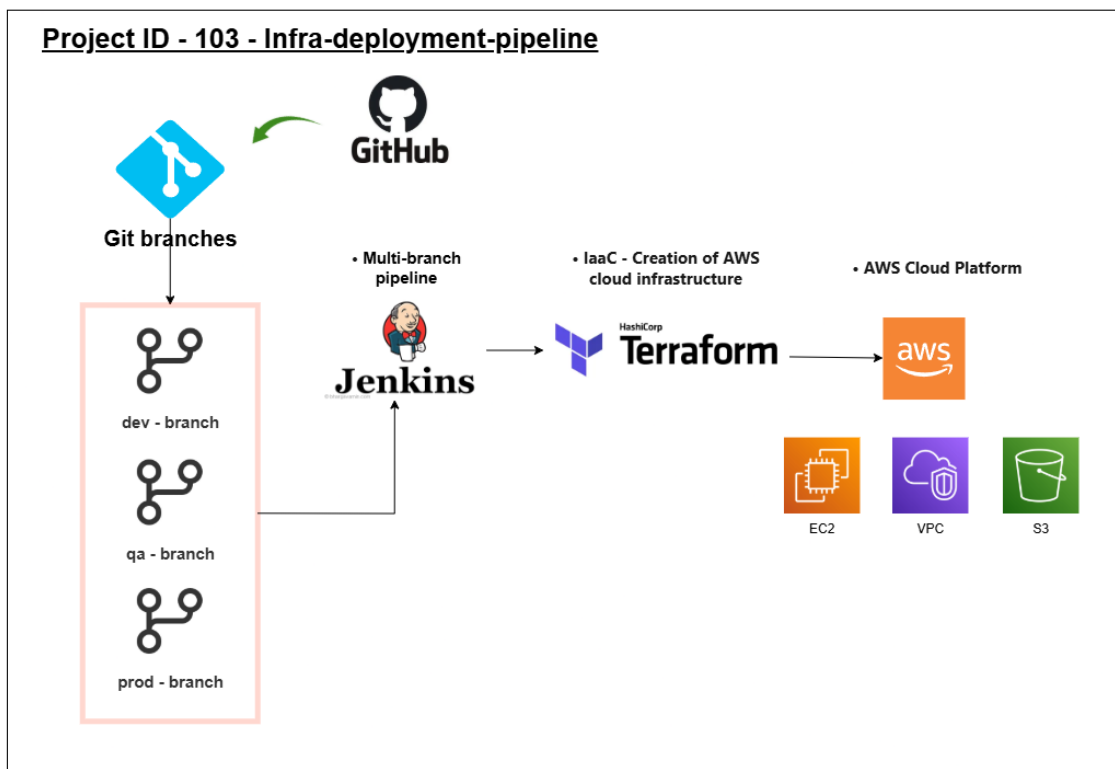


Project ID – 103 – Infra-deployment-pipeline

This project automates the deployment of AWS infrastructure using a CI/CD pipeline with Jenkins and Terraform, following a multi-environment branching strategy.

Project ID - 103: Infra-deployment-pipeline automates AWS infrastructure provisioning using a CI/CD pipeline. Code is managed in GitHub with separate branches for dev, QA, and prod. Jenkins triggers Terraform scripts through a multi-branch pipeline. The setup provisions AWS resources like EC2, VPC, and S3 using Infrastructure as Code.



Components:

- **Source Code Management:**
 - Code is maintained in **GitHub** with separate branches for:
 - dev branch
 - qa branch
 - prod branch
- **CI/CD Pipeline:**
 - **Jenkins** is used to set up a **multi-branch pipeline**, automatically triggering jobs based on the branch pushed to GitHub.
- **Infrastructure as Code (IaC):**
 - **Terraform** is used to define and provision AWS infrastructure.

- **Cloud Provider:**

- **AWS Cloud Platform** is the target environment for deployment.
- Resources managed include:
 - **EC2** (Virtual servers)
 - **VPC** (Virtual network)
 - **S3** – (Virtual Storage)

Base EC2:

t2.medium -> installed with Jenkins and Terraform - admin access - IAM role – SG – relevant port number allowed.



