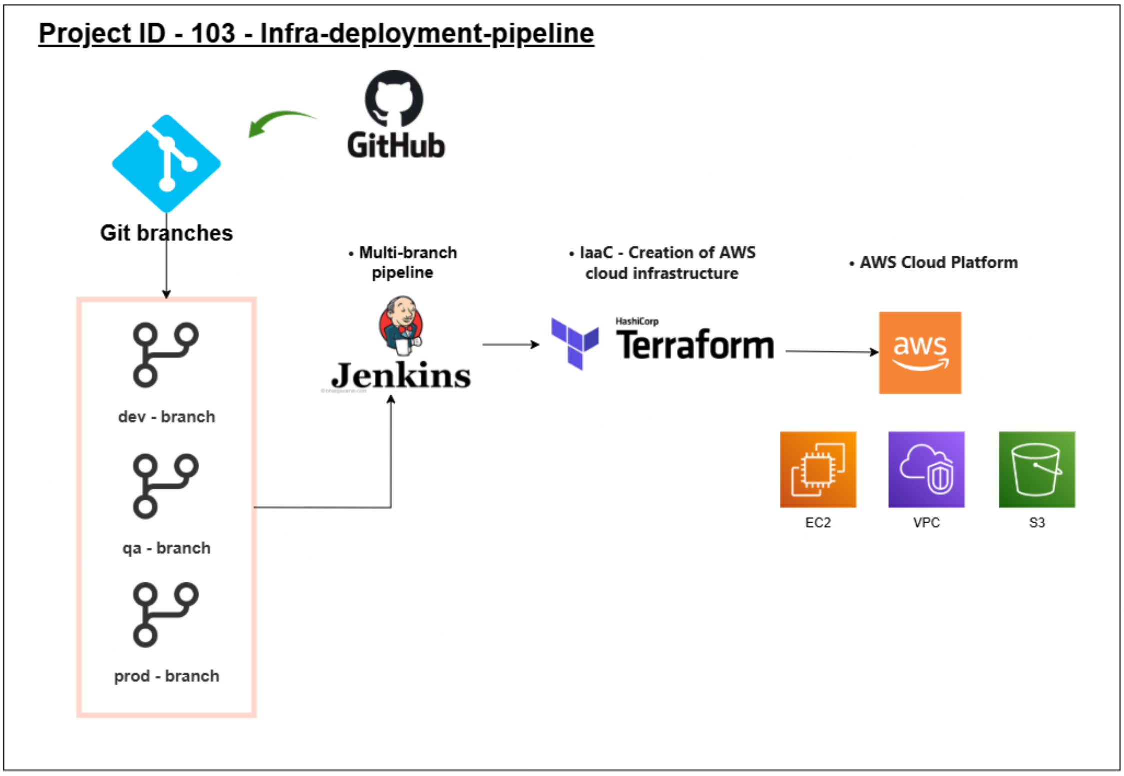
Project ID – 103 – Infra-deployment-pipeline

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.

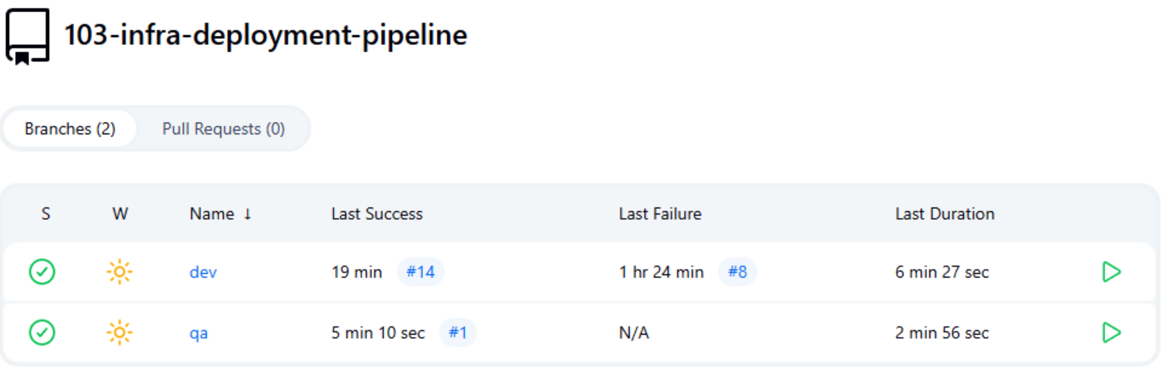


Components:

- **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.

