

Project Bedrock — Deployment Guide

Quick Navigation

Section	Description
1. Pipeline Triggers	How to run CI/CD workflows
2. Retail Store URL	Access the running application
3. Grading Credentials	Access Key and Secret Key for grading

1. How to Trigger the Pipeline

The repository uses GitHub Actions for Terraform plan, apply, and validation. Pipelines are in `.github/workflows/`.

Pipeline Workflows

Workflow	Trigger	Actions
Terraform Plan & Validation	Pull request to infra/**	terraform fmt, validate, tflint, plan
Terraform Apply	Push to main (infra changes)	terraform init, terraform apply -auto-approve
Terraform (dev/staging/prod)	Manual (Actions → Run workflow)	terraform init, validate, plan

Triggering a Full Deployment

```
# 1. Package Lambda (required before apply)
bash scripts/package_lambda_handler.sh

# 2. Commit and push to main (triggers Apply)
git add -A
git commit -m "Deploy Project Bedrock infrastructure"
git push origin main
```

2. Retail Store Application URL

How to Obtain the URL

The Retail Store UI is exposed via an ALB. After deployment:

```
# Configure kubectl
aws eks update-kubeconfig --region us-east-1 --name project-bedrock-cluster

# Get the Ingress address
kubectl get ingress -n retail-app
```

Use the ADDRESS column (e.g. `http://k8s-retail-xxxxxx-xxx.us-east-1.elb.amazonaws.com`).

The ALB may take 5–15 minutes to become available.

3. Grading Credentials (bedrock-dev-view)

The IAM user **bedrock-dev-view** has read-only AWS access plus EKS view and S3 PutObject.

Retrieving Access Key and Secret Key

Option A — Terraform CLI:

```
cd infra/envs/dev
terraform output bedrock_dev_view_access_key_id
terraform output bedrock_dev_view_secret_access_key
```

Option B — Grading JSON:

```
make output-dev
# Outputs written to infra/grading.json
```

Option C — JSON extract:

```
cd infra/envs/dev
terraform output -json | jq -r '{
  bedrock_dev_view_access_key_id: .bedrock_dev_view_access_key_id.value,
  bedrock_dev_view_secret_access_key: .bedrock_dev_view_secret_access_key.value
}'
```

Field	Example
Access Key ID	AKIA... (20 chars)
Secret Access Key	wJalr... (40 chars)

Resource Nomenclature

Resource	Value
AWS Region	us-east-1
EKS Cluster	project-bedrock-cluster
VPC Name	project-bedrock-vpc
Namespace	retail-app
IAM User	bedrock-dev-view
S3 Bucket	bedrock-assets-ALTSOE025-0347
Lambda	bedrock-asset-processor

Prerequisites & Bootstrap

Install Terraform, AWS CLI, kubectl. Configure AWS credentials. Create S3 bucket `project-bedrock-0347-tf-state` and DynamoDB table `project-bedrock-tf-lock` for Terraform state.

Step-by-Step Deployment

1. Clone repo, package Lambda (`bash scripts/package_lambda_handler.sh`)
 2. `cd infra/envs/dev && terraform init`
 3. `terraform plan -out=tfplan`
 4. `terraform apply tfplan`
 5. Configure kubectl and verify outputs
-

Architecture Overview

```
us-east-1
├─ project-bedrock-vpc (10.0.0.0/16)
│   ├── Public Subnets + NAT
│   └─ Private Subnets (EKS)
│       └─ project-bedrock-cluster
│           └─ retail-app namespace
├─ bedrock-assets-ALTS0E025-0347 → bedrock-asset-processor (Lambda)
└─ IAM: bedrock-dev-view
```