5-Day DevOps Hands-On Training Curriculum

Training Goal: Deploy Visa Validation Website on AWS EKS with DNS Routing

DAY 1: Linux Fundamentals & GitHub Basics

Session 1 (9:00 AM - 12:00 PM) - Linux Essentials

Theory (30 min)

- Linux file system hierarchy (`/`, `/home`, `/var`, `/opt`)
- Basic commands: `ls`, `cd`, `pwd`, `mkdir`, `rm`, `cp`, `mv`
- File permissions: `chmod`, `chown`, `umask`
- User & group management: `sudo`, `useradd`, `groupadd`

Hands-On Lab (30 min)

- Connect to Lab EC2 instance via SSH
- Navigate file system and create project directory structure
- Practice file permissions and ownership

Checkpoint: Create nested directories with specific permissions

Theory (30 min)

- Process management: `ps`, `top`, `kill`, `systemctl`
- Package managers: `apt`, `yum`
- Text editors: `nano`, `vi` basics

- Finding files: `find`, `grep`, `locate`

Hands-On Lab (30 min)

- Install packages (`curl`, `wget`, `git`)
- View running processes and system resources
- Search and filter files using `find` and `grep`

Checkpoint: Successfully install required tools

Session 2 (1:00 PM - 5:00 PM) - GitHub & Version Control

Theory (30 min)

- Git basics: repository, commit, branch, merge
- GitHub vs Git distinction
- SSH keys and authentication
- Clone, push, pull workflow
- `.gitignore` and `.gitattributes`

Hands-On Lab (30 min)

- Generate SSH keys on Lab instance
- Clone provided Visa Validation app repo from GitHub
- Explore repo structure and branches
- Checkpoint: Successfully clone and navigate repo

Theory (30 min)

- Branching strategy: `main`, `develop`, `feature` branches
- Commit best practices and messages

- Pull requests and code reviews
- Merge conflicts resolution

Hands-On Lab (30 min)

- Create feature branch for Day 1 documentation
- Make commits with meaningful messages
- Create and review a pull request
- Checkpoint: Merge PR to main branch

DAY 2: AWS Basics & EC2 Fundamentals

Session 1 (9:00 AM - 12:00 PM) - AWS Fundamentals

Theory (30 min)

- AWS account structure: regions, availability zones (AZs)
- AWS services overview: compute, storage, networking, database
- IAM basics: users, roles, policies, permissions
- AWS CLI setup and configuration

Hands-On Lab (30 min)

- Access Lab AWS account via Console
- Explore IAM roles and permissions
- Configure AWS CLI on Lab instance
- Test AWS CLI with `aws s3 ls`, `aws ec2 describe-instances`
- Checkpoint: Successfully authenticate via AWS CLI

Theory (30 min)

- EC2 concepts: instances, AMIs, instance types, security groups
- VPC basics: subnets, internet gateway, route tables
- Elastic IPs and DNS
- Key pairs and SSH access

Hands-On Lab (30 min)

- Launch pre-configured EC2 instance (already in Lab)
- Review security groups and inbound/outbound rules
- Connect via SSH using provided key pair
- Checkpoint: SSH into EC2 and run basic commands

Updated DAY 2: AWS Basics, EC2 Fundamentals & Jenkins CI/CD

Session 2 (1:00 PM - 5:00 PM) - EC2 Operations, Networking & Jenkins Setup

Theory (30 min)

- EC2 lifecycle: running, stopped, terminated
- Elastic Block Store (EBS): volumes, snapshots
- Security groups: stateful firewalls
- Network ACLs vs Security Groups
- EC2 metadata and user data scripts
- Jenkins CI/CD basics: pipeline, jobs, stages
- Jenkins architecture: master/agent model
- Jenkins on EC2: installation, configuration, security

Hands-On Lab (30 min)

- Stop and start EC2 instance, observe state changes
- Create and attach an EBS volume
- Modify security group rules (add HTTP, HTTPS, and Jenkins port 8080)
- View EC2 metadata: curl http://169.254.169.254/latest/meta-data/
- Install Jenkins on EC2 instance
- Configure Jenkins initial setup and admin user
- Create Jenkins security group inbound rule for port 8080
- Checkpoint: Successfully access Jenkins via http://<EC2-IP>:8080

DAY 3: Docker Containerization

Session 1 (9:00 AM - 12:00 PM) - Docker Basics

Theory (30 min)

- Containerization vs VMs
- Docker architecture: daemon, CLI, registry, images, containers
- Dockerfile: layers, instructions (`FROM`, `RUN`, `COPY`, `EXPOSE`, `CMD`, `ENTRYPOINT`)
- Image vs Container lifecycle
- Docker registries: Docker Hub, AWS ECR

Hands-On Lab (30 min)

- Verify Docker is installed on Lab instance
- Run pre-built Docker containers: `docker run -it alpine`
- Explore running containers: `docker ps`, `docker inspect`, `docker logs`
- View Docker images: `docker images`

Checkpoint: Successfully run and inspect containers

Theory (30 min)

- Dockerfile best practices: minimize layers, use `.dockerignore`
- Multi-stage builds
- Environment variables in containers
- Volume mounting and bind mounts
- Container networking: bridge, host, overlay

Hands-On Lab (30 min)

- Examine provided Dockerfile for Visa Validation app
- Build Docker image: `docker build -t visa-app:v1 .`
- Run container locally: `docker run -p 8080:3000 visa-app:v1`
- Test app via curl: `curl localhost:8080`

Checkpoint: Build and run app container locally

Session 2 (1:00 PM - 5:00 PM) - Docker Advanced & ECR

Theory (30 min)

- Docker Compose: multi-container orchestration
- `docker-compose.yml` structure: services, volumes, networks
- Environment files and override files
- Docker networking in Compose
- Container logs and debugging