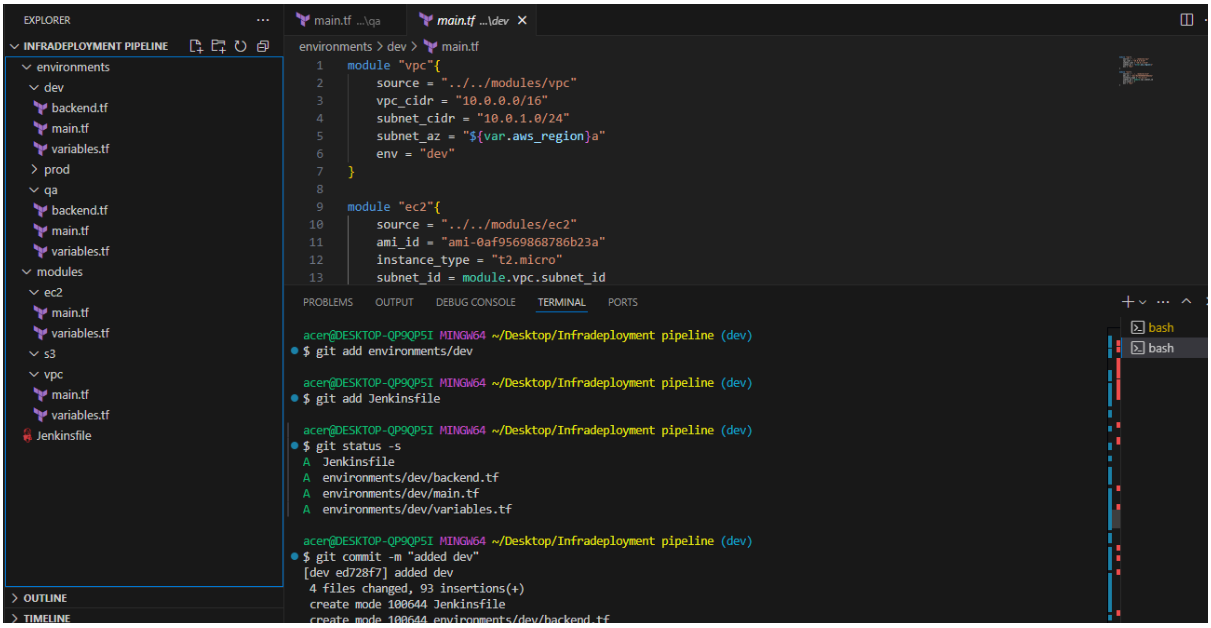
- **Cloud Provider:**
    - **AWS Cloud Platform** is the target environment for deployment.
    - Resources managed include:
        - **EC2** (Virtual servers)
        - **VPC** (Virtual network)
        - **S3 –** (Virtual Storage)

Base EC2:
t2.medium -> installed with Jenkins and Terraform - admin access - IAM role – SG – relevant port number allowed.



103-infra-deployment-pipeline

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



```
module "vpc"{
    source = "../../modules/vpc"
    vpc_cidr = "10.0.0.0/16"
    subnet_cidr = "10.0.1.0/24"
    subnet_az = "${var.aws_region}a"
    env = "dev"
}

module "ec2"{
    source = "../../modules/ec2"
    ami_id = "ami-0af9569868786b23a"
    instance_type = "t2.micro"
    subnet_id = module.vpc.subnet_id
```

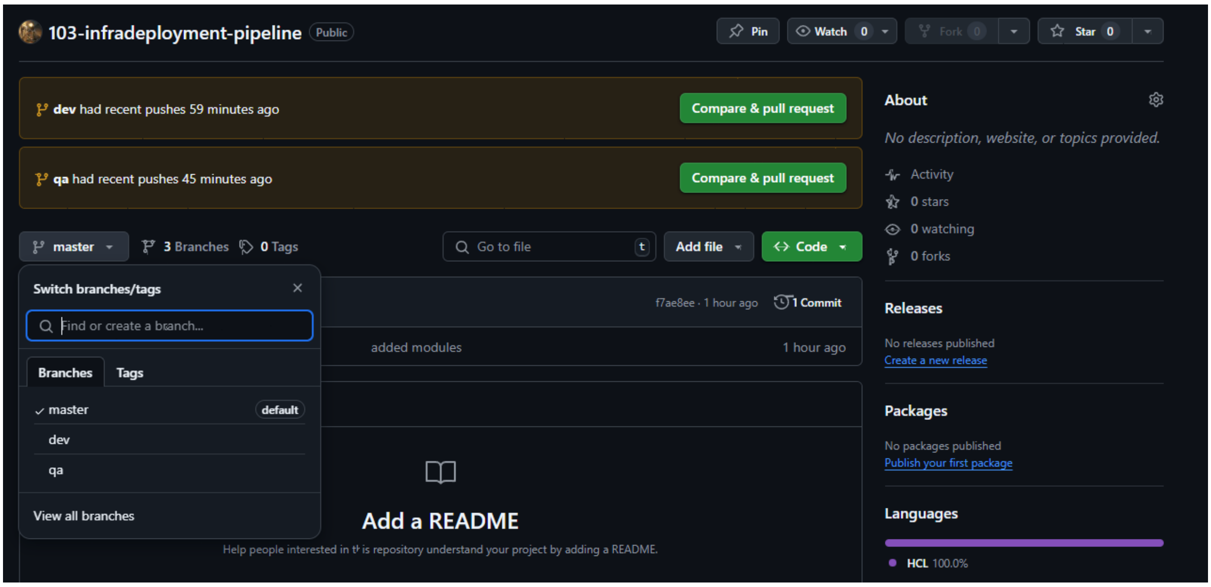
```
acer@DESKTOP-QP9QP5I MINGW64 ~/Desktop/Infradeployment pipeline (dev)
$ git add environments/dev

acer@DESKTOP-QP9QP5I MINGW64 ~/Desktop/Infradeployment pipeline (dev)
$ git add Jenkinsfile

acer@DESKTOP-QP9QP5I 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-QP9QP5I 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



dev had recent pushes 59 minutes ago

qa had recent pushes 45 minutes ago

added modules 1 hour ago

Add a README

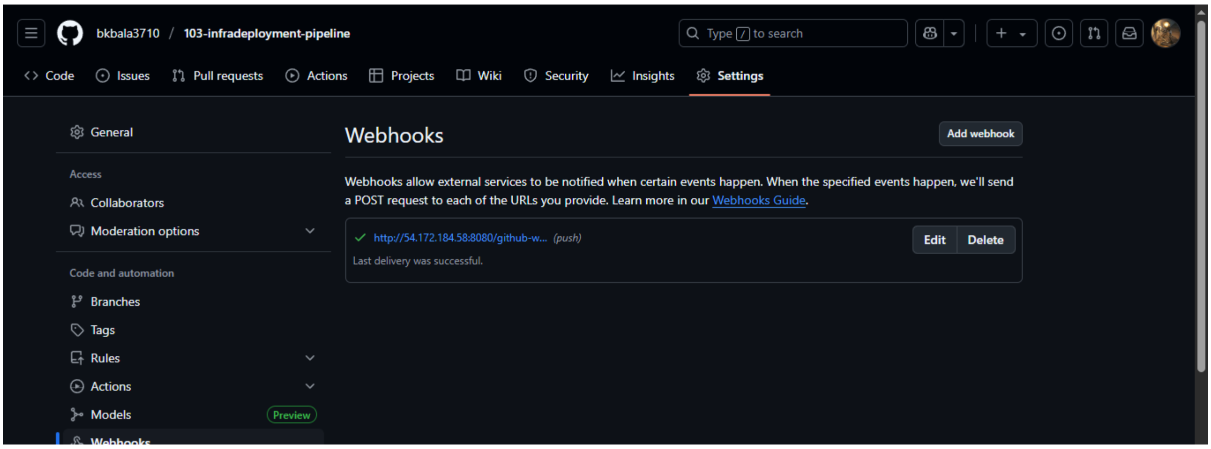
Help people interested in this repository understand your project by adding a README.

