AWS TERRAFORM PROJECT

Automatically apply terraform with AWS CodeBuild

In this project, I describe how to create a CodeBuild job that automatically applies terraform scripts whenever I commit changes to my GitHub repository.

I used the following resources for this project:

- AWS Codebuild.
- Amazon S3.
- EC2.
- IAM.
- AWS CLI.
- AWS Terraform.
- GitHub.

Lessons learned

During this project, I learned how to troubleshoot my Terraform Code e.g. (classic (directory not found) issue during the (PRE_BUILD) phase of my (AWS CodeBuild Project) and how to fix it.

The steps for this project are:

- Clone a private GitHub repository using SSH.
- Create an IAM user with programmatic Access.
- Create & clone a GitHub repository.
- Create Shell Scripts.
- Create a build.yml file for CodeBuild job.
- Create a Personal Access Token.
- Store terraform state file in an S3 Bucket.
- Create Build Project in CodeBuild.

The next stage of this project to clone a private GitHub repository using SSH.

Steps:

- Sign up into my GitHub account.
- Create a private repository.
- Install Git on my Computer.
- Create a Keypair on my Computer.
- Add the public key to GitHub.

- Clone the private repository.

1. Create a Private repository

- Click on (New Repository).
- Give my repository a name (TerraformPro).
- Under the Description, I will call it (My first terraformpro repository).
- Click (Private repository) – This means that this repository is not going to be available to the (Public).
- Under initialize this repository, click on (Add a README file).
- Under the (Add.gitgnore), scroll & add (Terraform) – This means that there are some terraform files that shouldn't be commited into my (repository).
- Then lastly, click (Create repository).

I have successfully created my (terraformpro) repository to store my project in.

Diagram 1

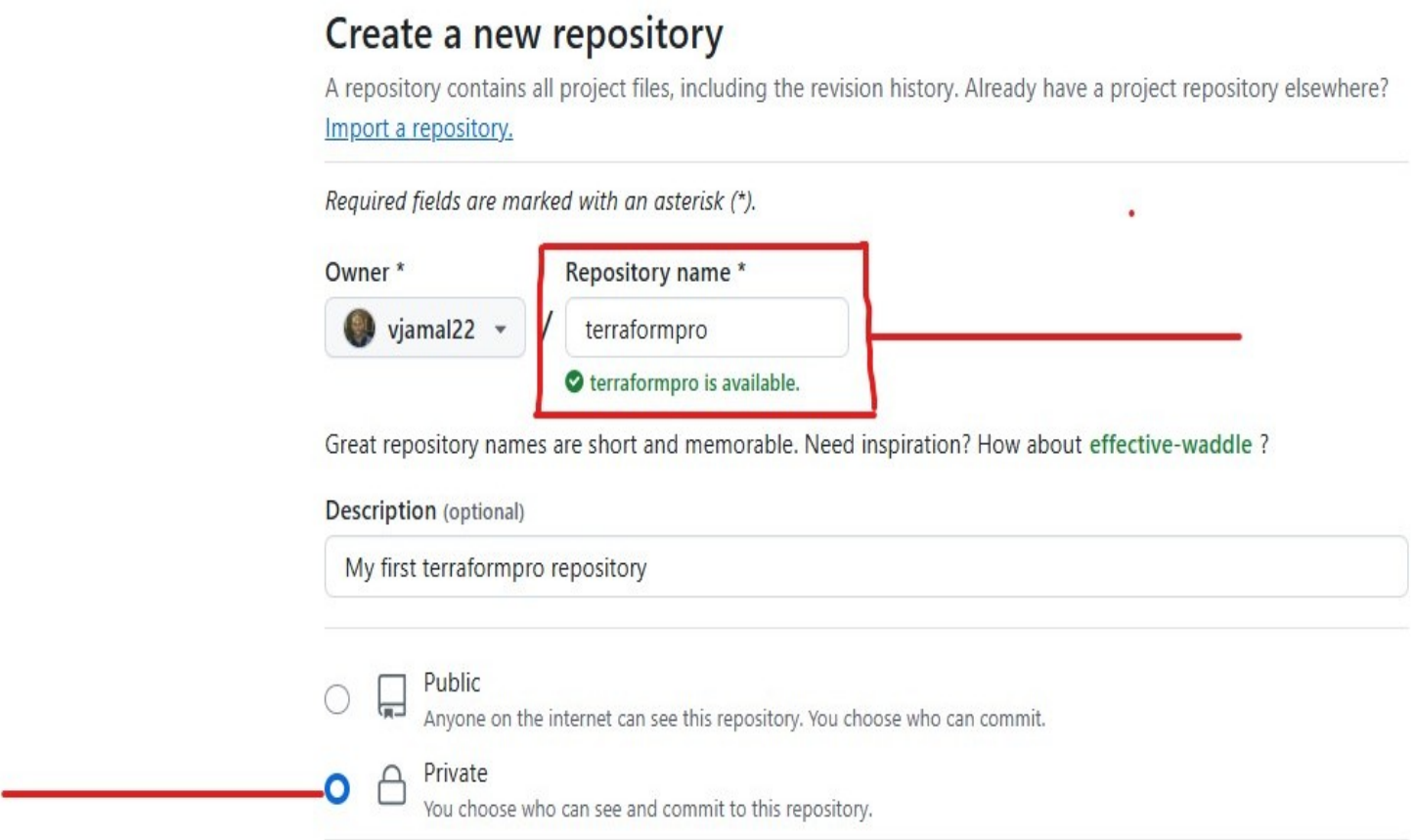
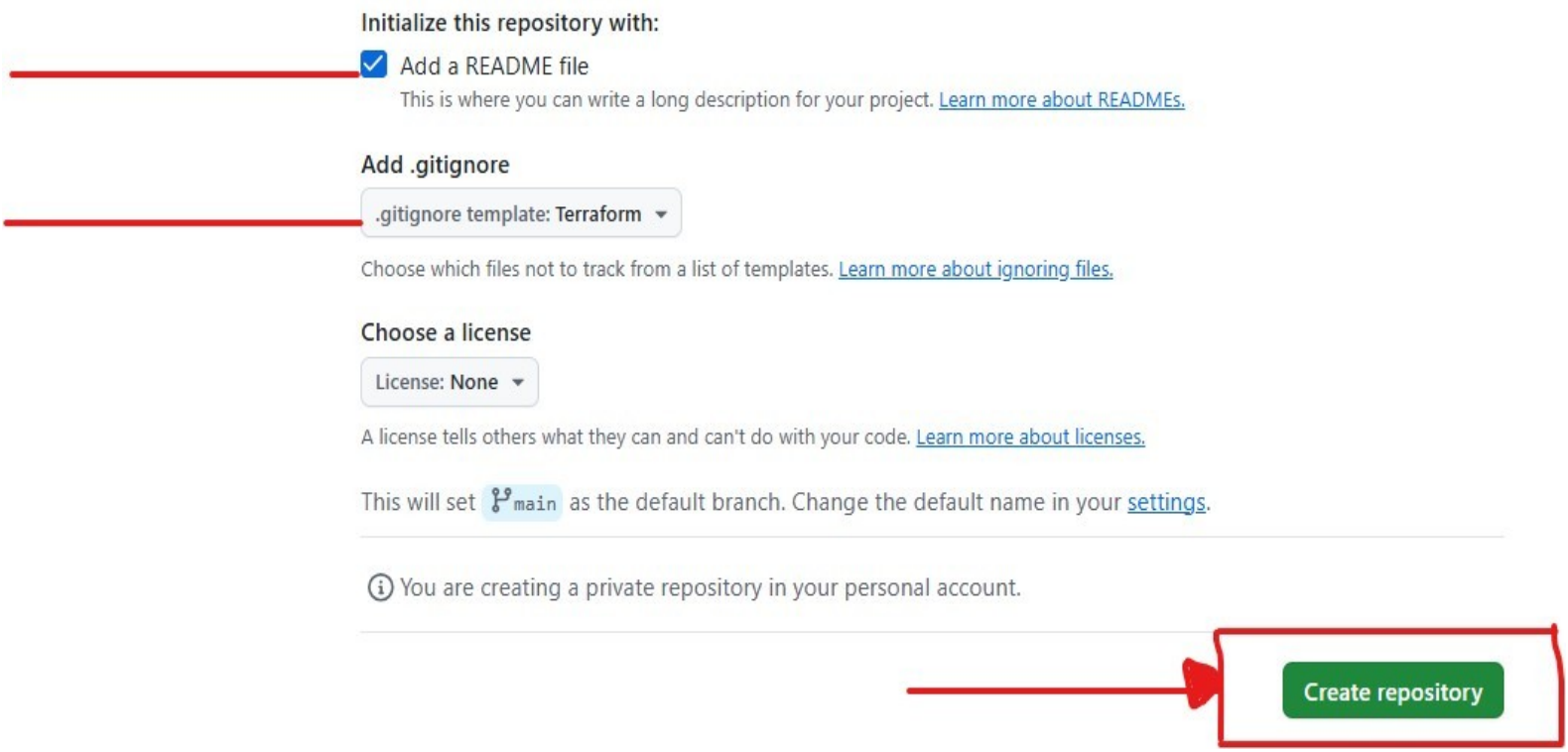


Diagram 1.1



2. Install Git on my Computer

- Cloning my repository to my Computer.
- When I clone my repository to my Computer, all the files in my repository are downloaded to my Computer to allow me easy access to them.

Steps:

- Go to (Google) and install (Git) on my Computer.
- Type (Git) and click (Downloads).
- Click (Windows) under (Downloads).
- Click here to download the latest (2.49.0, 64-bit version) for (Git) for Windows.
- Right click on (show in folder), and from the folder, start the (installation) process.
- Double click on the (Git-2.49.0-64-bit-exe) file to install it.
- Click on (view release notes).
- Click (finish).
- Verify that (Git) was successfully downloaded onto my Computer.
- I will do this by typing (Git) in the (Search) bar at the bottom left and if it pops up, then this is proof that it is downloaded onto my Computer.
- Click on the (Git bash app) and open it.
- This is to verify that (Git) is available in my Computer.
- I can also verify that (Git) is available on my Computer is through the Command line interface (CLI).

- I will do this by opening the (Command prompt) terminal.
- When it’s open, enter the command (git –version) to verify that (Git) is installed on my Computer
- I got an output which means that (Git) is installed on my Computer.
- Once I have verified that (Git) is on my Computer, the next thing I will do is to run this command in the terminal as well.
- This is the command (git config --global user.name) + Enter.
- The next command is (git config --global user.email).
- Inputting my username and email address and inputting them in the commands.
- That’s all needed to configure my details for authorization.

Diagram 1.2


Git 2.49.0 Setup
Completing the Git Setup Wizard
Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Git Bash
View Release Notes
Only show new options
Finish

Diagram 1.3

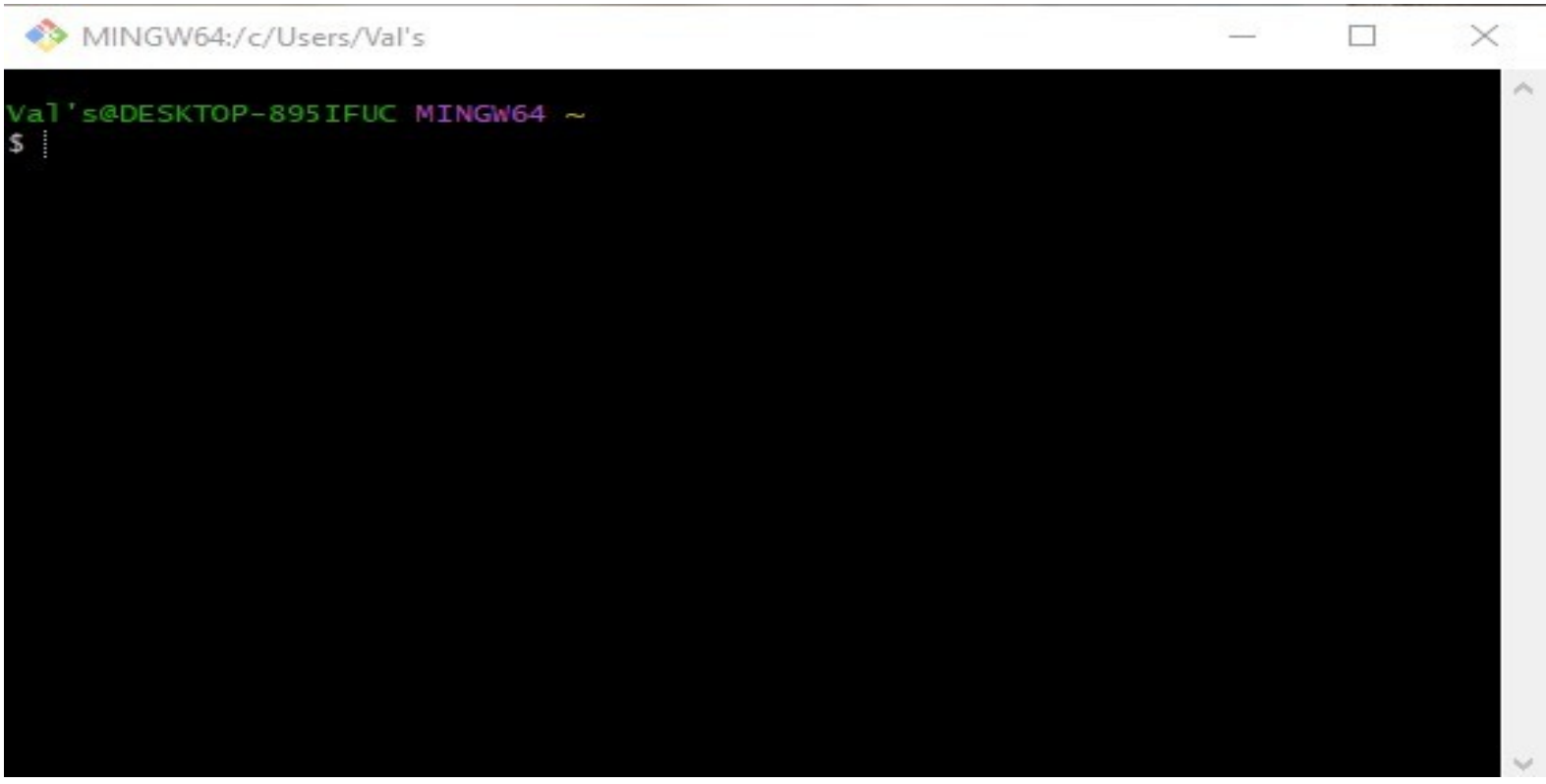

MINGW64:/c/Users/Val's
Val's@DESKTOP-895IFUC MINGW64 ~
$

Diagram 1.4

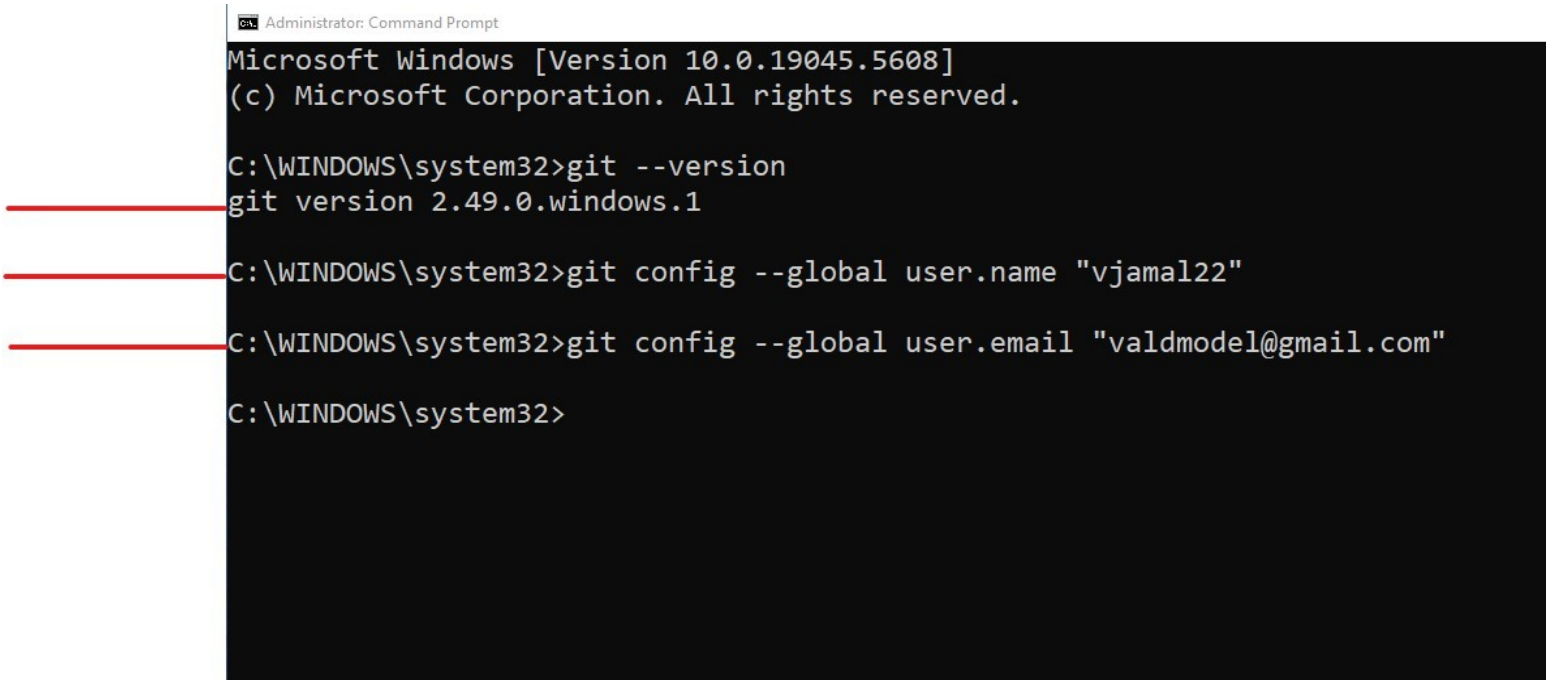
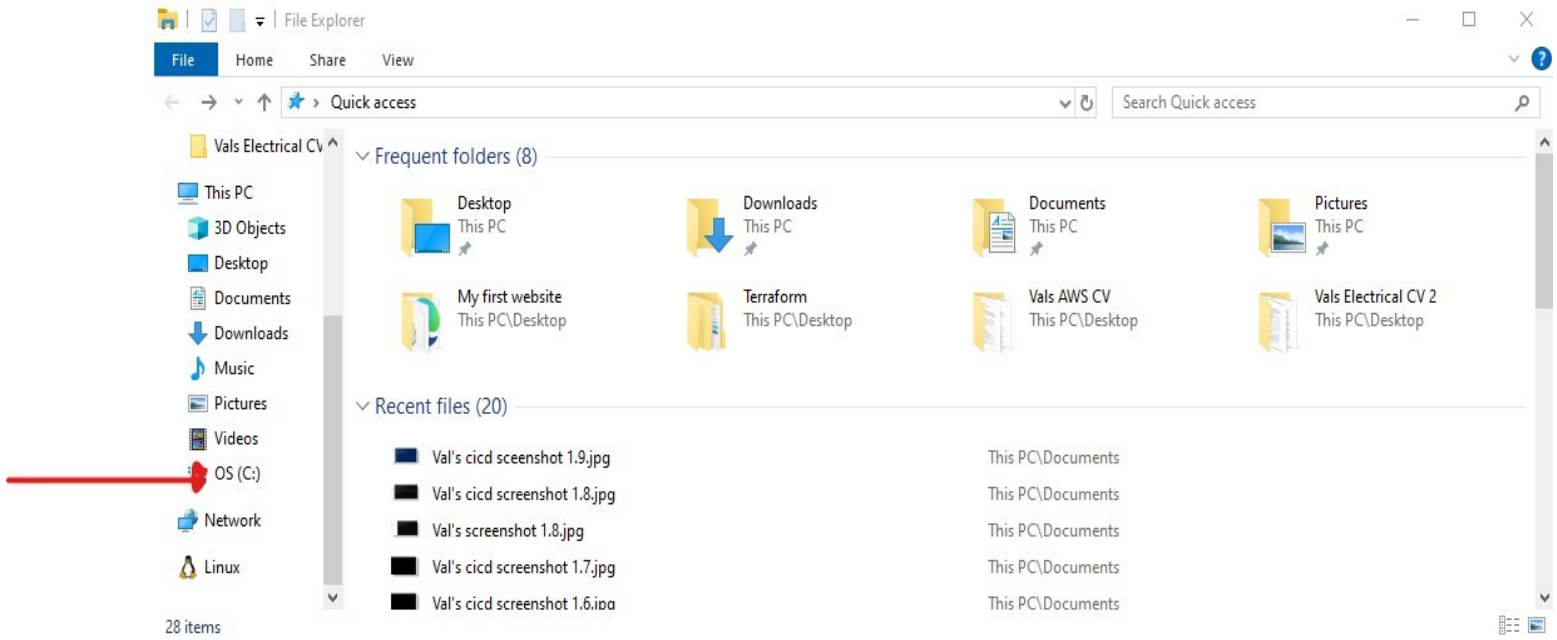


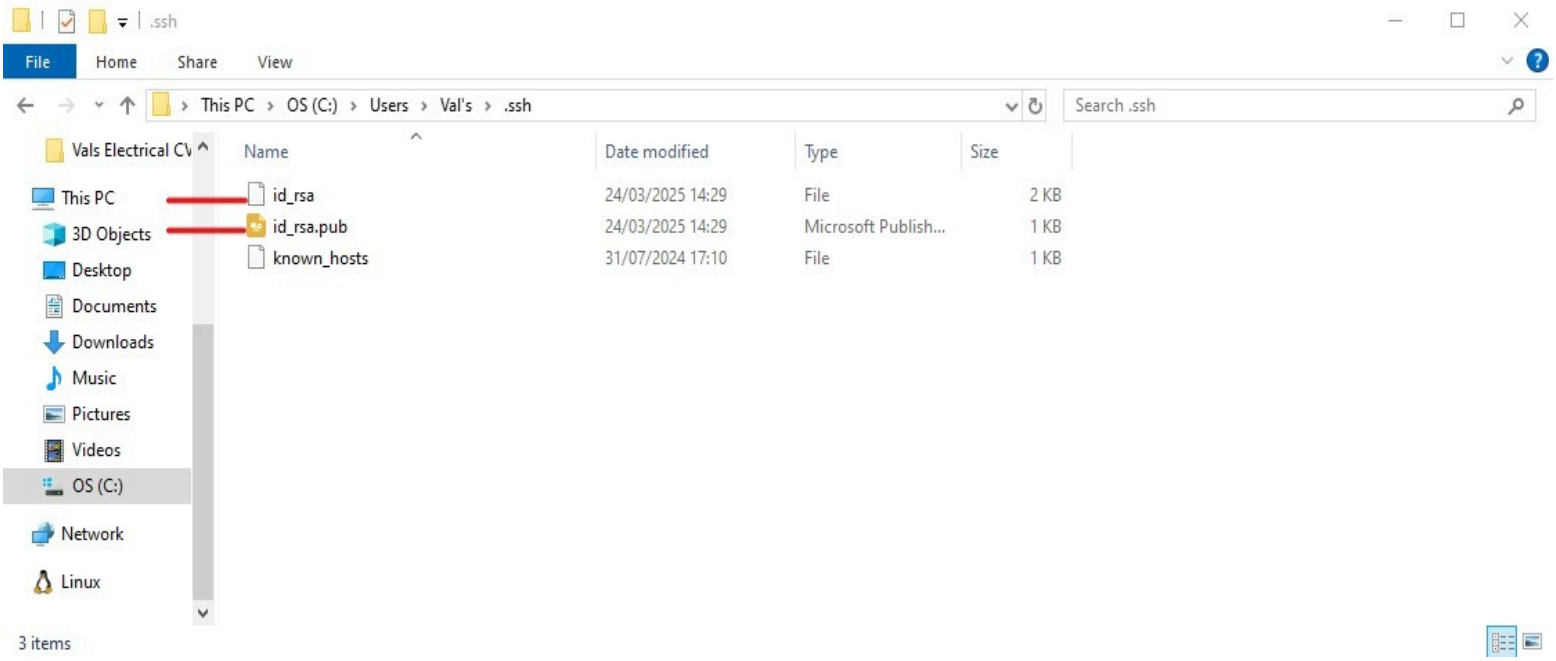
Diagram 1.5



3. Create a Keypair on your Computer

- To create a (Keypair) on my Computer, I have to open (PowerShell).
- To open power shell, click the (search) menu close to the (start) button and type (PowerShell).
- Click the Windows (PowerShell app), and it's going to open it.
- (Open) and (Run) it as (Administrator).
- To use (PowerShell) to create a (Keypair) on my Computer, I will type the command (ssh-keygen – t rsa – b 2048), and press (Enter).
- Note; this command means (t means type, and the type is rsa, - b means byte, and the byte is 2048).
- Next, it shows on the (PowerShell screen), (Generating Public/Private rsa Keypair) and next, "Enter file in which to save the key.
- Then it shows the (default directory), where my keys will be saved to, and it is always good to leave it in the directory (c:\users\val's/.ssh/id-rsa).
- Then press (Enter), then it shows the (keypair) and it's (random art image).
- So to confirm that the (Keypair) is saved in the directory (c:\users\val's/.ssh/id-rsa).
- To (verify) that my (Keypair) is saved in this directory, I will type (file explorer) in the (search menu).
- Select the C drive (local disk C:).
- Then select (Users).
- Then select (Admin or Name).
- Under (Admin or Name), look for and select (.ssh).
- In the (.ssh) folder, the Keypair is in there.
- So it is (confirmed) from following the directory path on my Computer.
- So the (Keypair) has been (Created, Saved and Verified).

