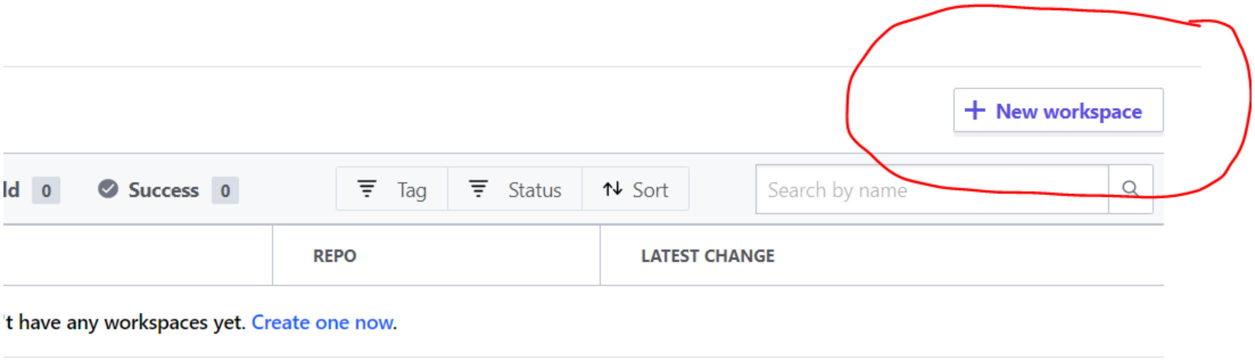
AmazingDevOps

Creating a Workspace in terraform Cloud

A workspace is important in terraform Cloud because it helps us to be organized and keep all our projects in specific workspaces.

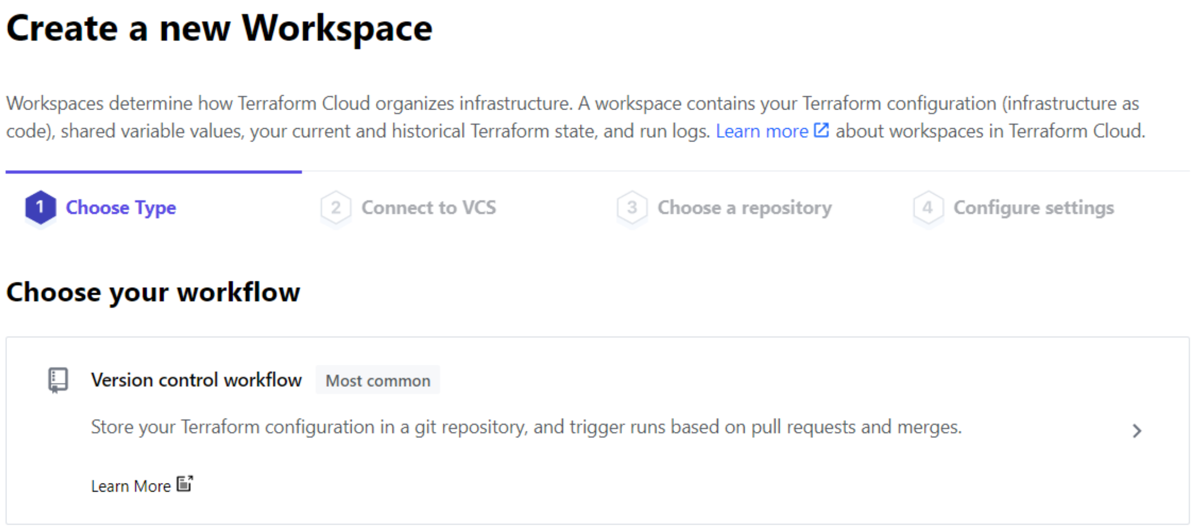
Each project occupies a specific workspace in terraform Cloud. This means that two projects cannot have the same workspace name.



t have any workspaces yet. Create one now.

There are three types of workflows in terraform Cloud

Version Control Workflow – This is closely related to the gitops concept, where everything is triggered from the version control system called Git. Everything runs remotely



Create a new Workspace

Workspaces determine how Terraform Cloud organizes infrastructure. A workspace contains your Terraform configuration (infrastructure as code), shared variable values, your current and historical Terraform state, and run logs. Learn more about workspaces in Terraform Cloud.

Choose your workflow

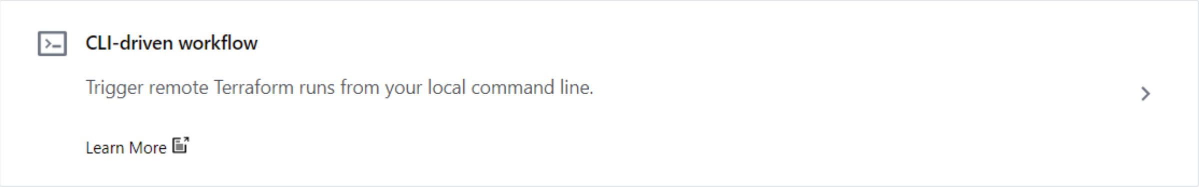
Version control workflow Most common

Store your Terraform configuration in a git repository, and trigger runs based on pull requests and merges.

Learn More

CLI -driven workflow

Everything runs locally through the command-line interface.



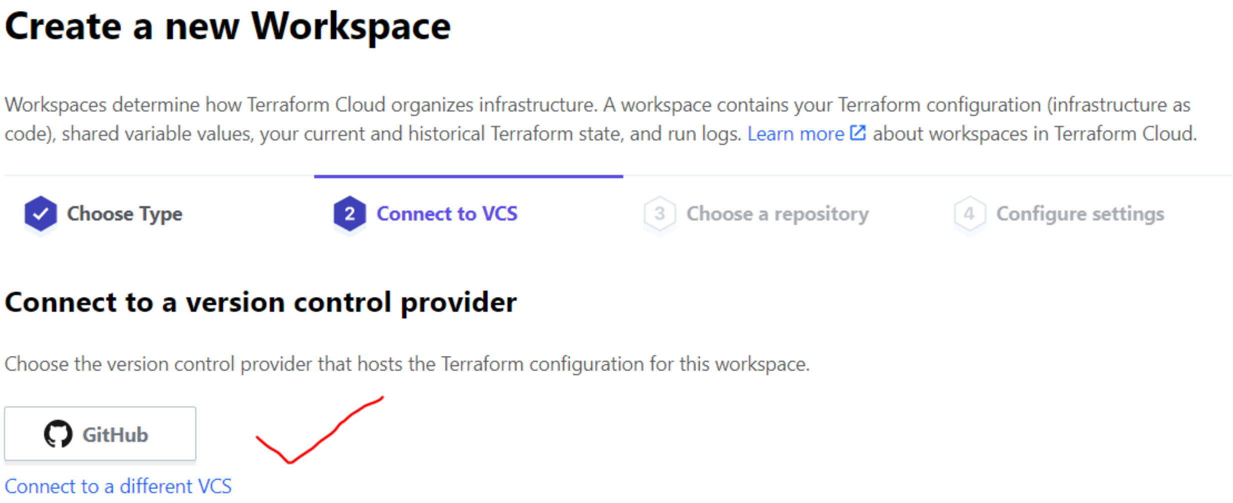
CLI-driven workflow

Trigger remote Terraform runs from your local command line.

Learn More

For now, we will focus on these two types of workflows.

Next Step ➔ Create a workflow.



Create a new Workspace

Workspaces determine how Terraform Cloud organizes infrastructure. A workspace contains your Terraform configuration (infrastructure as code), shared variable values, your current and historical Terraform state, and run logs. Learn more about workspaces in Terraform Cloud.

Connect to a version control provider

Choose the version control provider that hosts the Terraform configuration for this workspace.

The version control system we are going to use is the Github. There are different types of version control systems, such as

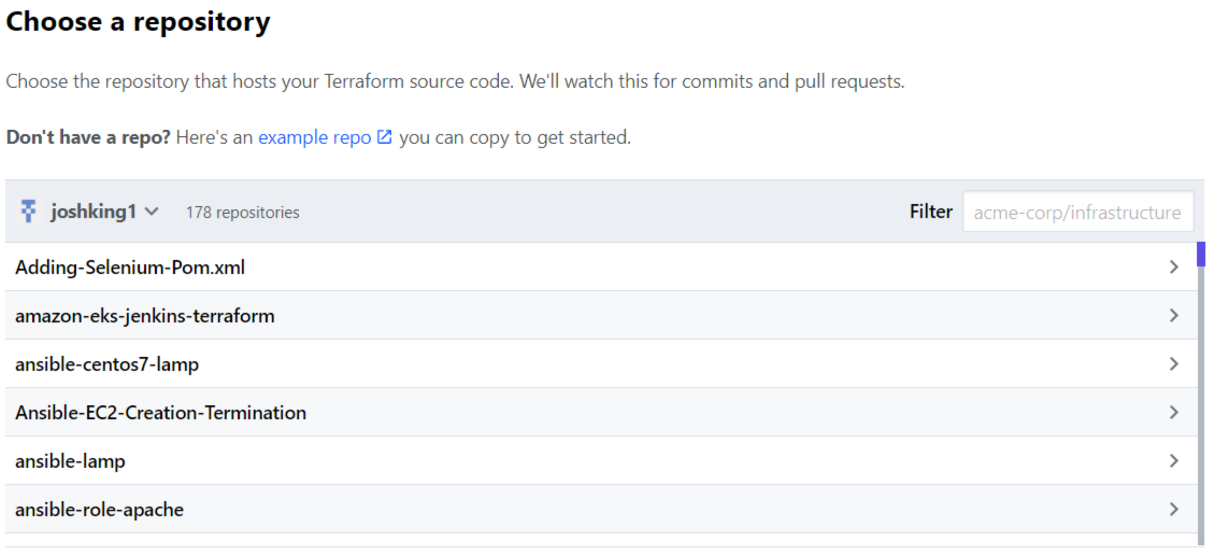
Atlassian Bitbucket and GitLab

Next step ➔ Click on GitHub

Make sure Github is API integrated with terraform Cloud.