No description, website, or topics provided.

No releases published

No packages published

Triggers pipeline from Jenkins

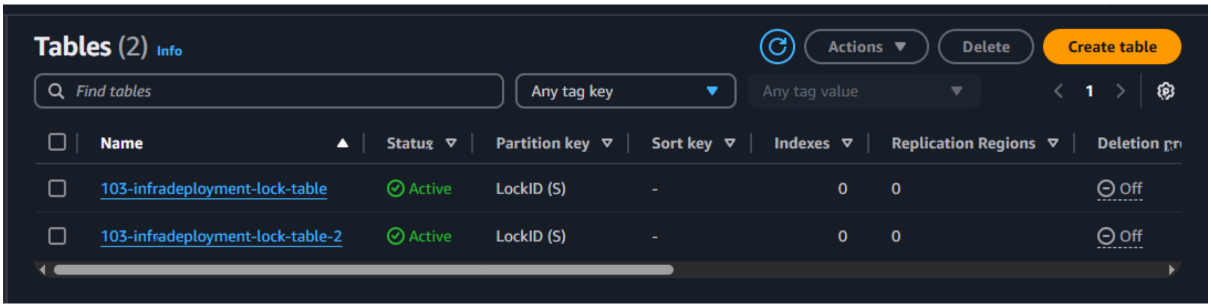


Webhooks

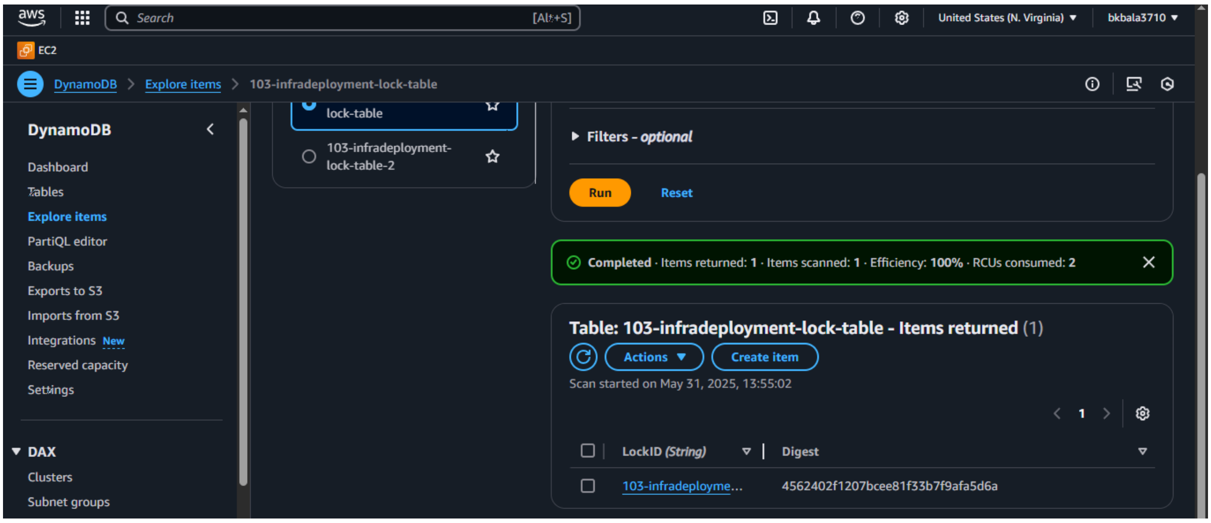
Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://54.172.184.58:8080/github-w... (push)

Last delivery was successful.