Diagram 1.6



4. **Add the (Public Keypair) to GitHub**

- To upload the (Public Keypair) to my GitHub.
- Once that is done, then I will be able to clone my GitHub repository.
- To (upload) the Public Key onto GitHub.
- Open the (File Explorer).
- Go to the (location) where I saved my (Keypair).
- It's on the (C:), and go back to the (User), and under the (.ssh).
- The (Key) I want to upload is the (Key) with the (Pub) at the end.
- That is the (id_rsa.pub) key.
- I right clicked on it.
- Came down to open it with (Libre Office).
- Once the (Keypair) is shown on (Libre Office), copy it by pressing (Ctrl A), or just manually copy it.
- Then go to my GitHub account.
- Go to my (Profile Pic) at the (top right) and select (Settings).
- Under (Settings), select (SSH and GPG keys).
- Then give the (SSH Key) a Name, I will call it (pubkey1).

- Then (paste) the Keypair under the Name tag (box).
- Click add (SSH Key).

5. **Clone repository to my Computer**

- To clone the repository, I have to open the (cmd prompt) on My Computer.
- Changed the directory I want to clone the (terraform project) repository in.
- I want to clone mine to my desktop.
- To change the directory to my (desktop), type (cd desktop), this is the command to change to desktop.
- Then press (Enter).
- The type (Git clone), then go back to my GitHub account.
- On my (GitHub) homepage, select the repository (terraformpro), that I want to clone to my Computer.
- In my repository, select (Code).
- I can clone a repository (HTTPS, SSH or GitHub CLI).
- I will be using (SSH) to clone my repository.
- Select (SSH).
- Click below to copy the address.
- Then go back to my (cmd prompt).

- On my (cmd prompt) after (Git clone) and paste the address copied from (GitHub) in there.
- Now minimize the (cmd prompt) screen and click (Enter).
- Now I can see that the (terrformpro) has been cloned to my Computer (desktop) successfully.

I can verify this clone by double clicking on the (terraformpro) folder cloned on my desktop, and I can find the (gitignore) and (README.md).

I have successfully cloned my repository.

The next stage of my project is to;

6. Apply Terraform & create an IAM User with full Access.

- In this stage, I will create an IAM User with programmatic access.
- AWS CodeBuild will use these credentials to authenticate with my AWS environment.

Creating an IAM User

- Select (IAM) from the search box of my (AWS console).
- On the (IAM dashboard).

- Select (Users).
- Then click add (Users).
- Give the user a name (CodeBuild-User).
- Click on (Next), and go onto the (Set permissions) page.
- Under (Set permissions), click on (Attach policies directly).
- Then under (Permissions Policies).
- Select (Administrator Access).
- Then click on (Next).
- On the (Permissions Summary) page, under tags, click on (Create User).
- To create (Programmatic Access), click on the new User (CodeBuild-user).
- Scroll down to (Access Keys), under Access keys, click (Create Access Key).
- On the (Access key best practices and alternatives) page, scroll down and click on (other) and click on (Next).
- On the (Set permissions tag) page, click on (create access key).
- Now that the (Access key) and (Secret Access key) have been (Created), I can (download) them with the (CSV file).

7. Apply Terraform, create and clone a GitHub repository.

- Clone the repository on my Computer.

- I will use a terraform project to complete this task.
- I want to create a (CodeBuild) job that will automatically apply my terraform script anytime I make a (Commit) into my GitHub repository.
- To start, I will create a repository in my GitHub account.
- I will use the (new-ec2-instance) repository.
- Adding two files to this repository (ec2.tf), (install_techmax.sh).
- By clicking and dropping them into the repository.
- Scroll down, once the files have been dropped, and click on (Commit Changes).
- I have successfully added these 2 files into the repository I just created.
- The next thing I have to do is to clone the repository into my Computer.
- To clone the repository, I will select the (drop down) of (Code), and click on the (SSH).
- Then copy the link below it.
- Once that is done, open the (command prompt) on my Computer.
- In the (command prompt), change to the (directory) where I want to clone this repository, which in this case is my Desktop.
- I achieve this by using the (command prompt), (cd desktop), and press (Enter).
- Next step, is to type (git clone) and paste the (directory) from (GitHub) that I copied and press (Enter).
- Now that I have cloned my repository (new-ec2-instance) to my Desktop, I can verify it by double clicking on the folder (new-ec2-instance) to see that the 2 files from the repository are in there.

8. Apply Terraform & Test my Terraform script locally.

- Before building my CodeBuild project, I have to test my (Terraform Script) to make sure it’s working properly.
- I will open the repository, and clone in (Visual studio code) which I downloaded a while ago on my Computer.
- Then click on (New file) and then select (Open folder) to grab the repository I cloned to my Computer.
- Then go to the location of my project folder, which is on my (Desktop).
- Select it, then click (Select folder).
- Once I have uploaded the (Terraform folder) from the Desktop, (new-ec2-instance).
- Select the (ec2.tf).
- Then make sure that I provide the right profile that I have configured in my Computer on AWS.

Before I move on, I have to figure out (How to configure a named profile on my Computer), as I will need to enter this in my (Terraform Code) on (Visual studio code).

To configure a named profile on my Computer, I must install (AWS CLI) on my Computer.

Steps:

- Log into the (AWS Console).
- Install (AWS CLI) on my Computer.
- Create an (IAM User) with programmatic access.
- Use the user's credentials to configure a named profile on my Computer.
- After installing an (AWS CLI).
- I opened the (Command prompt), and enter the command (aws –version) and it popped up.

To create an (IAM User):

- Search for (IAM) in the search box.
- Select IAM under services.
- Select (Users) on the left-hand side.
- Click (Create User).
- Give the (User) a name (terraform-1).
- Click on (Next).
- On this page, I will add (Permissions) by clicking (Attach policies directly).
- Then select (Administrator Access), then select (Next).

- Click (Create User).
- Giving my (User) account programmatic access can be done by following the steps described a few pages ago.
- The (Access Key) is created.
- To (Configure a named profile) = Create an (Access Key) with programmatic access and (Secret Access Key).
- Next, I downloaded these as (CSV files), copy and paste them onto my (Note pad).
- For Security purposes, I had to delete these after using them and can't post them.

- Now that I downloaded the (CSV file).
- Terraform will use the Users credentials with (Programmatic Access) and (Administrator process), to create resources in the IAM (User Account).

Now to (configure a named profile) on my Computer:

- Open my (Command prompt).
- I will use this command to create a (named profile) for my (IAM User).
- This is the command to do this; (aws configure -- profile name).
- I changed the name.
- I will choose a name similar to my (IAM User) name which is (terraform-1).

- So the full command becomes (aws configure – profile terraform-1).
- Type this in and press (Enter).
- Enter the (Access Key ID), then press (Enter).
- Enter your (Secret Access key), copy and paste it, then press (Enter).
- Enter the (Default region name) as (us-east-1), press (Enter).
- Default output format (None), keep this empty, press (Enter).
- I have successfully created a (named profile) for my (IAM User) and these credentials are stored stored on my Computer.
- To see where these (credentials) are stored on my Computer.
- Open to see where these IAM credentials are stored on my Computer.
- Open the (file explorer).
- Select the (C:) which is on my Computer.
- Select the (Users) folder.
- Select my name which is (Val's) here, then click the (.aws folder).
- To see the (credentials) of the file, I have to select and right click and click on (open), either the (config or credentials) file.
- Open with (Notepad).
- I also updated my User profile name e.t.c
- Now I can change the profile name on my terraform user which is (profile = iamadmin-general) if I want to.

Steps to apply Terraform:

- Open the (terraform script).
- Change the profile to the right one (iamadmin-general).
- Then scroll down to update my (Keypair) name which is (terraform-1Keypair) which is the (Keypair name) of my (AWS Account).
- I have to (Save) this file by selecting (File) and select (Save All).
- To (Run) terraform apply, right click anywhere at the top left, and select open in the (Integrated terminal).
- This opens the (PowerShell terminal).
- Next, I am going to (initialize) with my (AWS environment) and I am going to type (terraform init) to initialize.
- Once this is typed, press (Enter), it is now initializing.
- Now I successfully initialized with my(AWS environment).
- Once I have initialized it, the next thing that I will type is (terraform apply).
- Type (terraform apply) and press (Enter).

9. Apply Terraform & create shell scripts.

- When I create a (CodeBuild) project, (CodeBuild) will launch a (Container) that it will use for my build job.
- In this part of the project, I will create (Shell scripts) for the things I want to install in the Container.
- I will create (Shell scripts) to install terraform, configure a profile and (Run) the (terraform command).
- To create the (Shell script), open the project folder in (Visual Studio Code).
- Once the folder is opened, I am going to create another folder in the Project's folder.
- To create another folder, right click anywhere on the (VS Code) screen.
- Then select (New folder).
- Give the folder a name.
- I will call my folder (vals-cicd).
- I will put all of my (cicd scripts) in this folder.
- Then press (Enter).

- Now I have to install terraform on the (Container) that (CodeBuild) will use for the (Build job).
- The first (Shell script) I will create is the (Shell script) to install terraform on the (Container).
- To create the (Shell script), right click on the (cicd).
- Select (New file).
- Create a (file) in my (vals-cicd) folder.

- Call the file (install-terraform.sh), then press (Enter).
- Open the reference (script file) from the (GitHub link).
- Copy it onto my (Notepad) – This is link file to install terraform on a (Linux Container).
- Now paste the linked file onto the (VS Code) screen, under the (install-terraform.sh).
- Press (ctrl A) to select everything in the file.
- Right click to copy it.
- Then close it.
- In the (install-terraform.sh) file I created, paste it in there.
- This is the (Command) to install terraform on a (Linux Machine).

Note:

These commands already exist in the (AWS terraform documentation), so all I did was copy this command and paste it in my (Shell scripts) file.

Steps:

- Type (Download terraform).
- Go into the page and select (Linux).
- Select install (Amazon Linux).
- Then it comes up with the (Command) to install terraform on (Amazon Linux).
- So copy and paste it onto my (Notepad).
- This is how to make up my (Shell Script).

- I only added (terraform –version) to verify that terraform was successfully installed and the other command I added is on line 4 (set -eu), which will allow the (Shell Script) to fail if there is any error in it.
- To (Run) my script, I have to enter (#!/bin/bash).
- The next thing to do is to (Save) my file.
- I will do this by (Selecting file), click on it and select (Save All).
- The second (Shell script) I will create is the shell script that I will use to configure a (named profile) on the (Container).
- Right click on my (cicd) folder and select (New file).
- Give the file a name (pro-config.sh).
- Then open the reference file to configure a (named profile) from the previous page of this project.
- Enter these lines of Code, but I am going to be using the (Environmental variables), because I will only be able to interact with the (Container) with these (Environmental variables). e.g. ($AWS_ACCESS_KEY_ID).
- Now I'll pass my credentials onto my (Container).
- Next, save this second (Shell script) to be used in the (Build job).

- Use the same procedure from the previous process to save this (File) by clicking on the file and clicking on (Save All).
- The third Shell script) I will create with (Run) my (terraform init), (terraform apply) & (terraform destroy).
- Right click on my (cicd folder), select (New file), give the file a name and call it (apply-terraform.sh), the press (Enter).
- Then open the (reference file).
- Copy it and paste it in my (apply-terraform.sh) script.
- This is the (Shell script) I will use to (apply) my (terraform project) in the (Build job).
- Save this (File), by selecting (File) & select (Save All).

10. Apply Terraform and Create Builspecfile

- In this part of the project, I will be building the (buildspec.yaml) file.
- This file is a collection of (build commands) and related settings in (yaml) format, that (Codebuild) will use to run the (Build job).
- The (buildspec file) contains a list of (commands) that (CodeBuild) will use to run the build.
- Note: The (buildspec file) is always in (yaml format).

- To create the buildspec file for this project, I have to go back to (Visual Studio Code).
- In (Visual Studio Code), I will create a file in my (cicd) folder, which I label as (vals-cicd).
- Right click on my (vals-cicd) folder, and select (New file).
- Give the file a name calling it (buildspec.yml).
- Then press (Enter).
- Now that I have created the (buildspec) file, the next thing I will do is to open the (buildspec file download link) from (GitHub).
- Copy and paste this link onto (Visual Code).
- This is the link that I will use for my (CodeBuild) job.
- Next, I will select everything inside this folder on the screen by pressing (ctrl A) to select, right click to copy, then paste onto my (buildspec) file.
- To explain what's in the (buildspec) file:
- The first part is (version 0.2), recommended by AWS.
- The next section is the phases; which consists of the (install phase), (prebuild phase) and the (build phase).
- In the (install phase), I will install any software I want to (install) onto the (Container). For example, in this phase I have (python 3.x)-which I will install.
- The next phase is the (pre-build) phase.
- In this phase, I will install (terraform) and configure the (named profile) on the (Container).
- (CodeBuild) will clone the repository onto the (Container).
- Once (CodeBuild) clones the repository onto the (Container), it will be in the directory to the (top left-hand side of my (Visual code) screen.
- The first command on the reference file copied onto the (buildspec.yml) folder is (cd) into the (cicd folder).
- Once this is done, I will make the (Shell script) file executionable and this is done by the next command.
- This is the command to make my (Shell script) file executionable. This command is (chmod +x).
- Note: I have to change a couple of the names in my files before executing it e.g. from (configure - name - profile.sh) to (pro – config.sh), which is my own file name.
- Paste it onto my (Visual Code).

- Then the third file I want to make executionable is (apply-terraform.sh).
- Once these files are made executionable, the next command I will (Run) is (./install-terraform.sh), and this is the command to (install terraform).
- If I go into my (install-terraform.sh) file and click on it, it will (Run) all the (commands) in this file.

Note: I had to change the file name as I had some issues because I did not change the names correctly.

- Going back to my (buildspec.yml) file.
- The reason why I use (Shell script) file to (Run) my (commands) is because my (buildspec file) much clearer.
- The next (command) that I will (Run) is the (configure-named-profile.sh) command on the (Container) to (configure my named profile).
- Also, I have to change the (configure-named-profile.sh) on line 13.
- When I run this script, it is running in (pro-config.sh).
- This is because I am using an (Amazon Linux Container), AWS CLI already comes with it, that's why I am not installing it on the (VS Code Screen/Window).
- Once the (terraform) and the (configured-name-profile) are installed, the next stage/phase is the build phase.
- In this phase, the command I am using is (./apply-terraform.sh).
- By running this (Shell script), what will happen is it will run all the commands in the (apply-terraform.sh) file.
- Also, by opening my terraform apply file (apply-terraform.sh), I found out that I typed (error) as (erroe) on line 3 (#fail on any error) and that's why my terraform in this previous project (initialized), but didn't (apply).

- The first thing I will is change my (directory) back into my terraform projects (directory).
- Remember to run my (terraform commands), I have to be in my (terraform projects directory).

- (cd..) means that I want to exit out of the (cicd folder) to come back into the (NEW-EC2-INSTANCE) folder; this is the directory where my (terraform file) is in (ec2.tf).
- Once I am in my terraform folder, (CodeBuild) will run (terraform init) and run (terraform apply).
- This is how to use (AWS CodeBuild) to create the resources on a terraform project in an AWS account.

Going back to the (buildspec file) = (buildspec.yml) and save my file by selecting (File) and select (Save All).
Another thing I can do is to (Push) all the changes that I have made to my (GitHub repository).

Steps:

- Select (Source control).
- Type in (Created new files), then click (Commit).
- Once I have committed, click (sync changes), and it will (Push) all the changes into my (GitHub) repository.

11. How to create a Personal Access Token

- I will create this in (GitHub).
- The build job I will create will use the (Personal Access Token) to authenticate it with my (GitHub account).
- To create a (Personal Access Token), in my (GitHub Account), select my (Profile) at the top right corner of my (GitHub page).
- The select (Settings).
- On the (Settings) page, scroll down and select (Developer Settings).
- Click (Personal Access Token).
- On the (Personal Access Token) page.
- Select (Generate new token).
- Give the token a name (vals-token).
- Next, under (Expiration), select the amount of days I can use the token before it expires, and I will use (90 days).
- Under (Select scopes), select (repo permissions), then (full control of private repositories).
- Scroll down and give the token (admin: repo_hook).
- These are the (two permissions) I need.
- Scroll down and press (Generate token).
- I have successfully generated my (Personal Access Token).
- Next, I will (copy it), and (Save it).

- I have to (Save it) because once I close the page, I won't be able to see the (Personal Access Token) again.
- Save it on my (Notepad).

12. Apply Terraform & store terraform file in a Bucket.

- When I create a build job that will automatically apply my (Terraform Script) and create the resources in my AWS account, I have to store my (terraform state file) in an S3 Bucket.
- In this part of the project, I will show how to store my (terraform state file) in an S3 Bucket.

Steps:

- In the (AWS Management console), type (S3) in the (search bar).
- Select (S3) under services.
- On the next page, click on (Create Bucket).
- Remember that the name of the Bucket has to be (Globally unique).
- Give the Bucket a name (vals-terraform-state-bucket).
- The (Region) I chose to create the Bucket in is (us-east-1).
- Scroll down to (Bucket Versioning), leaving everything else as default.
- Enable (Bucket Versioning).
- Then scroll all the way down and click on (Create Bucket).
- Bucket is created.

This is the Bucket that I will store m (terraform state file) in.

- Select the Bucket (vals-terraform-state-bucket).
- The next thing is to go into my (terraform project).
- Open the (VS Studio Code).
- In the (terraform project), open the file (ec2.tf).
- Under the provider block, press (Enter) twice.
- Then open the reference (syntax) – GitHub URL link.
- Copy and paste it onto the (VS Code) of (ec2.tf).
- Then enter the bucket name.

Note: When terraform stores my (state file), I specified that first terraform will create a folder called (build), and in that folder it will store my (state file) in there ("build/terraform.tfstate"). This is a good option to use if I want to store the (state file) for (multiple projects) in the same (S3 Bucket).

- Region = (us-east-1), where the Bucket is stored.

- Profile is used to (authenticate) with my (AWS Environment) which is (terraform-1).
- Next, save this file by (Selecting file).
- And Select (Save All).
- Once this is saved, the next thing to do is run (terraform apply) to test that this script is working.
- Right click on the (ec2.tf) file.
- Select (Open in integrated terminal).
- In the terminal, the first command I will type is (terraform init) & press (Enter).
- It came up with a (Back-end configuration error), which I have to troubleshoot.

TROUBLESHOOTING

- A Back-end configuration error during (terraform init) usually means there's a problem with how you've defined your (terraform back-end block). This back-end is what terraform uses to store its state e.g. (S3) for AWS.

Steps:

- Checking for quotation marks around strings.
- Don't use commas in HCL – (HCL stands for HashiCorp Configuration Language).
- Correct indenting (terraform is strict with syntax).

Run Re-initialization

- In my project directory, run the command (terraform init-reconfigure) – This forces terraform to re-initialize the back-end.
- Accept your updated back-end configuration.
- Potentially prompt you to migrate state if needed.
- I did this and the terraform (successfully configured the back-end S3).
- Terraform is successfully i**nitialized**.

The next step to do is to run (terraform apply), and to make sure that my script is working properly.

- Using the command (terraform apply), and press (Enter).
- Then when the terraform shows me the plan, type (yes), and it should create the resources in my (AWS Environment).
- The process takes a bit of time so I will wait for it.

- I have come up with another error which this time is a (creating ec2 instance: operation error) – stating that my Keypair (terraform-1keypair) does not exist.

TROUBLESHOOTING THIS ERROR

- The (Invalid KeyPair) not found error means AWS can't find the KeyPair I am referencing in my (terraform code. step ly step fix).
- Check the Key name in my (Terraform Code).
- Open your ec2 instance resource and look for (key_name = your_key-name) – AWS is case sensitive.
- Verify if the KeyPair exists in AWS by running the command (aws ec2 describe-keypairs –key-names your-key-name).

SOLUTIONS

- Option A: Use an existing Key if (available).
- AWS Console, go to (EC2), go to (KeyPairs).
- Copy the correct (Key name).
- Paste it onto your (terraform config) like this – (key_name = "correct existing – key").
- I took the following steps to create a keyPair in the AWS Console:

- Click (Create KeyPair).
- Set – Name (vals-ec2-key).
- KeyPair created.
- Next, go to my (terraform file) to update my Key Name (vals-ec2-key).
- Then press (Save All) to save it, now the change is updated.
- Going back down to (Run) the command (terraform apply) and (Enter).
- Terraform is applied successfully after troubleshooting.

I can see that it has created resources in my (AWS Environment), which means that my (Terraform Script) is working.

- Going back to the (S3 Bucket) I created to verify that the state file (vals-terraform-state-bucket).
- Open (S3) in the (AWS Console).
- I can see the (build) folder in it.
- When I click on the (build) folder, I will find the (terraform.tfstate).
- Now that I have verified that my (terraform state file) is in my (S3 Bucket), I can delete my resources.

- Going back to my (Terraform project).
- In my (terraform project), I am going to type (terraform destroy) then press (Enter).
- Enter a value of (Yes).
- This is how I delete resources from my (AWS Account).
- Resources are successfully **Destroyed.**

13. Apply Terraform & Create Build project in CodeBuild

- In this, I will create a project in (CodeBuild) that will automatically apply my (terraform script), anytime I make a (Commit) to my (GitHub) repository.
- Before I create my build project, I have to make sure that any changes that I made to my project file, I push to GitHub repository.
- Once I have (pushed) the changes to my repository, the next thing is to (log) into my (AWS Console).

Steps:

- In the (AWS Console), type (CodeBuild) in the searchbox.
- Select (CodeBuild) under (Services).
- Then click (Create project).
- Give this project a name (new-ec2-instance-build).
- Next, give it a description (CI/CD build project to apply terraform scripts).
- Scroll down to (Source), this is where I select the (repository) that my script is in.
- My Script is in (GitHub), so I will select (GitHub) from the drop down menu.
- Then to connect, click on (Manage Account Credentials).
- Select (Personal Access Token).
- Grab my (Personal Access Token) from (GitHub) and paste it in the (AWS Console) and continue the process.
- Now I will post my (GitHub) repository link. This is the repository that I want to use for my build project which is (new-ec2-instance).

- Scroll down, under (Additional Configuration), I don’t need to select anything in here.
- Under (Primary Source webhook events), make sure (Rebuild every time a code change is pushed to this repository) is ticked.
- This is going to rebuild my code anytime I push a change/changes in my (GitHub repository).
- Scroll down, the opening up (webhook event filter groups), and under (Event type).
- Select (PUSH) and (PULL_REQUEST_MERGED) – This means that anytime I merge a (pull request) into my (repository), or (push) an update into my repository. It is going to rebuild my project.
- Select these two options & click out.
- Go down under (Environment Image), make sure (Managed Image) – (Use an image managed by AWS CodeBuild is ticked).
- Next, under (Operating System), select the drop down to choose (Amazon Linux).
- Next, scroll down under (image) make sure the latest version is selected.
- Under (Service role), go to (Role name), make sure the (Role name) is there, which in this case is (codebuild-new-ec2-instance-build-service-role).
- I can use a new name if I want, but I will keep the name the same.
- Next, under (Additional configuration), select the drop down and scroll down leaving the rest as (Default).

Note: The VPC option here is if the (CodeBuild) project will create a (Resource) in a VPC. In my case, I have specified the VPC in my terraform template – leaving them the way they are.

- Scroll down to (Environment Variables), this is where I would enter the values of the (Environment Variables) I specified in my (Shell script file).
- Now I have to go back to (Visual Studio Code) and open it.
- In (Visual Studio Code) and open it.
- In (Visual Studio Code), select my (cicd folder) and select the file (pro-config.sh).

- The first (Environment Variable) I have is my access key ID – (AWS_ACCESS_KEY_ID).
- Select and copy this, then go back to (CodeBuild), and paste it in the (name) section.
- Then click (Add Environment Variable), and click on it again.
- Next, go back to (Visual Studio Code) again.
- The second (Environment Variable) is (Secret Access Key), copy and paste it back on (CodeBuild).
- Next, I have to enter the values of this (Environment Variables).
- Go back to (VS Code).
- The third (Environmental Variable) is the (AWS Region); Copy and paste it back in (CodeBuild).
- In (CodeBuild), click on (Add Environment Variable), and paste it in there.
- The last (Environment Variable) I will add is my profile.
- Go back to (VS Code), then select (PROFILE_NAME).
- Copy and return to the (CodeBuild) to paste it.
- Click (Add Environments) and paste it in there.
- Next, In the (values) box, I will add the actual values.
- The first one is the (AWS_ACCESS_KEY_ID).
- Using the (CSV) file from the (IAM) created in previous page. This (CSV) file contains my users (Access key ID) and (Select Access key).
- Open the (CSV) file, the first value is the (Access key ID), copy and paste it onto my (AWS console) under the (Environment Variables Name).
- Paste the (Secret Access Key).
- The next option is (AWS_REGION) – This is the (Region) I want to deploy my terraform project in. To get to this value, I have to go back to (Visual Studio Code).
- In (VS Code), select my (ec2.tf) file.
- Copy the Region of the (Provider block) on the (ec2.tf). This is the Region that I want to deploy my (Terraform project).

- Paste it onto the (CodeBuild) in the (AWS Console).
- Next, for the (Profile Name).
- Going back to (VS Code).
- The (profile name), I am using to (authenticate) my (terraform-1).
- Copy (terraform-1), and then paste it onto the (CodeBuild) on the (AWS Console).
- This is how to specify my (Environment Variables).
- These are the variables I will use to (configure my profile) on the (Container).
- This way, terraform can use it to (authenticate) with my (AWS Environment).
- Next, scroll down onto (Buildspec), this is where I specify where my (Buildspec) is located.
- Going back to my (VS Code), then to my (buildspec file) which is located in my (cicd folder).
- Now I have to tell (CodeBuild) where my (buildspec file) is located and the name is (buildspec.yml).
- Going back to the (CodeBuild).
- Under (Buildspec), click on (Use a builspec file).
- Then (Enter) in the name in the (Buildspec name) section – (vals-cicd/buildspec.yml).
- Scroll down, leaving all the others as (Default), and click (Create build project).
- Done: I have successfully created the (Build Project).
- The name of the build project is (new-ec2-instance-build).
- Under (Configuration), I can see the current configuration of my (Build Project).
- My (Source provider) is (GitHub), and my primary repository where my (Script) is located.
- Scroll down under (build history), any build that I run, will show up under (build history).
- Under (Project details), all the details of my (Build Project) are there.

