GitLab CI/CD Pipeline for Terraform Infrastructure Management

Overview

The pipeline manages the full lifecycle of Terraform infrastructure:

- **Validation** of configuration
- **Planning** infrastructure changes
- **Applying** changes
- **Destroying** resources when needed

Each stage is executed in an isolated GitLab CI job with secure AWS credentials and Terraform backend configured on S3.

Pre-requisites

1. AWS Resources

- An **S3 bucket** for remote Terraform state (must be globally unique)
- (Optional) Enable **versioning** on the bucket to track tfstate changes
- IAM user with appropriate permissions:
  - `s3:*` on the Terraform bucket
  - EC2, VPC, IAM, etc., permissions as per the infra

2. GitLab Configuration

- **GitLab Runner**: Use a shell or Docker runner with internet access
- **CI/CD Variables**:
  - `AWS_ACCESS_KEY_ID`
  - `AWS_SECRET_ACCESS_KEY`
  - `AWS_DEFAULT_REGION` (e.g., `ap-south-1`)

Terraform Configuration

```:

terraform {
  backend "s3" {
    bucket = "your-terraform-bucket"
    key    = "Gitlab/terraform.tfstate"
    region = "ap-south-1"
```

```
  }
}
```

``:

```
provider "aws" {
  region = "ap-south-1"
}

resource "aws_instance" "web" {
  ami           = "ami-0abcdef1234567890" # Update with region-specific AMI
  instance_type = "t2.micro"
  tags = {
    Name = "GitLab-TF-Demo"
  }
}
```

GitLab CI/CD Pipeline: `.gitlab-ci.yml`

```yaml
stages:
  - validate
  - plan
  - apply
  - destroy

image:
  name: hashicorp/terraform:light
  entrypoint:
    - '/usr/bin/env'
    - 'PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin'

before_script:
  - export AWS_ACCESS_KEY_ID=${AWS_ACCESS_KEY_ID}
  - export AWS_SECRET_ACCESS_KEY=${AWS_SECRET_ACCESS_KEY}
  - export AWS_DEFAULT_REGION=${AWS_DEFAULT_REGION:-ap-south-1}

validate:
  stage: validate
  script:
    - terraform init
    - terraform validate

plan:
  stage: plan
  script:
    - terraform init
    - terraform plan -out=tfplan
  artifacts:
```

```yaml
    paths:
      - tfplan
    expire_in: 1 hour

apply:
  stage: apply
  script:
    - terraform init
    - |
      if terraform apply -auto-approve tfplan; then
        echo "Applied tfplan successfully."
      else
        echo "tfplan is stale. Re-planning and applying fresh."
        terraform plan -out=tfplan
        terraform apply -auto-approve tfplan
      fi
  dependencies:
    - plan

destroy:
  stage: destroy
  script:
    - terraform init
    - terraform destroy -auto-approve
  when: manual
  dependencies:
    - apply
```

Security Best Practices

- Do **not** hardcode AWS keys in Terraform files or pipeline YAML
- Use **GitLab CI/CD Variables** and mark secrets as “Masked”
- Rotate access keys periodically
- Use **IAM least privilege policies** for CI users

Conclusion

This GitLab CI/CD setup ensures safe and automated Terraform workflows with clear separation of plan, apply, and destroy stages. It leverages AWS S3 for secure remote state and GitLab’s pipeline capabilities for scalable, repeatable infrastructure management.

Ayush-tf Info

Delete

Summary

ARN
arn:aws:iam::269855572870:user/Ayush-tf

Created
July 06, 2025, 19:57 (UTC+05:30)

Console access
Enabled with MFA

Last console sign-in
Never

Access key 1
Create access key

Permissions | Groups | Tags | Security credentials | Last Accessed

Console sign-in

Manage console access

Console sign-in link
https://269855572870.signin.aws.amazon.com/console

Console password
Updated 1 hour ago (2025-07-06 19:57 GMT+5:30)

