### **Linux and Shell scripting**

### **🔹 Beginner Projects**

1. **System Information Script** – Display OS, kernel version, CPU, RAM, and disk usage.
2. **File Backup Script** – Copy files from one directory to another with timestamps.
3. **User Account Information Tool** – Show currently logged-in users and their processes.
4. **Directory Organizer** – Automatically sort files into folders (e.g., images, documents, music).
5. **Basic To-Do List** – Store and manage tasks using text files and shell commands.

### **🔹 Intermediate Projects**

1. **Log Analyzer** – Parse /var/log files to detect errors, warnings, or failed logins.
2. **Automated Installer Script** – Install and configure commonly used packages on a fresh Linux system.
3. **Process Monitor** – Display top CPU- or memory-consuming processes and alert if limits are exceeded.
4. **Network Scanner** – Use ping or nmap to check active devices on the local network.
5. **Disk Usage Report** – Generate a summary of storage usage and send it via email.

### **🔹 Advanced Projects**

1. **Automated Backup with Compression & Encryption** – Backup directories, compress them with tar/gzip, and encrypt using gpg.
2. **Cron Job Management Tool** – Script to list, add, and remove cron jobs easily.
3. **Web Server Log Visualizer** – Parse Apache/Nginx logs and generate summaries (top IPs, request counts, errors).
4. **Custom Command-Line Utilities** – Write scripts that behave like mini Linux commands (e.g., mygrep, myfind).
5. **System Health Dashboard** – Collect CPU, memory, disk, and network stats and display them in a formatted report.

### **AWS**

### **🔹 Beginner**

1. **Static Website Hosting on S3 + CloudFront** — S3, IAM, CloudFront, Route 53
2. **EC2 Web Server Bootstrap** — EC2 (user data), IAM roles, Security Groups
3. **S3 Backup & Lifecycle Policies** — S3 (versioning, lifecycle), IAM
4. **Basic IAM Hardening Kit** — IAM users/groups/roles, MFA, password policy
5. **RDS Quickstart with Read Replica** — RDS (MySQL/Postgres), Security Groups
6. **SNS Email/SMS Alerts** — SNS topics/subscriptions, IAM
7. **CloudWatch Alarms for EC2** — CloudWatch metrics/alarms, SNS notifications

### **🔸 Intermediate**

1. **CloudFormation One-Click VPC + EC2 Stack** — CloudFormation, EC2, IAM, SGs
2. **S3 → Lambda → SNS Event Pipeline** — S3 event notifications, Lambda, SNS, IAM
3. **Centralized Log Collection** — CloudWatch Agent on EC2, Log groups, Metric filters, Alarms
4. **Blue/Green Deployment on ECS (Fargate)** — ECS, ECR, ALB, CodeDeploy, IAM, CloudWatch
5. **EKS Cluster with Managed Node Groups** — EKS, IAM roles for Service Accounts (IRSA), ALB Ingress, CloudWatch Container Insights
6. **RDS Automated Backups + PITR Demo** — RDS automated backups, snapshots, restore test
7. **Private S3 Access via VPC Endpoint** — S3, Gateway Endpoint, IAM policies, Bucket policies
8. **Cost Guardrails** — CloudWatch billing alarms, SNS, Cost Explorer saved reports, IAM least privilege

### **🔺 Advanced**

1. **Highly Available 3-Tier App** — ALB + Auto Scaling EC2 (web), RDS Multi-AZ (db), S3 (assets), CloudWatch (dashboards & alarms), IAM
2. **Serverless Image Processing** — S3 (uploads) → Lambda (resize) → S3 (outputs) with SNS failure alerts; IaC via CloudFormation
3. **Multi-Region S3 DR Strategy** — S3 replication, KMS, bucket policies, failover plan doc + validation script
4. **Event-Driven Data Ingestion** — S3 → SNS → SQS → ECS/EKS consumer; observability in CloudWatch (dashboards/logs/alarms)
5. **GitOps on EKS** — EKS, ArgoCD/Flux, IRSA, Secrets management, CloudWatch Container Insights, HPA with metrics-server
6. **ECS Fargate Spot Cost Optimization** — ECS capacity providers (Fargate + Fargate Spot), ALB, auto scaling policies, CloudWatch alarms
7. **RDS Proxy + App Connection Pooling** — RDS, RDS Proxy, ECS/EKS app, IAM auth, performance testing
8. **Cross-Account Access Pattern** — IAM roles with trust policies, S3 bucket policies, CloudWatch cross-account dashboards
9. **CloudFormation Macros/Modules Library** — Reusable stacks for VPC, ALB, ECS/EKS services, RDS; CI checks for drift & linting
10. **Full Observability Pack** — CloudWatch Agent (metrics/logs/traces), Container Insights, synthetic canaries, alarms → SNS/Slack webhook