- I can modify any options underneath by checking (Edit).
- Under (Build triggers) and (Metrics), there’s no information.

- Before building, clicking on (Start build), (CodeBuild) will clone my repository where the (Script) is located. My (Script) is located in the (GitHub) repository.
- (CodeBuild) will clone this repository onto my (Container), then it is going to use my (buildspec file) to run all the commands I have specified on it.
- The (Build Project) I am creating will apply my (terraform script) and create any (resources) that I have specified in it to my (AWS Environment).
- In this project, my (terraform script) will create an (ec2 instance) and install the (techmax) website on it. Basically, (CodeBuild) will create the (AWS resources) and install them on the (Website).
- So when I click (Start Project), (CodeBuild) is going to apply my (terraform script) and create my (ec2 instance) and install the website on it.
- To elaborate on this, open a (new AWS tab).
- I will type (EC2) in the search bar of the (AWS console).
- Select (EC2) under (Services).
- I can (verify) that I have no (EC2 instances) running.
- So when I click (Start build) and my build is (successful), I should have an (EC2 instance) running and visible under instances and, I should be able to access the (website) that I installed on the EC2 instance.
- Next, going back to the other (first tab).
- Click (Start build).
- Now the build is in (progress).
- Under (build logs), I will see all the logs; what will happen is the same output I will see in the (management console), when I (SSH) into my (EC2 instance) and run my (command) under (build logs).
- It’s also going to show me those outputs on (view entire log).
- Right now, it’s installing the (Container) and doing all the things it needs to do before it starts running the (command) on my (buildspec).

- (Start build) has come up with an (error) so the (build failed) which can happen.
- I have to (troubleshoot) this to fix this error.

TROUBLESHOOTING

- It is coming up with I am running into a classic (directory not found) issue during the (PRE_BUILD) phase of my (AWS CodeBuild) project.
- The (buildspec.yml) instructs (CodeBuild) to (cd cicd), but there's no (cicd) folder in the working directory.

HOW TO FIX

- Correct the folder path.
- Make sure the directory I am trying to (cd) into actually exists in my repository.
- Check my repository and confirm whether named (cicd) exists.
- If the folder is named something else like (vals-cicd) based on the line: (yaml location is /vals-cicd/buildspec.yml), update my (buildspec.yml to match.
- Checking through these steps;
- Checking my (new-ec2-instance) repository.
- Looking through my (buildspec) folder, I noticed that I had a typo error of (-cd cicd) instead of (-cd vals-cicd).
- Now that I have successfully updated my (buildspec.yml) folder, by committing it to the repository after updating my (VS Code).
- I have to re-build this CodeBuild.
- Next, there's another (error screenshot).
- This means my script is expecting the Environment Variable (AWS_ACCESS_KEY_ID), but it's not set and the script doesn't handle thay gracefully.

- After going through all the troubleshooting, the issue for the failure was me selecting the wrong (Environmental details) and (Amazon image version).

- Also, I can show the information by checking on the (phase details).
- Now that the (Container) is launched, it has cloned the repository under (DOWNLOAD_SOURCE) – This is where it cloned my repository from (GitHub).
- In my (install phase), it's installing my (python 3.13).
- All my (commands) are running from terraform.
- By typing the (ec2_public_ipv4_url) = ("http://44.197.123.79"), I launched the (techmax website).
- Also, I deployed the Website by using the (CodeBuild) to apply my (terraform script).
- Also, now if I go to the (EC2 management console) and click (refresh), I can see that the (EC2 instance) that my (terraform) built.
- My resources are built.
- I can use this (build project) to apply any (terraform script).

14. Run Terraform Destroy in CodeBuild Project

- Now that I have tested my (Build Project) and verified that everything is working well, I will clean up my environment.
- I will have to clean up my (AWS Environment). I will destroy the resources that I created with (terraform).
- I will use (CodeBuild) to destroy the resources created instead of manually destroying them in the (AWS Console).
- To do that, I have to go back to (Visual Studio Code).
- In (Visual Studio Code), select the (apply-terraform.sh) file.
- On line 13 of the code (terraform apply-auto-approve), comment it out.
- To (comment it) out, press (ctrl + /) or add the (#) sign in the beginning manually to (comment it) out.

- Next, on line 16 (terraform destroy -auto-approve), remove the comment by using the mouse at the front, pressing (ctrl + /) or just remove the (#) sign manually.

- When I make the changes and (commit) them to my (GitHub repository), that's going to automatically start my (build job), and my (build job) is going to run everything again but this time instead of (running terraform apply), it will run (terraform destroy).
- Next, (Save the file), then click on the (commit) button and type a message (update file).
- Click (commit), then click (sync changes).
- Now that I have commited the changes into my (GitHub repository), I will go back to (CodeBuild).
- In (CodeBuild), select (Build Projects).
- When I (pushed) my changes into my (GitHub repository), that has triggered the build.
- If I select (new-ec2-instance-build), I can see that the build is on (Build number 4).
- If I select the (Build name), and (phase details), I can see that the build has been submitted, and it's currently in the (queue).
- Now it's (provisioning) the (Container) that it will use to create my (Build job).
- Going back and clicking on (Build logs) to see the logs. Once the (Container) is (created), I will see the logs.
- From the logs, I can see all the installations and I can confirm them with the dates and times which match with the time of (Destroy/Build terraform).
- Scrolling down, I can see that it's (Destroying the terraform).
- Done/Complete – Terraform has (**successfully deleted**) all the (resources), and the build is successful.

Now if I go back to the (Website Techmax), I shouldn't be able to access it as I have (deleted) the (EC2 instance) that hosts it.

The next thing I have to do is go back to the (AWS Console), to see whether I still have the (EC2 instance-Techmax server) there.

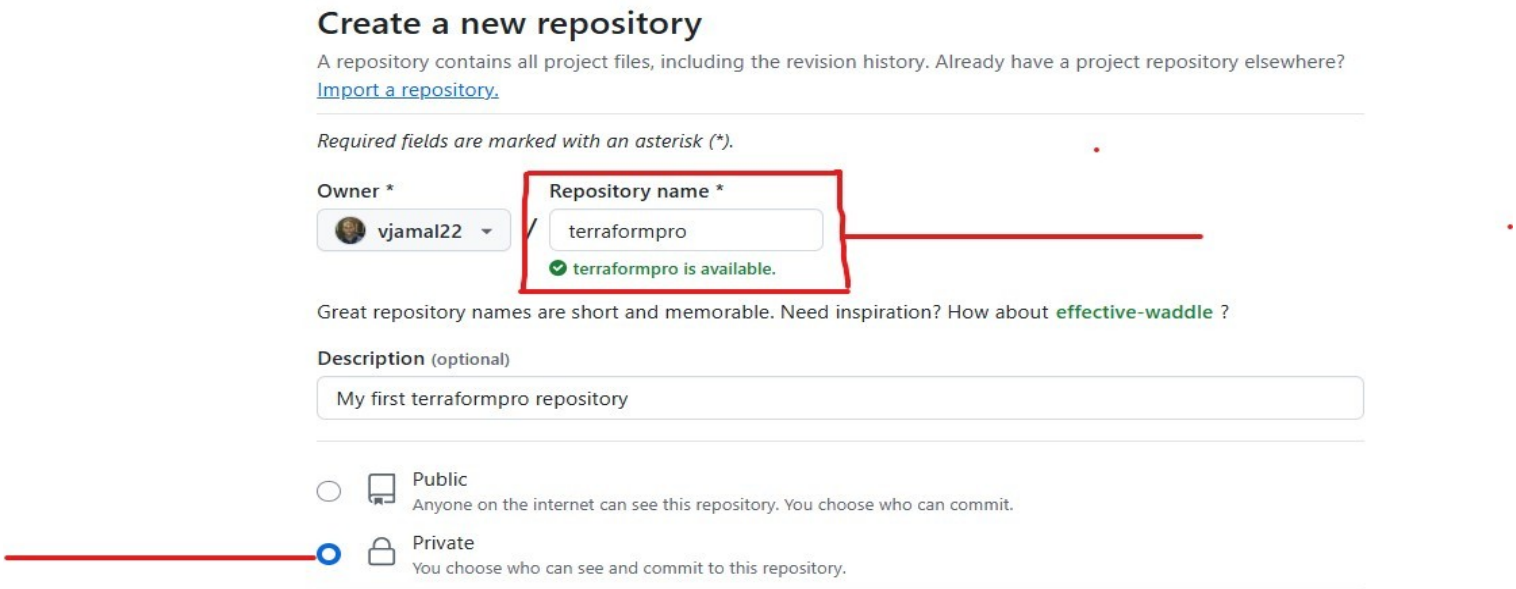
I can see it there but I when I click (refresh), the (EC2 instance) has also (terminated).

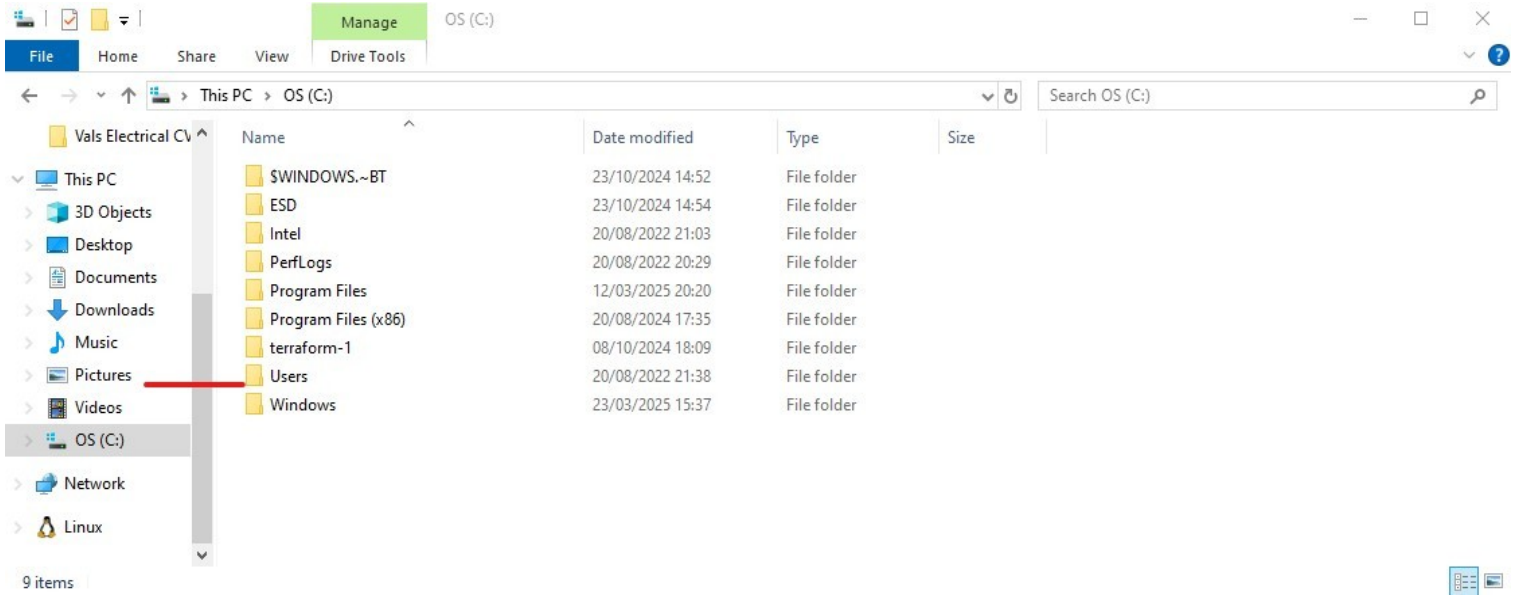
Going to (CodeBuild), I can (Delete) my (CodeBuild Project).

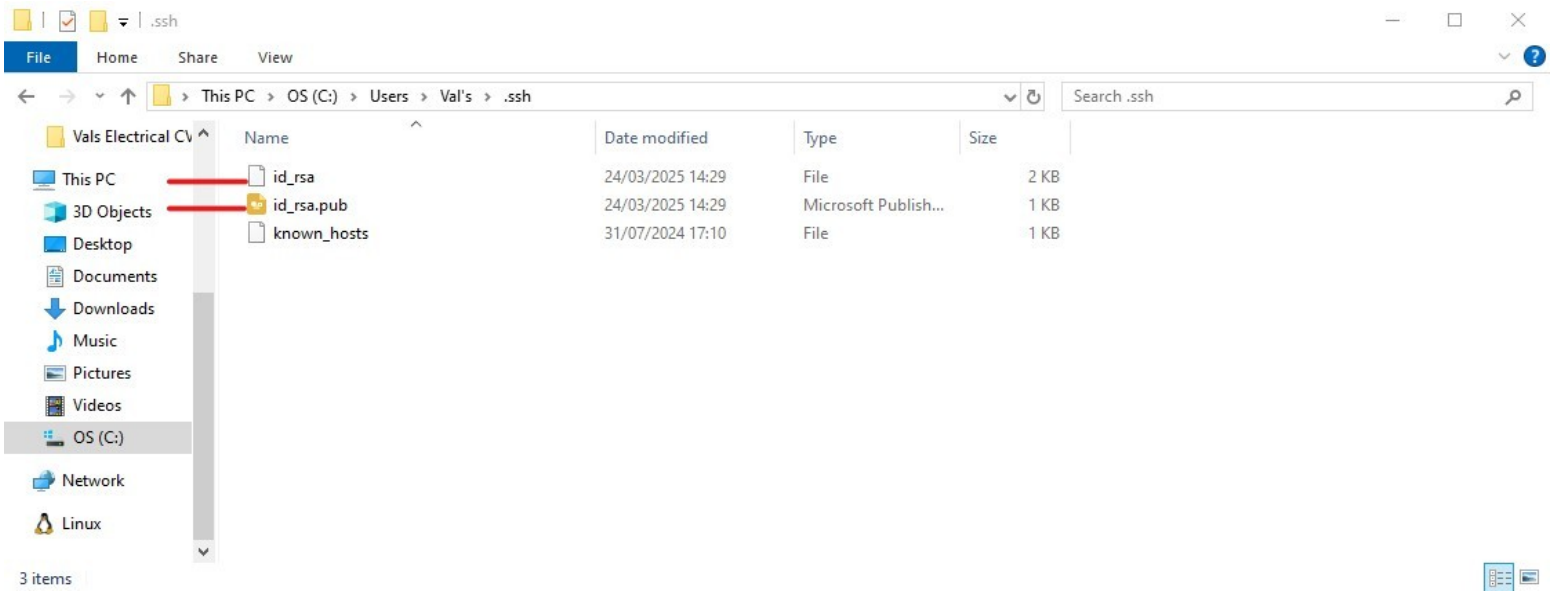
This is how I create a Build Project) in (CodeBuild) to automatically apply my (Terraform Project) and I can use this (Build Project) to apply my (terraform script).

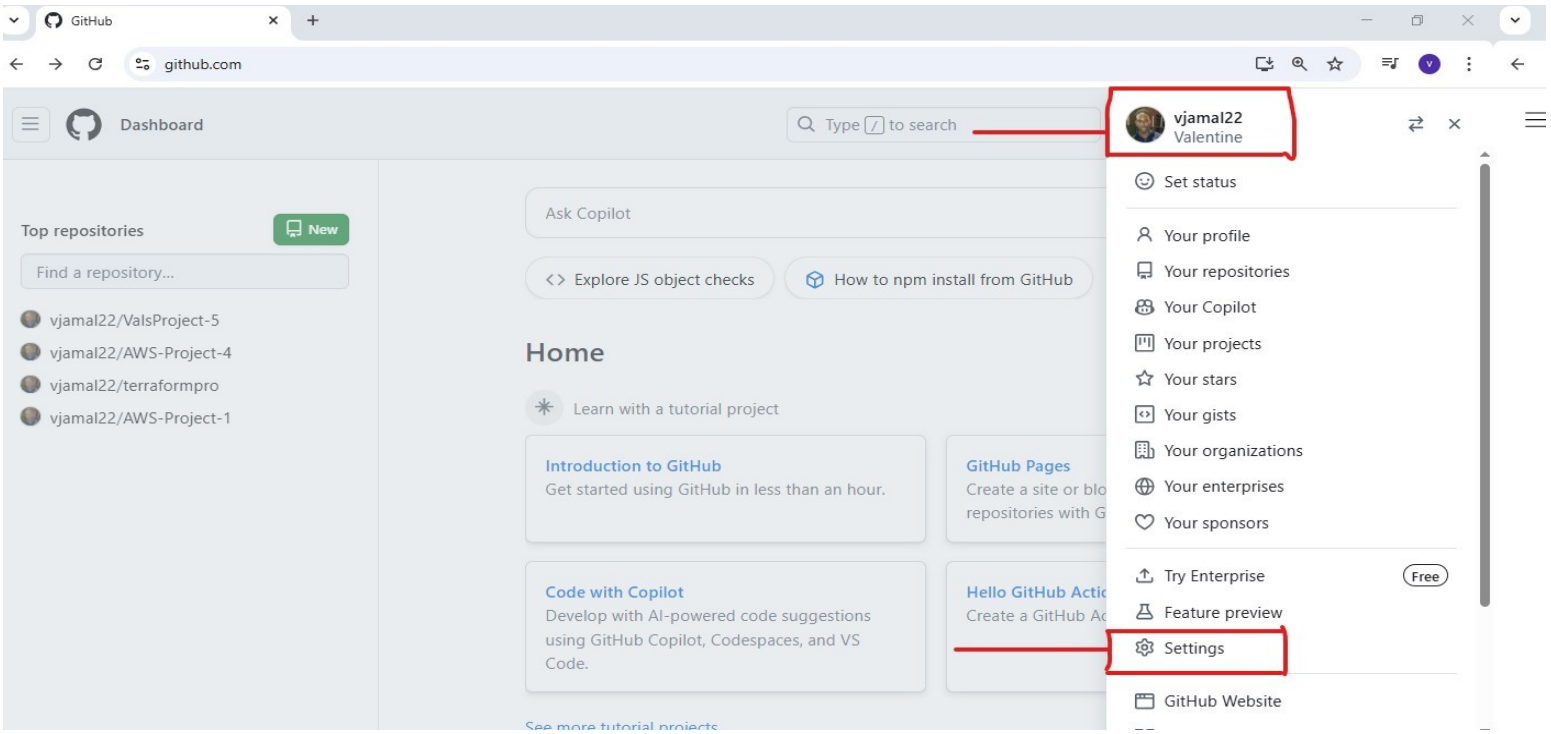
I have a lot of pictures/screenshots of this Project that I decided to paste below:

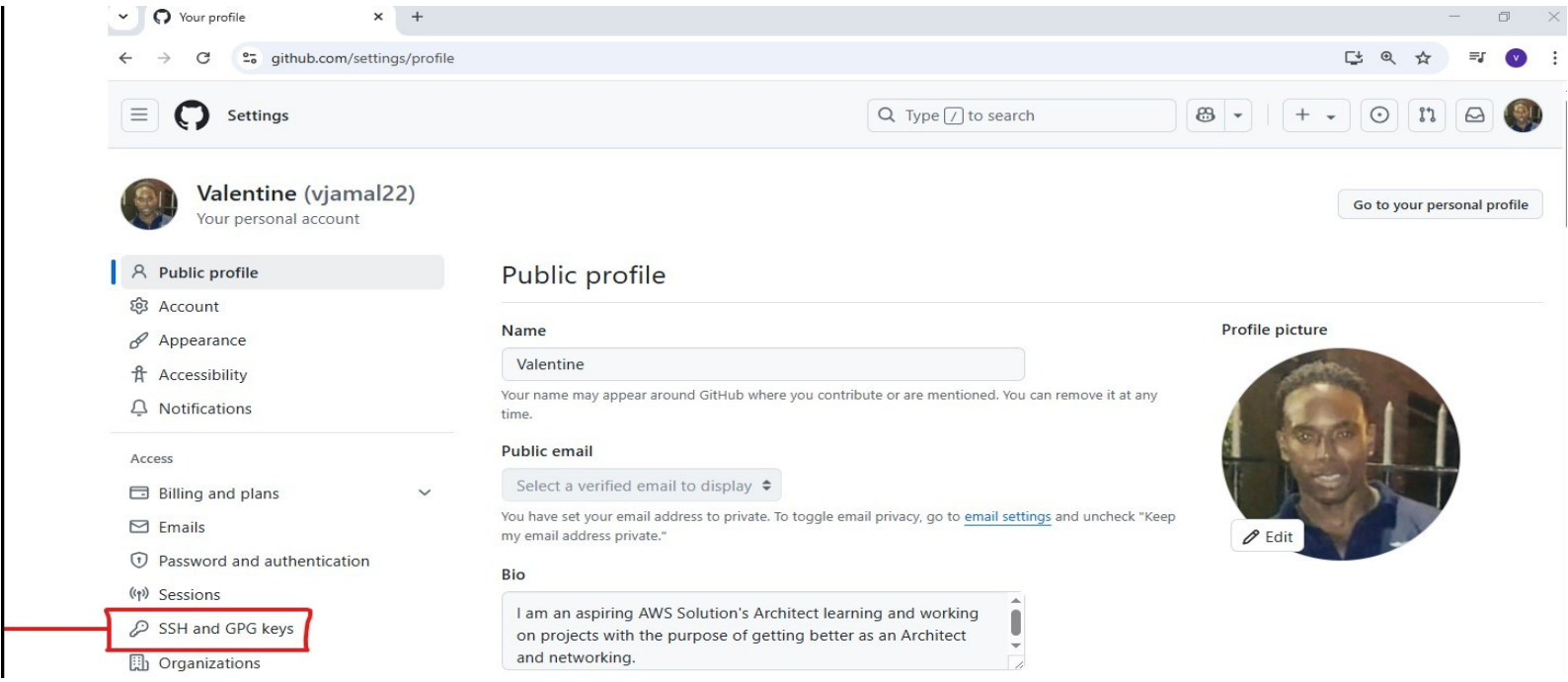
Diagram 1.7

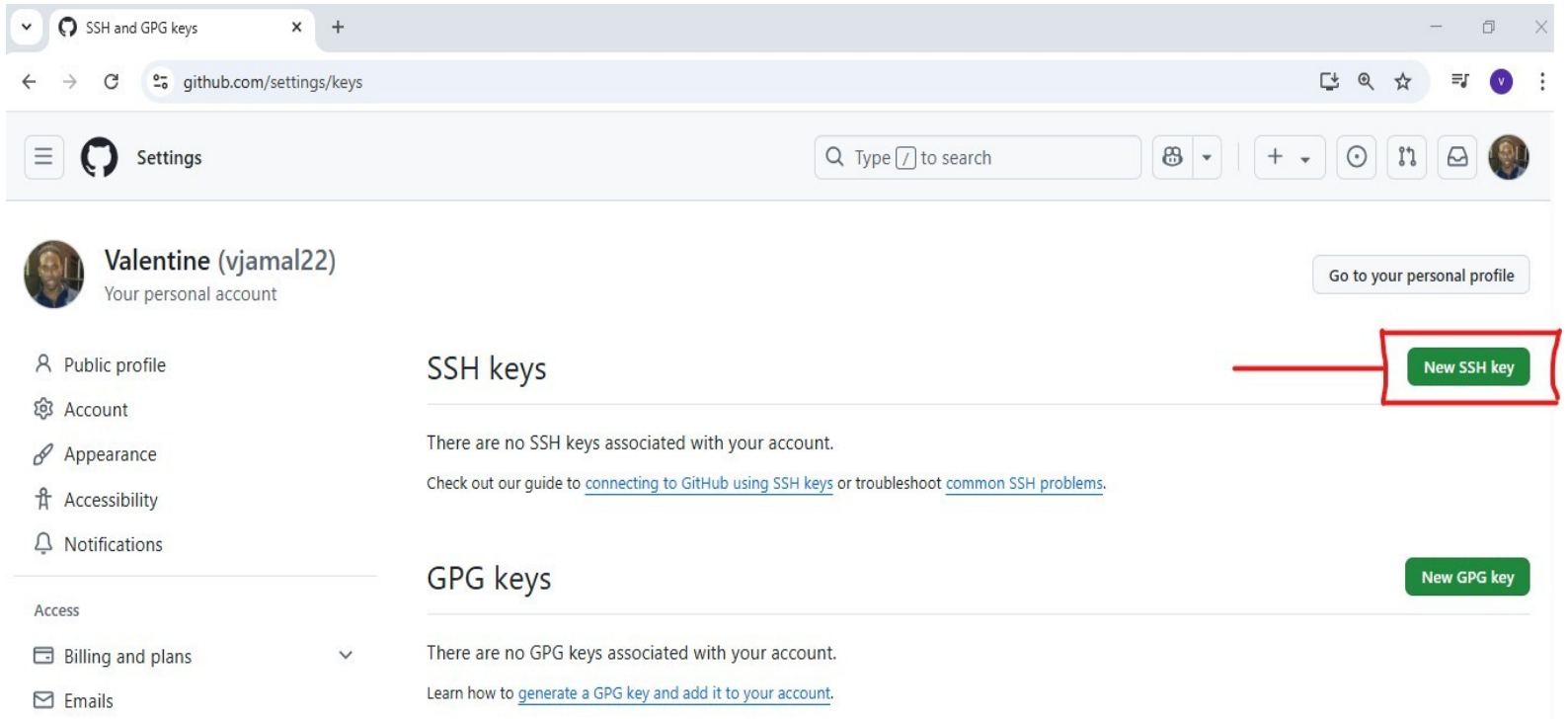


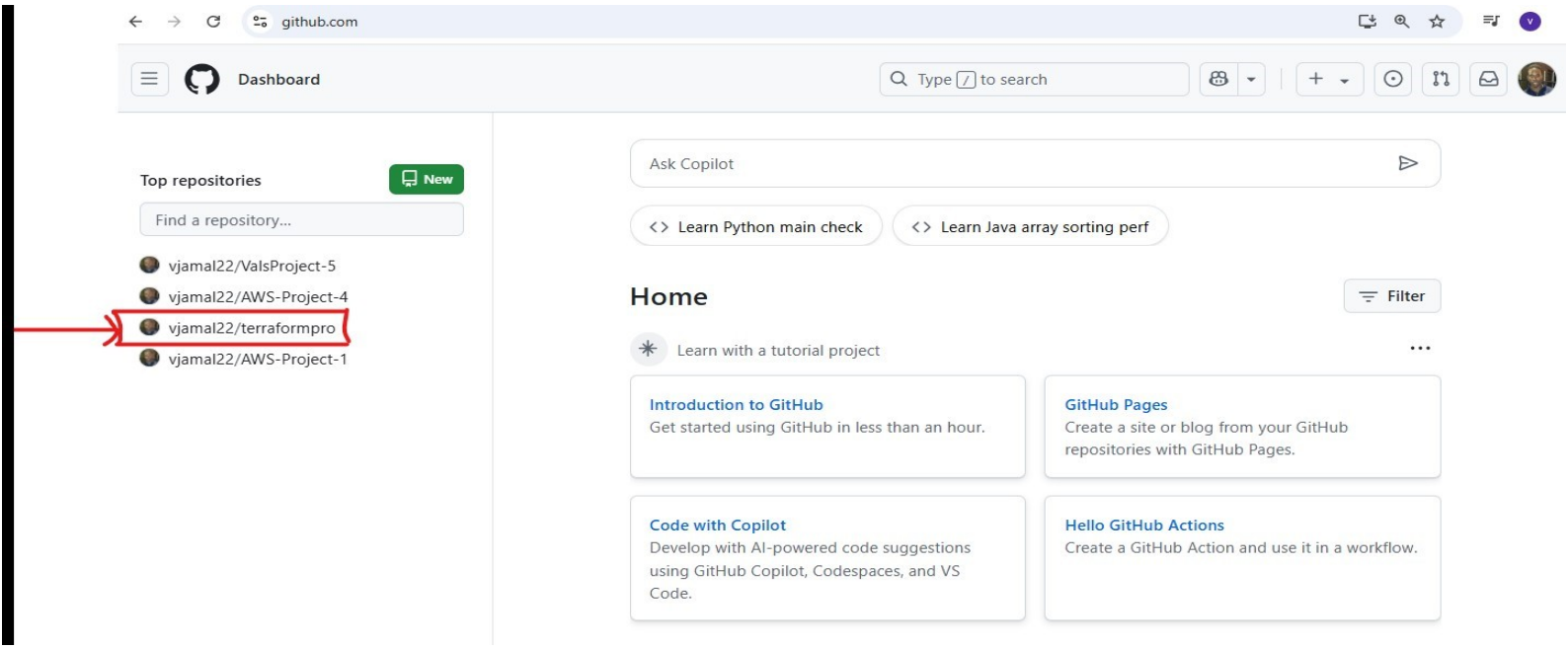


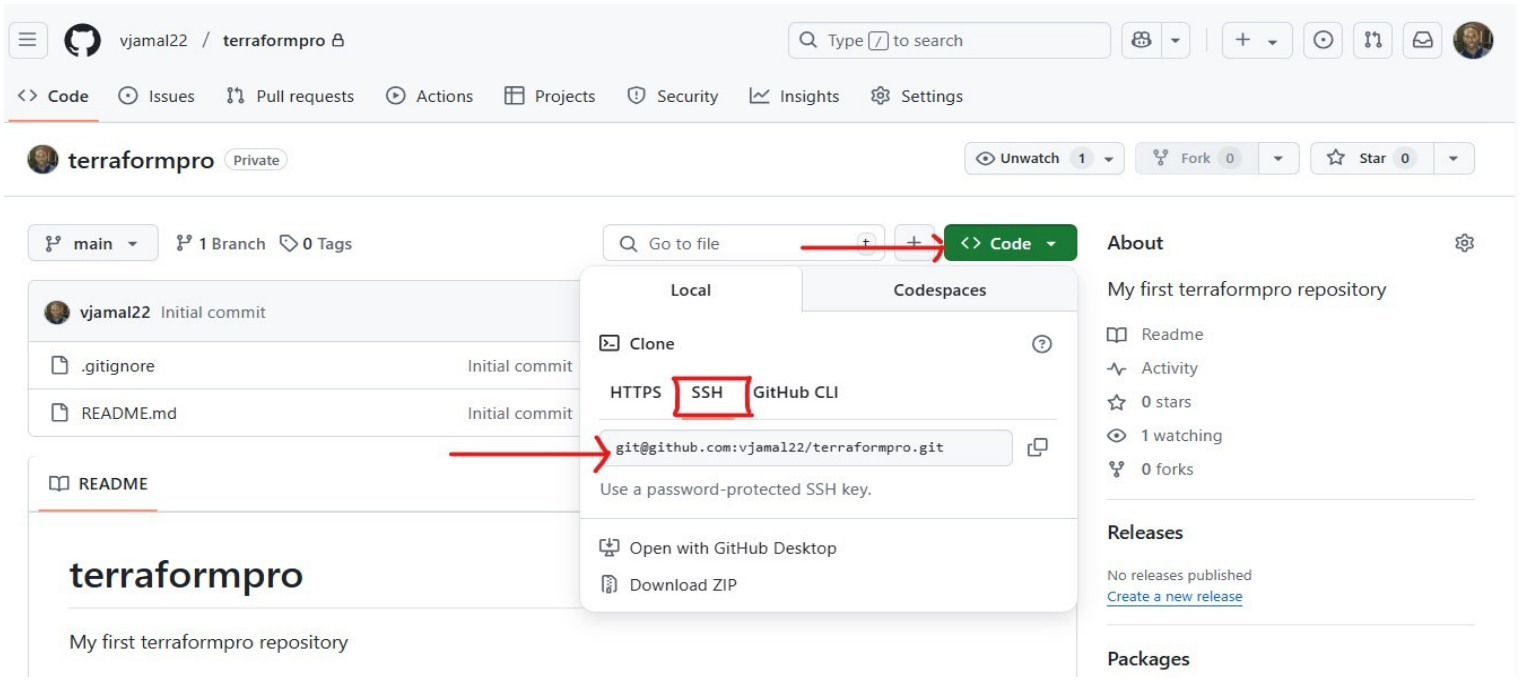

.ssh
File
Home
Share
View
This PC › OS (C:) › Users › Val's › .ssh
Search .ssh
Vals Electrical CV
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
Network
Linux
Name
Date modified
Type
Size
id_rsa
24/03/2025 14:29
File
2 KB
id_rsa.pub
24/03/2025 14:29
Microsoft Publish...
1 KB
known_hosts
31/07/2024 17:10
File
1 KB
3 items

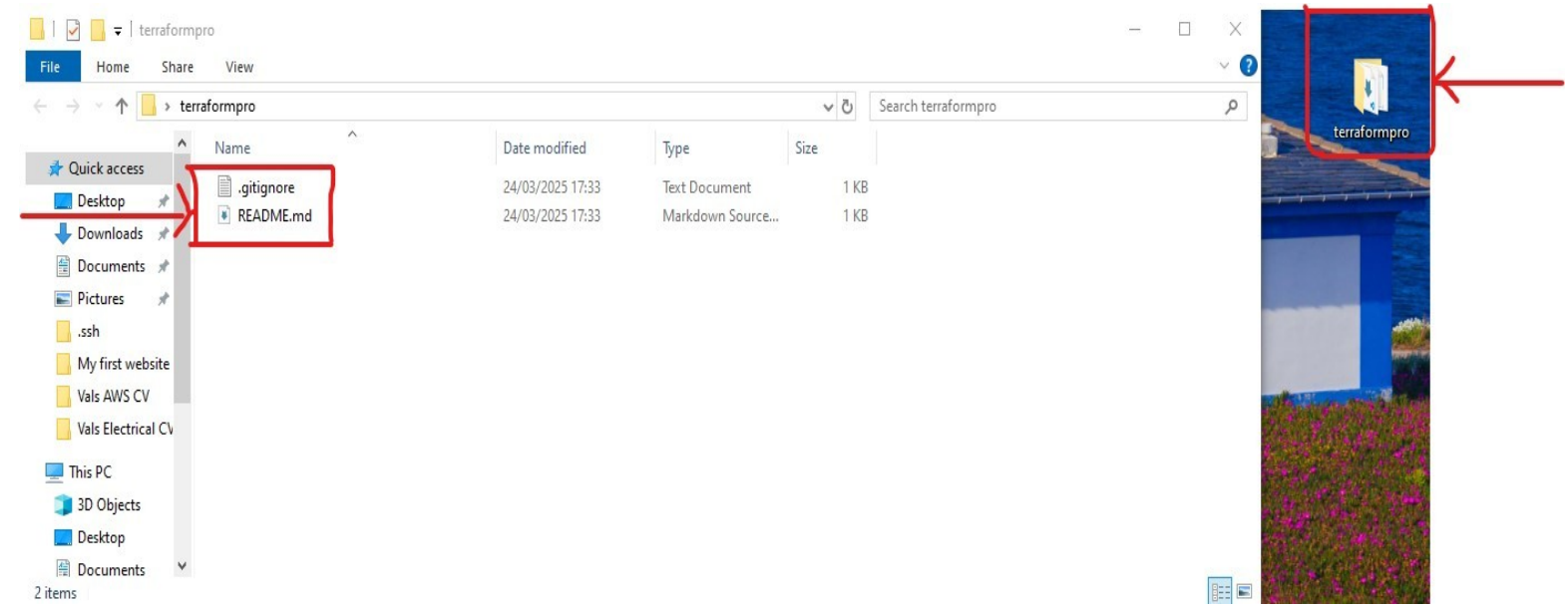

GitHub
github.com
Dashboard
Type / to search
vjamal22
Valentine
Set status
Your profile
Your repositories
Your Copilot
Your projects
Your stars
Your gists
Your organizations
Your enterprises
Your sponsors
Try Enterprise
Free
Feature preview
Settings
GitHub Website
Top repositories
New
Find a repository...
vjamal22/ValsProject-5
vjamal22/AWS-Project-4
vjamal22/terraformpro
vjamal22/AWS-Project-1
Ask Copilot
Explore JS object checks
How to npm install from GitHub
Home
Learn with a tutorial project
Introduction to GitHub
Get started using GitHub in less than an hour.
GitHub Pages
Code with Copilot
Develop with AI-powered code suggestions using GitHub Copilot, Codespaces, and VS Code.

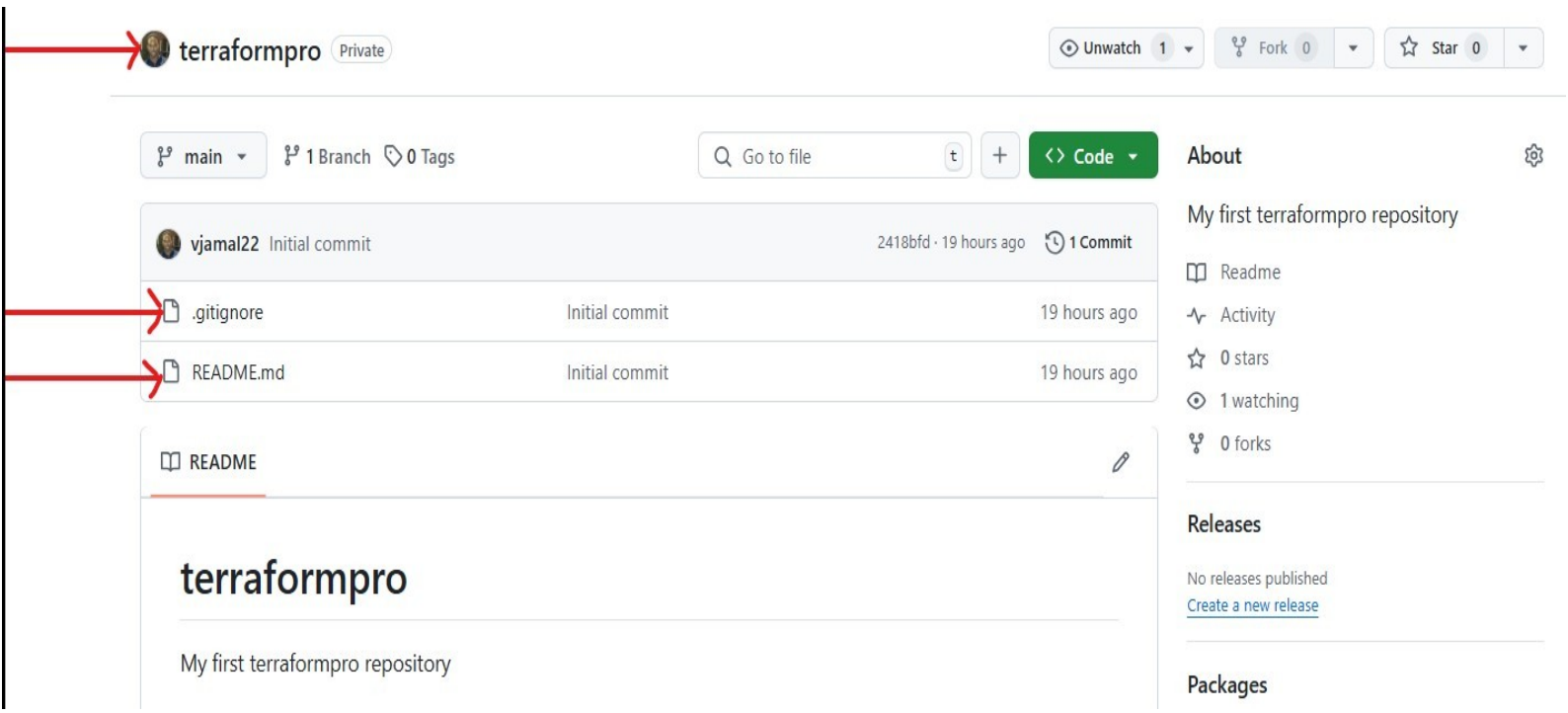

Your profile
github.com/settings/profile
Settings
Type / to search
Valentine (vjamal22)
Your personal account
Go to your personal profile
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Public profile
Name
Valentine
Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.
Public email
Select a verified email to display
You have set your email address to private. To toggle email privacy, go to email settings and uncheck "Keep my email address private."
Bio
I am an aspiring AWS Solution's Architect learning and working on projects with the purpose of getting better as an Architect and networking.
Profile picture
Edit

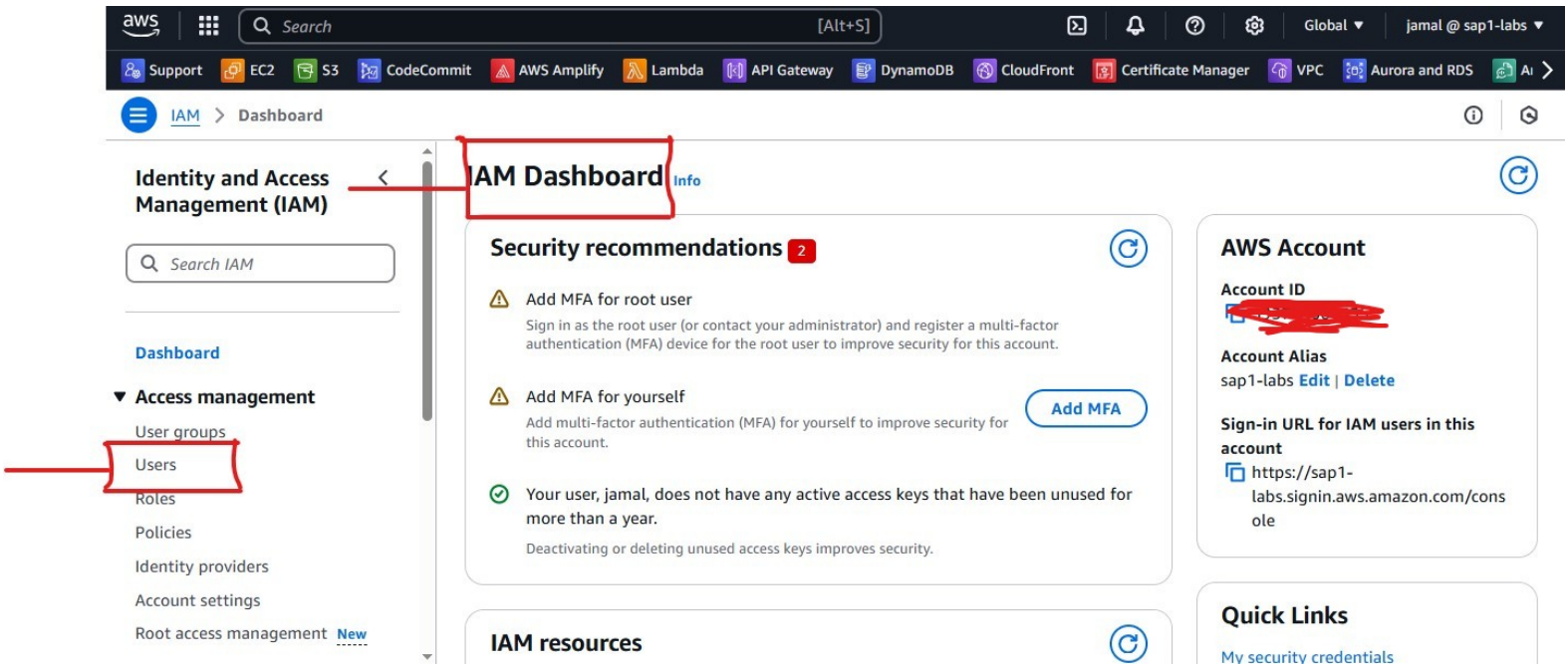

SSH and GPG keys
github.com/settings/keys
Settings
Type / to search
Valentine (vjamal22)
Your personal account
Go to your personal profile
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and plans
Emails
SSH keys
New SSH key
There are no SSH keys associated with your account.
Check out our guide to connecting to GitHub using SSH keys or troubleshoot common SSH problems.
GPG keys
New GPG key
There are no GPG keys associated with your account.
Learn how to generate a GPG key and add it to your account.

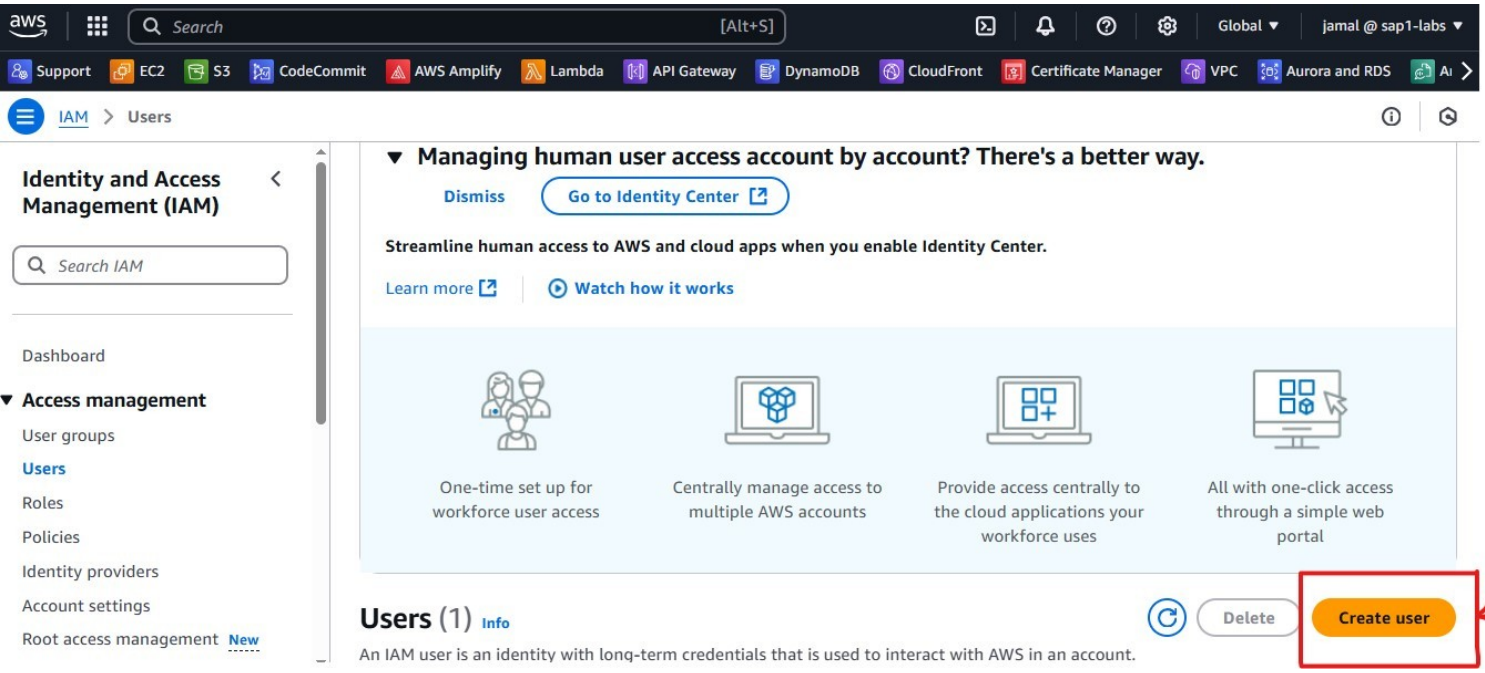

github.com
Dashboard
Type / to search
Top repositories
New
Find a repository...
vjamal22/ValsProject-5
vjamal22/AWS-Project-4
vjamal22/terraformpro
vjamal22/AWS-Project-1
Ask Copilot
Learn Python main check
Learn Java array sorting perf
Home
Filter
Learn with a tutorial project
Introduction to GitHub
Get started using GitHub in less than an hour.
GitHub Pages
Create a site or blog from your GitHub repositories with GitHub Pages.
Code with Copilot
Develop with AI-powered code suggestions using GitHub Copilot, Codespaces, and VS Code.
Hello GitHub Actions
Create a GitHub Action and use it in a workflow.


vjamal22 / terraformpro
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
terraformpro
Private
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Code
vjamal22 Initial commit
.gitignore Initial commit
README.md Initial commit
README
terraformpro
My first terraformpro repository
Local
Codespaces
Clone
HTTPS
SSH
GitHub CLI
git@github.com:vjamal22/terraformpro.git
Use a password-protected SSH key.
Open with GitHub Desktop
Download ZIP
About
My first terraformpro repository
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release
Packages


terraformpro
File
Home
Share
View
terraformpro
Search terraformpro
Quick access
Desktop
Downloads
Documents
Pictures
.ssh
My first website
Vals AWS CV
Vals Electrical CV
This PC
3D Objects
Desktop
Documents
2 items
Name
Date modified
Type
Size
.gitignore
24/03/2025 17:33
Text Document
1 KB
README.md
24/03/2025 17:33
Markdown Source...
1 KB
terraformpro


terraformpro
Private
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Code
About
My first terraformpro repository
vjamal22 Initial commit
2418bfd · 19 hours ago
1 Commit
.gitignore
Initial commit
19 hours ago
README.md
Initial commit
19 hours ago
Readme
Activity
0 stars
1 watching
0 forks
README
terraformpro
My first terraformpro repository
Releases
No releases published
Create a new release
Packages


Search [Alt+S]
Global
jamal @ sap1-labs
Support EC2 S3 CodeCommit AWS Amplify Lambda API Gateway DynamoDB CloudFront Certificate Manager VPC Aurora and RDS
IAM > Dashboard
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
IAM Dashboard Info
Security recommendations 2
Add MFA for root user
Sign in as the root user (or contact your administrator) and register a multi-factor authentication (MFA) device for the root user to improve security for this account.
Add MFA for yourself
Add multi-factor authentication (MFA) for yourself to improve security for this account.
Add MFA
Your user, jamal, does not have any active access keys that have been unused for more than a year.
Deactivating or deleting unused access keys improves security.
IAM resources
AWS Account
Account ID
Account Alias
sap1-labs Edit | Delete
Sign-in URL for IAM users in this account
https://sap1-labs.signin.aws.amazon.com/console
Quick Links
My security credentials


Search [Alt+S]
Global
jamal @ sap1-labs
Support EC2 S3 CodeCommit AWS Amplify Lambda API Gateway DynamoDB CloudFront Certificate Manager VPC Aurora and RDS
IAM > Users
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Root access management New
Managing human user access account by account? There's a better way.
Dismiss
Go to Identity Center
Streamline human access to AWS and cloud apps when you enable Identity Center.
Learn more
Watch how it works
One-time set up for workforce user access
Centrally manage access to multiple AWS accounts
Provide access centrally to the cloud applications your workforce uses
All with one-click access through a simple web portal
Users (1) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Delete
Create user

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

- Add user to group
  Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
  Copy all group memberships, attached managed policies, and inline policies from an existing user.
- **Attach policies directly**
  Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1336)

Create policy

Choose one or more policies to attach to your new user.