Last console sign-in
Never

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

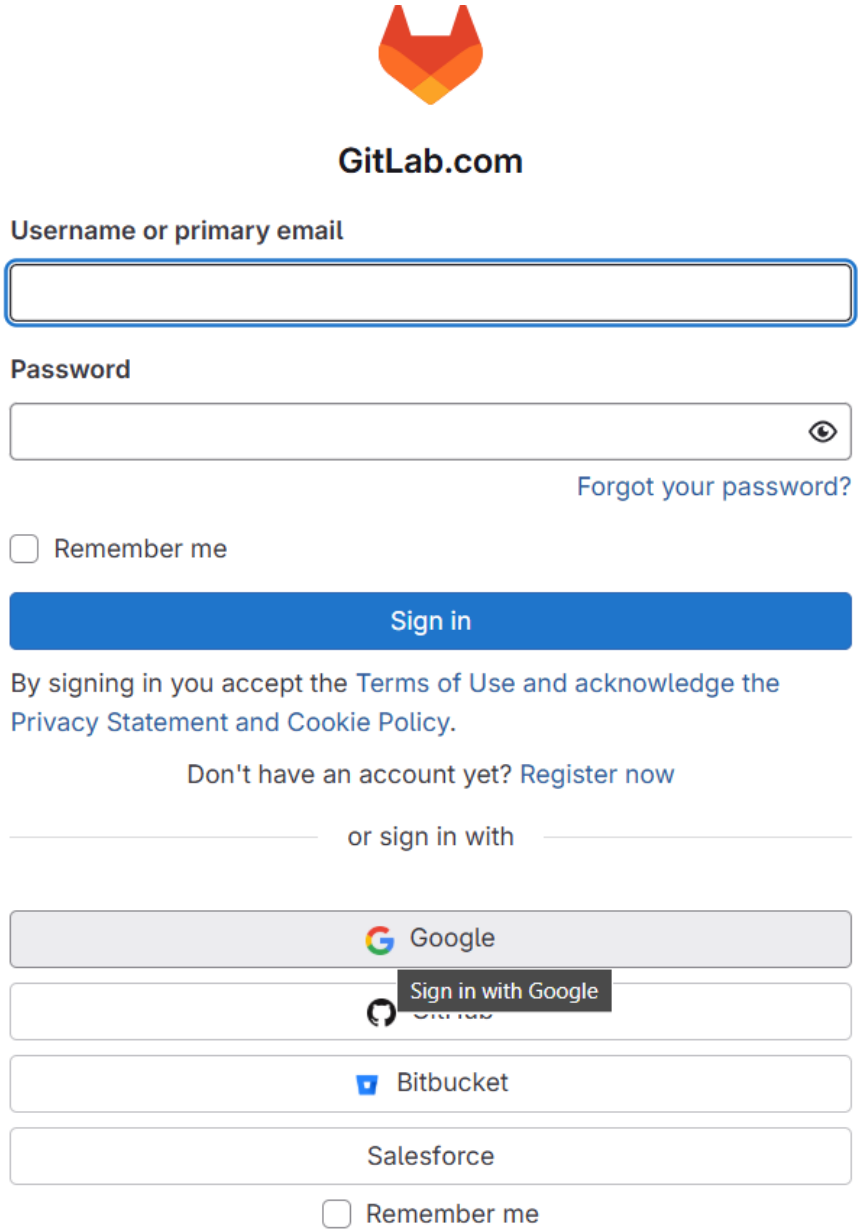
Access key	Secret access key
AKIAT5VFFIODKK33DXN3	Yu8og+t9uy/GEzpd54r1KKNcjziv47wFkNoANqJK Hide