103-infra-deployment-pipeline

Branches (2)

Pull Requests (0)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	dev	19 min #14	1 hr 24 min #8	6 min 27 sec	▶
✓	☀	qa	5 min 10 sec #1	N/A	2 min 56 sec	▶

The screenshot shows a code editor with the following structure:

- EXPLORER:**
 - INFRADEPLOYMENT PIPELINE
 - environments
 - dev
 - backend.tf
 - main.tf
 - variables.tf
 - prod
 - qa
 - backend.tf
 - main.tf
 - variables.tf
 - modules
 - ec2
 - main.tf
 - variables.tf
 - s3
 - main.tf
 - variables.tf
 - vpc
 - main.tf
 - variables.tf
 - Jenkinsfile

- main.tf (environments > dev > main.tf):**

```

1 module "vpc" {
2   source = "../../modules/vpc"
3   vpc_cidr = "10.0.0.0/16"
4   subnet_cidr = "10.0.1.0/24"
5   subnet_az = "${var.aws_region}a"
6   env = "dev"
7 }
8
9 module "ec2" {
10  source = "../../modules/ec2"
11  ami_id = "ami-0af9569868786b23a"
12  instance_type = "t2.micro"
13  subnet_id = module.vpc.subnet_id

```
- TERMINAL:**

```

acer@DESKTOP-QP9QP51 MINGW64 ~/Desktop/InfraDeployment pipeline (dev)
• $ git add environments/dev
acer@DESKTOP-QP9QP51 MINGW64 ~/Desktop/InfraDeployment pipeline (dev)
• $ git add Jenkinsfile
acer@DESKTOP-QP9QP51 MINGW64 ~/Desktop/InfraDeployment pipeline (dev)
• $ git status -s
A Jenkinsfile
A environments/dev/backend.tf
A environments/dev/main.tf
A environments/dev/variables.tf
acer@DESKTOP-QP9QP51 MINGW64 ~/Desktop/InfraDeployment pipeline (dev)
• $ git commit -m "added dev"
[dev ed728f7] added dev
4 files changed, 93 insertions(+)
create mode 100644 Jenkinsfile
create mode 100644 environments/dev/backend.tf

```

Github – 3 branches

The screenshot shows the GitHub repository page for '103-infradeployment-pipeline'. The repository is public and has 3 branches (master, dev, qa) and 0 tags. The 'dev' branch has recent pushes 59 minutes ago, and the 'qa' branch has recent pushes 45 minutes ago. The repository has 0 stars, 0 watching, and 0 forks. The 'About' section is empty. The 'Releases' section shows no releases published. The 'Packages' section shows no packages published. The 'Languages' section shows HCL at 100.0%. The 'Add a README' section is visible.

Triggers pipeline from Jenkins

The screenshot shows the Jenkins 'Webhooks' settings page for the '103-infradeployment-pipeline'. The page has a sidebar with navigation links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The 'Webhooks' section is active, showing a list of webhooks. One webhook is listed with the URL 'http://54.172.184.58:8080/github-w... (push)' and a status of 'Last delivery was successful'. The 'Add webhook' button is visible.

The screenshot shows the 'Tables (2)' page in a database management tool. The page has a search bar and filters for 'Any tag key' and 'Any tag value'. The table list shows two tables: '103-infradeployment-lock-table' and '103-infradeployment-lock-table-2'. Both tables are 'Active' and have a 'LockID (S)' partition key. The 'Indexes' column shows 0 indexes for both tables. The 'Replication Regions' column shows 0 regions for both tables. The 'Deletion protection' column shows 'Off' for both tables.