Filter by Type

Search | All types | 1 2 3 4 5 6 7 ... 67

Policy name	Type	Attached entities
AccessAnalyzerServiceRo...	AWS managed	0
AdministratorAccess	**AWS managed - job function**	**1**
AdministratorAccess-Am...	AWS managed	0
AdministratorAccess-AW...	AWS managed	0

Permissions summary

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags - *optional*

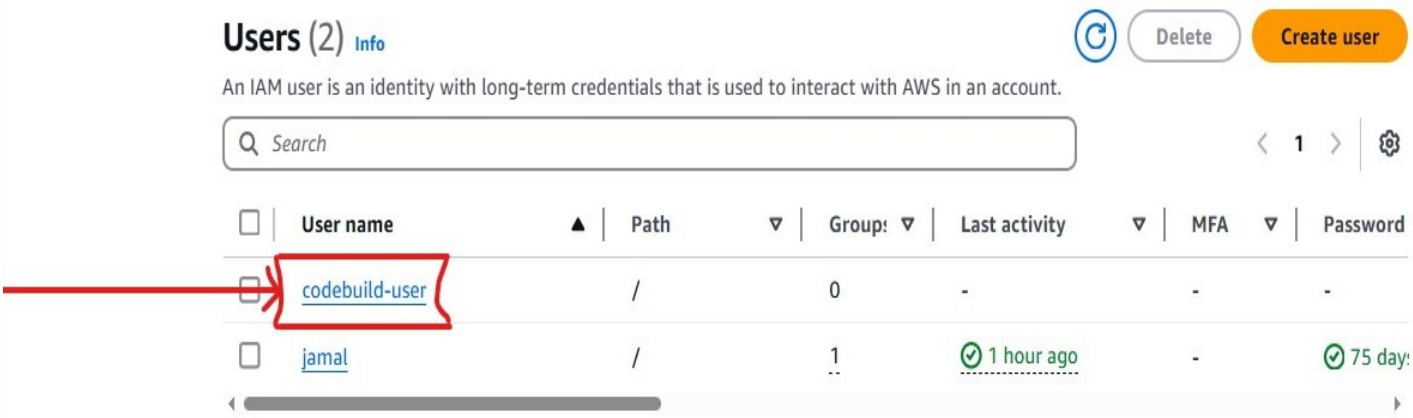
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

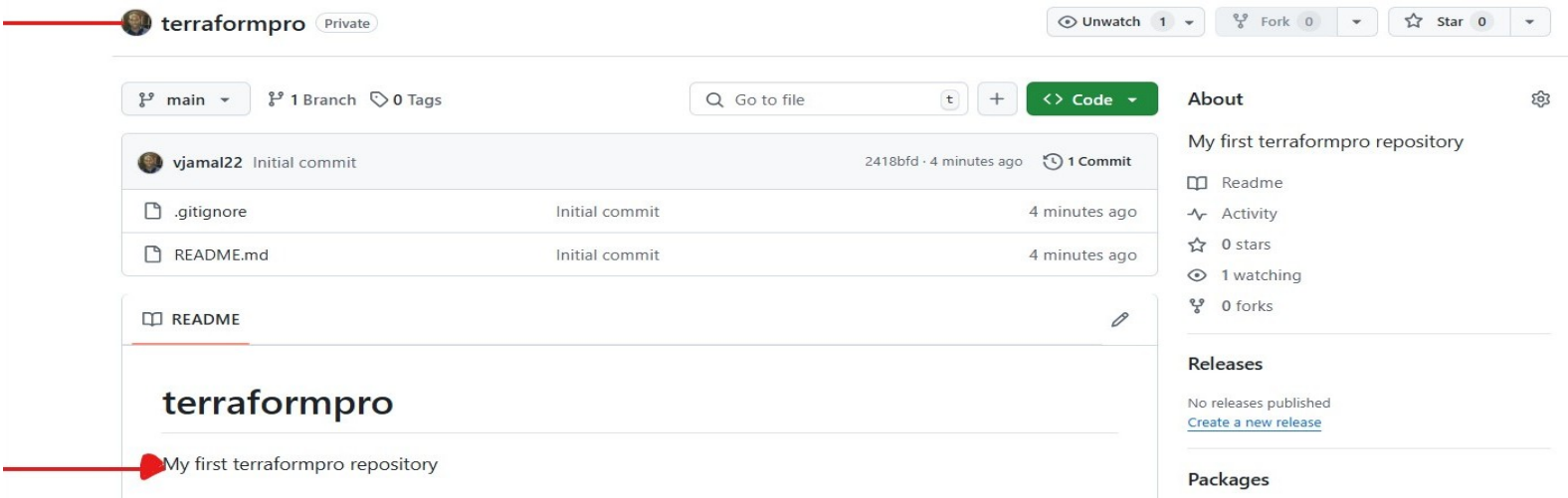
Add new tag

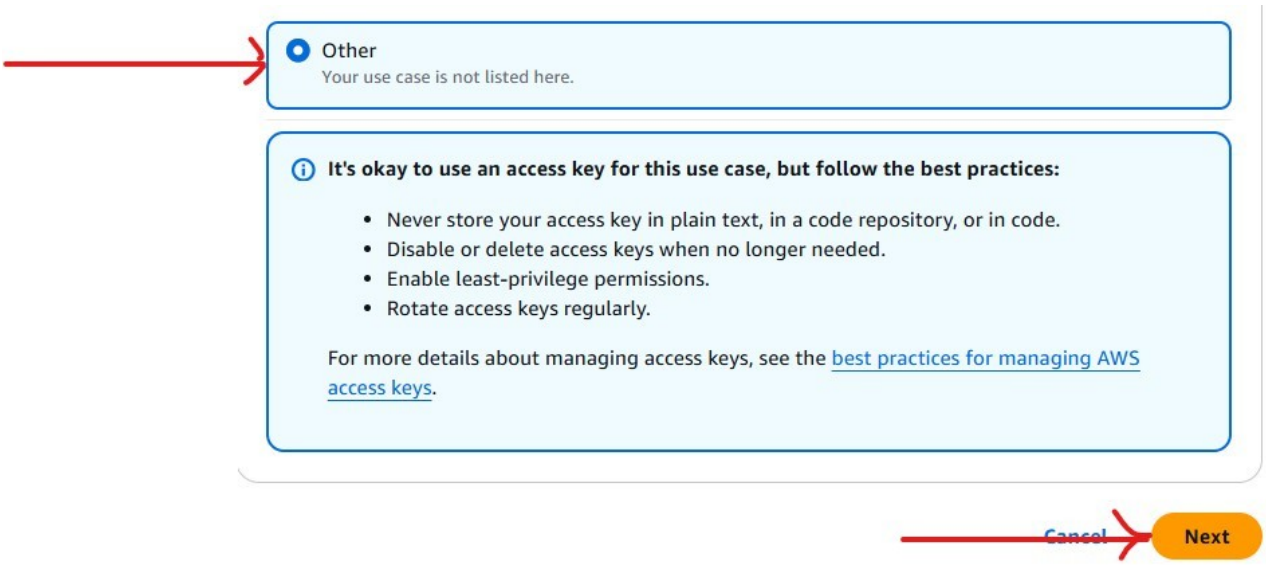
You can add up to 50 more tags.

Cancel | Previous | **Create user**


Users (2) Info
Delete
Create user
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Search
1
User name
Path
Groups
Last activity
MFA
Password
codebuild-user
/
0
-
-
-
jamal
/
1
1 hour ago
-
75 days


Access keys (0)
Create access key
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more
No access keys. As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more
Create access key


terraformpro
Private
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Code
About
My first terraformpro repository
vjamal22 Initial commit
2418bfd · 4 minutes ago
1 Commit
.gitignore
Initial commit
4 minutes ago
README.md
Initial commit
4 minutes ago
Readme
Activity
0 stars
1 watching
0 forks
README
terraformpro
My first terraformpro repository
Releases
No releases published
Create a new release
Packages



Set description tag - *optional* Info

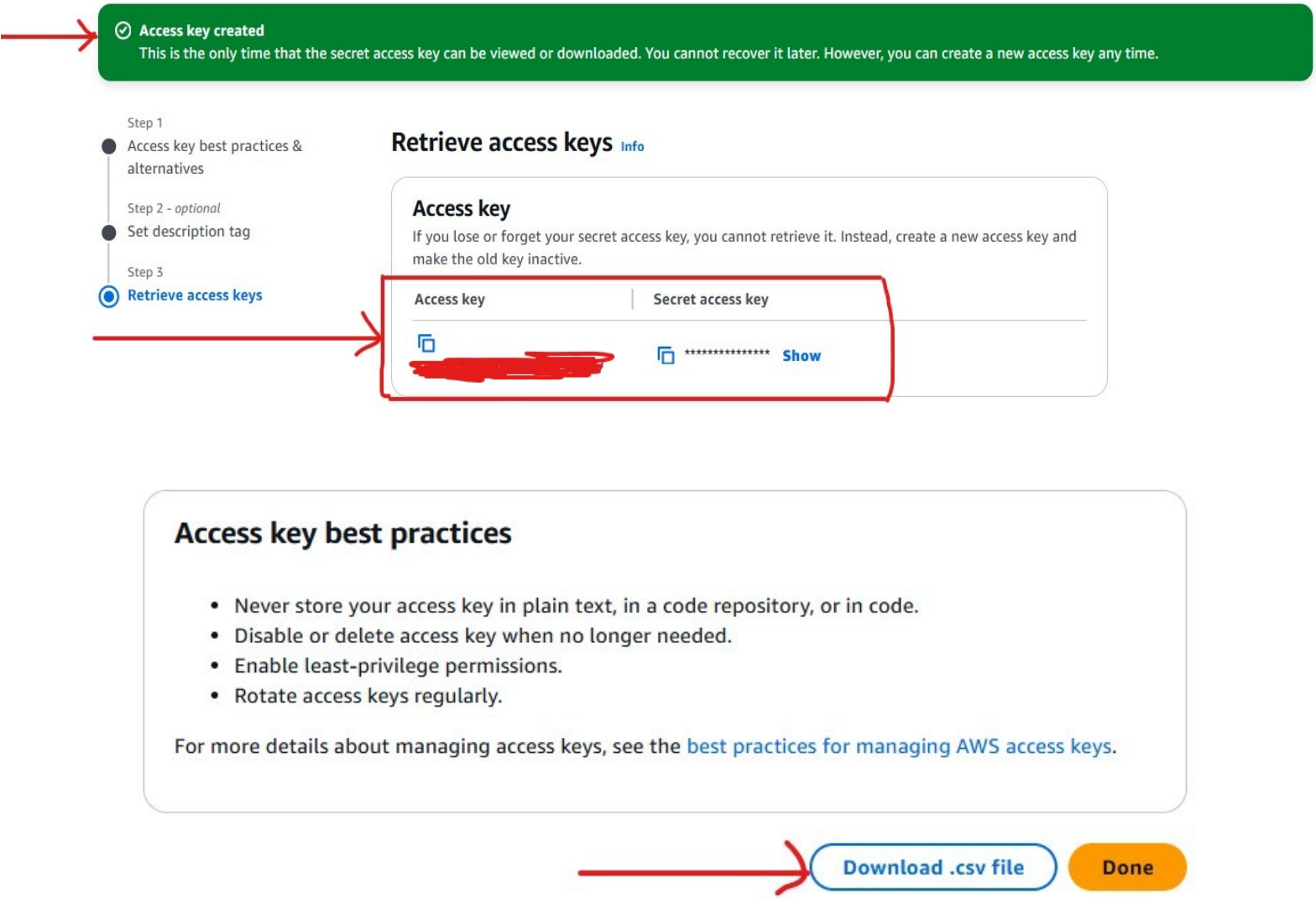
The description for this access key will be attached to this user as a tag and shown alongside the access key.

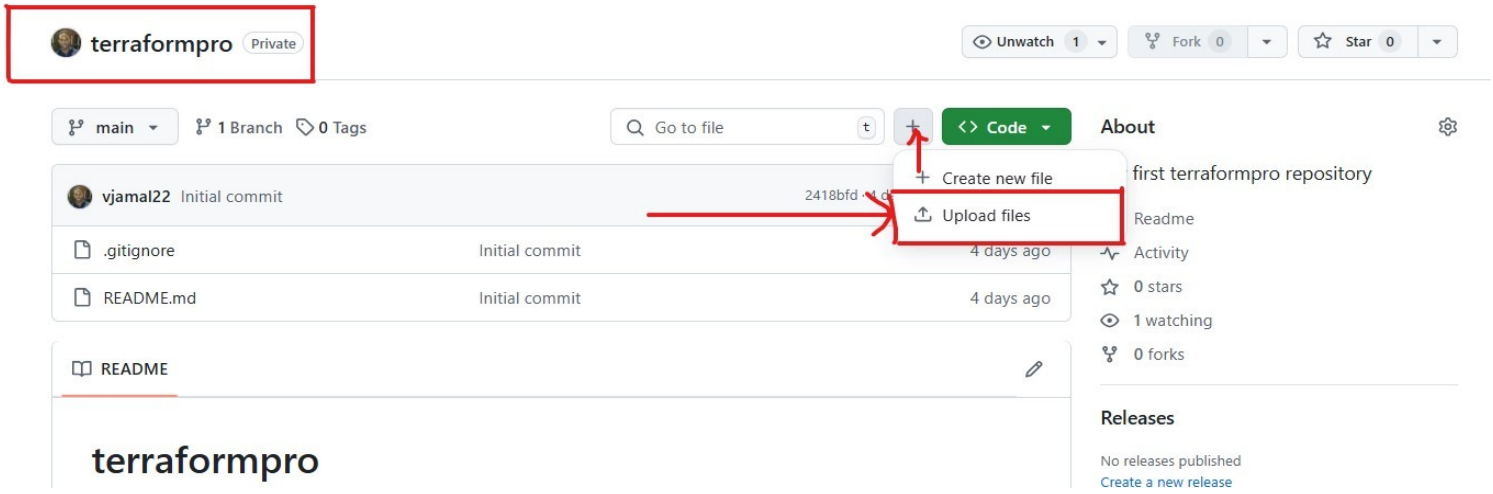
Description tag value

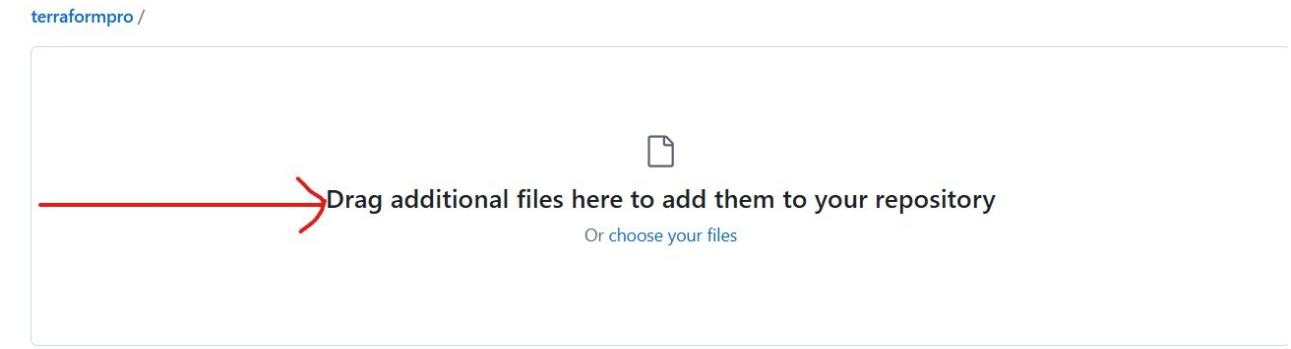
Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

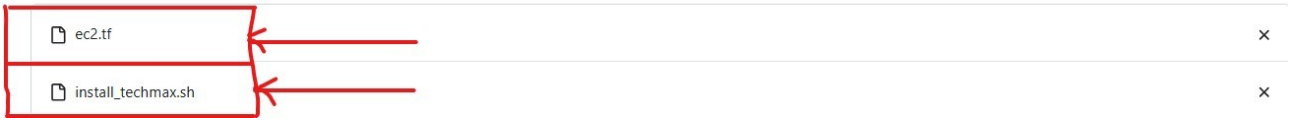
Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . : / = + - @

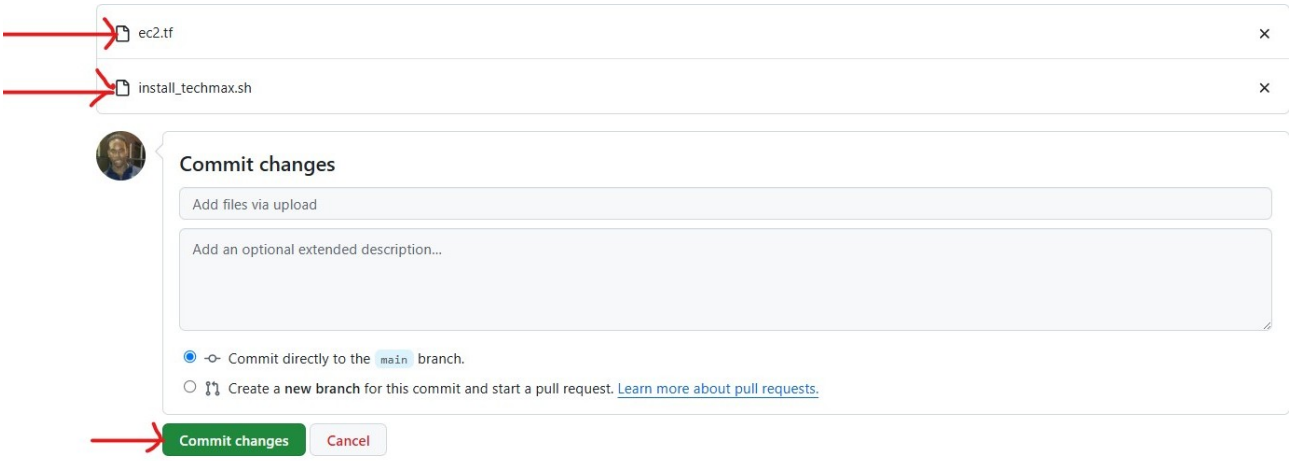
Cancel Previous Create access key

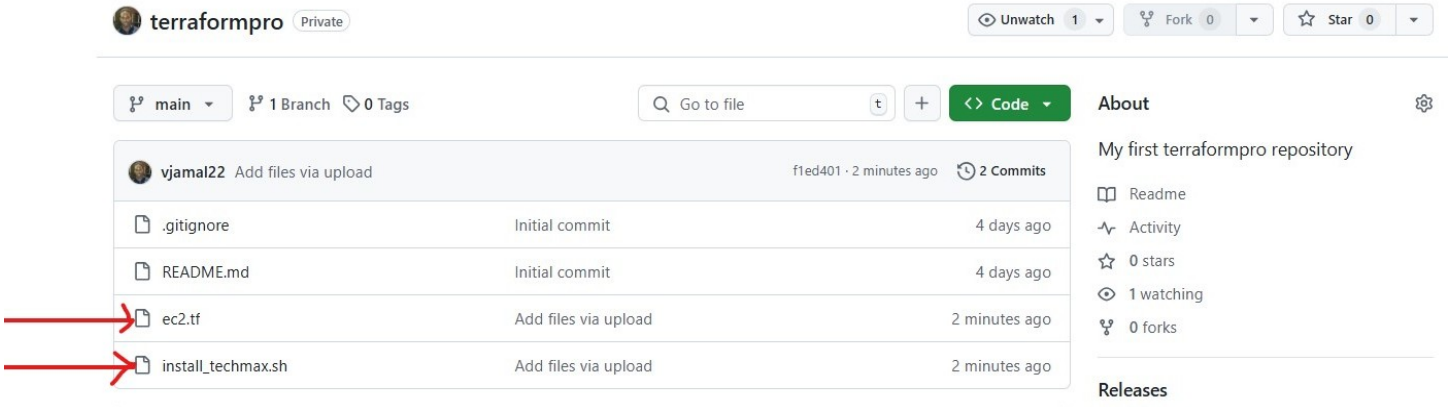

Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.
Step 1
Access key best practices & alternatives
Step 2 - optional
Set description tag
Step 3
Retrieve access keys
Retrieve access 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
********** Show
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

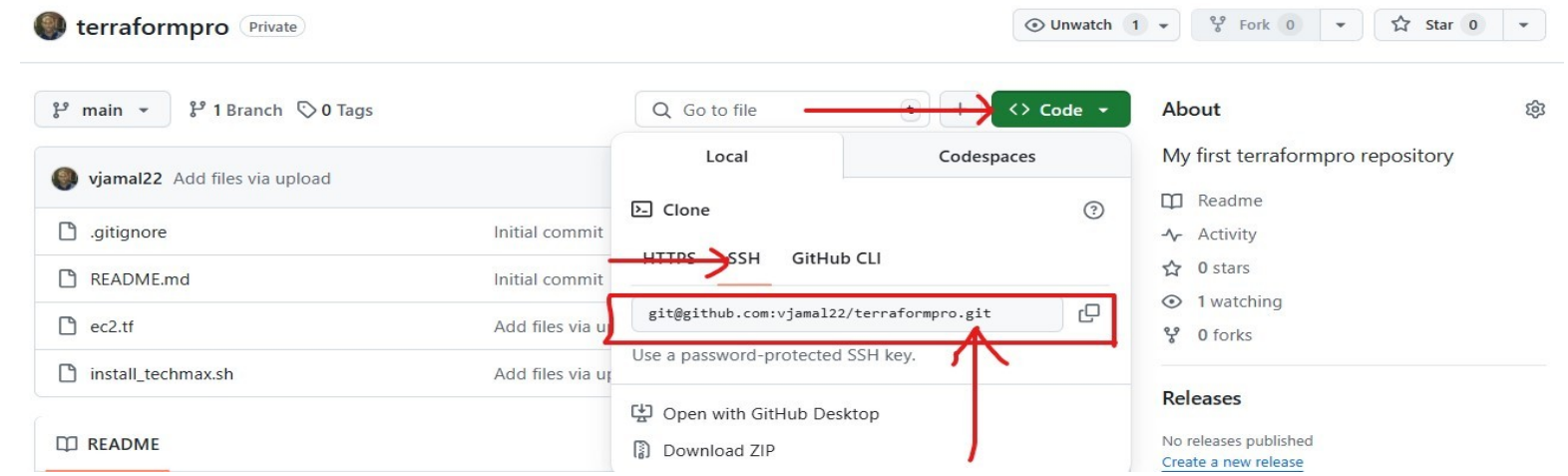

terraformpro
Private
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Code
About
vjamal22 Initial commit
2418bfd
Create new file
Upload files
.gitignore
Initial commit
4 days ago
README.md
Initial commit
4 days ago
first terraformpro repository
Readme
Activity
0 stars
1 watching
0 forks
README
terraformpro
Releases
No releases published
Create a new release

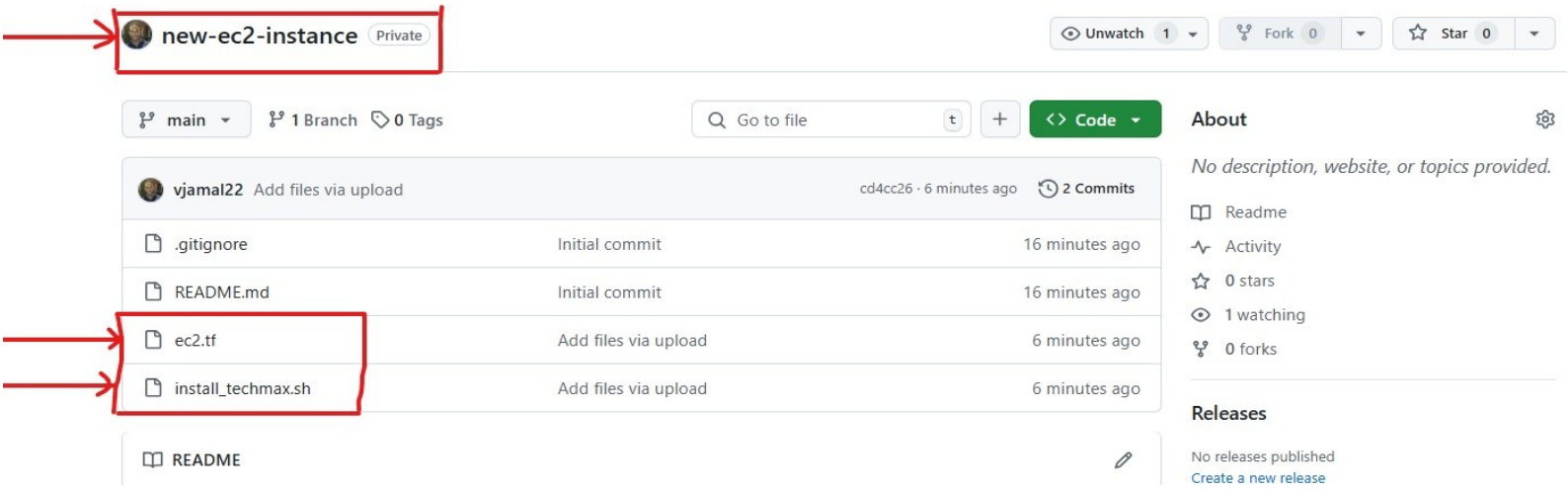

terraformpro /
Drag additional files here to add them to your repository
Or choose your files