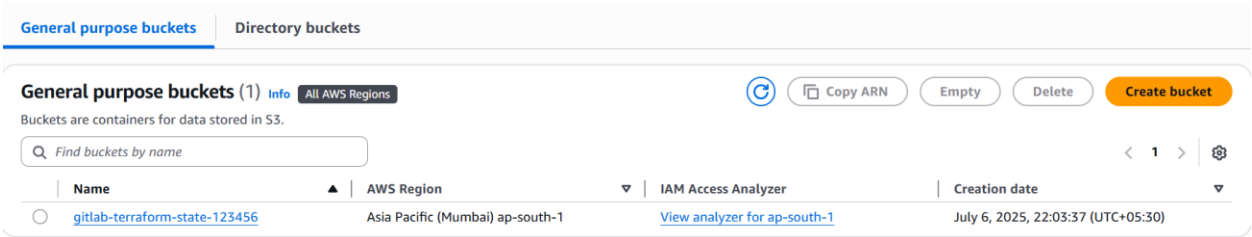
Access key best practices

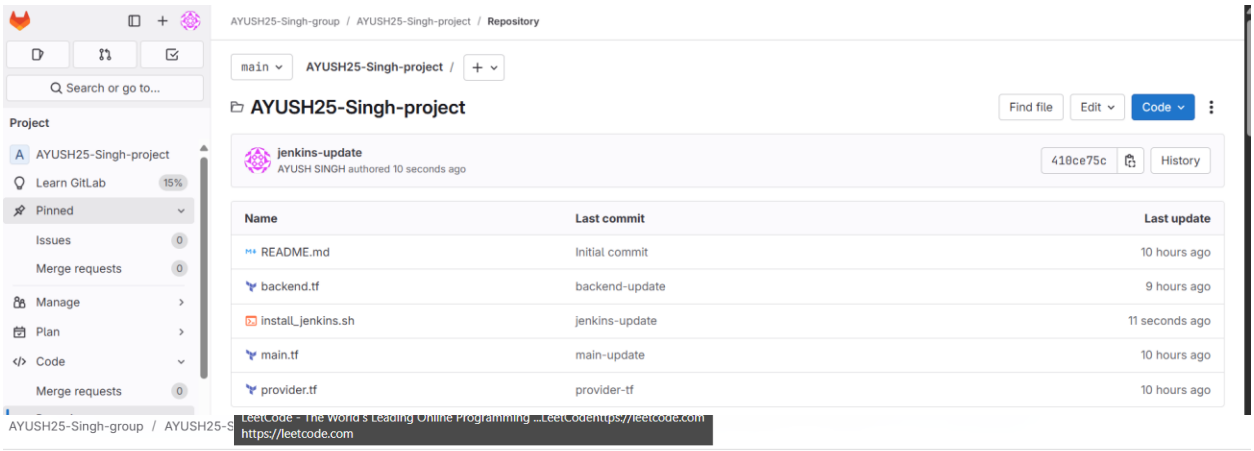
- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the best practices for managing AWS access keys.

Download .csv file | Done


GitLab.com
Username or primary email
Password
Forgot your password?
Remember me
Sign in
By signing in you accept the Terms of Use and acknowledge the Privacy Statement and Cookie Policy.
Don't have an account yet? Register now
or sign in with
Google
Sign in with Google
Bitbucket
Salesforce
Remember me


General purpose buckets
Directory buckets
General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1
Name
AWS Region
IAM Access Analyzer
Creation date
gitlab-terraform-state-123456
Asia Pacific (Mumbai) ap-south-1
View analyzer for ap-south-1
July 6, 2025, 22:03:37 (UTC+05:30)



Runners

Runners are processes that pick up and execute CI/CD jobs for GitLab. What is GitLab Runner?

Artifacts

A job artifact is an archive of files and directories saved by a job when it finishes.

Variables

Variables store information that you can use in job scripts. Each project can define a maximum of 8000 variables. Learn more.

Pipeline trigger tokens

Trigger a pipeline for a branch or tag by generating a trigger token and using it with an API call. The token impersonates a user's project access and permissions. Learn more.

Deploy freezes

Add a freeze period to prevent unintended releases during a period of time for a given environment. You must update the deployment jobs in `.gitlab-ci.yml` according to the deploy freezes added here. Learn more. Specify deploy freezes using cron syntax.

Job token permissions

Control which projects can use CI/CD job tokens to authenticate with this project.

Secure files

Use secure files to store files used by your pipelines such as Android keystores, or Apple provisioning profiles and signing certificates. Learn more

Project variables

Variables can be accidentally exposed in a job log, or maliciously sent to a third party server. The masked variable feature can help reduce the risk of accidentally exposing variable values, but is not a guaranteed method to prevent malicious users from accessing variables. How can I make my variables more secure?

CI/CD Variables 2

Reveal values | Add variable

Key	Value	Environments	Actions
AWS_ACCESS_KEY_ID Protected Masked Expanded	•••••	All (default)	
AWS_SECRET_ACCESS_KEY Protected Masked Expanded	•••••	All (default)	