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

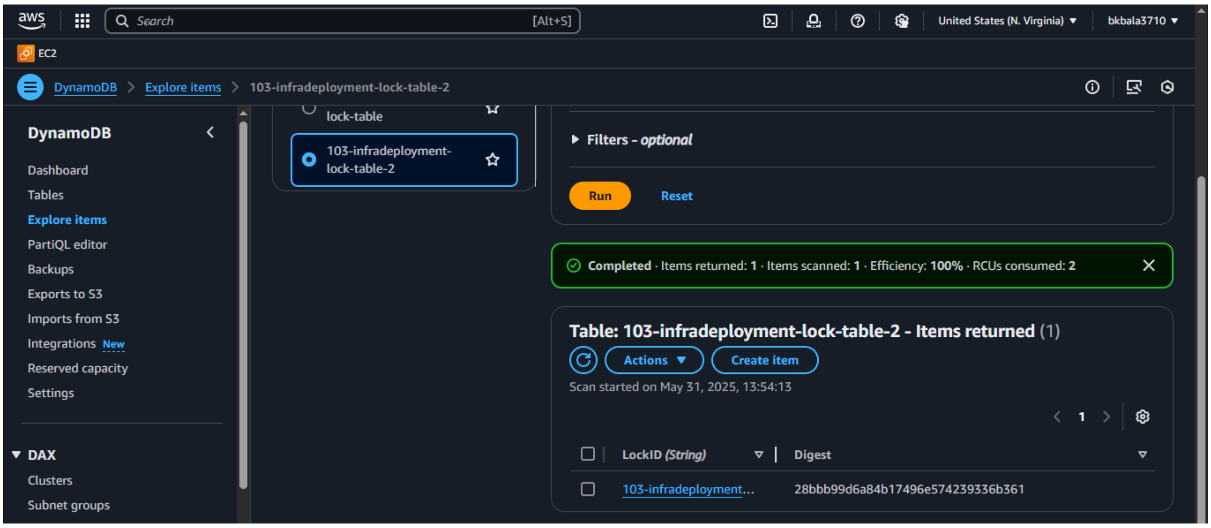


Completed · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 2

Table: 103-infradeployment-lock-table - Items returned (1)

Scan started on May 31, 2025, 13:55:02

□	LockID *(String)*	Digest
□	103-infradeployme...	4562402f1207bcee81f33b7f9afa5d6a



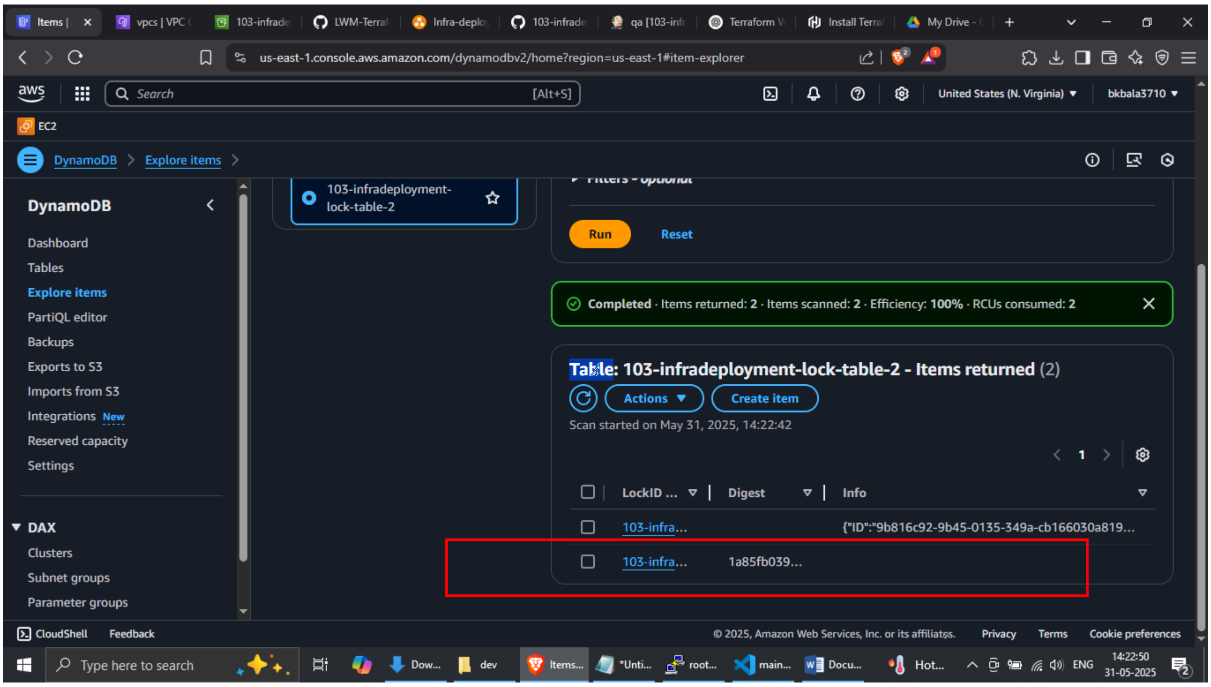
Completed · Items returned: 1 · Items scanned: 1 · Efficiency: 100% · RCUs consumed: 2

Table: 103-infradeployment-lock-table-2 - Items returned (1)

Scan started on May 31, 2025, 13:54:13

□	LockID *(String)*	Digest