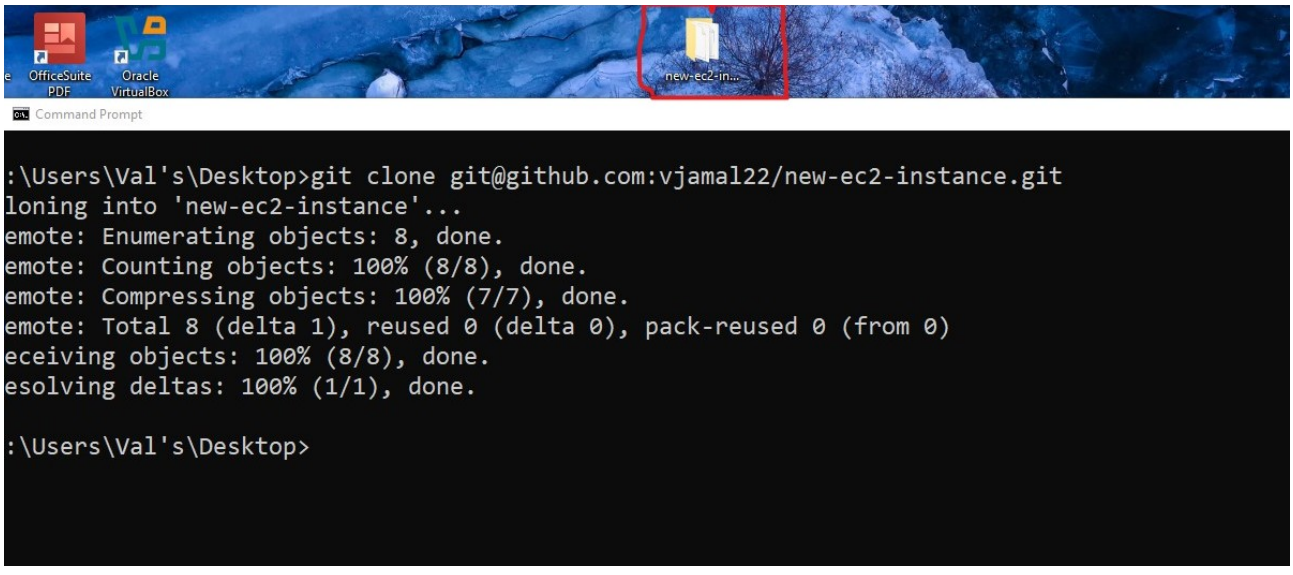

ec2.tf
install_techmax.sh

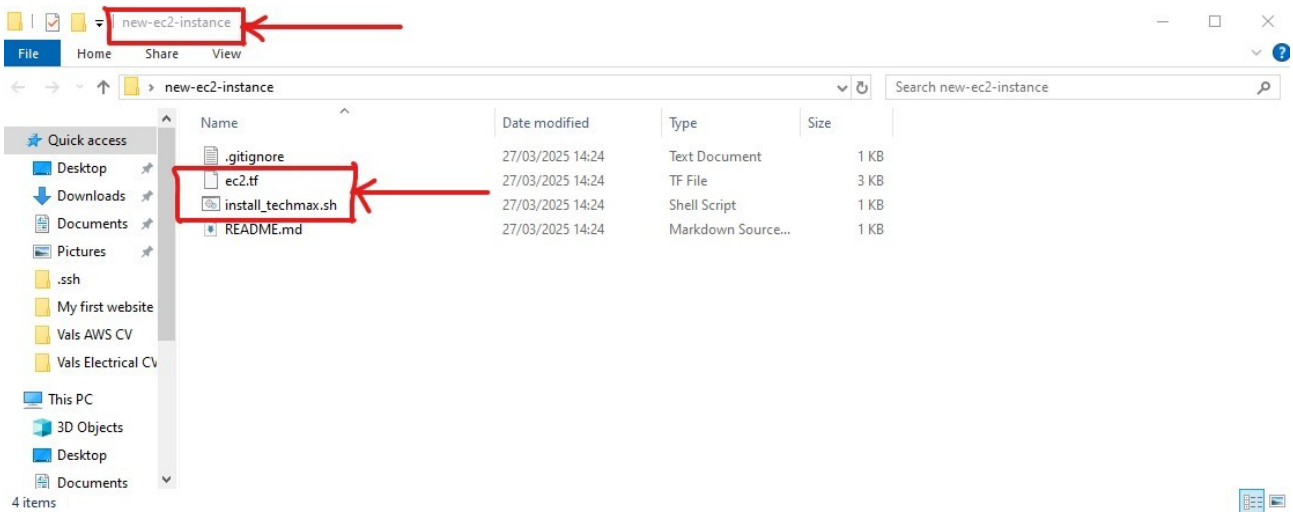

ec2.tf
install_techmax.sh
Commit changes
Add files via upload
Add an optional extended description...
Commit directly to the main branch.
Create a new branch for this commit and start a pull request. Learn more about pull requests.
Commit changes
Cancel

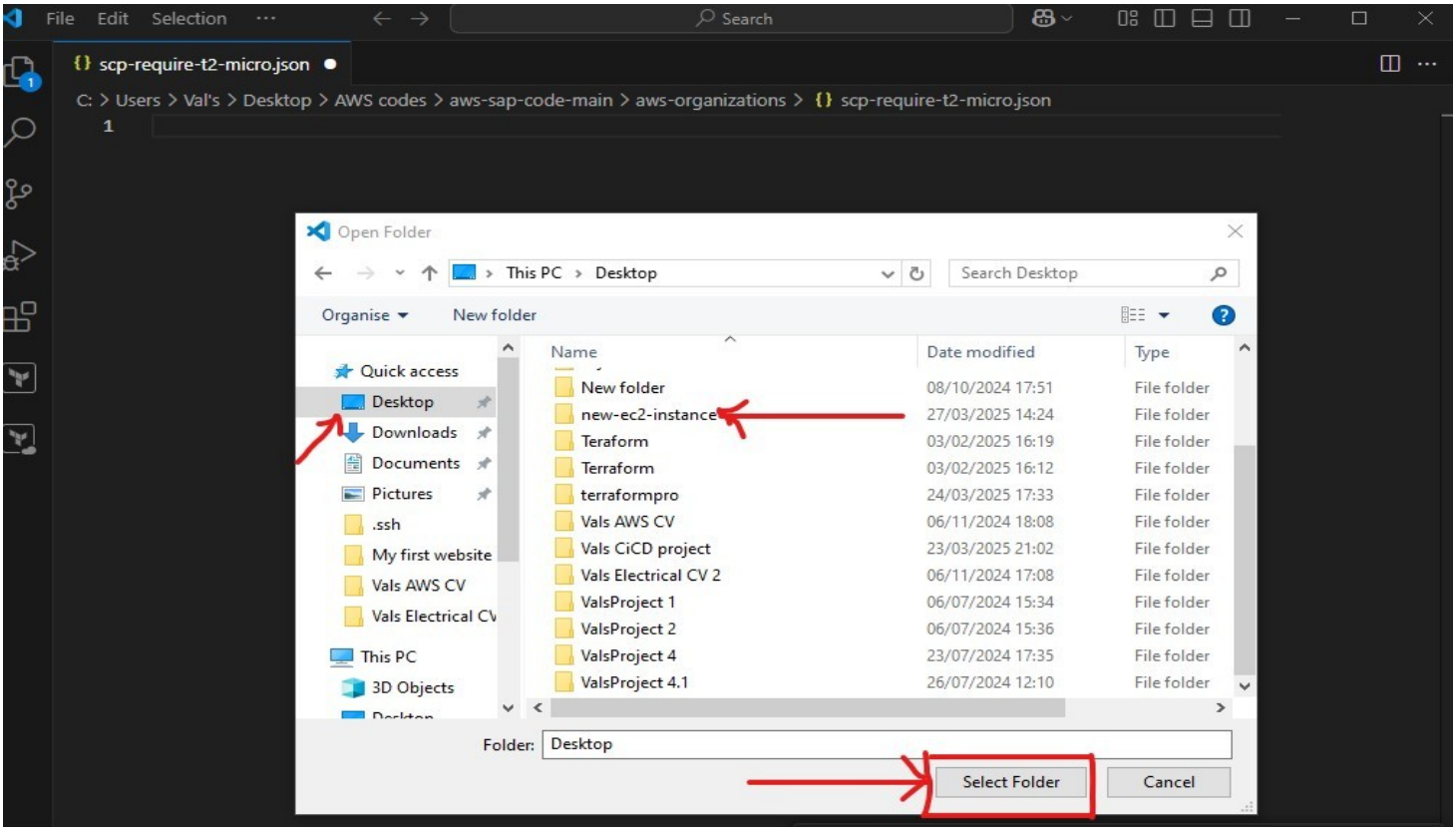

terraformpro
Private
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Code
About
My first terraformpro repository
vjamal22 Add files via upload
f1ed401 · 2 minutes ago
2 Commits
.gitignore
Initial commit
4 days ago
README.md
Initial commit
4 days ago
ec2.tf
Add files via upload
2 minutes ago
install_techmax.sh
Add files via upload
2 minutes ago
Readme
Activity
0 stars
1 watching
0 forks
Releases

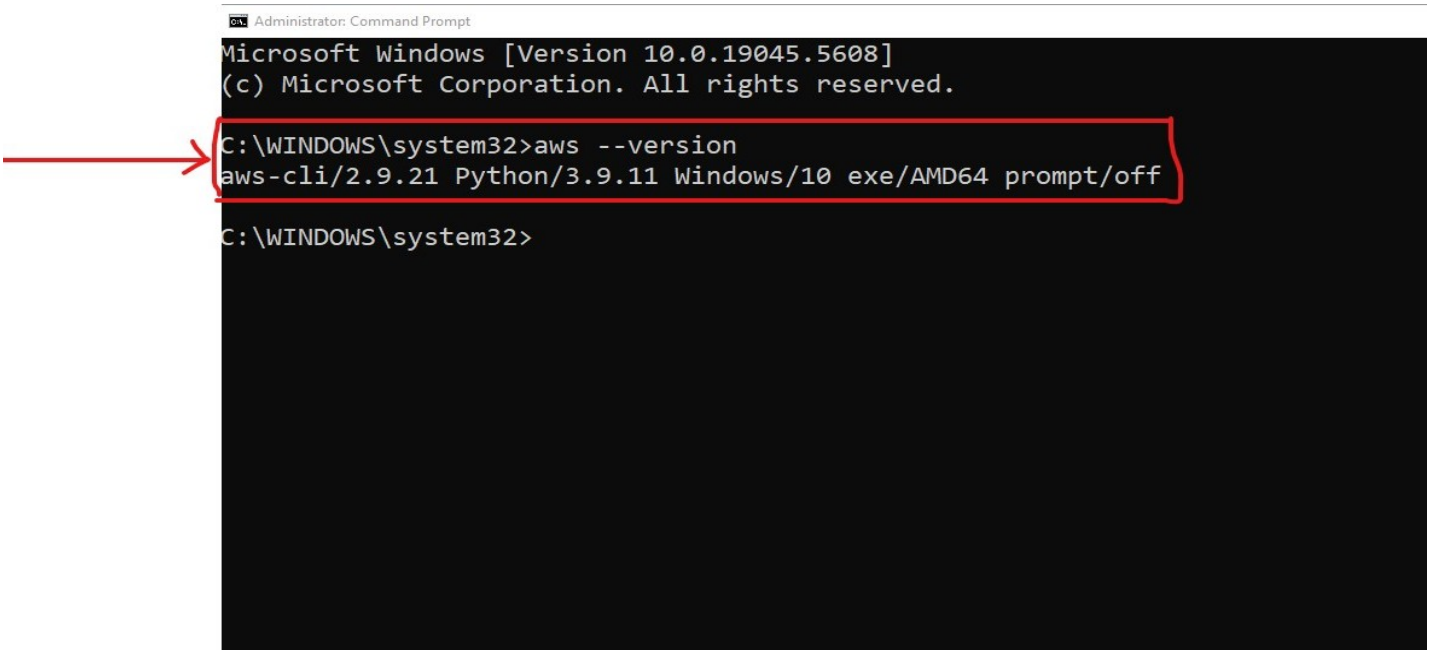

terraformpro
Private
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Code
About
My first terraformpro repository
vjamal22 Add files via upload
.gitignore Initial commit
README.md Initial commit
ec2.tf Add files via u
install_techmax.sh Add files via up
Local
Codespaces
Clone
HTTPS
SSH
GitHub CLI
git@github.com:vjamal22/terraformpro.git
Use a password-protected SSH key.
Open with GitHub Desktop
Download ZIP
README
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release

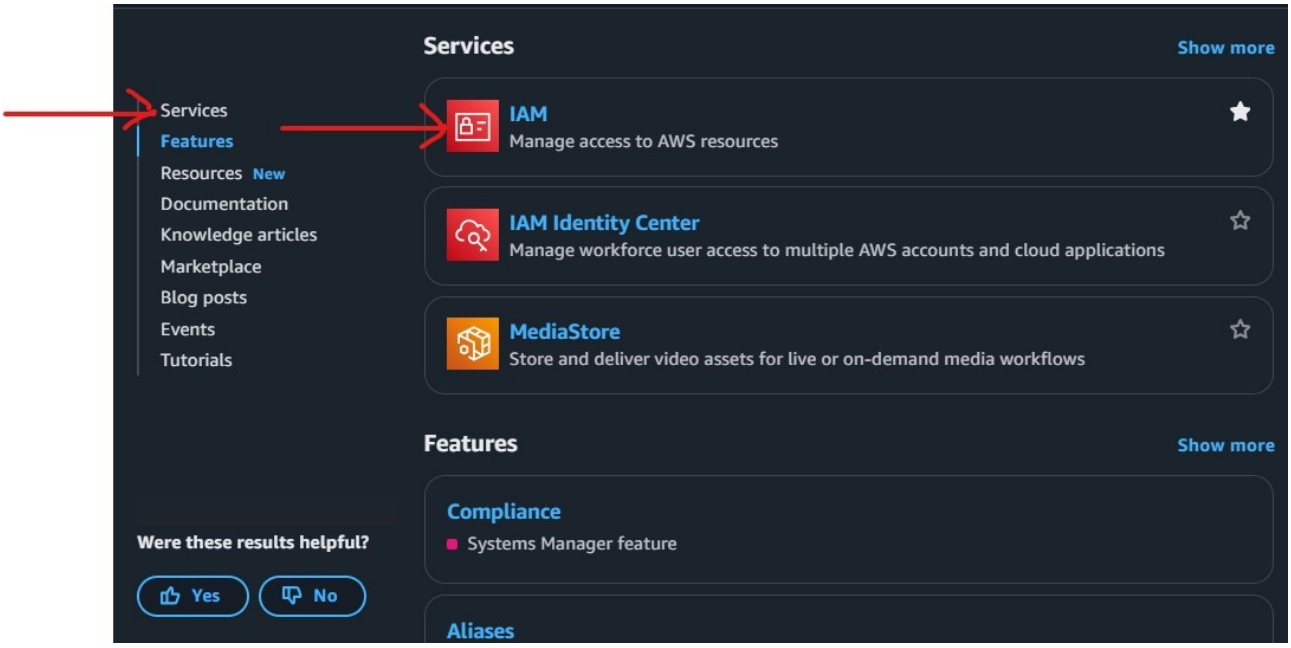

new-ec2-instance
Private
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
t
Code
About
vjamal22 Add files via upload
cd4cc26 · 6 minutes ago
2 Commits
.gitignore Initial commit 16 minutes ago
README.md Initial commit 16 minutes ago
ec2.tf Add files via upload 6 minutes ago
install_techmax.sh Add files via upload 6 minutes ago
README
No description, website, or topics provided.
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release

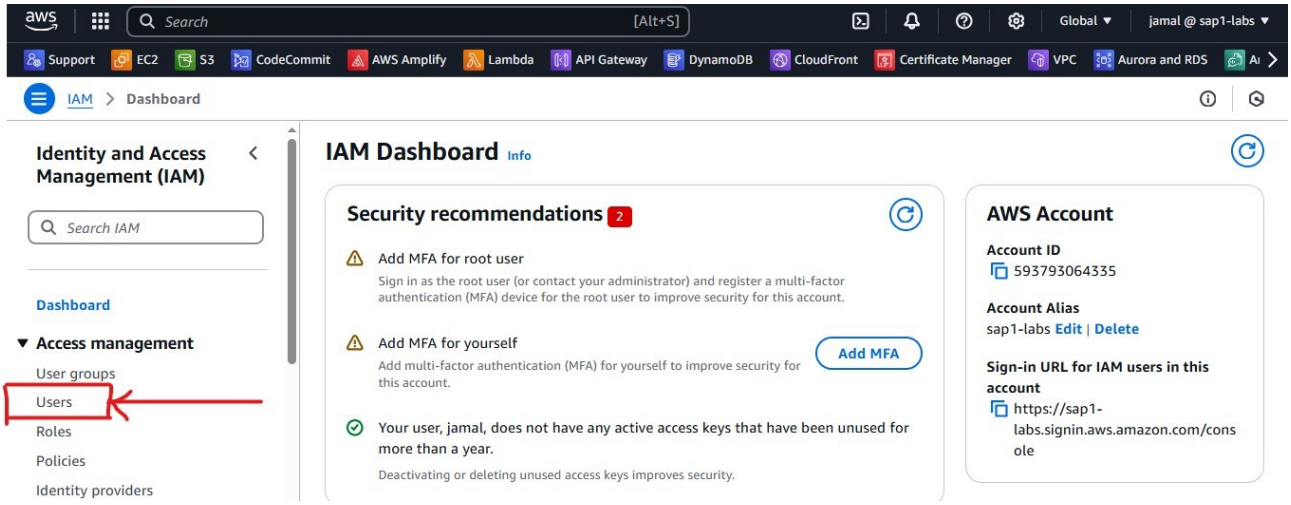

OfficeSuite PDF
Oracle VirtualBox
new-ec2-in...
Command Prompt
:\Users\Val's\Desktop>git clone git@github.com:vjamal22/new-ec2-instance.git
loning into 'new-ec2-instance'...
emote: Enumerating objects: 8, done.
emote: Counting objects: 100% (8/8), done.
emote: Compressing objects: 100% (7/7), done.
emote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
eceiving objects: 100% (8/8), done.
esolving deltas: 100% (1/1), done.
:\Users\Val's\Desktop>

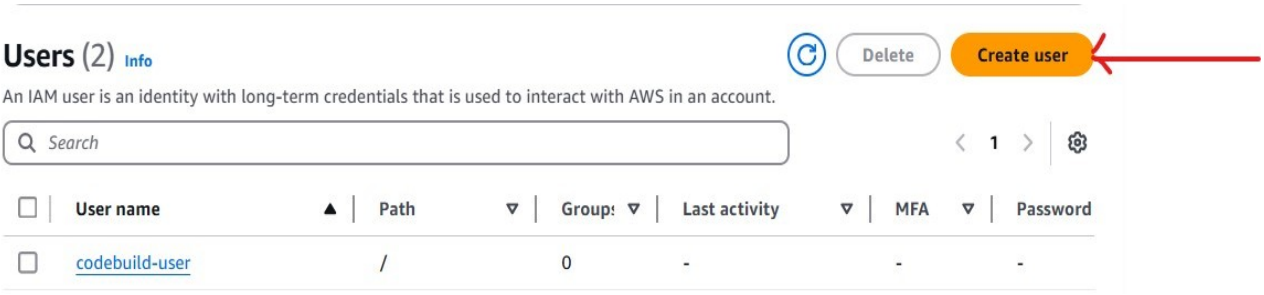

new-ec2-instance
File
Home
Share
View
new-ec2-instance
Search new-ec2-instance
Quick access
Desktop
Downloads
Documents
Pictures
.ssh
My first website
Vals AWS CV
Vals Electrical CV
This PC
3D Objects
Desktop
Documents
4 items
Name
Date modified
Type
Size
.gitignore
27/03/2025 14:24
Text Document
1 KB
ec2.tf
27/03/2025 14:24
TF File
3 KB
install_techmax.sh
27/03/2025 14:24
Shell Script
1 KB
README.md
27/03/2025 14:24
Markdown Source...
1 KB

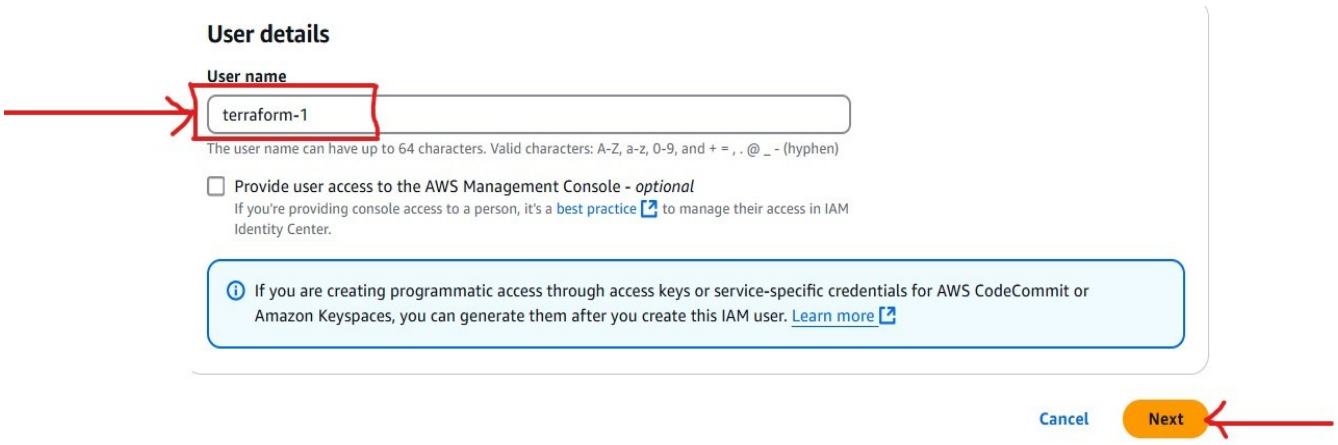

File
Edit
Selection
Search
scp-require-t2-micro.json
C: > Users > Val's > Desktop > AWS codes > aws-sap-code-main > aws-organizations > scp-require-t2-micro.json
1
Open Folder
This PC > Desktop
Search Desktop
Organise
New folder
Quick access
Desktop
Downloads
Documents
Pictures
.ssh
My first website
Vals AWS CV
Vals Electrical CV
This PC
3D Objects
Name
Date modified
Type
New folder
08/10/2024 17:51
File folder
new-ec2-instance
27/03/2025 14:24
File folder
Teraform
03/02/2025 16:19
File folder
Terraform
03/02/2025 16:12
File folder
terraformpro
24/03/2025 17:33
File folder
Vals AWS CV
06/11/2024 18:08
File folder
Vals CiCD project
23/03/2025 21:02
File folder
Vals Electrical CV 2
06/11/2024 17:08
File folder
ValsProject 1
06/07/2024 15:34
File folder
ValsProject 2
06/07/2024 15:36
File folder
ValsProject 4
23/07/2024 17:35
File folder
ValsProject 4.1
26/07/2024 12:10
File folder
Folder:
Desktop
Select Folder
Cancel


Administrator: Command Prompt
Microsoft Windows [Version 10.0.19045.5608]
(c) Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>aws --version
aws-cli/2.9.21 Python/3.9.11 Windows/10 exe/AMD64 prompt/off
C:\WINDOWS\system32>


Services
Features
Resources New
Documentation
Knowledge articles
Marketplace
Blog posts
Events
Tutorials
Were these results helpful?
Yes
No
Services
Show more
IAM
Manage access to AWS resources
IAM Identity Center
Manage workforce user access to multiple AWS accounts and cloud applications
MediaStore
Store and deliver video assets for live or on-demand media workflows
Features
Show more
Compliance
Systems Manager feature
Aliases


Search
[Alt+S]
Global
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and RDS
IAM
Dashboard
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
IAM Dashboard Info
Security recommendations 2
Add MFA for root user
Sign in as the root user (or contact your administrator) and register a multi-factor authentication (MFA) device for the root user to improve security for this account.
Add MFA for yourself
Add multi-factor authentication (MFA) for yourself to improve security for this account.
Add MFA
Your user, jamal, does not have any active access keys that have been unused for more than a year.
Deactivating or deleting unused access keys improves security.
AWS Account
Account ID
593793064335
Account Alias
sap1-labs Edit | Delete
Sign-in URL for IAM users in this account
https://sap1-labs.signin.aws.amazon.com/console


Users (2) Info
Delete
Create user
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Search
1
User name
Path
Groups
Last activity
MFA
Password
codebuild-user
/
0
-
-
-


User details
User name
terraform-1
The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)
Provide user access to the AWS Management Console - optional
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more
Cancel
Next

Set permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Permissions options

- Add user to group
  Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions
  Copy all group memberships, attached managed policies, and inline policies from an existing user.
- **Attach policies directly**
  Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

☑ AdministratorAccess	AWS managed - job function	2
☐ AdministratorAccess-Am...	AWS managed	0
☐ AdministratorAccess-AW...	AWS managed	0
☐ AIOpsAssistantPolicy	AWS managed	0
☐ AIOpsConsoleAdminPolicy	AWS managed	0
☐ AIOpsOperatorAccess	AWS managed	0
☐ AIOpsReadOnlyAccess	AWS managed	0

Permissions summary

1

Name	Type	Used as
AdministratorAccess	AWS managed - job function	Permissions policy

Tags - *optional*

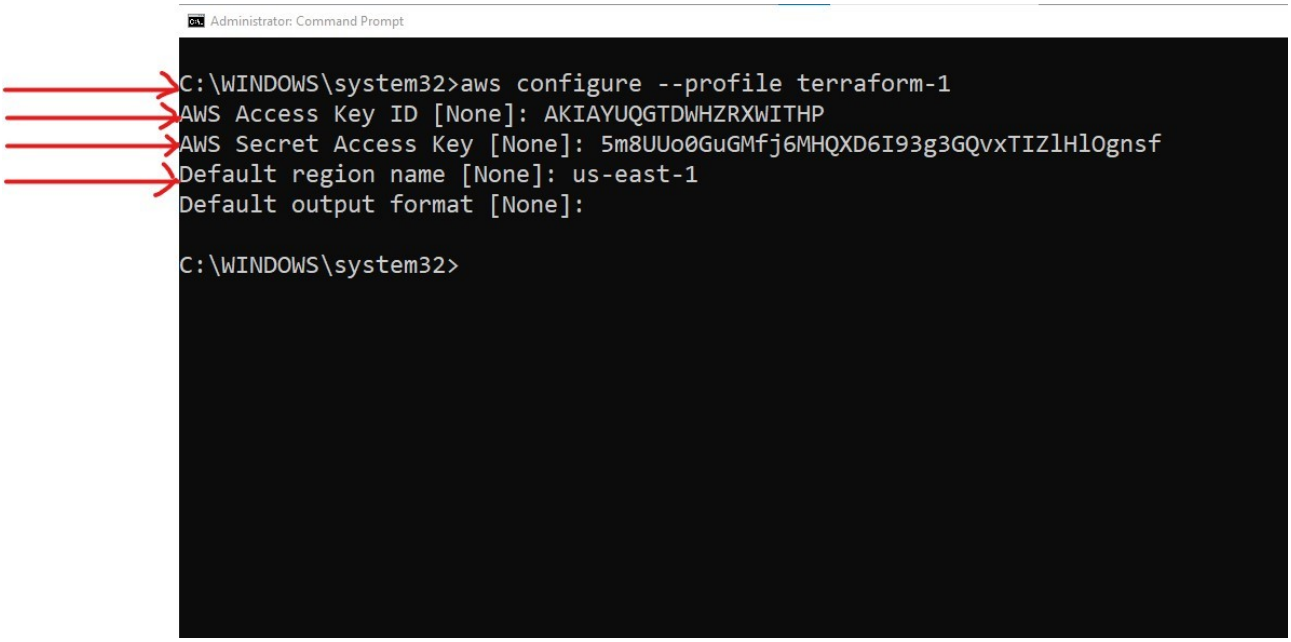
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

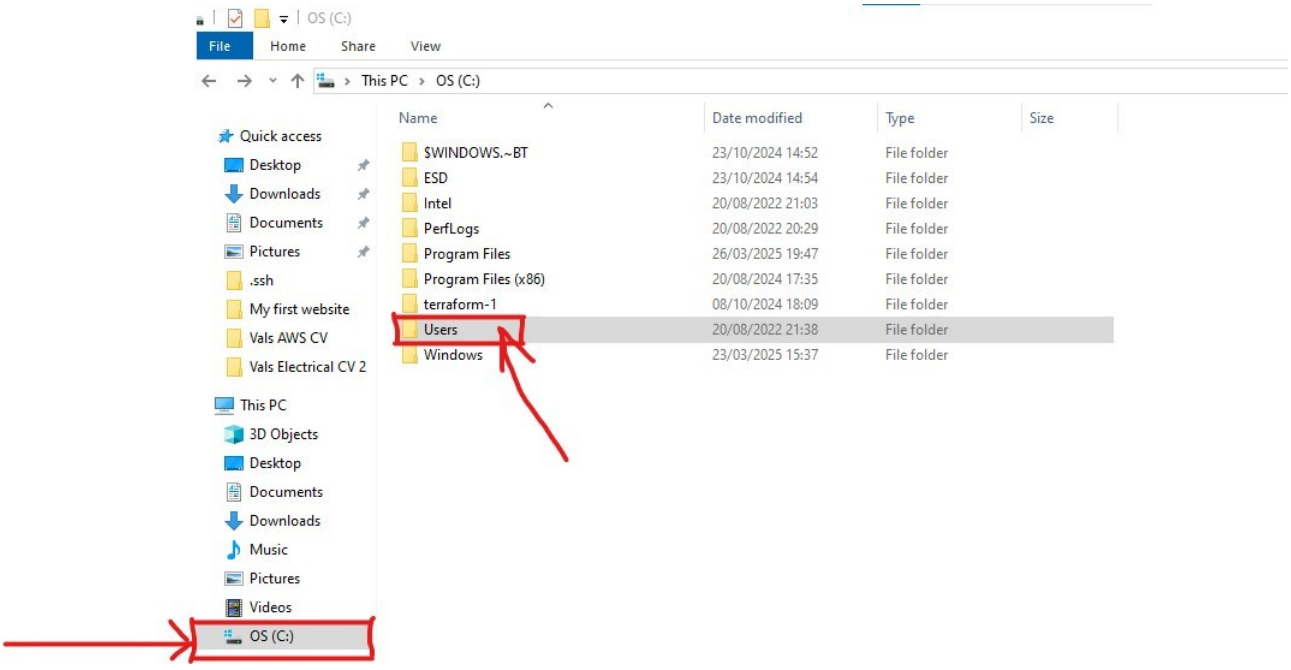
No tags associated with the resource.