**Docker**

## **🔹 Beginner Projects**

1. **Hello World Container** – Build and run a simple container with a custom message.
2. **Dockerized To-Do App** – Wrap a simple Python/Node/Go app in a container.
3. **Nginx Web Server** – Run a static website in an Nginx container with mounted volume for content.
4. **PostgreSQL/MySQL in Docker** – Spin up a database container with persistent storage.
5. **Multi-Container with Docker Compose** – Basic app + database setup using docker-compose.yml.
6. **Custom Base Image** – Build a minimal container from alpine or ubuntu with useful tools installed.

## **🔸 Intermediate Projects**

1. **Log Aggregation Setup** – Dockerize an ELK stack (Elasticsearch, Logstash, Kibana) or EFK (Fluentd).
2. **Reverse Proxy with Traefik or Nginx** – Route multiple apps in containers under different paths/ports.
3. **CI/CD Pipeline with GitHub Actions + Docker** – Auto-build and push images to Docker Hub/ECR.
4. **Containerized Monitoring Stack** – Use Prometheus + Grafana in Docker Compose to monitor containers.
5. **Docker Secrets & Configs** – Securely inject API keys/configs into running containers.
6. **Multi-Stage Docker Build** – Build a Go/Java app with reduced image size.
7. **Networked Microservices** – Two or more services (e.g., API + DB + frontend) communicating in Docker network.

## **🔺 Advanced Projects**

1. **Kubernetes Local Lab** – Use Docker Desktop + Minikube/Kind to deploy an app to K8s.
2. **ECS/EKS Deployment** – Push Dockerized app to AWS ECR and run it in ECS or EKS.
3. **Blue/Green or Canary Deployment** – Use Docker Compose or K8s manifests to simulate rollout strategies.
4. **Service Mesh Experiment** – Istio/Linkerd over Dockerized microservices.
5. **Distributed Data App** – Run Cassandra/Redis Cluster in Docker.
6. **Build Your Own CI System** – Use Jenkins in Docker to build/test/deploy another app.
7. **Auto-Scaling Lab** – Combine Dockerized app + metrics exporter + orchestration (ECS/K8s HPA).
8. **Security Hardening** – Create a hardened Dockerfile (non-root user, seccomp profile, slim image).
9. **Custom Docker Swarm Cluster** – Deploy replicated services, overlay network, and secrets.
10. **Chaos Engineering Sandbox** – Simulate container crashes and recovery policies.

**Kubernetes**

## **🔹 Beginner Projects**

1. **Minikube Setup & First Deployment** – Install Minikube, deploy a simple Nginx/Apache pod.
2. **Multi-Pod Deployment** – Deploy a sample web app with replicas using a Deployment + Service.
3. **ConfigMaps & Secrets Demo** – Inject configs and credentials into pods.
4. **Rolling Update vs Recreate** – Deploy an app, update its version, and test rollout strategies.
5. **Kubernetes Dashboard** – Install and explore the official dashboard UI.
6. **Persistent Volume Example** – Attach a volume to a pod (e.g., MySQL pod with persistent storage).
7. **Namespace Isolation** – Create separate namespaces for dev, staging, prod.

## **🔸 Intermediate Projects**

1. **Ingress Controller Setup** – NGINX/Traefik Ingress for routing multiple services under one domain.
2. **Horizontal Pod Autoscaler (HPA)** – Deploy a CPU-bound app and scale pods based on load.
3. **StatefulSet Example** – Run a database cluster (PostgreSQL/Redis) with stable network IDs.
4. **Custom Helm Chart** – Package and deploy your own chart for a sample app.
5. **RBAC Demo** – Create custom roles and service accounts with limited access.
6. **Blue/Green or Canary Deployments** – Use labels/selectors or Argo Rollouts to manage rollouts.
7. **Secrets Management with External Systems** – Integrate Kubernetes with HashiCorp Vault or AWS Secrets Manager.
8. **Centralized Logging** – Deploy EFK (Elasticsearch, Fluentd, Kibana) or Loki + Promtail + Grafana.