□	103-infradeployment...	28bbb99d6a84b17496e574239336b361

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

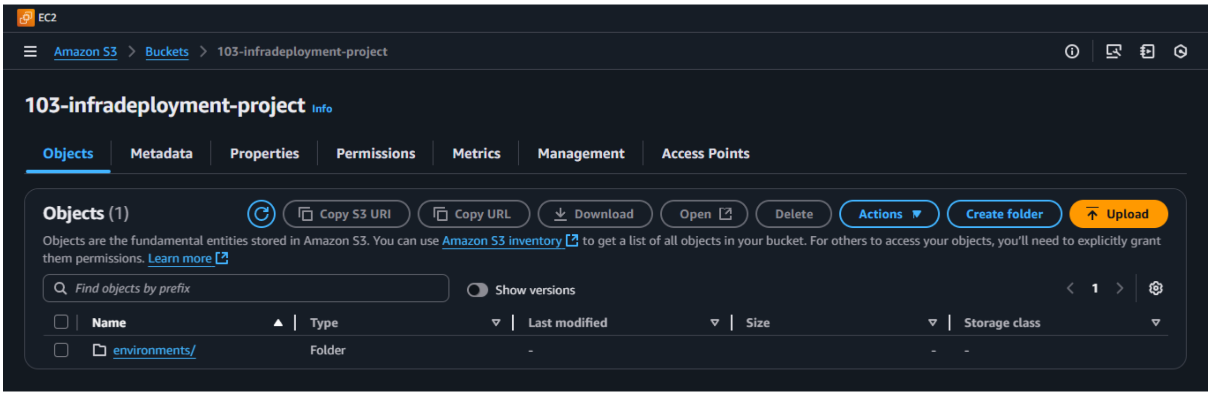


Completed · Items returned: 2 · Items scanned: 2 · Efficiency: 100% · RCUs consumed: 2

Table: 103-infradeployment-lock-table-2 - Items returned (2)

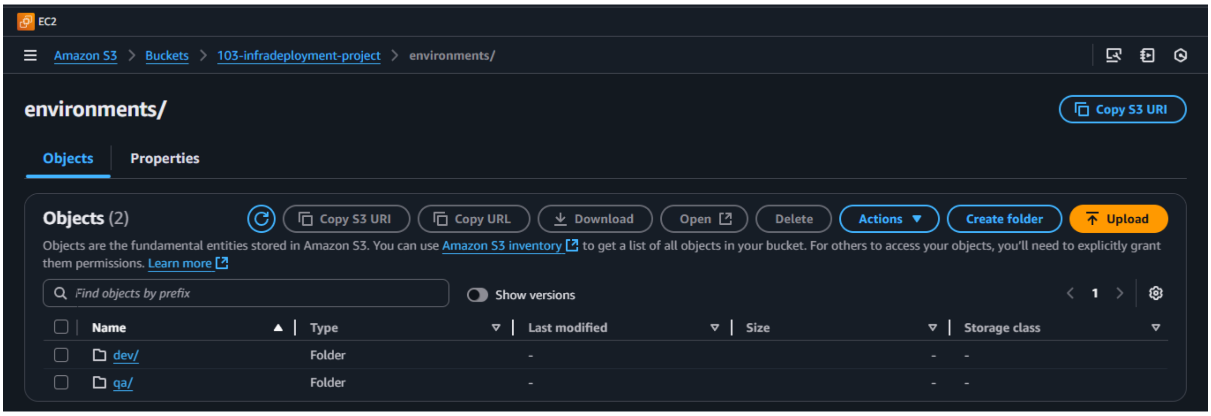
Scan started on May 31, 2025, 14:22:42

□	LockID ...	Digest	Info
□	103-infra...		{"ID":"9b816c92-9b45-0135-349a-cb166030a819...
□	103-infra...	1a85fb039...	



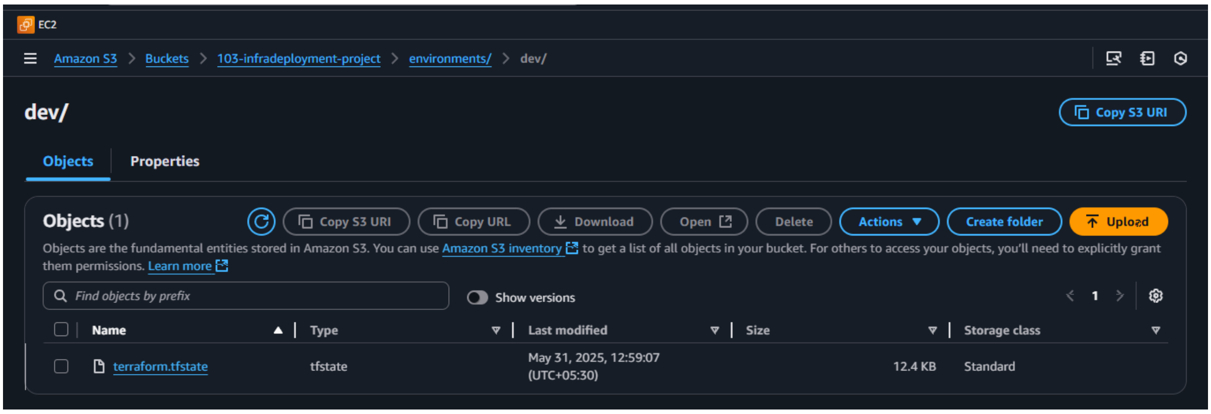
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
environments/	Folder	-	-	-



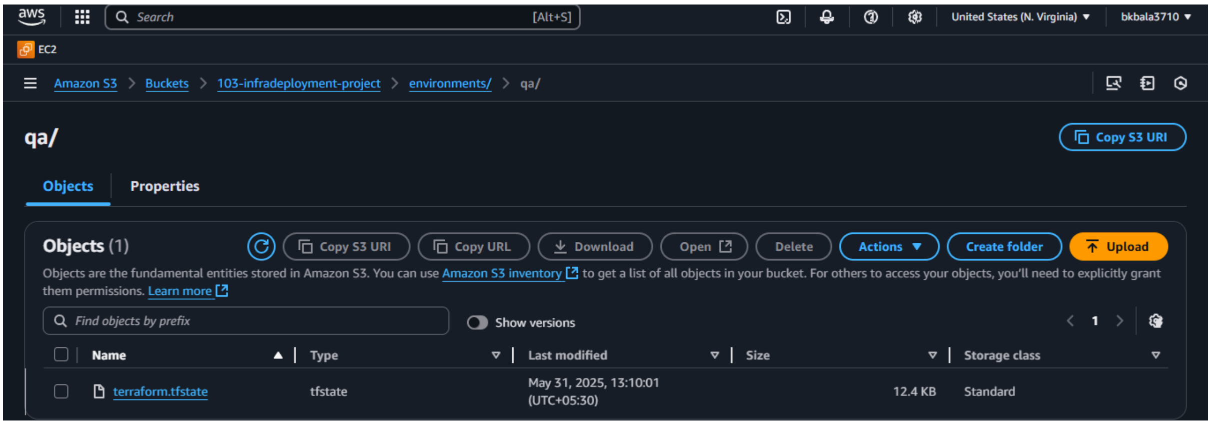