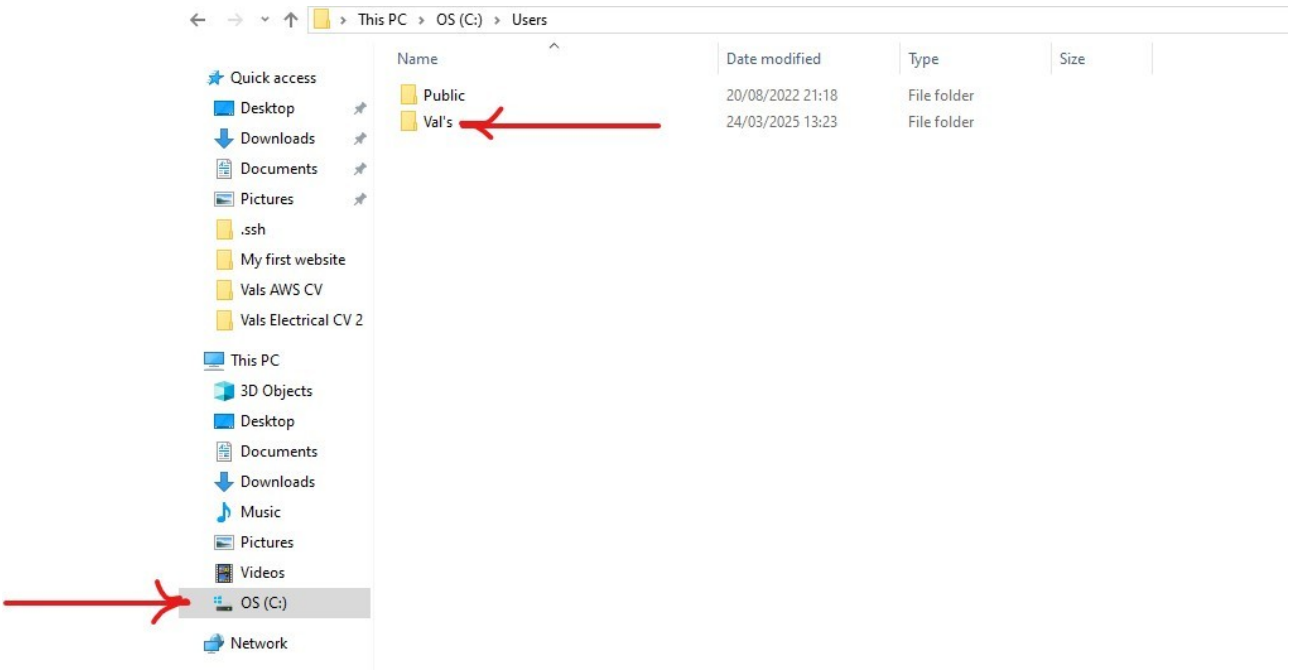
Add new tag

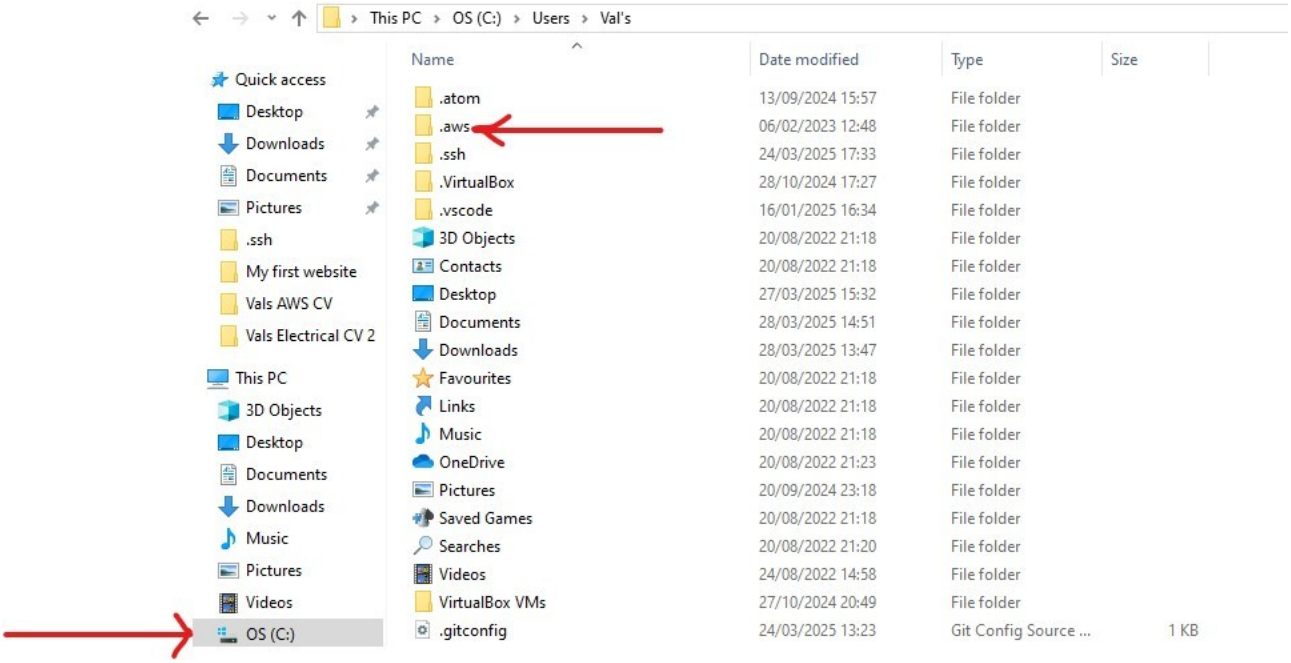
You can add up to 50 more tags.

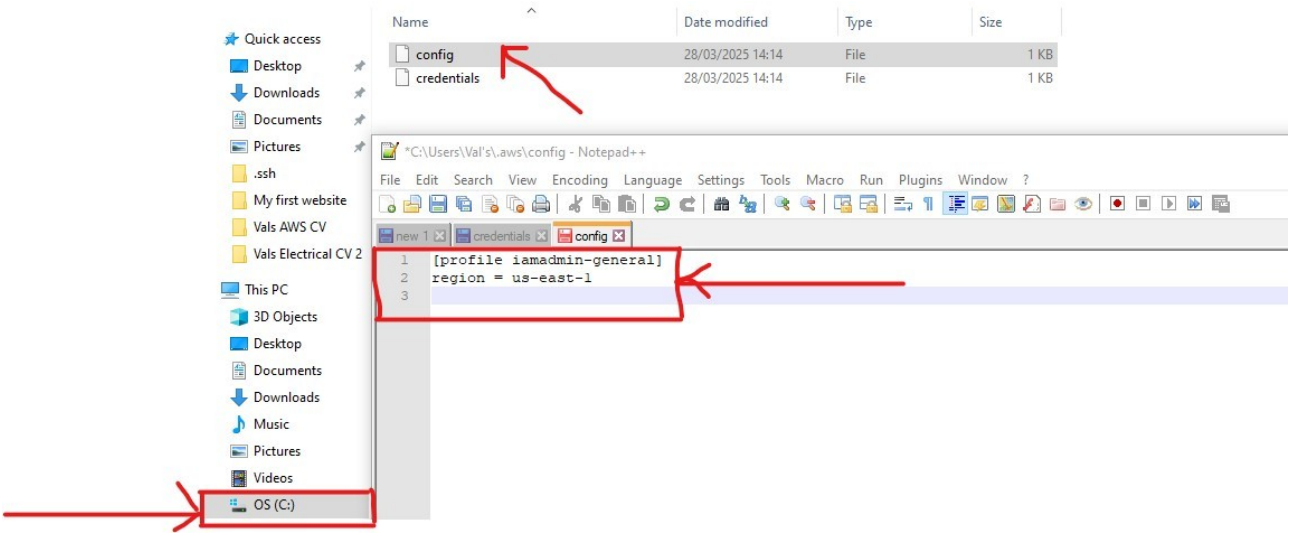
Cancel | Previous | **Create user**

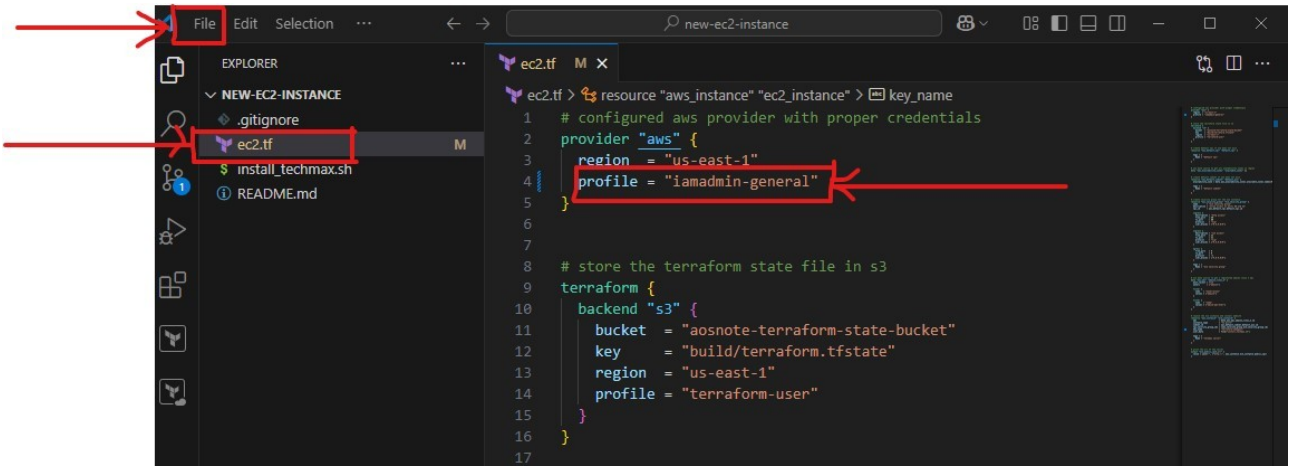

Administrator: Command Prompt
C:\WINDOWS\system32>aws configure --profile terraform-1
AWS Access Key ID [None]: AKIAYUQGTDWHZRXWITHP
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
C:\WINDOWS\system32>

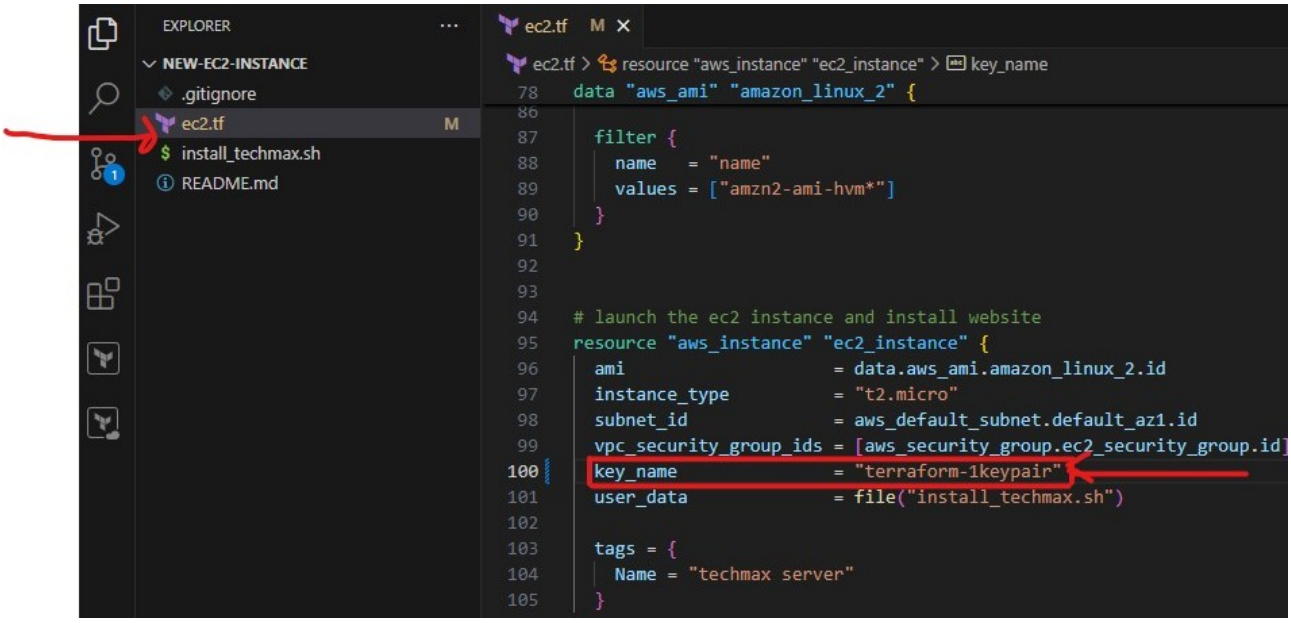

OS (C:)
File
Home
Share
View
This PC › OS (C:)
Quick access
Desktop
Downloads
Documents
Pictures
.ssh
My first website
Vals AWS CV
Vals Electrical CV 2
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
Name
Date modified
Type
Size
$WINDOWS.~BT 23/10/2024 14:52 File folder
ESD 23/10/2024 14:54 File folder
Intel 20/08/2022 21:03 File folder
PerfLogs 20/08/2022 20:29 File folder
Program Files 26/03/2025 19:47 File folder
Program Files (x86) 20/08/2024 17:35 File folder
terraform-1 08/10/2024 18:09 File folder
Users 20/08/2022 21:38 File folder
Windows 23/03/2025 15:37 File folder

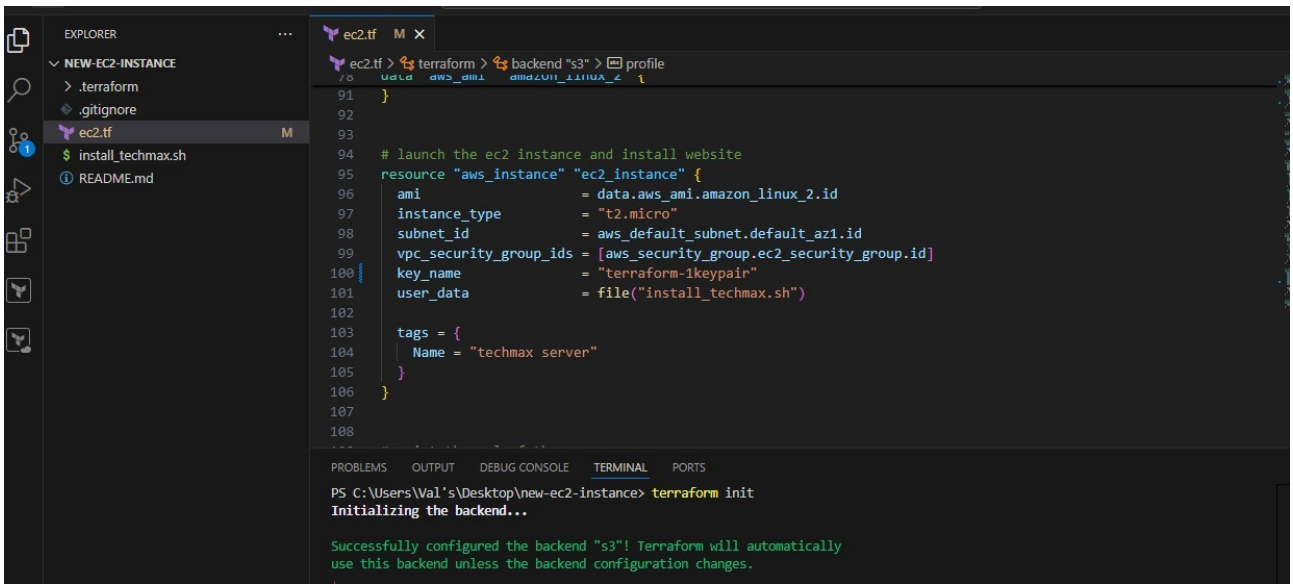

This PC › OS (C:) › Users
Quick access
Desktop
Downloads
Documents
Pictures
.ssh
My first website
Vals AWS CV
Vals Electrical CV 2
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
Network
Name
Date modified
Type
Size
Public
20/08/2022 21:18
File folder
Val's
24/03/2025 13:23
File folder

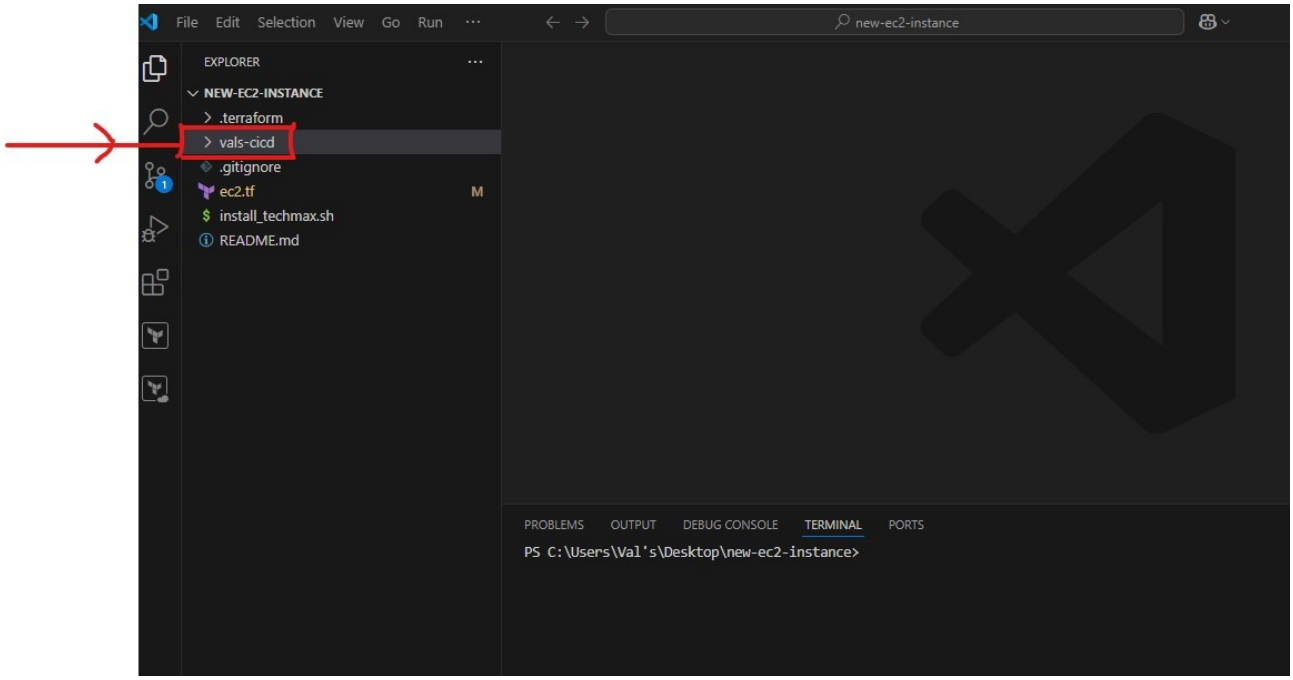

This PC › OS (C:) › Users › Val's
Quick access
Desktop
Downloads
Documents
Pictures
.ssh
My first website
Vals AWS CV
Vals Electrical CV 2
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
Name
Date modified
Type
Size
.atom
13/09/2024 15:57
File folder
.aws
06/02/2023 12:48
File folder
.ssh
24/03/2025 17:33
File folder
.VirtualBox
28/10/2024 17:27
File folder
.vscode
16/01/2025 16:34
File folder
3D Objects
20/08/2022 21:18
File folder
Contacts
20/08/2022 21:18
File folder
Desktop
27/03/2025 15:32
File folder
Documents
28/03/2025 14:51
File folder
Downloads
28/03/2025 13:47
File folder
Favourites
20/08/2022 21:18
File folder
Links
20/08/2022 21:18
File folder
Music
20/08/2022 21:18
File folder
OneDrive
20/08/2022 21:23
File folder
Pictures
20/09/2024 23:18
File folder
Saved Games
20/08/2022 21:18
File folder
Searches
20/08/2022 21:20
File folder
Videos
24/08/2022 14:58
File folder
VirtualBox VMs
27/10/2024 20:49
File folder
.gitconfig
24/03/2025 13:23
Git Config Source ...
1 KB

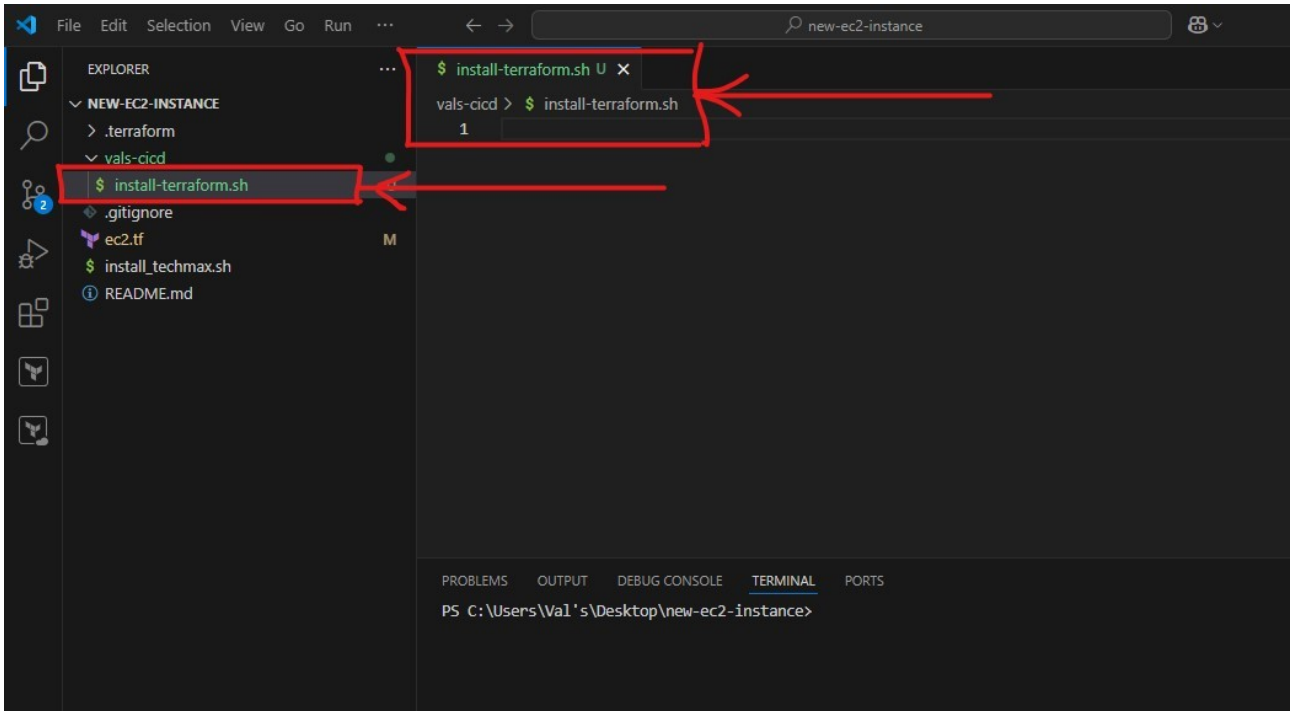

Name
Date modified
Type
Size
config
28/03/2025 14:14
File
1 KB
credentials
28/03/2025 14:14
File
1 KB
Quick access
Desktop
Downloads
Documents
Pictures
.ssh
My first website
Vals AWS CV
Vals Electrical CV 2
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
OS (C:)
*C:\Users\Val's\.aws\config - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1
credentials
config
[profile iamadmin-general]
region = us-east-1


File Edit Selection
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.gitignore
ec2.tf
install_techmax.sh
README.md
ec2.tf M
ec2.tf > resource "aws_instance" "ec2_instance" > key_name
# configured aws provider with proper credentials
provider "aws" {
region = "us-east-1"
profile = "iamadmin-general"
}
# store the terraform state file in s3
terraform {
backend "s3" {
bucket = "aosnote-terraform-state-bucket"
key = "build/terraform.tfstate"
region = "us-east-1"
profile = "terraform-user"
}
}


EXPLORER
NEW-EC2-INSTANCE
.gitignore
ec2.tf
install_techmax.sh
README.md
ec2.tf M
ec2.tf > resource "aws_instance" "ec2_instance" > key_name
data "aws_ami" "amazon_linux_2" {
filter {
name = "name"
values = ["amzn2-ami-hvm*"]
}
}
# launch the ec2 instance and install website
resource "aws_instance" "ec2_instance" {
ami = data.aws_ami.amazon_linux_2.id
instance_type = "t2.micro"
subnet_id = aws_default_subnet.default_az1.id
vpc_security_group_ids = [aws_security_group.ec2_security_group.id]
key_name = "terraform-1keypair"
user_data = file("install_techmax.sh")
tags = {
Name = "techmax server"
}


EXPLORER
NEW-EC2-INSTANCE
.terraform
.gitignore
ec2.tf
M
install_techmax.sh
README.md
ec2.tf M
ec2.tf > terraform > backend "s3" > profile
}
# launch the ec2 instance and install website
resource "aws_instance" "ec2_instance" {
ami = data.aws_ami.amazon_linux_2.id
instance_type = "t2.micro"
subnet_id = aws_default_subnet.default_az1.id
vpc_security_group_ids = [aws_security_group.ec2_security_group.id]
key_name = "terraform-1keypair"
user_data = file("install_techmax.sh")
tags = {
Name = "techmax server"
}
}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\Val's\Desktop\new-ec2-instance> terraform init
Initializing the backend...
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.