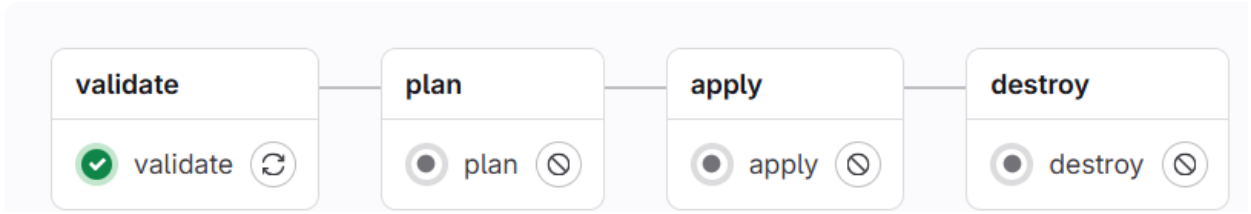
Update .gitlab-ci.yml file

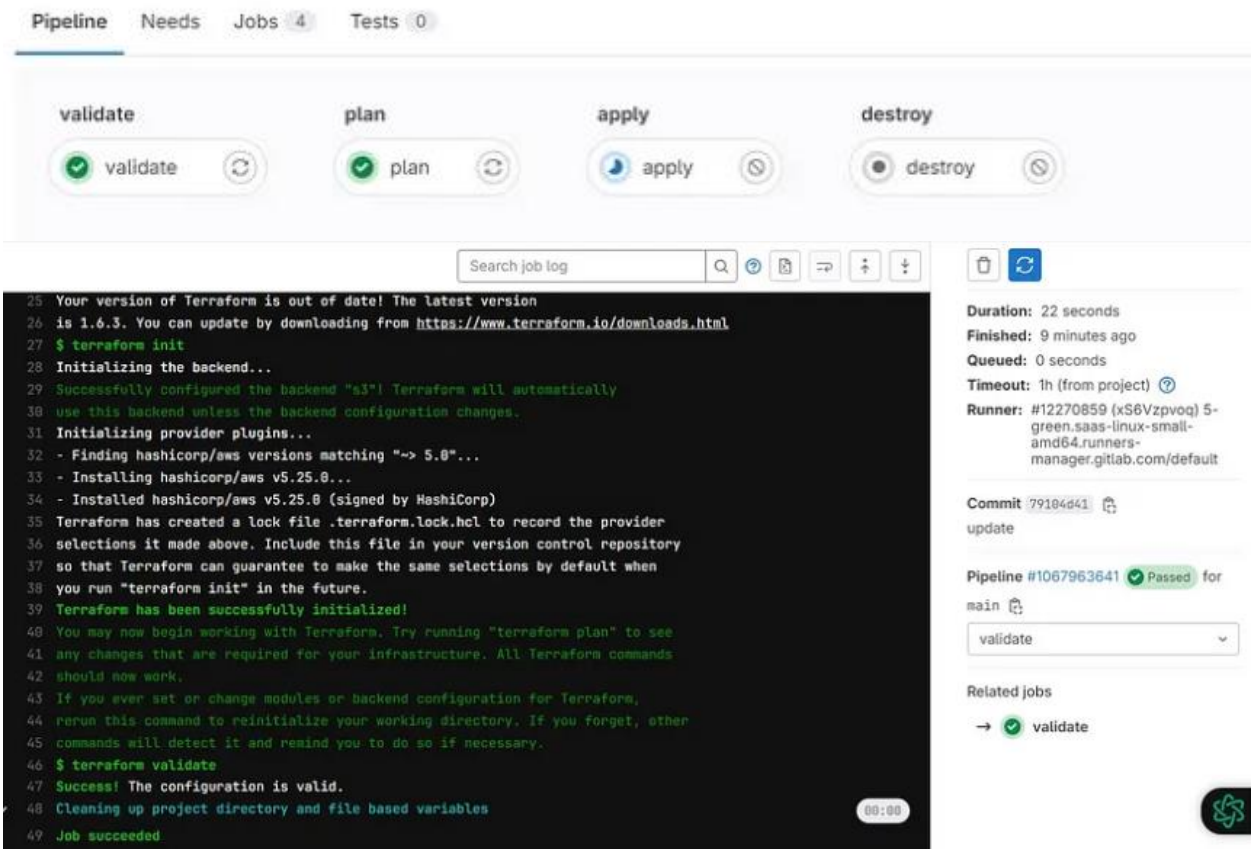
Running **AYUSH SINGH** created pipeline for commit `07c0d48e` 32 seconds ago