If the process is successful, you should view all your repositories in GitHub being displayed in terraform Cloud.

My Github is integrated with terraform Cloud, and this is why I can see all my repositories in Github in my terraform cloud workspace.



Choose a repository

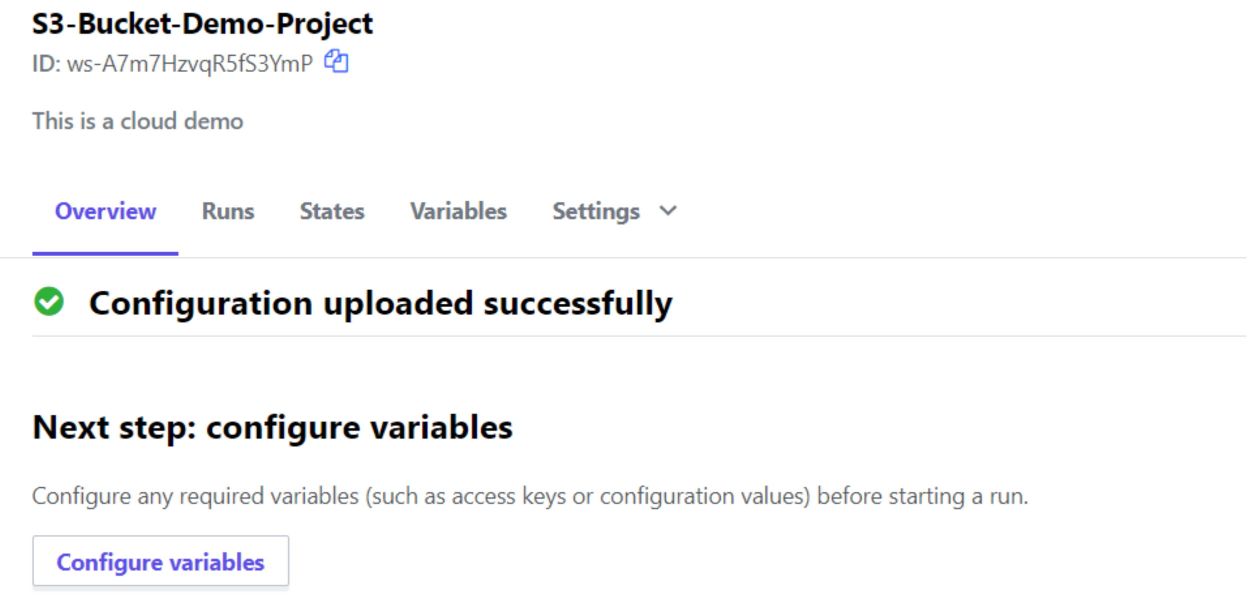
Choose the repository that hosts your Terraform source code. We'll watch this for commits and pull requests.

Don't have a repo? Here's an example repo you can copy to get started.

- Adding-Selenium-Pom.xml
- amazon-eks-jenkins-terraform
- ansible-centos7-lamp
- Ansible-EC2-Creation-Termination
- ansible-lamp
- ansible-role-apache

Next step ➔ Select the repository containing your infrastructure code.

Next step ➔ Finish setting up your workspace



S3-Bucket-Demo-Project

ID: ws-A7m7HzvqR5fS3YmP

This is a cloud demo

Configuration uploaded successfully

Next step: configure variables

Configure any required variables (such as access keys or configuration values) before starting a run.

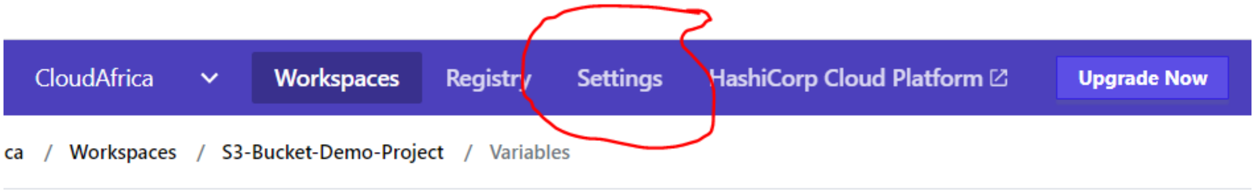
Next step ➔ Create variable sets.

It would be advisable to create an organization variable set if your entire organization uses just one cloud provider.

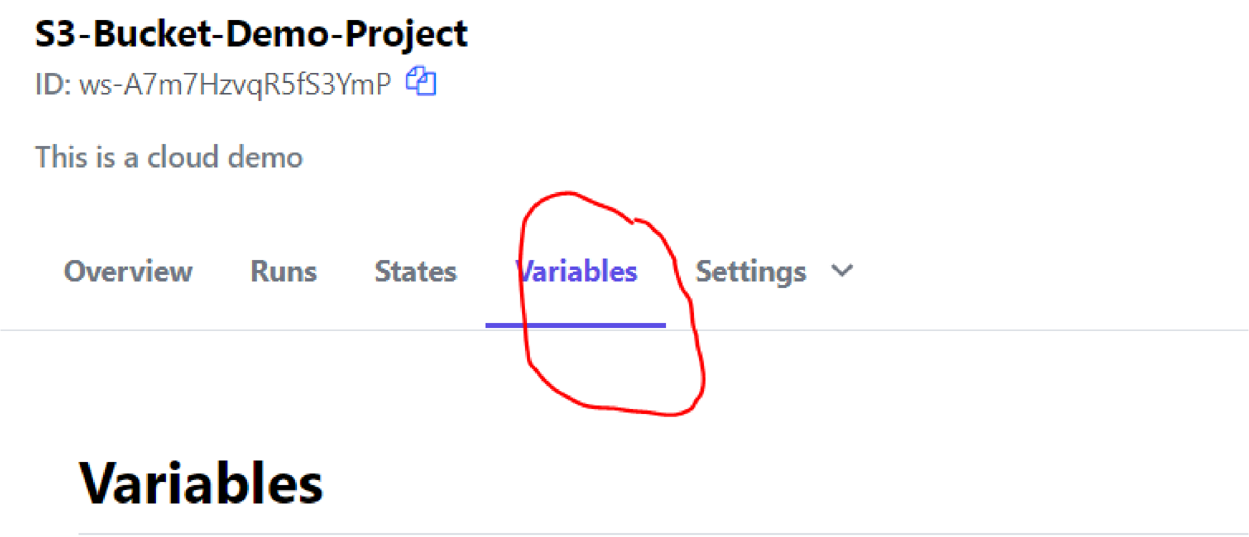
However, if you are using different cloud providers, creating workspace variable sets that are specific to that workspace only would be advisable.

In our case, we will use organization workspace because we are only using AWS as our provider.

Workspace variable set vs. Organization variable sets



Vs.



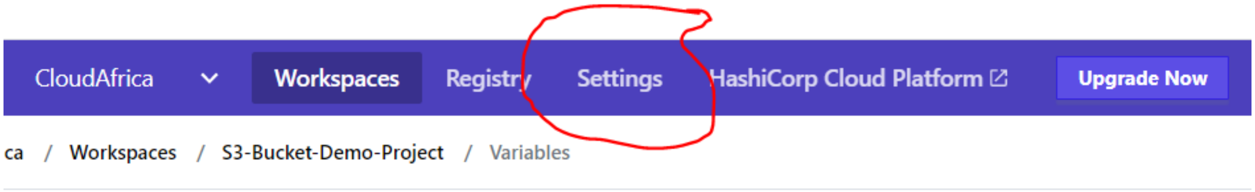
Workspace variable sets are specific to the workspace they are established for and cannot be used for the entire organization.

The entire organization uses organization variable sets. They are used by every workspace created under the organization.

