Week 1 and 2 Goal: Master Linux, shell scripting, networking, Git.

- **Week 1:**
    - **Project 1:**
        - Provision a Ubuntu server and configure it for a new user with SSH access, sudo privileges, firewall rules, and audit logs.
    - Deliverable1:
        - Git repo: Bash scripts for user creation, SSH config, UFW setup
        - Wiki: User management policy + Linux hardening checklist
    - Project 2:
        - Write a backup script to compress and rotate logs daily. Set up with `cron`.
    - Deliverable 2:
        - Git repo: `log-backup.sh` with parameters and `crontab` file
        - Wiki: Diagram + how the automation works + logs location

- **Week 2:**
    - **Project 3:**
        - Create a script to perform diagnostics: DNS lookup, port scanning, latency test, HTTP status check.
    - Deliverable 3:
        - Git repo: `network-diagnose.sh`
        - Wiki: Example output, test scenarios (e.g., broken DNS, firewall block)
    - Project 4:
        - Simulate a team with feature/bugfix branches and PR process using GitHub. Use a dummy app.
    - Deliverable 4:
        - Git repo: `dev`, `feature/xyz`, `bugfix/xyz` branches + PRs
        - Wiki: Branching strategy, PR approval process, merge conflict resolution

Week 3 and 4 Goal: Core AWS Services

- Week 3:
    - Project 5

    - Create IAM policies for an EC2 admin, S3 uploader, and a read-only auditor role.
  - Deliverable 5
    - Git repo: IAM JSON policies + Terraform if used
    - Wiki: Role-based access control design + MFA enforcement

  - Project 6
    - Launch EC2 instance with Nginx + host a static site using cloud-init script
  - Deliverable 6
    - Git repo: `user-data.sh`, terraform or CLI script
    - Wiki: Web server architecture + launch walkthrough
- Week 4:
  - Project 7
    - Setup versioned S3 bucket with lifecycle policy for logs. Integrate log archiving via script.
  - Deliverable 7
    - Git repo: `upload-logs.sh`, lifecycle policy JSON
    - Wiki: Lifecycle0 diagram, cost optimization strategy

  - Project 8
    - Provision a VPC with private/public subnets, NAT Gateway, EC2 in private subnet with internet access.
  - Deliverable 8
    - Git repo: Terraform + subnet map
    - Wiki: VPC design diagram + routing table explanatio

Week 5 and 6 Goal: CICD and Build Automation

- Week 5
  - Project 9
    - Setup Jenkins on EC2, create a basic Freestyle job to clone, build, and test a repo.
  - Deliverable 9
    - Git repo: Job config + helper scripts
    - Wiki: Jenkins setup steps, plugins used, job description

  - Project 10

- Create a Jenkins pipeline to build, test, and deploy a Node.js and Python app to EC2
- Deliverable 10
  - Git repo: `Jenkinsfile`, deploy script
  - Wiki: Pipeline architecture, stage-by-stage breakdown

- Week 6
  - Project 11
    - Setup GitHub Actions to test PRs and deploy on push to `main`
  - Deliverable 11
    - Git repo: `.github/workflows/main.yml`
    - Wiki: Workflow trigger description + security context

  - Project 12
    - Create build artifacts (e.g., `.zip`, `.jar`) and upload to S3 using CI tool
  - Deliverable 12
    - Git repo: Artifact naming + upload script
    - Wiki: Artifact strategy + versioning method

Week 7 and 8 Goal: IAAC

- Project 13
  - Use Terraform to provision EC2 and S3
- Deliverable 13
  - Git repo: `main.tf`, `outputs.tf`, `variables.tf`
  - Wiki: Setup steps, state file explanation

- Project 14
  - Create reusable VPC module and manage state in S3 backend
- Deliverable 14
  - Git repo: VPC module + backend config
  - Wiki: Module inputs/outputs, backend config steps
- Project 15
  - Deploy a 2-tier app (Web + DB) using CloudFormation
- Deliverable 15
  - Git repo: YAML templates
  - Wiki: Stack walkthrough + rollback scenarios

- Project 16
    - Jenkins pipeline to run `terraform plan` and `apply`
- Deliverable 16
    - Git repo: `Jenkinsfile`, `main.tf`
    - Wiki: CI + IaC integration architecture

Week 9 and 10 Goal: Container and Monitoring.

- Project 17
    - Dockerize a Flask app, push to Docker Hub
- Deliverable 17
    - Git repo: `Dockerfile`, `docker-compose.yml`
    - Wiki: Docker build + deployment strategy

- Project 18
    - Deploy app to ECS Fargate with task definition + ALB
- Deliverable 18
    - Git repo: ECS definition files + terraform/CLI script
    - Wiki: ECS service architecture

- Project 19
    - Setup alarms for ECS and EC2 + SNS email notifications
- Deliverable 19
    - Git repo: CloudWatch alarm JSON + SNS setup
    - Wiki: Monitoring strategy + alert thresholds

- Project 20
    - Create CloudWatch dashboard for ECS + EC2 metrics
- Deliverable 20
    - Git repo: Dashboard template/script
    - Wiki: Dashboard layout explanation + examples

Week 11 and 12 Goal: AWS Cloud Practitioner Preparation.

- Project 21
  - Present a cost-optimized solution for a sample workload
- Deliverable 21
  - Git repo: Sample architecture and pricing sheet
  - Wiki: Use case breakdown + AWS pricing strategy

- Project 22
  - Show how IAM prevents unauthorized access in a simulated attack
- Deliverable 22
  - Git repo: IAM policies + incident scenario
  - Wiki: IAM best practices documentation

- Project 23
  - Create a mini cloud architecture (EC2 + S3 + RDS) with use cases
- Deliverable 23
  - Git repo: Architecture diagram + provisioning script
  - Wiki: Explanation of why each service was chosen

- Project 24
  - Complete and present 50-question mock test results
- Deliverable 24
  - Git repo: Answers + explanations (markdown)
  - Wiki: Summary of weak areas + final study notes