For `main`

latest branch 4 jobs In progress, queued for 12 seconds

Pipeline Jobs 4 Tests 0





```
189       + revoke_rules_on_delete = false
190       + tags                   = {
191           + "Name" = "Jenkins-sg"
192         }
193       + tags_all               = {
194           + "Name" = "Jenkins-sg"
195         }
196       + vpc_id                 = (known after apply)
197     }
198 Plan: 2 to add, 0 to change, 0 to destroy.
199
200 Saved the plan to: tfplan
201 To perform exactly these actions, run the following command to apply:
202     terraform apply "tfplan"
203 Uploading artifacts for successful job                                  00:02
204 Uploading artifacts...
205 tfplan: found 1 matching artifact files and directories
206 WARNING: Upload request redirected                 location=https://gitlab.com/api/v4/jobs/5507876896/artifacts?artifact_format=zip&artifact_type=archive new-url=https://gitlab.com
207 WARNING: Retrying...                               context=artifacts-uploader error=request redirected
208 Uploading artifacts as "archive" to coordinator... 201 Created  id=5507876896 responseStatus=201 Created token=64_7p_kS
209 Cleaning up project directory and file based variables                  00:01
210 Job succeeded
```

Duration: 29 seconds
Finished: 10 minutes ago
Queued: 0 seconds
Timeout: 1h (from project)
Runner: #12270859 (xS6Vzpvoq) 5-green.saas-linux-small-amd64.runners-manager.gitlab.com/default

Job artifacts
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

Keep | Download | Browse

Commit 79104d41
update

Pipeline #1067963641 Passed for main

plan

Related jobs

→ plan

EC2 CloudWatch Billing

```
ubuntu@ip-172-31-1-248:~$ cd /
ubuntu@ip-172-31-1-248:/$ ls
Ansible  bin  boot  dev  etc  home  lib  lib32  lib64  libx32  lost+found  media  mnt  opt  proc  root  run  sbin  snap  srv  sys  tmp  usr  var
ubuntu@ip-172-31-1-248:/$ cd Ansible
ubuntu@ip-172-31-1-248:/Ansible$ ls
ANSIBLE
ubuntu@ip-172-31-1-248:/Ansible$ cd ANSIBLE/
ubuntu@ip-172-31-1-248:/Ansible/ANSIBLE$ ls
Jenkins-Ec2  Jenkins-playbook.yml  Jenkinsfile  README.md  ec2.yaml
ubuntu@ip-172-31-1-248:/Ansible/ANSIBLE$
```

i-0d6be153a0b768be9 (Jenkins-sonar)
PublicIPs: 13.234.30.58 PrivateIPs: 172.31.1.248

```
39 selections it made above. Include this file in your version control repository
40 so that Terraform can guarantee to make the same selections by default when
41 you run "terraform init" in the future.
42 Terraform has been successfully initialized!
43 You may now begin working with Terraform. Try running "terraform plan" to see
44 any changes that are required for your infrastructure. All Terraform commands
45 should now work.
46 If you ever set or change modules or backend configuration for Terraform,
47 rerun this command to reinitialize your working directory. If you forget, other
48 commands will detect it and remind you to do so if necessary.
49 $ terraform apply -auto-approve tfplan
50 aws_instance.web: Modifying... [id=i-08f9c60a955133f4f]
51 aws_instance.web: Still modifying... [id=i-08f9c60a955133f4f, 10s elapsed]
52 aws_instance.web: Still modifying... [id=i-08f9c60a955133f4f, 20s elapsed]
53 aws_instance.web: Still modifying... [id=i-08f9c60a955133f4f, 30s elapsed]
54 aws_instance.web: Still modifying... [id=i-08f9c60a955133f4f, 40s elapsed]
55 aws_instance.web: Still modifying... [id=i-08f9c60a955133f4f, 50s elapsed]
56 aws_instance.web: Still modifying... [id=i-08f9c60a955133f4f, 1m0s elapsed]
57 aws_instance.web: Modifications complete after 1m6s [id=i-08f9c60a955133f4f]
58 Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
59 Cleaning up project directory and file based variables
60 Job succeeded
```

00:01

Duration: 1 minute 31 seconds
Finished: 7 minutes ago
Queued: 0 seconds
Timeout: 1h (from project)
Runner: #12270845 (JLgUopmMV) 1-green.saas-linux-small-amd64.runners-manager.gitlab.com/default

Commit b5068bfe
Update install_jenkins.sh

Pipeline #1067196920 Passed for main

apply

Related jobs

→ apply

Instances (1/1) Info

Connect | Instance state | Actions | Launch instances

Find Instance by attribute or tag (case-sensitive)

Instance state = running | Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
Jenkins-sonar	i-08f9c60a955133f4f	Running	t2.medium	Initializing	No alarms	ap-south-1b	ec2-13-233-33-