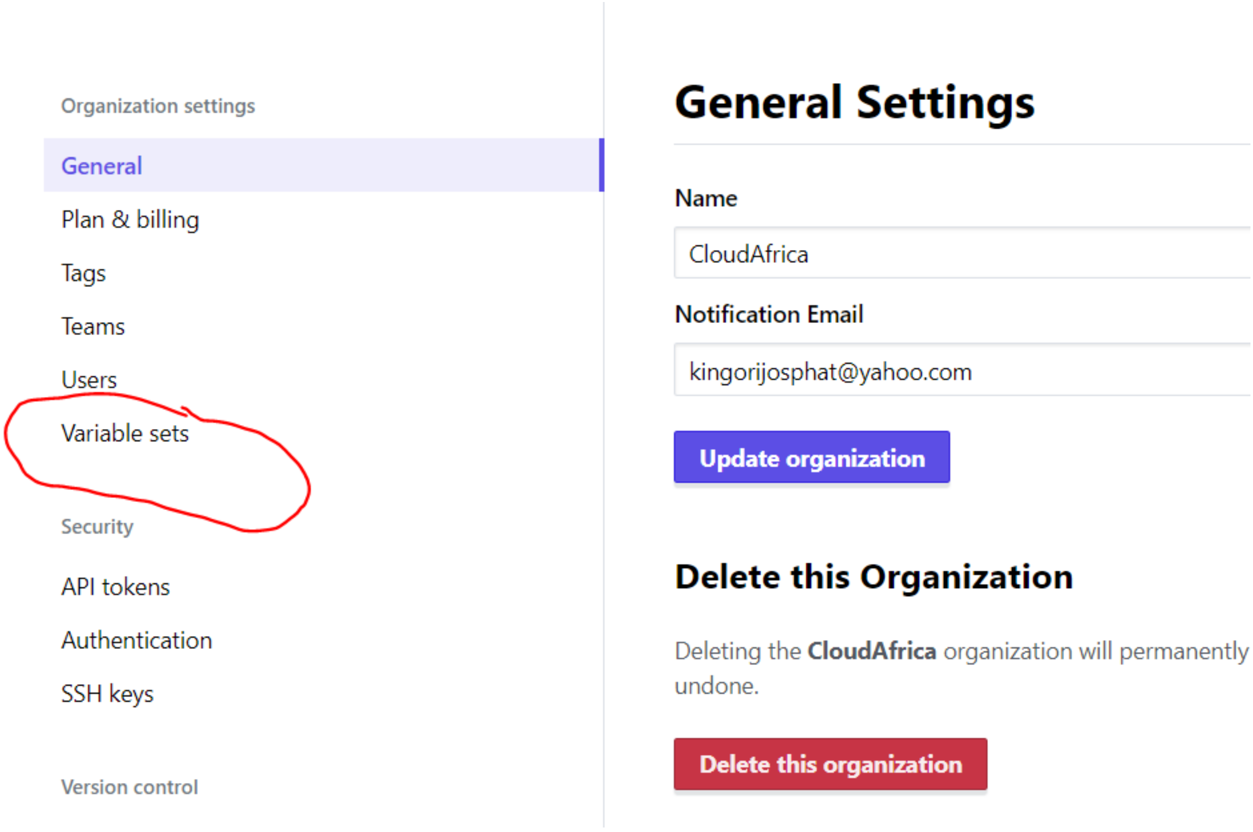
They are essential in reducing the need to keep creating variable sets for every workspace created under the same organization.

Next step ➔ Configure our workspace.

Click on settings on the bluish row.



Next step ➔ Click on settings and then select variable sets.



General Settings

Name

CloudAfrica

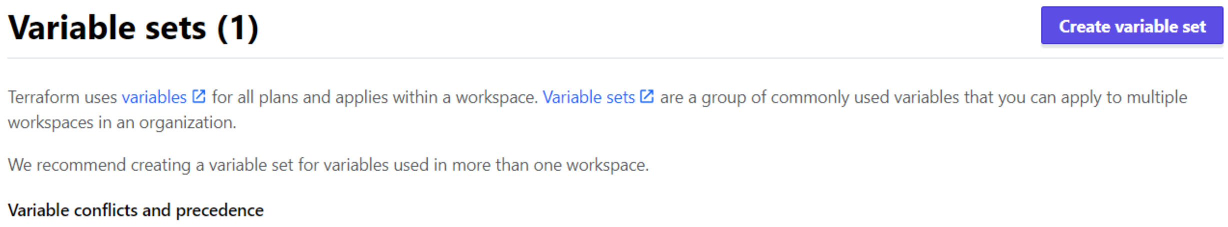
Notification Email

kingorijosphat@yahoo.com

Delete this Organization

Deleting the **CloudAfrica** organization will permanently undone.

Next step ➔ Click on create variable sets



Variable sets (1)

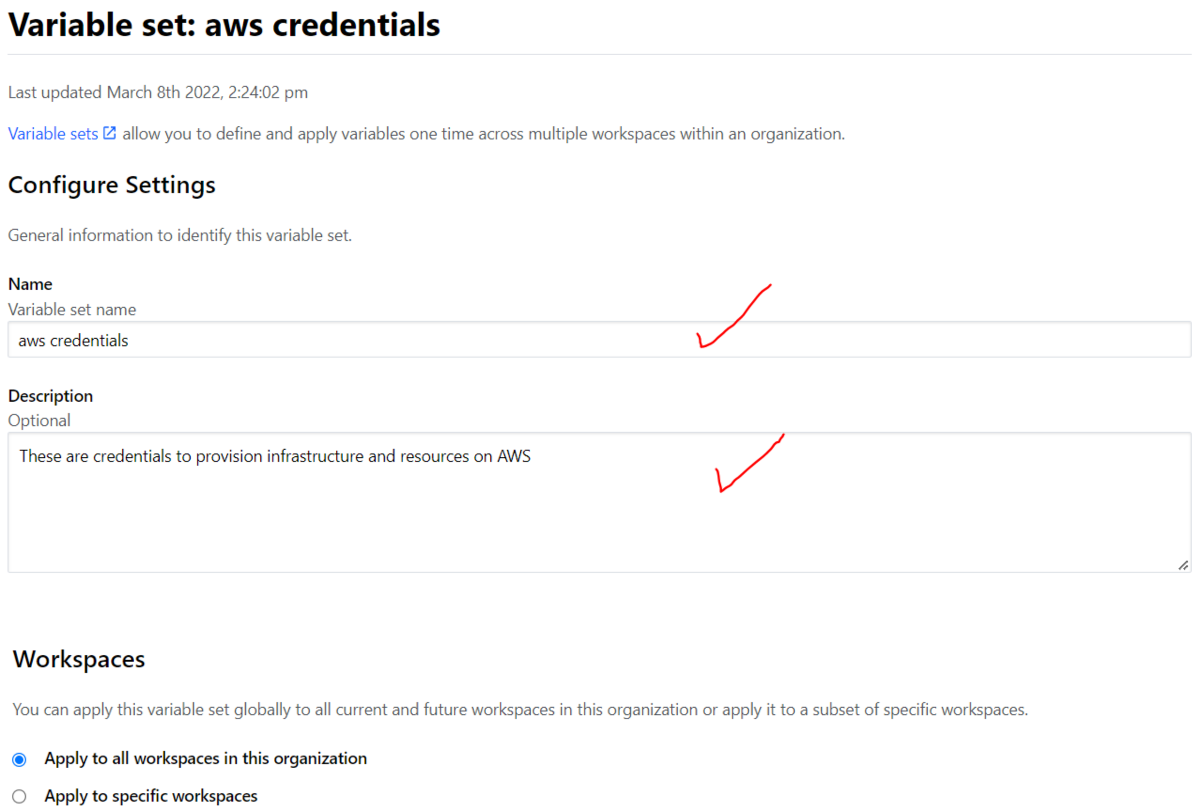
Create variable set

Terraform uses variables for all plans and applies within a workspace. Variable sets are a group of commonly used variables that you can apply to multiple workspaces in an organization.

We recommend creating a variable set for variables used in more than one workspace.

Variable conflicts and precedence

Next step ➔ Fill in the information required as shown below.



Variable set: aws credentials

Last updated March 8th 2022, 2:24:02 pm

Variable sets allow you to define and apply variables one time across multiple workspaces within an organization.

Configure Settings

General information to identify this variable set.

Name
Variable set name

aws credentials ✓

Description
Optional

These are credentials to provision infrastructure and resources on AWS ✓

Workspaces

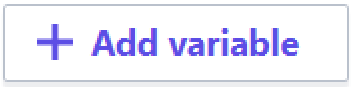
You can apply this variable set globally to all current and future workspaces in this organization or apply it to a subset of specific workspaces.

● Apply to all workspaces in this organization

○ Apply to specific workspaces

Make sure you select these variable sets to apply to all the workspaces in the organization you created

Next step ➔ Click on additional variables.



