

Week 1 Step-by-Step Guide — AWS EKS Platform + Secure CI/CD

This guide walks you step by step through building a production-style AWS EKS platform with secure CI/CD using Jenkins and GitHub Actions.

What You Will Have by the End of Week 1

- EKS cluster (dev) provisioned with Terraform (remote state)
- Jenkins running and deploying to the cluster
- GitHub Actions handling PR checks and security scans
- ECR image build and push
- Image signing with Cosign (optional admission policy enforcement)

Day 0 — Setup: Accounts & Local Tooling

0.1 Create AWS CLI Profile

```
aws configure --profile devops
export AWS_PROFILE=devops
aws sts get-caller-identity
```

0.2 Install Tools (Mac)

```
brew install terraform kubectl helm awscli ansible jq trivy cosign
docker version
```

0.3 Create Terraform Remote State (S3 + DynamoDB)

Pick unique names, for example: • S3 bucket: nick-devops-tfstate • DynamoDB table: terraform-locks

```
aws s3api create-bucket --bucket nick-devops-tfstate --region us-east-1

aws dynamodb create-table \
  --table-name terraform-locks \
  --attribute-definitions AttributeName=LockID,AttributeType=S \
  --key-schema AttributeName=LockID,KeyType=HASH \
  --billing-mode PAY_PER_REQUEST

aws s3 ls | grep tfstate
```

```
aws dynamodb describe-table --table-name terraform-locks | jq '.Table.TableStatus'
```

Day 1 — Terraform EKS (Dev Environment)

1.1 Repository Structure

```
platform/  
  infra/terraform/  
    modules/  
      envs/dev/
```

1.2 Required Terraform Files

backend.tf, providers.tf, vpc.tf, eks.tf, outputs.tf, variables.tf

backend.tf

```
terraform {  
  backend "s3" {  
    bucket      = "nick-devops-tfstate"  
    key         = "platform/dev/terraform.tfstate"  
    region     = "us-east-1"  
    dynamodb_table = "terraform-locks"  
    encrypt     = true  
  }  
}
```

Apply Steps

```
cd platform/infra/terraform/envs/dev  
terraform init  
terraform fmt -recursive  
terraform validate  
terraform plan -out tfplan  
terraform apply tfplan  
  
aws eks list-clusters --region us-east-1
```

1.3 Configure kubectl

```
aws eks update-kubeconfig --region us-east-1 --name <cluster-name>  
kubectl get nodes  
kubectl get ns
```

Day 2 — Enterprise Add-ons (Helm)

2.1 Add Helm Repositories

```
helm repo add eks https://aws.github.io/eks-charts
helm repo add metrics-server https://kubernetes-sigs.github.io/metrics-server/
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update
```

2.2 Install Metrics Server

```
helm upgrade --install metrics-server metrics-server/metrics-server -n kube-system
kubectl get deployment metrics-server -n kube-system
```

2.3 Install AWS Load Balancer Controller

Steps: 1. Create IAM policy 2. Create IRSA role 3. Install Helm chart

```
kubectl get pods -n kube-system | grep aws-load-balancer
```

Day 3 — Containerize & Deploy Sample App

```
docker build -t demo-app:1.0 .

aws ecr create-repository --repository-name demo-app --region us-east-1

aws ecr get-login-password --region us-east-1 | \
docker login --username AWS --password-stdin <acct>.dkr.ecr.us-east-1.amazonaws.com

docker tag demo-app:1.0 <acct>.dkr.ecr.us-east-1.amazonaws.com/demo-app:1.0
docker push <acct>.dkr.ecr.us-east-1.amazonaws.com/demo-app:1.0

kubectl create ns demo
helm upgrade --install demo-app ./charts/demo-app -n demo
kubectl get pods -n demo
kubectl get svc -n demo
```

Day 4 — GitHub Actions

Pipeline should include: • terraform fmt / validate • docker build • trivy scan • SBOM generation

Day 5 — Jenkins

Recommended setup: Jenkins on EC2. Install docker, terraform, kubectl, helm, trivy, cosign.

Pipeline stages: • checkout • terraform plan/apply (dev) • build + push • sign image • helm deploy

Day 6 — Break / Fix Day

Break: • Readiness probe path • IAM permissions • Security group rules

Fix using kubectl describe, kubectl logs. Document symptom → root cause → fix.

Day 7 — Documentation & Interview Script

Deliverables: • Architecture diagram • CI/CD flow diagram • Runbook (deploy & rollback) • 2-minute explanation script

Resources

- AWS EKS Workshop
- Terraform AWS Provider Docs
- Kubernetes Official Docs
- Jenkins Pipeline Docs
- Trivy & Cosign Docs