage — GCP equivalents: Compute Engine, VPC, GKE, Cloud Storage

Video Resource(s)

- AWS – EC2 in Tamil *— Learn With Mithran*
- AWS ECS, ECR & Fargate Full Hands-On Container Deployment (Part 15) *— Learn With Mithran*
- Complete API Gateway guide in 2 Hours (Part 17) *— Learn With Mithran*
- AWS - EC2 Advanced in Tamil *— Greens Technologys*
- AWS - RDS & DynamoDB in Tamil *— Greens Technologys*
- AWS - SQS, SNS & CloudWatch in Tamil *— Greens Technologys*
- AWS - RedShift & CloudTrail in Tamil *— Greens Technologys*
- AWS - Lambda & EFS in Tamil *— Greens Technologys*

Practice On

- **KodeKloud AWS Learning Path:** https://kodekloud.com/learning-path/aws
- **KodeKloud AWS Playground (real AWS services, sandboxed):** https://kodekloud.com/cloud-playgrounds/aws
- **KodeKloud 100 Days of Cloud (free challenge):** https://kodekloud.com/100-days-of-cloud
- **AWS “What's New” — official running log of new services & features:** https://aws.amazon.com/new/

My Notes

PHASE 11

Monitoring, Logging & Observability

You can't fix what you can't see. Learn to instrument systems and read the signals before things break.

What to Learn

- The three pillars: metrics, logs, and traces
- Prometheus — metrics collection & PromQL basics
- Grafana — building dashboards and alerts
- Centralized logging: the ELK/EFK stack (Elasticsearch, Logstash/Fluentd, Kibana)
- Alerting strategy: what to alert on, on-call basics, avoiding alert fatigue
- Application performance monitoring (APM) concepts

Video Resource(s)

- DevOps - AppDynamics in Tamil *— Greens Technologys*

Note: No confirmed Prometheus/Grafana-specific upload was found on either channel during this pass — check each channel's Playlists tab, and use the resources below in the meantime.

Practice On

- **KodeKloud DevOps Learning Path (covers Prometheus/Grafana):** https://kodekloud.com/learning-path/devops
- **Prometheus official docs:** https://prometheus.io/docs/introduction/overview/
- **Grafana official docs:** https://grafana.com/docs/grafana/latest/

My Notes

PHASE 12

Security, GitOps & Going Pro

Bake security into the pipeline (DevSecOps) and adopt GitOps practices used by senior/lead DevOps engineers.

What to Learn

- DevSecOps mindset: shift-left security
- Container image scanning (e.g., Trivy) and dependency scanning
- Static/dynamic code analysis (e.g., SonarQube)
- Secrets management (Vault, AWS Secrets Manager, Ansible Vault)
- GitOps: declarative, Git-as-source-of-truth deployments with Argo CD / Flux
- Kubernetes security: RBAC, network policies, pod security standards
- Building an end-to-end capstone project: Git → Jenkins → Docker → Trivy scan → Argo CD → Kubernetes
- Certifications to target: CKA, CKAD, CKS, AWS Certified DevOps Engineer / Solutions Architect, Terraform Associate

Video Resource(s)

- Complete CI/CD + GitOps Pipeline Tutorial — Jenkins, Argo CD, Docker, SonarQube, Trivy, Kubernetes *— Learn With Mithran*

Practice On

- **KodeKloud Certified Kubernetes Security Specialist (CKS):** https://kodekloud.com/courses/certified-kubernetes-security-specialist-cks
- **KodeKloud Kubestronaut path (all 5 CNCF K8s certs):** https://kodekloud.com/learning-path/kubestronaut
- **KodeKloud 100 Days of DevOps (free challenge):** https://kodekloud.com/100-days-of-devops
- **roadmap.sh DevOps projects:** https://roadmap.sh/devops/projects

My Notes

Practice Platforms — Where to Get Hands-On

Watching videos teaches concepts; these platforms are where the skill actually gets built. Use them alongside every phase above.

KodeKloud

The most complete DevOps/Cloud/Kubernetes learning platform, combining structured courses with browser-based hands-on labs, sandboxed cloud playgrounds, and free daily-challenge tracks.