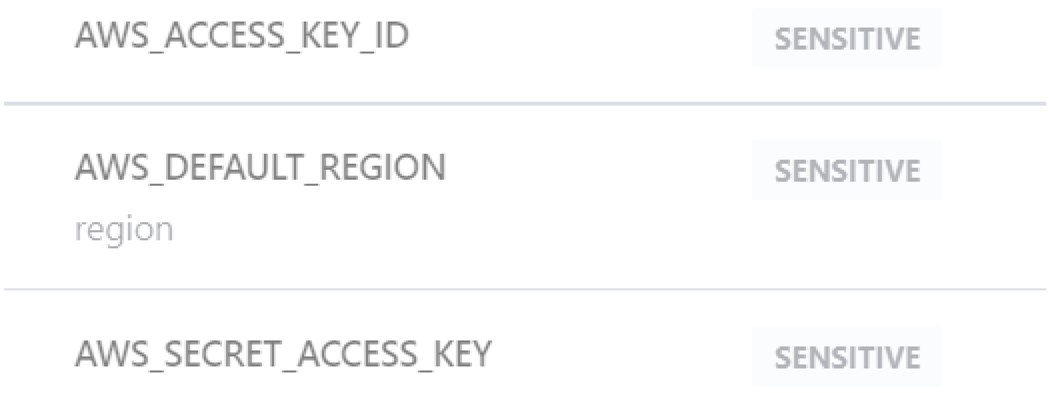
+ Add variable

AmazingDevOps

Next step ➔ create your AWS credentials.

They are vital in helping terraform to connect to the AWS console and provisioning the infrastructure in our desired state stored as a git repository in GitHub.

What are these credentials?



AWS_ACCESS_KEY_ID SENSITIVE

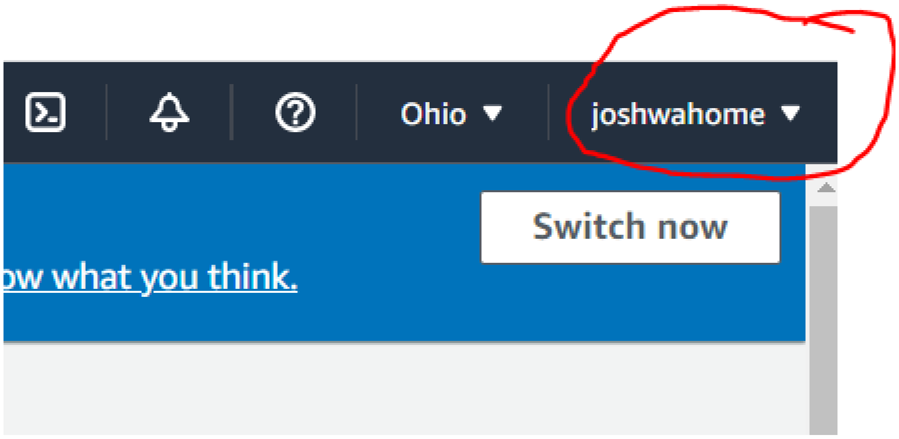
AWS_DEFAULT_REGION SENSITIVE
region

AWS_SECRET_ACCESS_KEY SENSITIVE

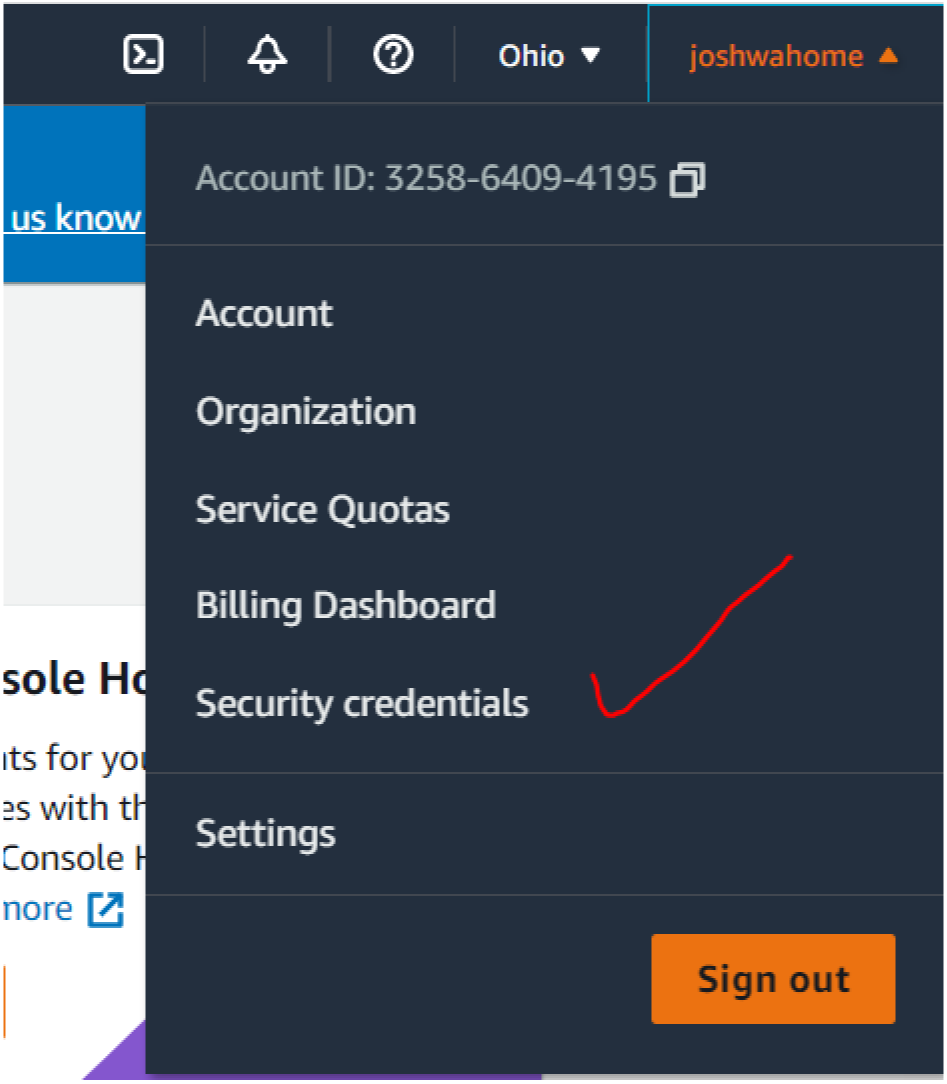
Next step ➔ let us create the AWS credentials

Login to your AWS account

Go to where the name of your account appears

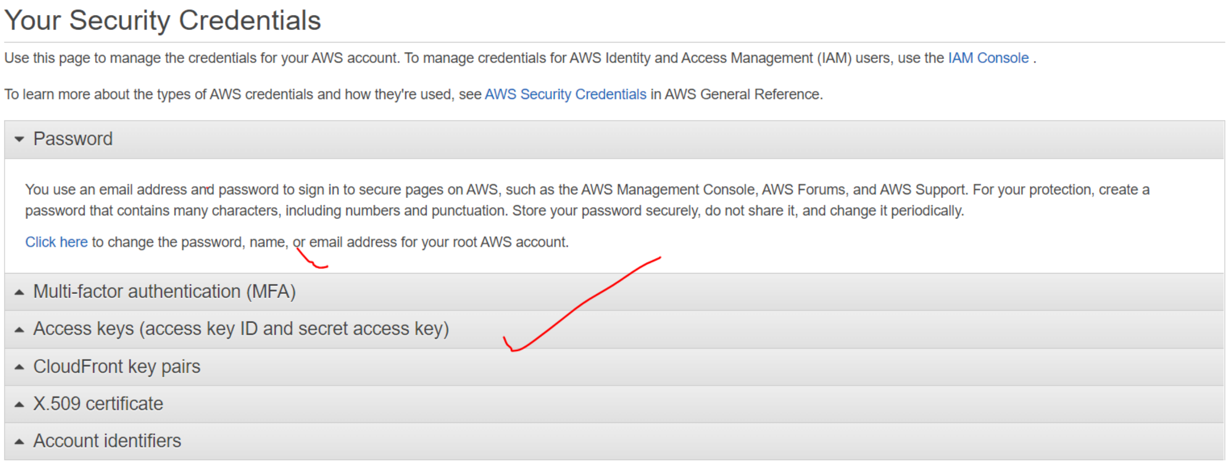


Click on the arrow pointing down.



Account ID: 3258-6409-4195

Select Security Credentials



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console .

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS General Reference.

▾ Password

You use an email address and password to sign in to secure pages on AWS, such as the AWS Management Console, AWS Forums, and AWS Support. For your protection, create a password that contains many characters, including numbers and punctuation. Store your password securely, do not share it, and change it periodically.

Click here to change the password, name, or email address for your root AWS account.