Hands-On Lab (30 min)

- Create `docker-compose.yml` for Visa Validation app (if multi-container)
- Run with `docker-compose up`
- View logs: `docker-compose logs -f`

- Stop and clean up: `docker-compose down`

Checkpoint: Multi-container app running via Compose

Theory (30 min)

- AWS ECR: private Docker registry
- ECR authentication and permissions
- Pushing images to ECR
- Image tagging and versioning
- ECR lifecycle policies

Hands-On Lab (30 min)

- Create ECR repository for Visa Validation app via CLI
- Authenticate Docker daemon to ECR: `aws ecr get-login-password`
- Tag image: `docker tag visa-app:v1 <account-id>.dkr.ecr.<region>.amazonaws.com/visa-app:v1`
- Push to ECR: `docker push <ecr-repo-url>`
- Verify in ECR console

Checkpoint: Image successfully pushed to ECR

DAY 4: Kubernetes (K8s) & EKS Deployment

Session 1 (9:00 AM - 12:00 PM) - Kubernetes Fundamentals

Theory (30 min)

- K8s architecture: control plane, worker nodes, kubelet, etcd
- Key objects: Pods, Deployments, Services, ConfigMaps, Secrets
- YAML manifest structure
- Namespaces and labels/selectors
- kubectl basics: `get`, `describe`, `apply`, `delete`, `logs`

Hands-On Lab (30 min)

- Access pre-configured EKS cluster via kubectl
- View nodes: `kubectl get nodes`
- Explore namespaces: `kubectl get namespaces`
- Create simple Pod YAML and deploy: `kubectl apply -f pod.yaml`

Checkpoint: Successfully deploy a test Pod

Theory (30 min)

- Deployments: replicas, rolling updates, rollback
- ReplicaSets and pod selectors
- Resource requests and limits (CPU, memory)
- Probes: liveness, readiness, startup
- Deployment strategies: rolling, recreate, blue-green

Hands-On Lab (30 min)

- Create Deployment manifest for Visa Validation app
- Specify resource limits and requests
- Deploy: `kubectl apply -f deployment.yaml`
- Scale replicas: `kubectl scale deployment visa-app --replicas=3`
- Check pod status: `kubectl get pods`

Checkpoint: Deployment with 3 replicas running

Session 2 (1:00 PM - 5:00 PM) - K8s Services & EKS Advanced

Theory (30 min)

- Services: ClusterIP, NodePort, LoadBalancer
- Service discovery and DNS
- Ingress resources and Ingress controllers
- Ingress rules: path-based, host-based routing
- SSL/TLS termination in Ingress

Hands-On Lab (30 min)

- Create Service manifest (ClusterIP first): `kubectl apply -f service.yaml`
- Test service DNS: `kubectl exec -it <pod> -- curl visa-app:8080`
- Create Ingress resource for external access
- Verify Ingress status: `kubectl get ingress`

Checkpoint: Service and Ingress created, app accessible

Theory (30 min)

- EKS-specific concepts: IAM roles, IRSA (IAM Roles for Service Accounts)
- ConfigMaps and Secrets for app configuration
- Persistent volumes and claims (if needed)
- Monitoring: CloudWatch Container Insights
- EKS logging and debugging

Hands-On Lab (30 min)

- Create ConfigMap for app environment variables
- Apply to Deployment: update manifest and redeploy

- Create Secrets for sensitive data (API keys, DB creds)
- Update Deployment to use Secrets
- View logs: `kubectl logs <pod-name>`

Checkpoint: App using ConfigMaps and Secrets

DAY 5: Routing, Load Balancing & Full Deployment

Session 1 (9:00 AM - 12:00 PM) - Advanced Routing & DNS

Theory (30 min)

- DNS resolution: A records, CNAME, Alias records
- Route 53: hosted zones, routing policies
- Application Load Balancer (ALB) integration with K8s
- AWS Load Balancer Controller for EKS
- Health checks and traffic distribution

Hands-On Lab (30 min)

Configure Route 53 hosted zone (pre-created in Lab)

Create A record pointing to Ingress ALB IP

Test DNS resolution: `nslookup visa-app.yourdomain.com`

Configure health check on Route 53

Checkpoint: DNS resolving to EKS app

Theory (30 min)

- Multi-region and multi-AZ routing with Route 53
- Weighted routing and latency-based routing
- Failover and failover routing

- SSL/TLS certificates: AWS Certificate Manager (ACM)
- Enforcing HTTPS and redirects

Hands-On Lab (30 min)

- Associate ACM certificate with ALB
- Configure HTTPS listener on ALB
- Update Ingress to use HTTPS
- Test HTTPS access: `curl https://visa-app.yourdomain.com`
- Verify SSL certificate: `openssl s_client -connect visa-app.yourdomain.com:443`

Checkpoint: HTTPS working end-to-end

Session 2 (1:00 PM - 5:00 PM) - Full Deployment & Production Readiness

Theory (30 min)

- Deployment best practices: resource limits, health probes, affinity
- Security: RBAC, network policies, pod security policies
- Cost optimization: spot instances, rightsizing
- Backup and disaster recovery
- GitOps principles and continuous deployment

Hands-On Lab (30 min)

- Review final Visa Validation app manifests
- Add readiness/liveness probes to Deployment
- Set resource requests and limits
- Implement network policies (basic)
- Deploy updated manifests

Checkpoint: Production-ready deployment configuration

Theory (30 min)

- Monitoring and alerting: CloudWatch, Prometheus, Grafana overview
- Log aggregation: CloudWatch Logs, ELK
- Cost tracking and optimization
- Troubleshooting checklist: pods, services, ingress, DNS
- Post-deployment validation and testing

Hands-On Lab (30 min)