Instance: i-08f9c60a955133f4f (Jenkins-sonar)

```
TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [Update all packages to their latest version] ****************************
changed: [localhost]

TASK [download jenkins key] ****************************************************
changed: [localhost]

TASK [Add Jenkins repo] ********************************************************
changed: [localhost]

TASK [Update all packages to their latest version] ****************************
ok: [localhost]

TASK [Install fontconfig] ******************************************************
changed: [localhost]

TASK [Install java] ************************************************************
changed: [localhost]

TASK [Install the Jenkins] *****************************************************
changed: [localhost]

TASK [Make sure a service unit is running] *************************************
ok: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=9    changed=6    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-1-248:/var/log$
```

aws Services Search [Alt+S]

EC2 CloudWatch Billing

```
ubuntu@ip-172-31-1-248:/$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
f3d7e7f9f9c24e48a894737f2aa64265
ubuntu@ip-172-31-1-248:/$
```

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

```
** Token Macro
Build Timeout
** Credentials
** Plain Credentials
** Trilead API
** SSH Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins API
Timestamper
** Caffeine API
** Script Security
** JAXB
** SnakeYAML API
** Jackson 2 API
** commons-text API
** Pipeline: Supporting APIs
** Plugin Utilities API
** Font Awesome API
** Bootstrap 5 API
** JQuery3 API
** ECharts API
** Display URL API
** - required dependency
```

Jenkins 2.414.3

Jenkins | Search (CTRL+K) | admin | log out

Dashboard >

- New Item
- People
- Build History
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Add description