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
dev/	Folder	-	-	-
qa/	Folder	-	-	-



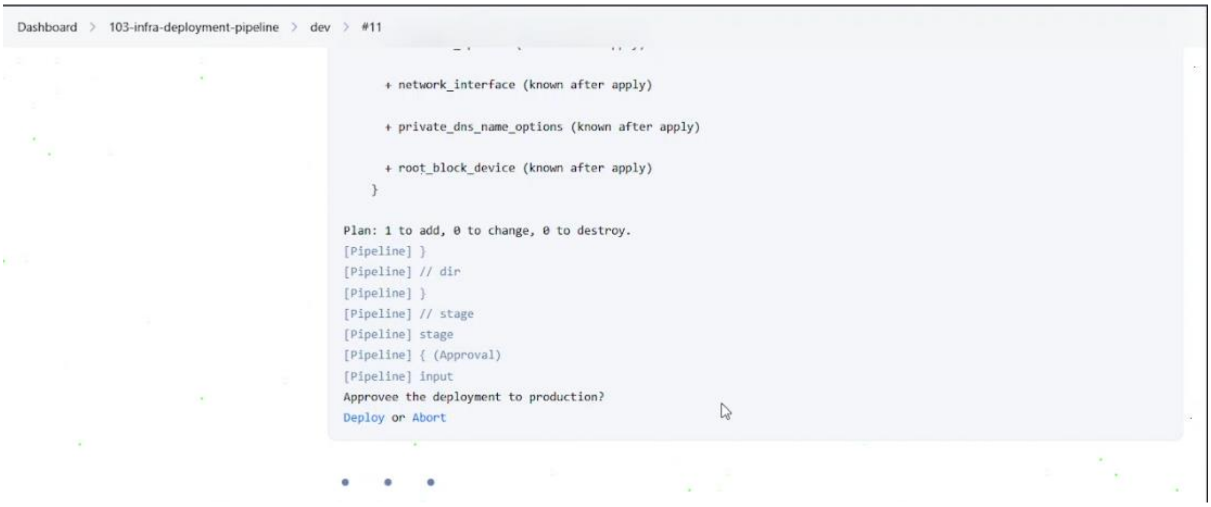
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
terraform.tfstate	tfstate	May 31, 2025, 12:59:07 (UTC+05:30)	12.4 KB	Standard



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

Name	Type	Last modified	Size	Storage class
terraform.tfstate	tfstate	May 31, 2025, 13:10:01 (UTC+05:30)	12.4 KB	Standard

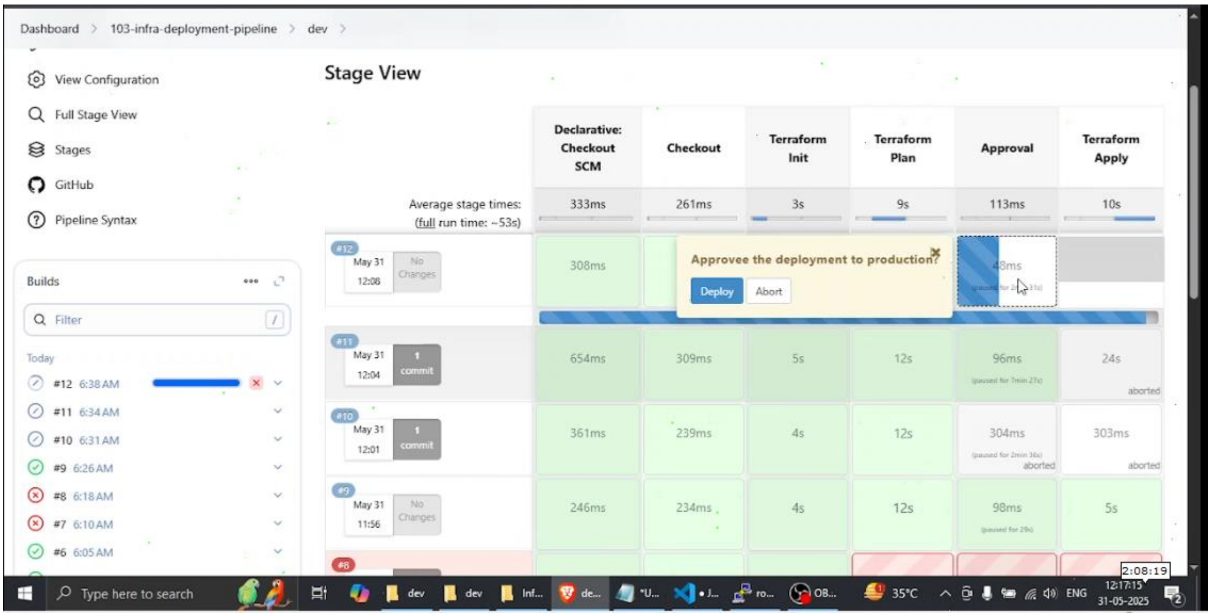


```
      + network_interface (known after apply)

      + 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
Approvee the deployment to production?