- **Homepage:** https://kodekloud.com/
- **All Hands-on Labs (Docker, Kubernetes, Linux, Git, Terraform...):** https://kodekloud.com/studio/labs
- **Free Playgrounds (no-cost sandboxes: AWS, Azure, GCP, Linux, Kubernetes):** https://kodekloud.com/playgrounds
- **100 Days of DevOps (free daily challenge):** https://kodekloud.com/100-days-of-devops
- **100 Days of Cloud (free daily challenge):** https://kodekloud.com/100-days-of-cloud
- **DevOps Engineer learning path:** https://kodekloud.com/learning-path/devops-engineer

Killercoda

Free, browser-based interactive scenarios for Linux, Docker, Kubernetes and Git — widely used by candidates preparing for the CKA, CKAD and CKS Kubernetes certification exams because it mimics the real exam's terminal-only environment.

- **Homepage:** https://killercoda.com/

iximiuz Labs

A hands-on platform focused specifically on Linux, Docker, Kubernetes and Networking — combining a large bank of bite-sized troubleshooting challenges with real playgrounds (VMs, Docker hosts, Kubernetes clusters) you can spin up in seconds via browser or SSH.

- **Homepage:** https://labs.iximiuz.com/
- **Docker: Zero to Hero roadmap:** https://labs.iximiuz.com/roadmaps/docker
- **Challenge catalog (Linux / Networking / Containers / Kubernetes):** https://labs.iximiuz.com/challenges
- **Free playgrounds (Linux, Docker, Kubernetes):** https://labs.iximiuz.com/playgrounds

Platform Comparison

Platform	Best For
KodeKloud	Structured courses + certification prep (CKA/CKAD/CKS/AWS) + free 100-day challenges
Killercoda	Free, exam-style terminal scenarios — great for last-mile Kubernetes cert practice
iximiuz Labs	Deep Linux/Docker/Networking internals, bite-sized troubleshooting challenges

Extra: Networking Study Materials

Networking trips up more DevOps beginners than any other topic because it is rarely taught hands-on. These resources go deeper than Phase 2 above.

- **iximiuz Labs — Computer Networking Fundamentals course:** https://labs.iximiuz.com/courses/computer-networking-fundamentals
- **Cisco Networking Academy (free Introduction to Networks course):** https://www.netacad.com
- **A-to-Z of Networking for DevOps (community notes on GitHub):** https://github.com/iam-veeramalla/a-to-z-of-networking
- **Subnetting practice drills (timed quizzes):** https://subnettingpractice.com
- **iximiuz Labs — Networking challenge bank:** https://labs.iximiuz.com/challenges?category=networking

Core Concepts Checklist

- Can you explain the difference between a switch and a router in one sentence?
- Can you calculate the usable host range for a /26 subnet by hand?
- Can you trace what happens between typing a URL and a page loading (DNS → TCP handshake → TLS → HTTP request/response)?
- Do you know the difference between a Security Group and a Network ACL in AWS?

My Networking Notes

Appendix A — Video Index (Both Channels)

Channel 1: Learn With Mithran — Cloud Computing, DevOps, Linux, Python & AI tutorials (personal channel).

- **Channel (all videos):** https://www.youtube.com/@LearnWithMithran/videos
- **Subscribe via RSS:** https://www.youtube.com/feeds/videos.xml?channel_id=UCcW1l2Ke8b8yD0DdojP6Kuw

Channel 2: Greens Technologys — official training institute channel (Chennai), structured numbered DevOps + AWS series in Tamil, aimed at freshers/job seekers.

- **Channel (all videos):** https://www.youtube.com/@greenstechnologys3238/videos
- **Subscribe via RSS:** https://www.youtube.com/feeds/videos.xml?channel_id=UCeKXvwdX6-oDbchcp36rY9g

Important: both channels load their video lists dynamically and could not be fully auto-scraped in this session. Every video below was individually verified through search as genuinely belonging to the named channel, then mapped to the phase it fits. This is a solid, confirmed starting set covering most phases on both channels — open the channel links above to browse the full, current catalogs (including additional numbered parts, e.g. more AWS/DevOps episodes) and add any missing links to the tables below yourself.