▴ Multi-factor authentication (MFA)

▴ Access keys (access key ID and secret access key)

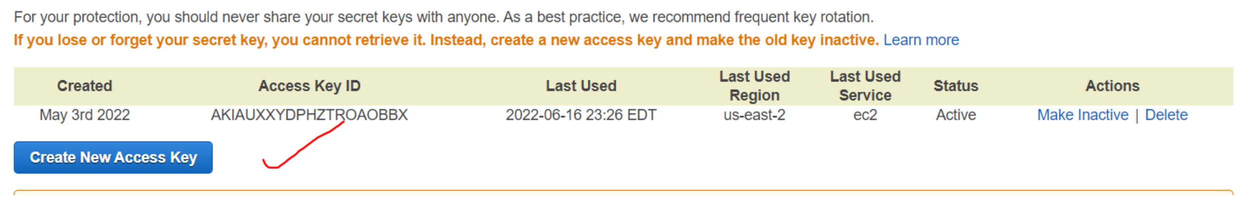
▴ CloudFront key pairs

▴ X.509 certificate

▴ Account identifiers

- ⇨ Click on Access Keys

Create New Access Key



For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.

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

Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
May 3rd 2022	AKIAUXXYDPHZTROAOBBX	2022-06-16 23:26 EDT	us-east-2	ec2	Active	Make Inactive | Delete

These are sensitive ➔ do not expose them in GitHub or terraform Cloud

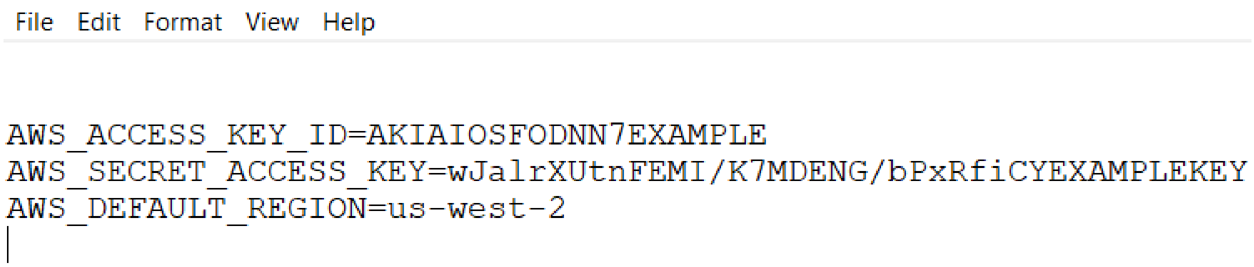
Example of AWS credentials

AWS_ACCESS_KEY_ID=*AKIAIOSFODNN7EXAMPLE*

AWS_SECRET_ACCESS_KEY=*wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY*

AWS_DEFAULT_REGION=*us-west-2*

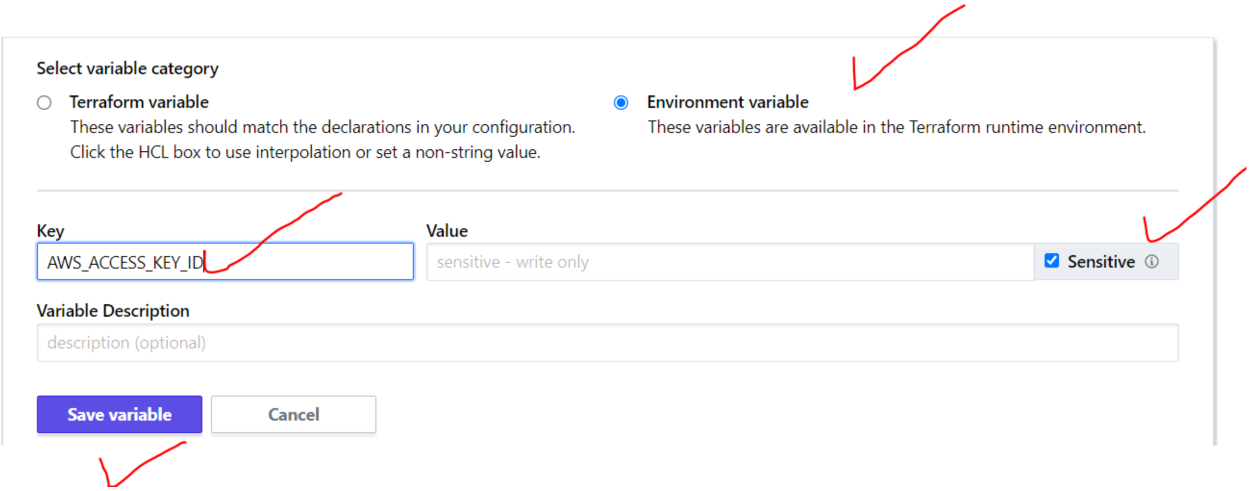
Next step ➔ Open a notepad



```
AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
AWS_DEFAULT_REGION=us-west-2
```

Next step ➔ go to terraform Cloud and fill in the name of the variables and their associated value.

Select the *environment variable,* and do not forget to select *sensitive*



Select variable category

○ Terraform variable
These variables should match the declarations in your configuration. Click the HCL box to use interpolation or set a non-string value.

● Environment variable
These variables are available in the Terraform runtime environment.

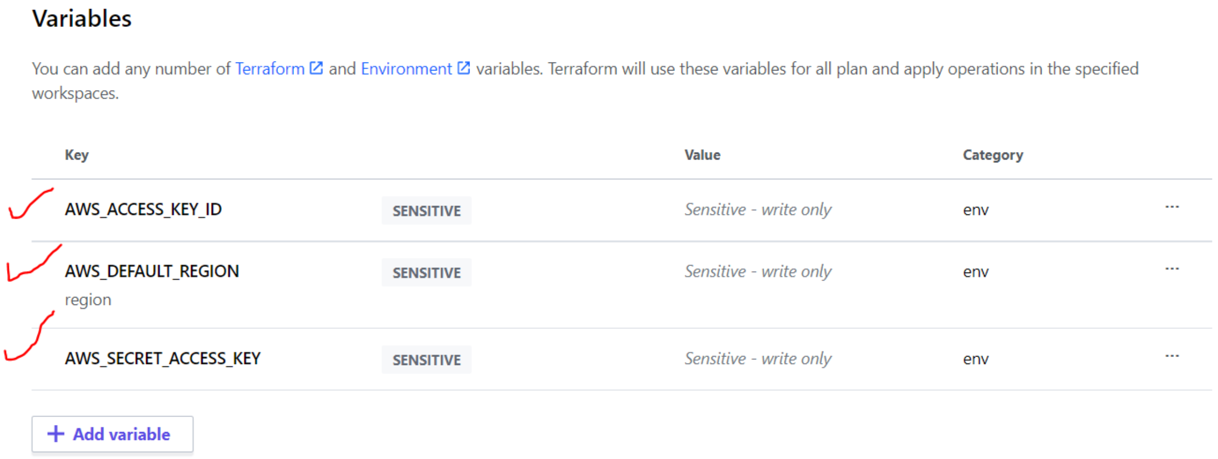
Key: AWS_ACCESS_KEY_ID

Value: sensitive - write only

☑ Sensitive ⓘ

Variable Description: description (optional)

Do the same for the remaining two variables. You should have the following variables.