Deploy or Abort
```

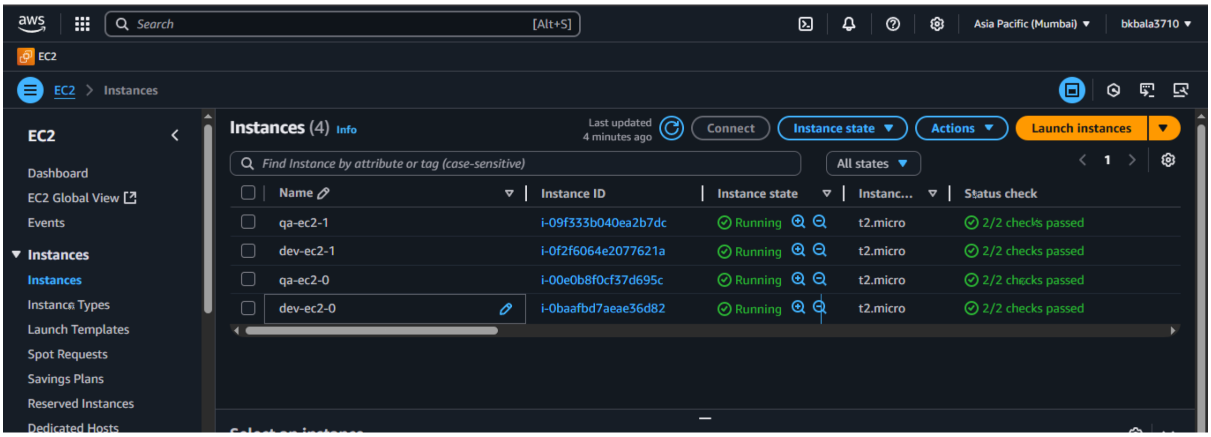


Stage View

	Declarative: Checkout SCM	Checkout	Terraform Init	Terraform Plan	Approval	Terraform Apply
Average stage times: (full run time: ~53s)	333ms	261ms	3s	9s	113ms	10s
#12 May 31 12:08 No Changes	308ms				48ms	
#11 May 31 12:04 1 commit	654ms	309ms	5s	12s	96ms (paused for 7min 27s)	24s aborted
#10 May 31 12:01 1 commit	361ms	239ms	4s	12s	304ms (paused for 2min 36s) aborted	303ms aborted
#9 May 31 11:56 No Changes	246ms	234ms	4s	12s	98ms (paused for 29s)	5s
#8						

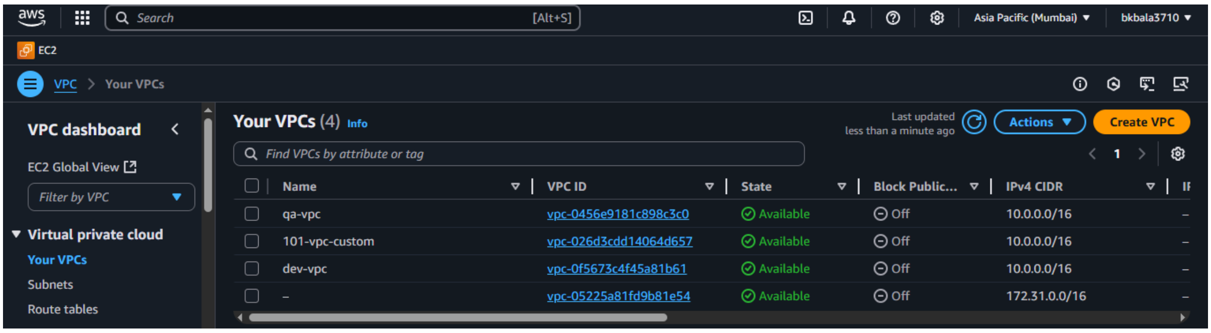
Approvee the deployment to production?