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection
103-infradeployment-lock-table	Active	LockID (S)	-	0	0	Off
103-infradeployment-lock-table-2	Active	LockID (S)	-	0	0	Off

The screenshot shows the AWS DynamoDB console interface. The left sidebar contains navigation options like Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. The main content area displays the 'lock-table' item selected from the '103-infradeployment-lock-table'. A green status bar indicates 'Completed - Items returned: 1 - Items scanned: 1 - Efficiency: 100% - RCUs consumed: 2'. Below this, a table titled 'Table: 103-infradeployment-lock-table - Items returned (1)' shows a single item with the key 'LockID (String)' and value '103-infradeployment...' and a digest '4562402f1207bcee81f33b7f9afa5d6a'.

This screenshot shows the AWS DynamoDB console with the '103-infradeployment-lock-table-2' item selected. The status bar shows 'Completed - Items returned: 1 - Items scanned: 1 - Efficiency: 100% - RCUs consumed: 2'. The table 'Table: 103-infradeployment-lock-table-2 - Items returned (1)' displays one item with the key 'LockID (String)' and value '103-infradeployment...' and a digest '28bbb99d6a84b17496e574239336b361'.

DynamoDB – locking the state file to avoid concurrent access from different user

This screenshot shows the AWS DynamoDB console with the '103-infradeployment-lock-table-2' table. The status bar indicates 'Completed - Items returned: 2 - Items scanned: 2 - Efficiency: 100% - RCUs consumed: 2'. The table 'Table: 103-infradeployment-lock-table-2 - Items returned (2)' shows two items. The first item has a key 'LockID ...' and a value '103-infra...' with a digest '(*ID*:9b816c92-9b45-0135-349a-cb166030a819...'. The second item, highlighted with a red box, has a key 'LockID ...' and a value '103-infra...' with a digest '1a85fb039...'.

EC2

Amazon S3

Buckets

103-infradeployment-project

Info

Objects

Metadata

Properties

Permissions

Metrics

Management

Access Points

Objects (1)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Show versions

< 1 >

Name

Type

Last modified

Size

Storage class

environments/

Folder

-

-

-

EC2

Amazon S3

Buckets

103-infradeployment-project

environments/

Copy S3 URI

Objects

Properties

Objects (2)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Show versions

< 1 >

Name

Type

Last modified

Size

Storage class

dev/

Folder

-

-

-

qa/

Folder

-

-

-

EC2

Amazon S3

Buckets

103-infradeployment-project

environments/

dev/

Copy S3 URI

Objects

Properties

Objects (1)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Show versions

< 1 >

Name

Type

Last modified

Size

Storage class

terraform.tfstate

tfstate

May 31, 2025, 12:59:07 (UTC+05:30)

12.4 KB

Standard

aws

Search

[Alt+S]

United States (N. Virginia)

bkbala3710

EC2

Amazon S3

Buckets

103-infradeployment-project

environments/

qa/

Copy S3 URI

Objects

Properties

Objects (1)

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Show versions

< 1 >

Name

Type

Last modified

Size

Storage class

terraform.tfstate

tfstate

May 31, 2025, 13:10:01 (UTC+05:30)

12.4 KB

Standard

Dashboard > 103-infra-deployment-pipeline > dev > #11

```
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}
```

Plan: 1 to add, 0 to change, 0 to destroy.

[Pipeline] }

[Pipeline] // dir

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Approval)

[Pipeline] input

Approve the deployment to production?

Deploy or Abort

Dashboard > 103-infra-deployment-pipeline > dev >

View Configuration

Full Stage View

Stages

GitHub

Pipeline Syntax

Stage View

Average stage times: (full run time: ~53s)

	Declarative: Checkout SCM	Checkout	Terraform Init	Terraform Plan	Approval	Terraform Apply
#12 May 31 12:08 No Changes	308ms				48ms	
#11 May 31 12:04 1 commit	654ms	309ms	5s	12s	96ms (paused for 2min 27s)	24s aborted
#10 May 31 12:01 1 commit	361ms	239ms	4s	12s	304ms (paused for 2min 30s)	303ms aborted
#9 May 31 11:56 No Changes	246ms	234ms	4s	12s	98ms (paused for 29s)	5s