Variables

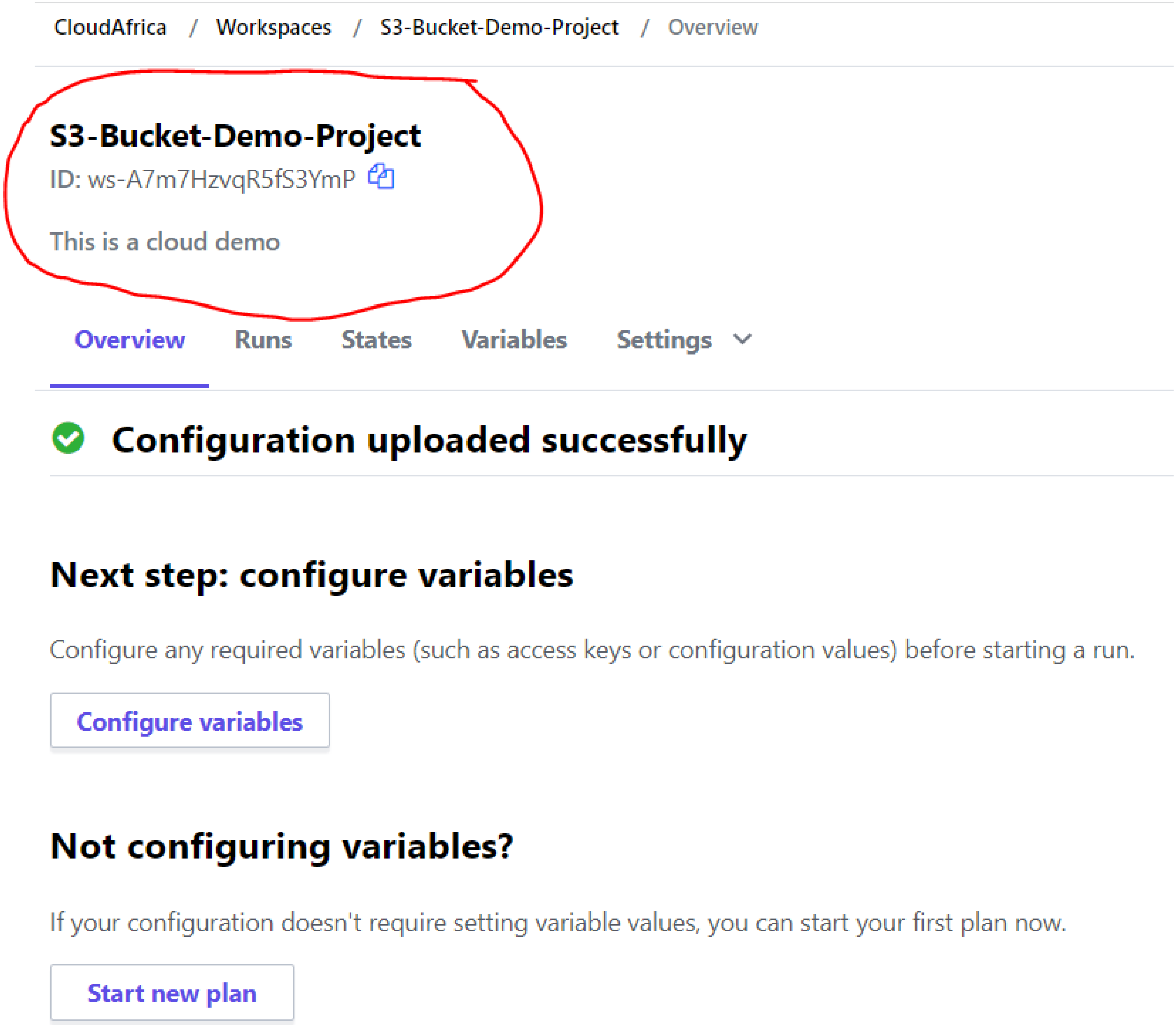
You can add any number of Terraform and Environment variables. Terraform will use these variables for all plan and apply operations in the specified workspaces.

Key		Value	Category	
AWS_ACCESS_KEY_ID	SENSITIVE	*Sensitive - write only*	env	···
AWS_DEFAULT_REGION region	SENSITIVE	*Sensitive - write only*	env	···
AWS_SECRET_ACCESS_KEY	SENSITIVE	*Sensitive - write only*	env	···

Next step ➔ Save the variable sets.

Save variable set
Delete variable set

Next step → go back to the namespace you created



CloudAfrica / Workspaces / S3-Bucket-Demo-Project / Overview

S3-Bucket-Demo-Project

ID: ws-A7m7HzvqR5fS3YmP

This is a cloud demo

Overview
Runs
States
Variables
Settings

Configuration uploaded successfully

Next step: configure variables

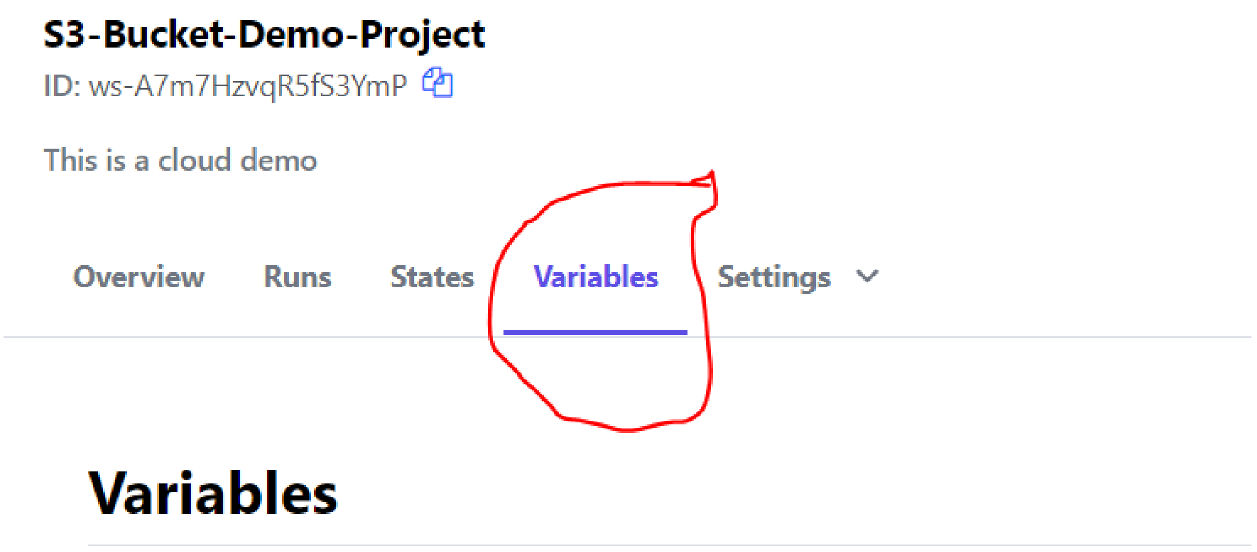
Configure any required variables (such as access keys or configuration values) before starting a run.

Not configuring variables?

If your configuration doesn't require setting variable values, you can start your first plan now.

Next step → Check the variables.

AmazingDevOps



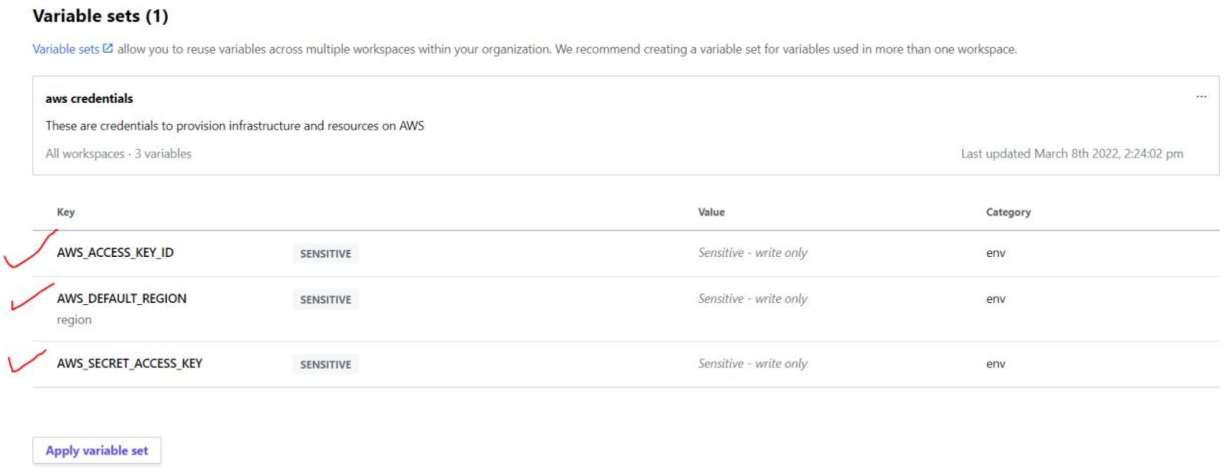
S3-Bucket-Demo-Project

ID: ws-A7m7HzvqR5fS3YmP

This is a cloud demo

Variables

Next step ➔ Scroll down and check the variable sets connected to your workspace.



Variable sets (1)

Variable sets allow you to reuse variables across multiple workspaces within your organization. We recommend creating a variable set for variables used in more than one workspace.

aws credentials

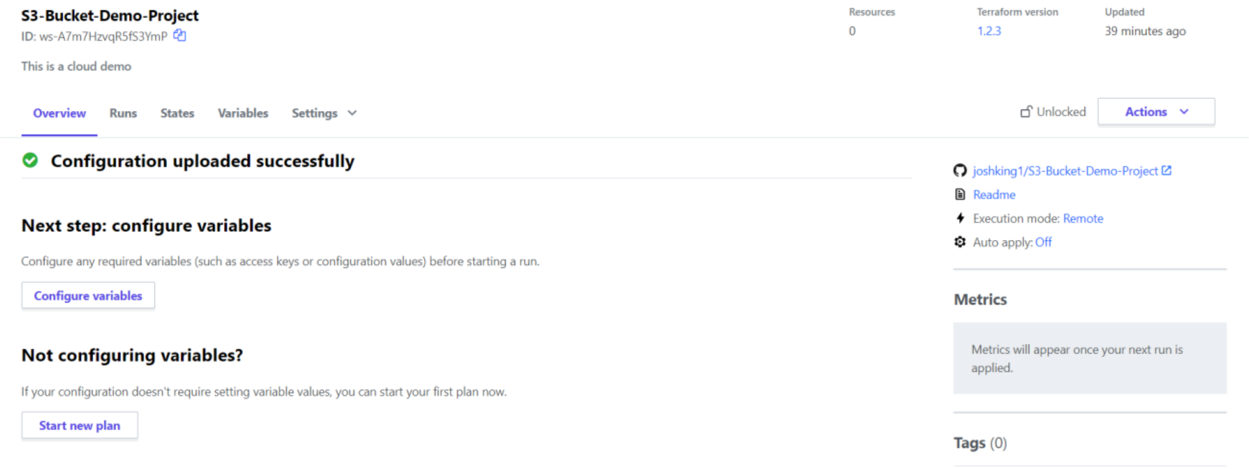
These are credentials to provision infrastructure and resources on AWS

All workspaces · 3 variables

Last updated March 8th 2022, 2:24:02 pm

Key		Value	Category
AWS_ACCESS_KEY_ID	SENSITIVE	*Sensitive - write only*	env
AWS_DEFAULT_REGION region	SENSITIVE	*Sensitive - write only*	env
AWS_SECRET_ACCESS_KEY	SENSITIVE	*Sensitive - write only*	env

We are all set!!!