## **🔺 Advanced Projects**

1. **Kubernetes on Cloud (EKS/GKE/AKS)** – Provision a managed K8s cluster and deploy apps.
2. **CI/CD with GitOps** – Use ArgoCD or Flux to manage deployments from Git.
3. **Service Mesh Lab** – Deploy Istio/Linkerd for traffic routing, observability, and mTLS.
4. **Multi-Cluster Federation** – Connect multiple clusters with kube-federation.
5. **Chaos Engineering on K8s** – Use LitmusChaos/Chaos Mesh to inject pod/node failures.
6. **Custom Operators (CRDs)** – Write an Operator in Go or Python to manage a custom resource.
7. **KEDA Autoscaling** – Autoscale workloads based on external metrics (e.g., queue length, Kafka lag).
8. **Kubernetes Security Lab** – Test PodSecurityPolicies (or PodSecurity admission), OPA/Gatekeeper policies.
9. **Observability Pack** – Deploy Prometheus, Grafana, Alertmanager, Jaeger/Tempo for metrics, logs, traces.
10. **Hybrid Deployment** – Run workloads across on-prem and cloud with KubeVirt or Rancher.

**Jenkins**

## **🔹 Beginner Projects**

1. **Jenkins Hello World Job** – Freestyle job that prints “Hello Jenkins” to the console.
2. **Simple Build Job** – Clone a GitHub repo and compile a small Java/Python/Node project.
3. **Automated Test Job** – Run unit tests from a repo and publish results.
4. **Parameterized Job** – Build a job with parameters (branch, environment).
5. **Scheduled Jobs (CRON)** – Run a job daily/weekly using Jenkins’ scheduler.

## **🔸 Intermediate Projects**

1. **Jenkins Pipeline (Declarative)** – Write a Jenkinsfile to build, test, and package an app.
2. **CI/CD for Dockerized App** – Build Docker image from repo, push to Docker Hub/ECR.
3. **Multi-Branch Pipeline** – Create a pipeline that automatically builds for every branch.
4. **Pipeline Notifications** – Send build status to Slack, Teams, or email.
5. **Blue Ocean UI Pipelines** – Use Blue Ocean for visual pipeline management.
6. **Parallel Builds** – Run multiple test suites in parallel using pipeline stages.
7. **Artifact Storage** – Archive build outputs and publish them for download.
8. **Basic Security Setup** – Create users, roles, and matrix-based authorization.

## **🔺 Advanced Projects**

1. **Full CI/CD for Microservices** – Multi-pipeline setup for multiple services (frontend + backend + db migrations).
2. **Jenkins + Kubernetes (Jenkins on K8s)** – Deploy Jenkins in Kubernetes and run dynamic agents as pods.
3. **Infrastructure as Code** – Use Jenkins pipeline to apply Terraform/CloudFormation templates.
4. **Pipeline Libraries** – Build and use shared libraries across multiple Jenkins pipelines.
5. **Jenkins + GitOps** – Automate deployment to Kubernetes using ArgoCD/Flux triggers from Jenkins.
6. **Security Hardening** – Implement role-based access control, credential binding, and secret management.
7. **Blue/Green & Canary Deployments** – Jenkins pipeline deploying apps with rollout strategies.
8. **Chaos Engineering Pipeline** – Add fault injection stages (e.g., kill pods/instances) to test resilience.
9. **Monitoring Jenkins with Prometheus** – Expose Jenkins metrics and visualize in Grafana.
10. **Distributed Jenkins Setup** – Master-agent architecture for scaling build workloads.
11. **Jenkins + AWS/GCP/Azure** – Pipeline that deploys apps on cloud infra (EC2, ECS, EKS, etc.).
12. **ML Model CI/CD** – Automate model training/testing/deployment pipeline (Jenkins + Docker + K8s).