Builds

Filter

Today

- #12 6:38 AM
- #11 6:34 AM
- #10 6:31 AM
- #9 6:26 AM
- #8 6:18 AM
- #7 6:10 AM
- #6 6:05 AM

aws

Search [Alt+S]

Asia Pacific (Mumbai) bkbala3710

EC2

EC2 > Instances

Instances (4) Info

Last updated 4 minutes ago

Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

Name	Instance ID	Instance state	Instanc...	Status check
qa-ec2-1	i-09f333b040ea2b7dc	Running	t2.micro	2/2 checks passed
dev-ec2-1	i-0f2f6064e2077621a	Running	t2.micro	2/2 checks passed
qa-ec2-0	i-00e0b8f0cf37d695c	Running	t2.micro	2/2 checks passed
dev-ec2-0	i-0baafb7aee36d82	Running	t2.micro	2/2 checks passed

aws

Search

[Alt+S]

EC2

VPC > Your VPCs

VPC dashboard

EC2 Global View

Filter by VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Your VPCs (4) Info

Last updated less than a minute ago

Actions

Create VPC

Find VPCs by attribute or tag

	Name	VPC ID	State	Block Public...	IPv4 CIDR	IF
<input type="checkbox"/>	qa-vpc	vpc-0456e9181c898c3c0	Available	Off	10.0.0.0/16	-
<input type="checkbox"/>	101-vpc-custom	vpc-026d3cdd14064d657	Available	Off	10.0.0.0/16	-
<input type="checkbox"/>	dev-vpc	vpc-0f5673c4f45a81b61	Available	Off	10.0.0.0/16	-
<input type="checkbox"/>	-	vpc-05225a81fd9b81e54	Available	Off	172.31.0.0/16	-

Jenkins

Balakumaran T

log out

Dashboard > 103-infra-deployment-pipeline > dev

Status

Changes

Build Now

View Configuration

Full Stage View

Stages

GitHub

Pipeline Syntax

dev

Full project name: 103-infra-deployment-pipeline/dev

Stage View

Average stage times: (full run time: ~2min 42s)

	Declarative: Checkout SCM	Checkout	Terraform Init	Terraform Plan	Approval	Terraform Apply
#14 May 31 12:52 3 commits	274ms	272ms	4s	13s	94ms (paused for 23s)	20s
#13 May 31 12:18 1 commit	312ms	281ms	4s	12s	102ms (paused for 3min 44s)	20s
#12 May 31 12:06 No Changes	308ms	334ms	5s	12s	93ms (paused for 3min 7s) aborted	91ms aborted
#11 May 31 12:04 1 commit	654ms	309ms	5s	12s	96ms (paused for 7min 27s)	24s

Jenkins

Balakumaran T

log out

Dashboard > 103-infra-deployment-pipeline > qa

Status

Changes

Build Now

View Configuration

Full Stage View

Stages

GitHub

Pipeline Syntax

qa

Full project name: 103-infra-deployment-pipeline/qa

Stage View

Average stage times: (full run time: ~3min 32s)

	Declarative: Checkout SCM	Checkout	Terraform Init	Terraform Plan	Approval	Terraform Apply
#4 May 31 14:21 1 commit	273ms	248ms	4s	17s	91ms (paused for 57s)	26s
#3 May 31 14:12 1 commit	284ms	236ms	4s	17s	95ms (paused for 53s)	471ms failed
#2 May 31 13:56 1 commit	328ms	311ms	4s	17s	92ms (paused for 3s)	9s
#1 May 31 13:07 No Changes	318ms	267ms	4s	8s	98ms (paused for 2min 16s)	25s

This project streamlines infrastructure deployment by integrating GitHub, Jenkins, and Terraform into a fully automated pipeline. It ensures consistent provisioning across dev, QA, and prod environments. Using Infrastructure as Code enhances scalability, reliability, and version control. Overall, it simplifies AWS resource management and accelerates delivery cycles.