S3-Bucket-Demo-Project

ID: ws-A7m7HzvqR5fS3YmP

This is a cloud demo

Resources	Terraform version	Updated
0	1.2.3	39 minutes ago

Configuration uploaded successfully

Next step: configure variables

Configure any required variables (such as access keys or configuration values) before starting a run.

Not configuring variables?

If your configuration doesn't require setting variable values, you can start your first plan now.

joshking1/S3-Bucket-Demo-Project

Readme

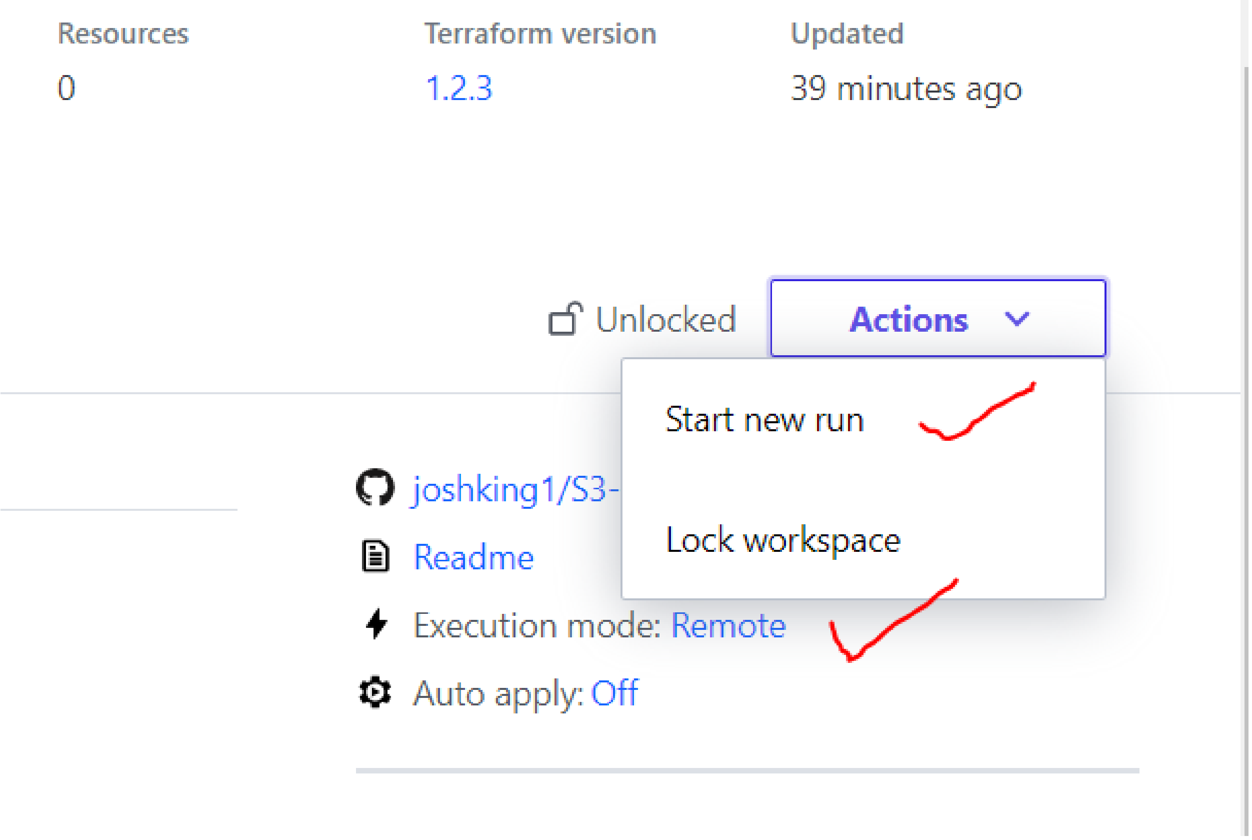
Execution mode: Remote

Auto apply: Off

Metrics

Metrics will appear once your next run is applied.

Tags (0)



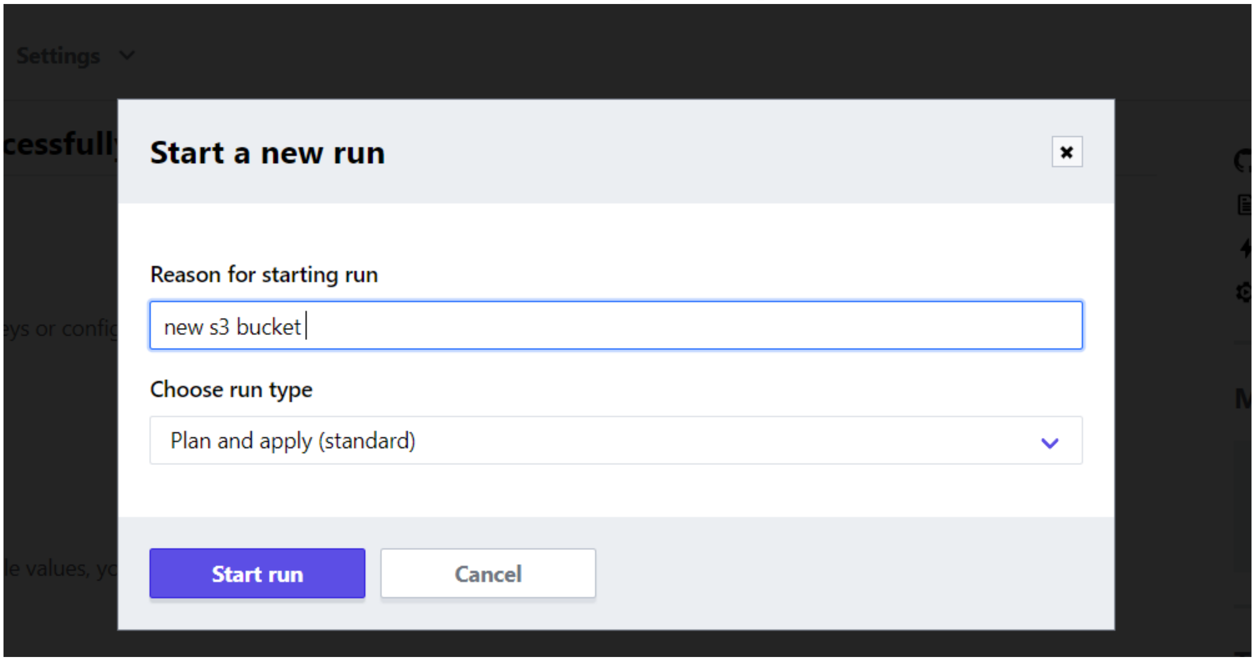
Resources	Terraform version	Updated
0	1.2.3	39 minutes ago

joshking1/S3-

Readme

Execution mode: Remote

Auto apply: Off

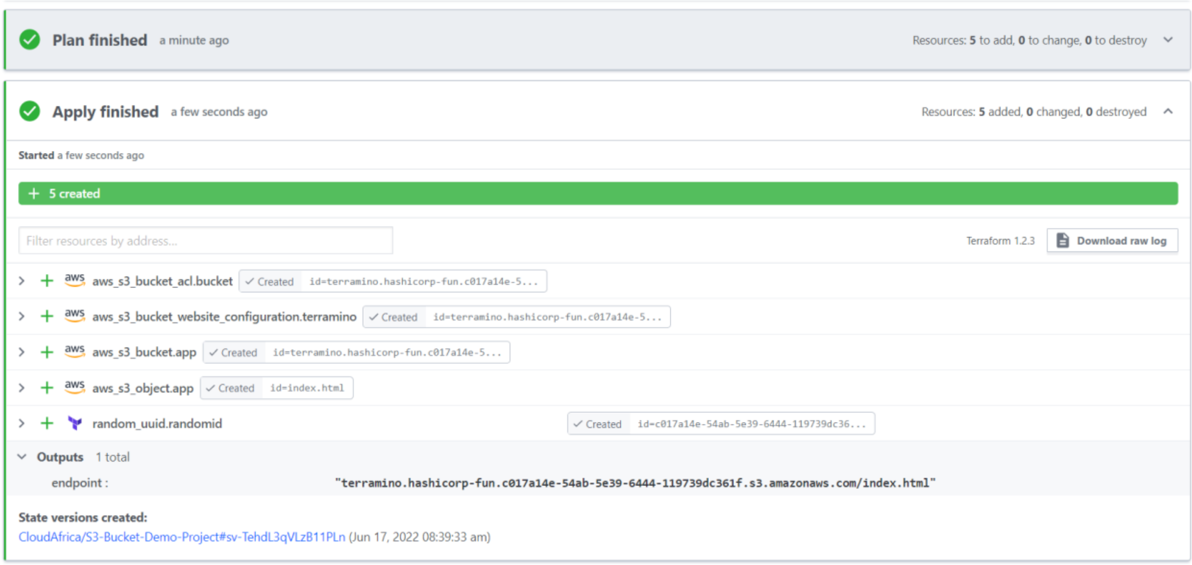


Start a new run

Reason for starting run: new s3 bucket

Choose run type: Plan and apply (standard)