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

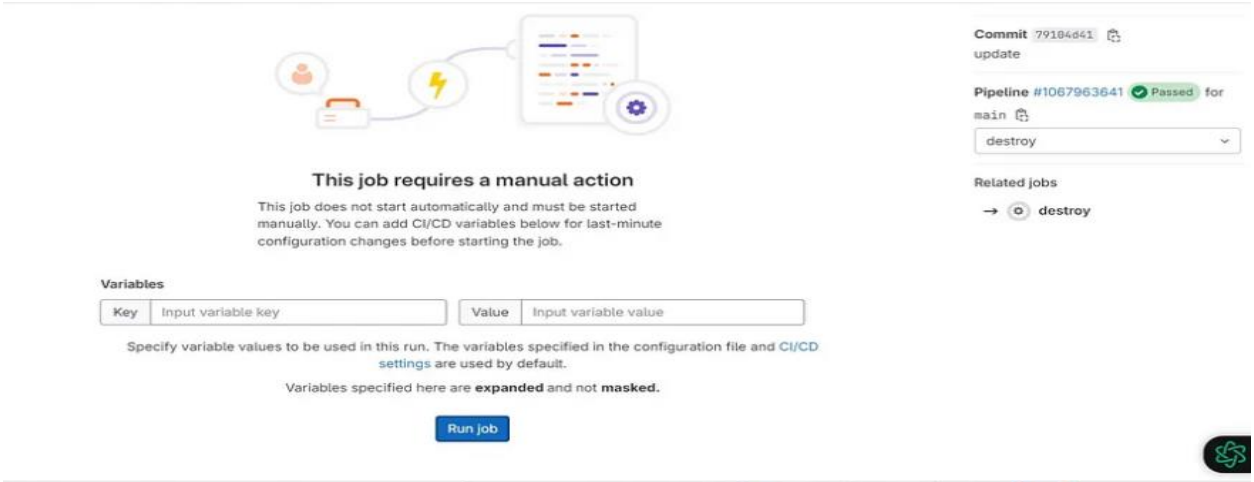
Create a job →

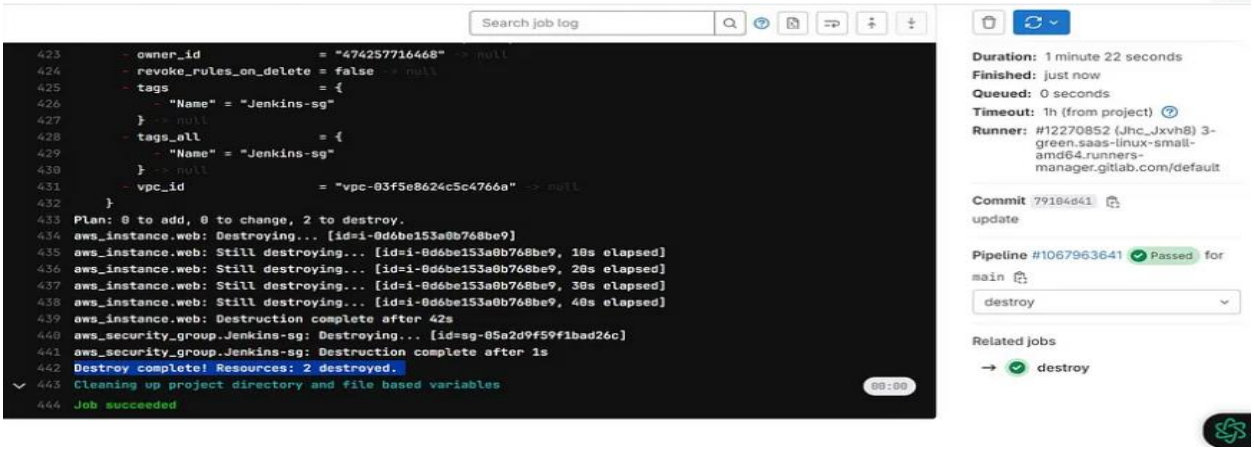
Set up a distributed build

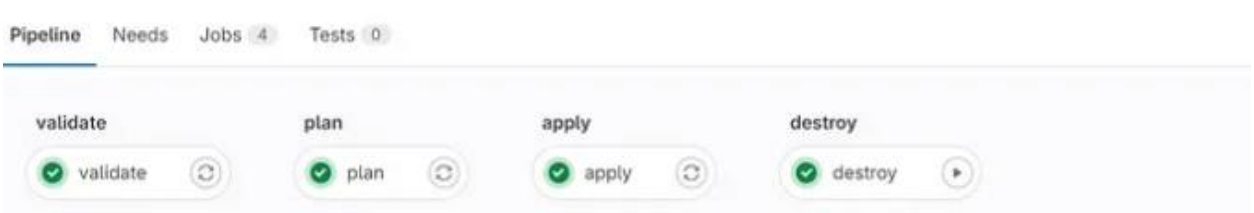
Set up an agent →

Configure a cloud →

Learn more about distributed builds


This job requires a manual action
This job does not start automatically and must be started manually. You can add CI/CD variables below for last-minute configuration changes before starting the job.
Variables
Key
Input variable key
Value
Input variable value
Specify variable values to be used in this run. The variables specified in the configuration file and CI/CD settings are used by default.
Variables specified here are expanded and not masked.
Run job
Commit 79104d41
update
Pipeline #1067963641 Passed for main
destroy
Related jobs
destroy


Search job log
423 - owner_id = "474257716468" -> null
424 - revoke_rules_on_delete = false -> null
425 - tags = {
426 - "Name" = "Jenkins-sg"
427 } -> null
428 - tags_all = {
429 - "Name" = "Jenkins-sg"
430 } -> null
431 - vpc_id = "vpc-03f5e8624c5c4766a" -> null
432 }
433 Plan: 0 to add, 0 to change, 2 to destroy.
434 aws_instance.web: Destroying... [id=i-0d6be153a0b768be9]
435 aws_instance.web: Still destroying... [id=i-0d6be153a0b768be9, 10s elapsed]
436 aws_instance.web: Still destroying... [id=i-0d6be153a0b768be9, 20s elapsed]
437 aws_instance.web: Still destroying... [id=i-0d6be153a0b768be9, 30s elapsed]
438 aws_instance.web: Still destroying... [id=i-0d6be153a0b768be9, 40s elapsed]
439 aws_instance.web: Destruction complete after 42s
440 aws_security_group.Jenkins-sg: Destroying... [id=sg-05a2d9f59f1bad26c]
441 aws_security_group.Jenkins-sg: Destruction complete after 1s
442 Destroy complete! Resources: 2 destroyed.
443 Cleaning up project directory and file based variables
444 Job succeeded
00:00
Duration: 1 minute 22 seconds
Finished: just now
Queued: 0 seconds
Timeout: 1h (from project)
Runner: #12270852 (Jhc_Jxvh8) 3-green.saas-linux-small-amd64.runners-manager.gitlab.com/default
Commit 79104d41
update
Pipeline #1067963641 Passed for main
destroy
Related jobs
destroy


Pipeline
Needs
Jobs 4
Tests 0
validate
validate
plan
plan
apply
apply
destroy
destroy