**Terraform**

## **🔹 Beginner**

1. **Provider & Remote Backend Bootstrap** — Set up aws provider + remote state in S3 with DynamoDB state lock. *(S3, DynamoDB, tfstate)*
2. **VPC Baseline** — Create a VPC with public/private subnets, IGW, NAT, route tables, NACLs. *(VPC, Subnets, NAT, Routes)*
3. **EC2 + Security Groups** — Launch an EC2 with user data, key pair, SG allowing HTTP/SSH. *(EC2, SG, user\_data)*
4. **S3 Website Bucket** — Static site bucket with versioning, lifecycle rules, block public access exceptions. *(S3, Policies)*
5. **IAM Least-Privilege Role** — IAM role + policy for EC2/S3 access, attached via instance profile. *(IAM, Policies)*

## **🔸 Intermediate**

1. **ALB + Auto Scaling Group** — ASG across 2 AZs behind an ALB with health checks & target groups. *(EC2, ASG, ALB)*
2. **RDS Module** — Parameterized RDS (MySQL/Postgres) with subnet group, security groups, snapshots. *(RDS, SG, Snapshots)*
3. **CloudFront in Front of S3** — OAC/OAI, custom domain via Route 53, caching behaviors. *(CloudFront, S3, Route53)*
4. **Lambda + API Gateway** — Serverless CRUD with IAM auth, stage variables, logs. *(Lambda, API GW, CW Logs)*
5. **Private S3 via VPC Endpoint** — Gateway endpoint + bucket policy restricting access to your VPC. *(S3, VPC Endpoint)*
6. **ECS Fargate Service** — ECR image + ECS service/TaskDef, ALB, autoscaling policies. *(ECR, ECS, ALB)*
7. **EKS Starter Cluster** — Managed node groups, IRSA, cluster auth, kubeconfig output. *(EKS, IAM, OIDC)*
8. **Eventing: S3 → SNS → SQS** — Fan-out pattern with DLQ and permissions wired via Terraform. *(S3, SNS, SQS)*
9. **CloudWatch Alarms Pack** — CPU/ALB/StatusCheck alarms → SNS; CW dashboards. *(CloudWatch, SNS)*
10. **Secrets Manager Integration** — Store DB creds; inject into ECS task or Lambda env. *(Secrets Manager, IAM)*

## **🔺 Advanced**

1. **Multi-Account Landing Zone (IaC)** — Orgs bootstrap, SCPs, account vending via Terraform. *(AWS Organizations, SCP, IAM)*
2. **Multi-Region DR** — S3 CRR, Route 53 failover records, RDS cross-region read replica. *(S3 Replication, Route53, RDS)*
3. **Network Hub-Spoke** — Central shared VPC + TGW + VPC attachments, route propagation. *(Transit Gateway, VPC)*
4. **Blue/Green with CodeDeploy** — ASG or ECS service with traffic-shift strategies and alarms. *(CodeDeploy, ALB, CW)*
5. **EKS Production Add-ons** — Deploy add-ons via Terraform: VPC CNI, Cluster Autoscaler, ALB Ingress, ExternalDNS, KEDA. *(EKS, Helm provider)*
6. **Data Lake Skeleton** — Tiered S3 buckets (raw/curated), Glue crawlers/catalog, Athena workgroup. *(S3, Glue, Athena)*
7. **Serverless ETL** — EventBridge schedule → Lambda → write to S3; DLQ, retries, metrics. *(EventBridge, Lambda, S3)*
8. **WAF + Shield Setup** — WAF rules for ALB/CloudFront, rate limits, IP sets. *(WAF, CloudFront/ALB)*
9. **Bastion + SSM Session Manager** — No public SSH; use SSM and session policies. *(SSM, IAM, EC2)*
10. **Cost Guardrails** — Budgets + alerts, tag policies, SCPs blocking expensive regions. *(Budgets, Organizations, SCP)*

## **🧰 Engineering Practice Projects (Terraform-specific)**

1. **Reusable VPC Module** — Build & publish your own opinionated VPC module with variables, outputs, examples. *(Modules)*
2. **Workspaces for Envs** — dev/stage/prod with per-workspace vars and state isolation. *(Workspaces, tfvars)*
3. **CI/CD for Terraform** — GitHub Actions pipeline: fmt/validate/plan (PR comment) → apply on main. *(CI/CD)*
4. **Policy as Code** — Add Conftest/OPA checks (e.g., “no public S3,” “no 0.0.0.0/0 SSH”). *(OPA, Security)*
5. **Testing with Terratest** — Go tests that apply, verify resources, and destroy. *(Terratest)*
6. **Module Registry + Versioning** — Host a private module registry; semantic version bumps. *(Registry, Versioning)*
7. **Drift Detection** — Scheduled terraform plan with Slack/SNS notifications on drift. *(Automation)*