After the plan, apply



Plan finished a minute ago — Resources: 5 to add, 0 to change, 0 to destroy

Apply finished a few seconds ago — Resources: 5 added, 0 changed, 0 destroyed

Started a few seconds ago

+ 5 created

Terraform 1.2.3

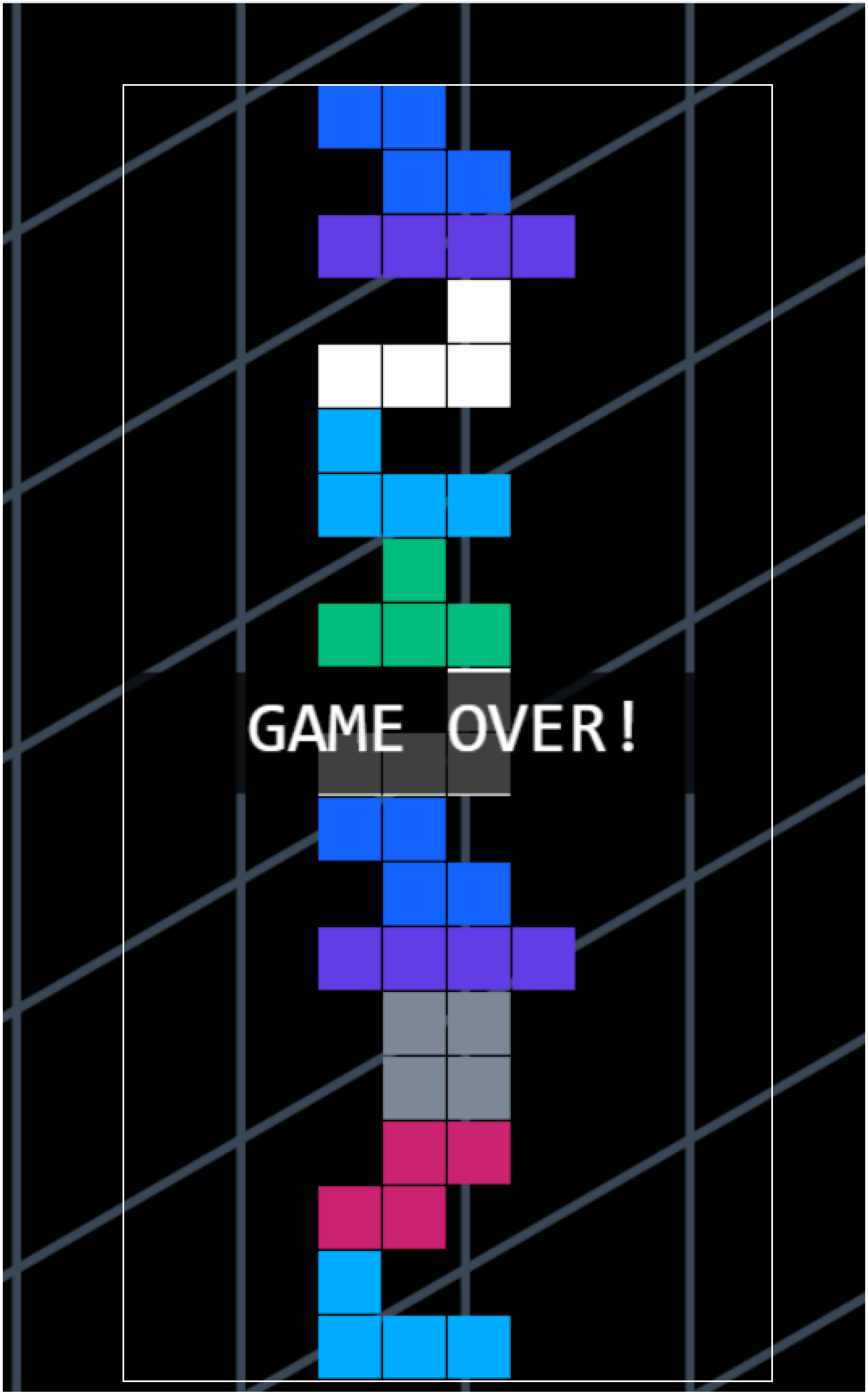
```
+ aws_s3_bucket_acl.bucket                        ✓ Created  id=terramino.hashicorp-fun.c017a14e-5...
+ aws_s3_bucket_website_configuration.terramino   ✓ Created  id=terramino.hashicorp-fun.c017a14e-5...
+ aws_s3_bucket.app                               ✓ Created  id=terramino.hashicorp-fun.c017a14e-5...
+ aws_s3_object.app                               ✓ Created  id=index.html
+ random_uuid.randomid                            ✓ Created  id=c017a14e-54ab-5e39-6444-119739dc36...
Outputs 1 total
  endpoint :   "terramino.hashicorp-fun.c017a14e-54ab-5e39-6444-119739dc361f.s3.amazonaws.com/index.html"
```

State versions created:
CloudAfrica/S3-Bucket-Demo-Project#sv-TehdL3qVLzB11PLn (Jun 17, 2022 08:39:33 am)

The result should be a game called terramino

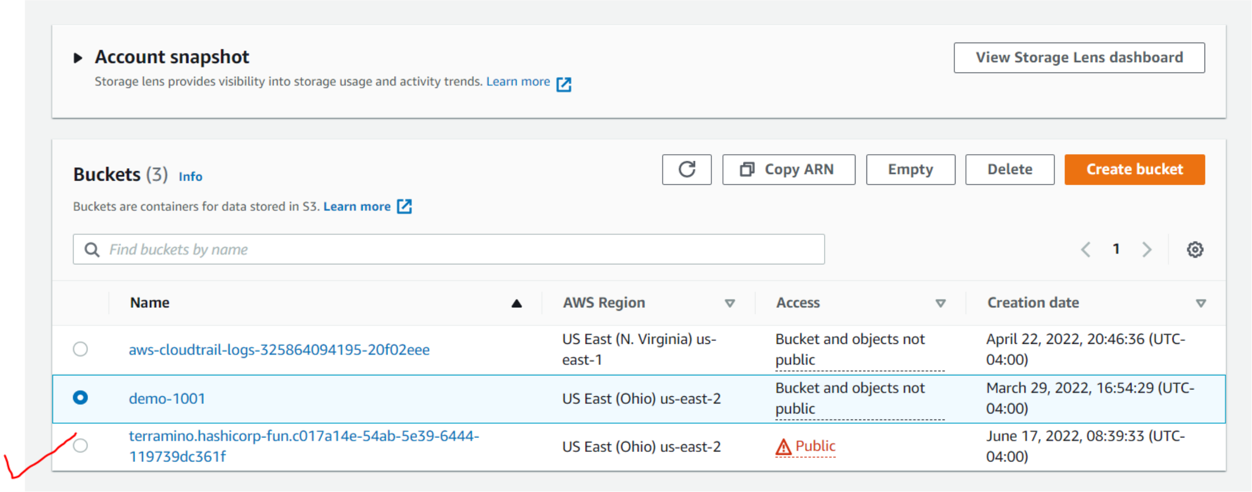
"terramino.hashicorp-fun.c017a14e-54ab-5e39-6444-119739dc361f.s3.amazonaws.com/index.html"

AmazingDevOps



GAME OVER!

Go to the AWS console and confirm the S3 bucket has been created.



Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (3) Info

Buckets are containers for data stored in S3. Learn more

	Name ▲	AWS Region	Access	Creation date
○	aws-cloudtrail-logs-325864094195-20f02eee	US East (N. Virginia) us-east-1	Bucket and objects not public	April 22, 2022, 20:46:36 (UTC-04:00)
●	demo-1001	US East (Ohio) us-east-2	Bucket and objects not public	March 29, 2022, 16:54:29 (UTC-04:00)
○	terramino.hashicorp-fun.c017a14e-54ab-5e39-6444-119739dc361f	US East (Ohio) us-east-2	⚠ Public	June 17, 2022, 08:39:33 (UTC-04:00)