Topic / Phase	Video (Learn With Mithran)
Intro	Intro to Cloud & DevOps in Tamil
Git	Master Git fetch, pull, clone, reset, revert & stash (Part 3)
Networking	VPC Prerequisites: Public & Private IP, CIDR Range (Part 6)
Ansible	Introduction to Ansible (Part 1)
Terraform	Learn Terraform with AWS from scratch (Part 1)
Kubernetes	Kubernetes from scratch — Intro & Installation (Part 1) Learn Kubernetes Deployments & Services from scratch Create EKS Cluster, IAM RBAC & IRSA (Part 4)
CI/CD & GitOps	Complete CI/CD + GitOps Pipeline (Jenkins, Argo CD, Docker, SonarQube, Trivy, K8s) Learn ArgoCD & GitOps from Scratch
AWS	AWS – EC2 in Tamil AWS ECS, ECR & Fargate Full Hands-On Deployment (Part 15) Complete API Gateway guide in 2 Hours (Part 17)

Topic / Phase	Video (Greens Technologys)
Git	DevOps - GitHub (Part 1) in Tamil
Networking	AWS - VPC in Tamil (Part 1)
Scripting (Python)	DevOps - Python (Part 1) in Tamil
Docker	DevOps - Docker (Part 1) in Tamil DevOps - Docker (Part 2) in Tamil
CI/CD	AWS - Codedeploy & Codepipeline in Tamil

Topic / Phase	Video (Greens Technologys)
Ansible (+ SaltStack)	DevOps - Ansible (Part -1) in Tamil DevOps - Ansible (Part -2) in Tamil DevOps - SaltStack (Part -1) in Tamil — optional alt. tool
Terraform	DevOps - Terraform in Tamil
AWS	AWS - EC2 Advanced in Tamil AWS - RDS & DynamoDB in Tamil AWS - SQS, SNS & CloudWatch in Tamil AWS - RedShift & CloudTrail in Tamil AWS - Lambda & EFS in Tamil
Monitoring / APM	DevOps - AppDynamics in Tamil

Appendix B — Additional Study Materials

Official Documentation (always the ground truth)

- **Docker Docs:** https://docs.docker.com/
- **Kubernetes Docs:** https://kubernetes.io/docs/home/
- **Ansible Docs:** https://docs.ansible.com/
- **Terraform Docs:** https://developer.hashicorp.com/terraform/docs
- **Jenkins Docs:** https://www.jenkins.io/doc/
- **AWS Documentation:** https://docs.aws.amazon.com/
- **Git Documentation:** https://git-scm.com/doc

Roadmaps & Structured Guides

- **roadmap.sh — DevOps Roadmap:** https://roadmap.sh/devops
- **roadmap.sh — DevOps Best Practices:** https://roadmap.sh/devops/best-practices
- **roadmap.sh — DevOps Tools:** https://roadmap.sh/devops/tools
- **roadmap.sh — DevOps Projects (build these to prove your skills):** https://roadmap.sh/devops/projects

Communities

- **KodeKloud Discord community:** https://kode.wiki/44ANwVH
- **iximiuz Labs Discord community:** https://discord.gg/p6nxaHm46p
- **r/devops on Reddit:** https://www.reddit.com/r/devops/

Certification Targets (once phases are complete)

Certification	Covers
CKA — Certified Kubernetes Administrator	Cluster setup, security, networking, storage, troubleshooting
CKAD — Certified Kubernetes Application Developer	Designing & deploying apps on Kubernetes
CKS — Certified Kubernetes Security Specialist	Hardening clusters, RBAC, pod/network security
AWS Certified Cloud Practitioner	Cloud fundamentals, entry-level AWS knowledge
AWS Certified Solutions Architect – Associate	Designing resilient, secure AWS architectures
HashiCorp Terraform Associate	Core Terraform workflow and IaC best practices

Final Notes / Career Log

Use the space below to track projects built, interviews taken, and skills still to revisit.