Deploy Abort



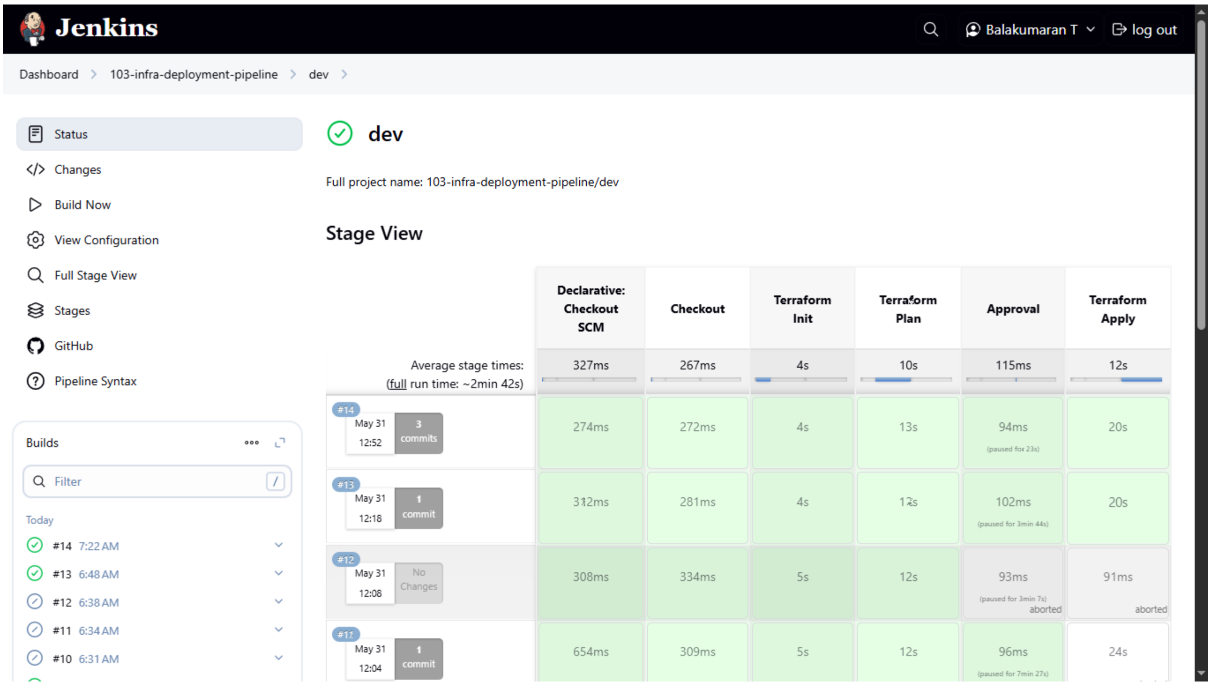
Instances (4) Info

□	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-0baafbd7aeae36d82	✓ Running	t2.micro	✓ 2/2 checks passed



Your VPCs (4) Info

□	Name	VPC ID	State	Block Public...	IPv4 CIDR	IF
□	qa-vpc	vpc-0456e9181c898c3c0	✓ Available	⊖ Off	10.0.0.0/16	–
□	101-vpc-custom	vpc-026d3cdd14064d657	✓ Available	⊖ Off	10.0.0.0/16	–
□	dev-vpc	vpc-0f5673c4f45a81b61	✓ Available	⊖ Off	10.0.0.0/16	–
□	–	vpc-05225a81fd9b81e54	✓ Available	⊖ Off	172.31.0.0/16	–



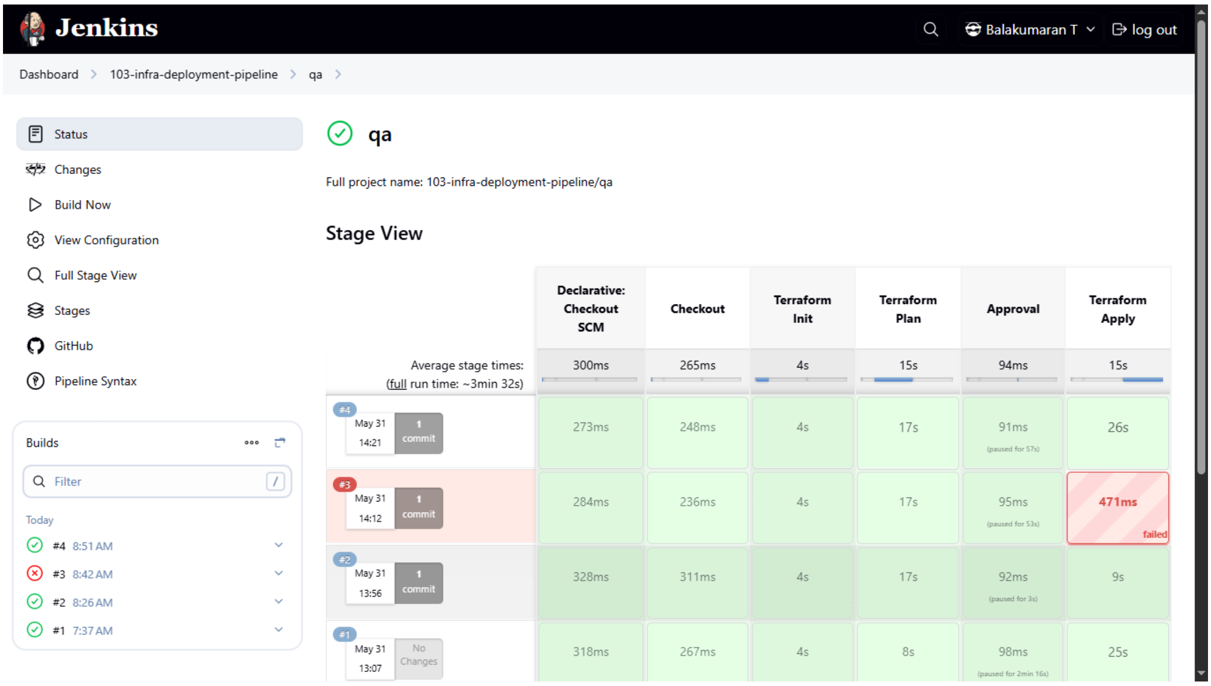
✓ dev

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

Stage View

	Declarative: Checkout SCM	Checkout	Terraform Init	Terraform Plan	Approval	Terraform Apply
Average stage times: (full run time: ~2min 42s)	327ms	267ms	4s	10s	115ms	12s
#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 12:08 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

Builds:
- ✓ #14 7:22 AM
- ✓ #13 6:48 AM
- ⊘ #12 6:38 AM
- ⊘ #11 6:34 AM
- ⊘ #10 6:31 AM



✓ qa

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

Stage View

	Declarative: Checkout SCM	Checkout	Terraform Init	Terraform Plan	Approval	Terraform Apply
Average stage times: (full run time: ~3min 32s)	300ms	265ms	4s	15s	94ms	15s
#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

Builds:
- ✓ #4 8:51 AM
- ⊗ #3 8:42 AM
- ✓ #2 8:26 AM
- ✓ #1 7:37 AM

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.