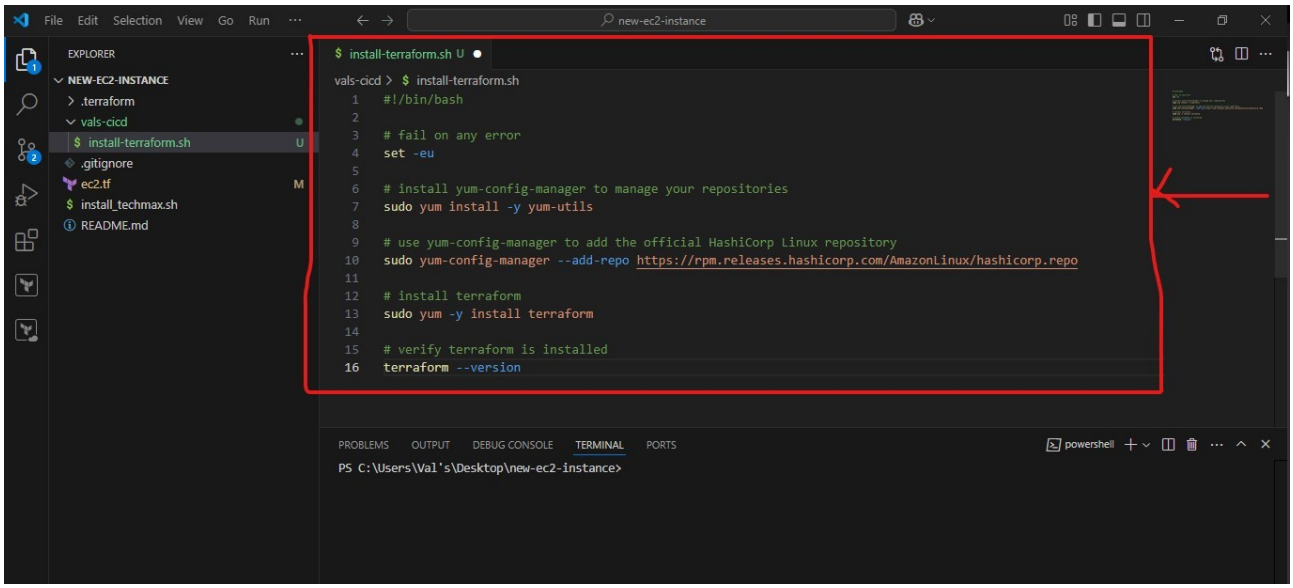

File
Edit
Selection
View
Go
Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
vals-cicd
.gitignore
ec2.tf
M
install_techmax.sh
README.md
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\Val's\Desktop\new-ec2-instance>



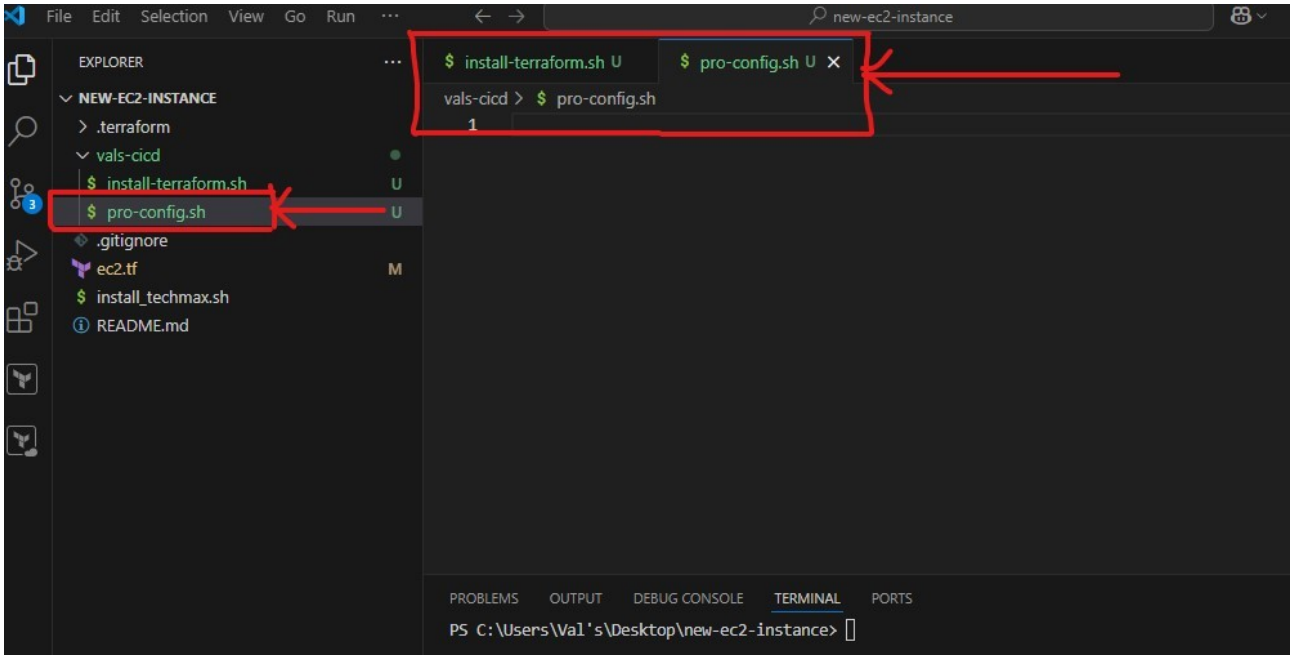
*Untitled - Notepad

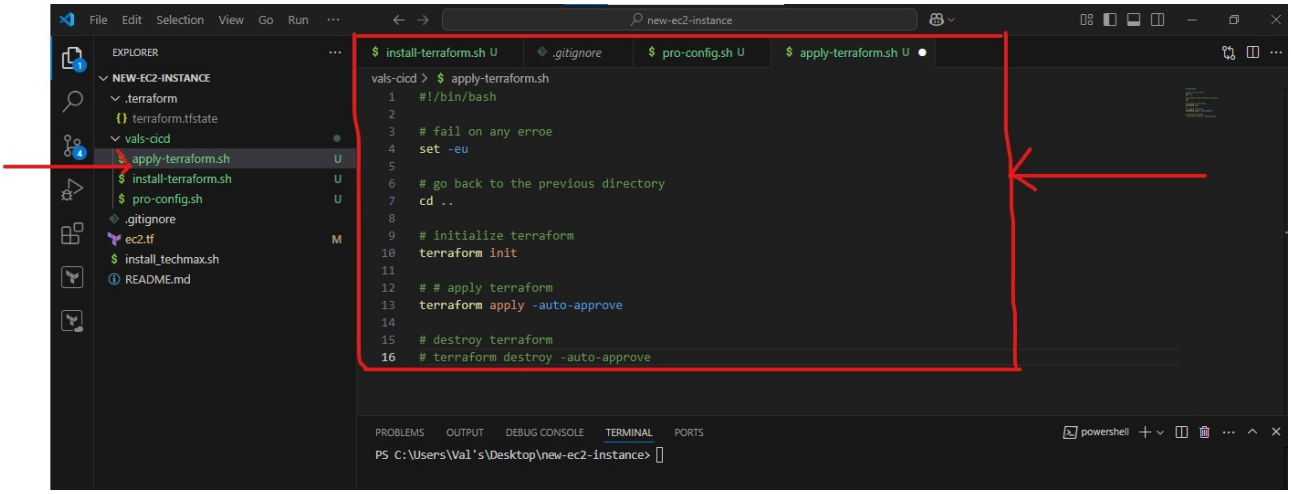
File Edit Format View Help

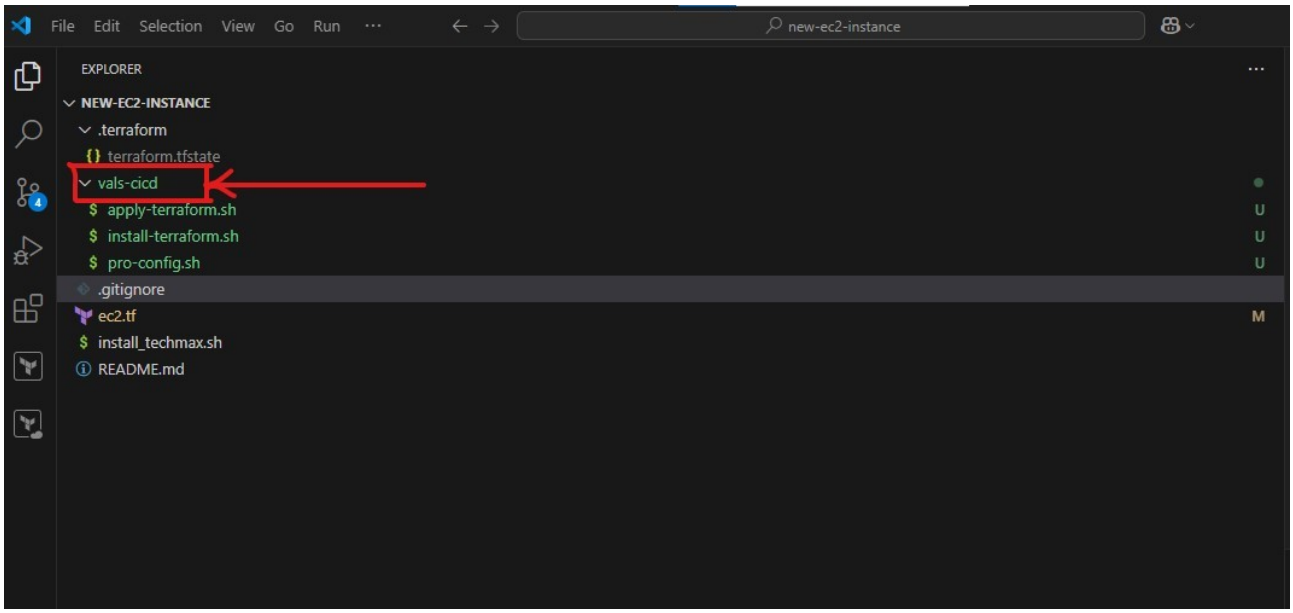
```
#!/bin/bash

# fail on any error
set -eu

# install yum-config-manager to manage your repositories
sudo yum install -y yum-utils

# use yum-config-manager to add the official HashiCorp Linux repository
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo

# install terraform
sudo yum -y install terraform

# verify terraform is installed
terraform --version
```

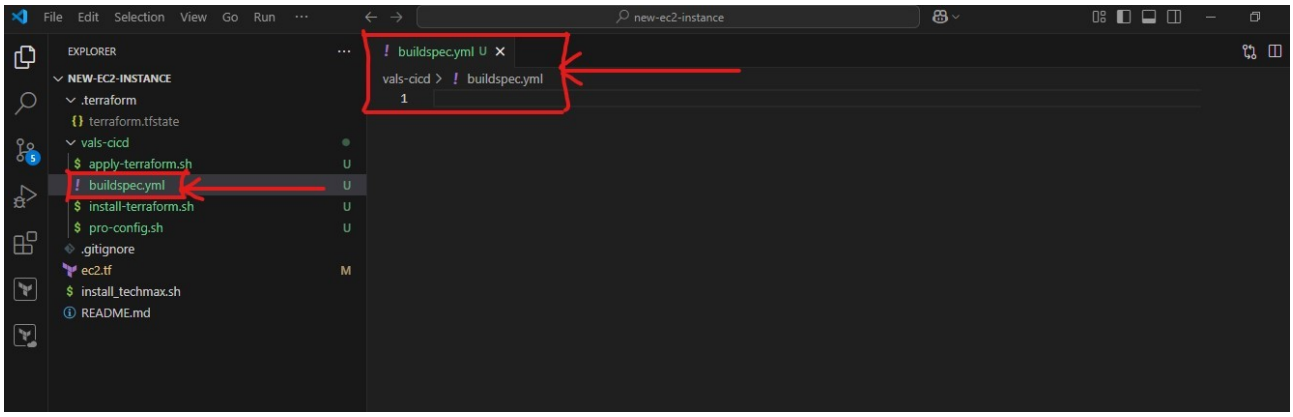

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
vals-cicd
install-terraform.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
install-terraform.sh U
vals-cicd > install-terraform.sh
#!/bin/bash
fail on any error
set -eu
# install yum-config-manager to manage your repositories
sudo yum install -y yum-utils
# use yum-config-manager to add the official HashiCorp Linux repository
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
# install terraform
sudo yum -y install terraform
# verify terraform is installed
terraform --version
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\Val's\Desktop\new-ec2-instance>

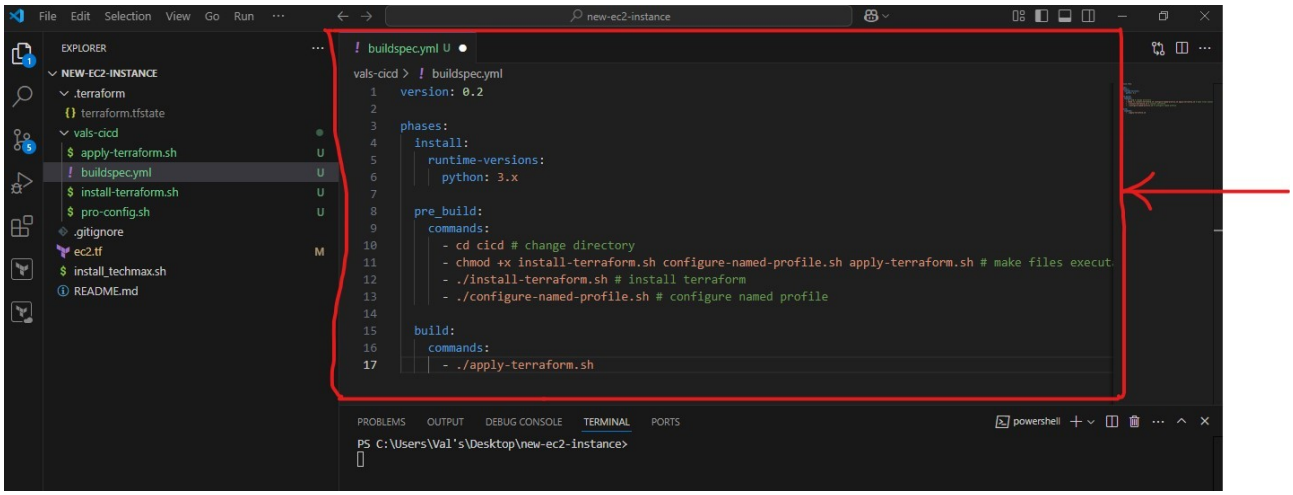

developer.hashicorp.com/terraform/install?product_intent=terraform#linux
Terraform Install Tutorials Documentation Registry Try Cloud
Search ⌘/ctrl K
Terraform Home
Install Terraform
Operating Systems
macOS
Windows
Linux
FreeBSD
OpenBSD
Solaris
Linux
Package manager
Ubuntu/Debian CentOS/RHEL Fedora Amazon Linux Homebrew
sudo yum install -y yum-utils shadow-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLi
sudo yum -y install terraform
Binary download
386
Version: 1.11.3
Download
AMD64
Version: 1.11.3
Download
About Terraform
Define cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share.
Featured docs
Introduction to Terraform
Configuration Language
Terraform CLI
HCP Terraform
Provider Use

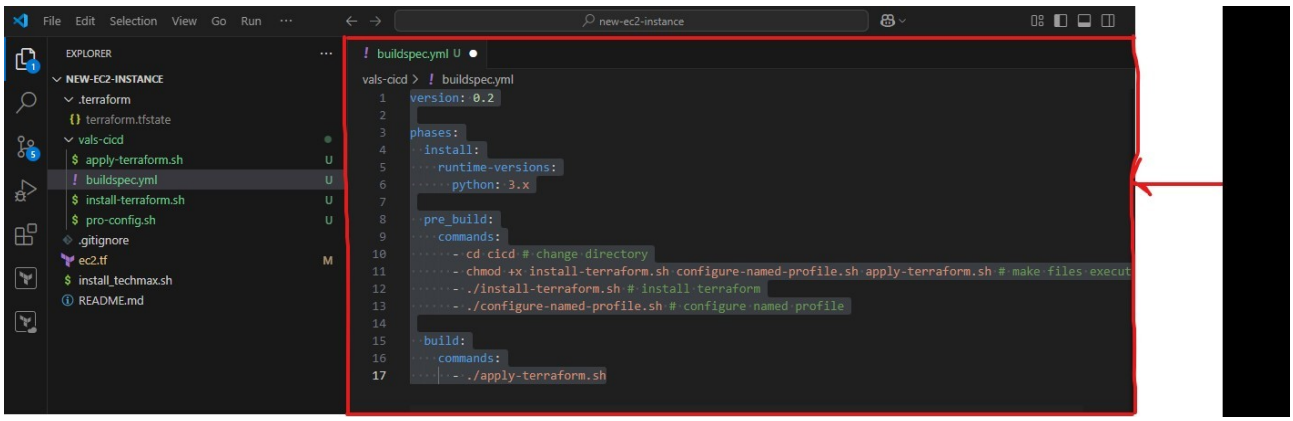

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
vals-cicd
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
install-terraform.sh U
pro-config.sh U
vals-cicd > pro-config.sh
1
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Val's\Desktop\new-ec2-instance>

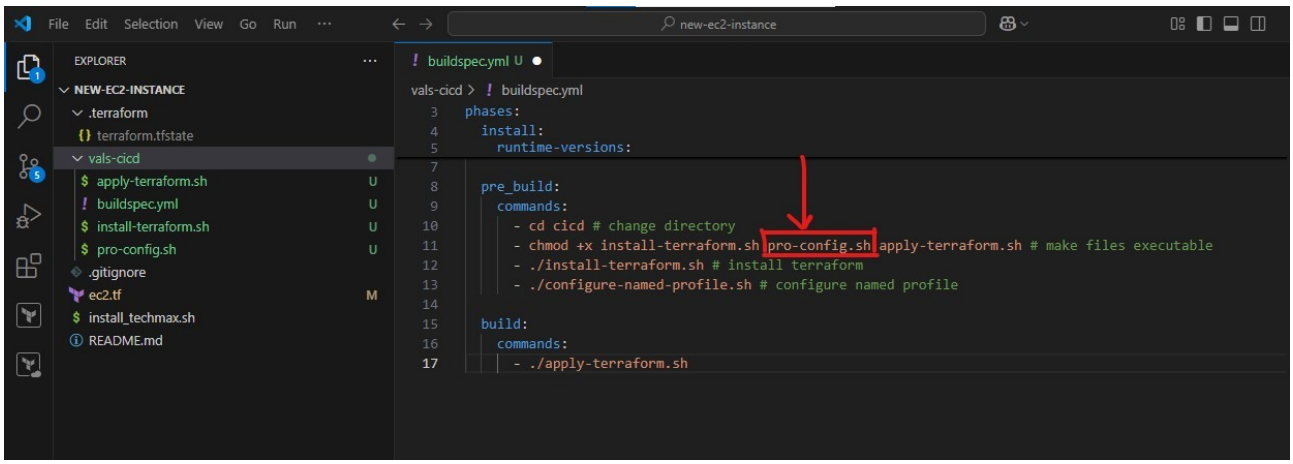

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh U
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
install-terraform.sh U
.gitignore
pro-config.sh U
apply-terraform.sh U
vals-cicd > apply-terraform.sh
#!/bin/bash
fail on any erroe
set -eu
go back to the previous directory
cd ..
initialize terraform
terraform init
apply terraform
terraform apply -auto-approve
destroy terraform
terraform destroy -auto-approve
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\Val's\Desktop\new-ec2-instance>

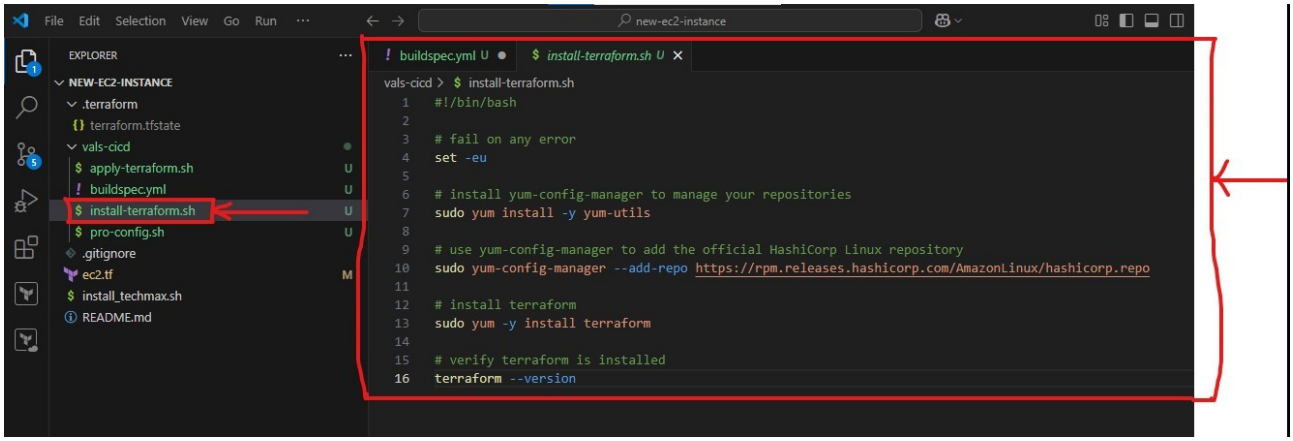

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh U
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md

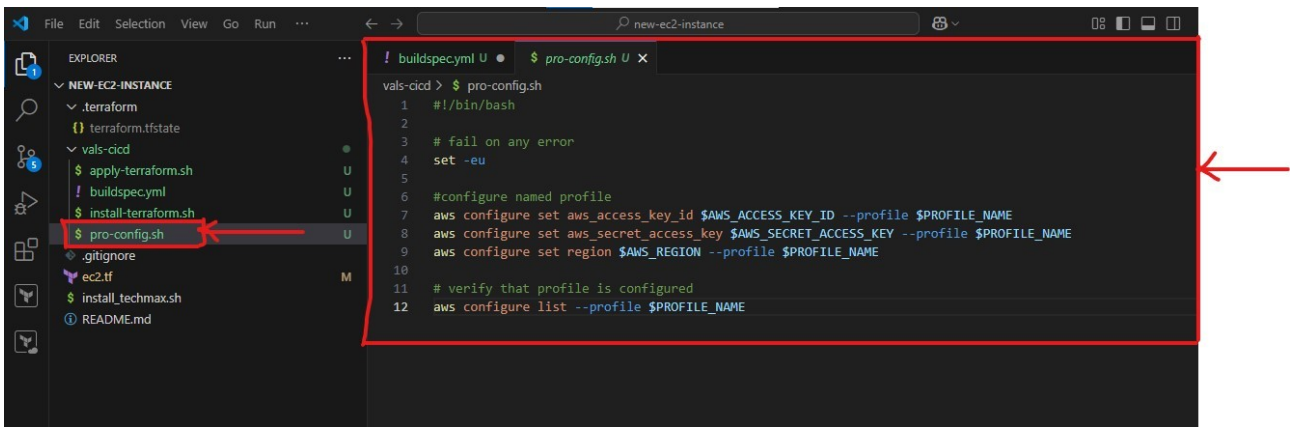

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh U
buildspec.yml U
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
buildspec.yml U
vals-cicd > buildspec.yml
1

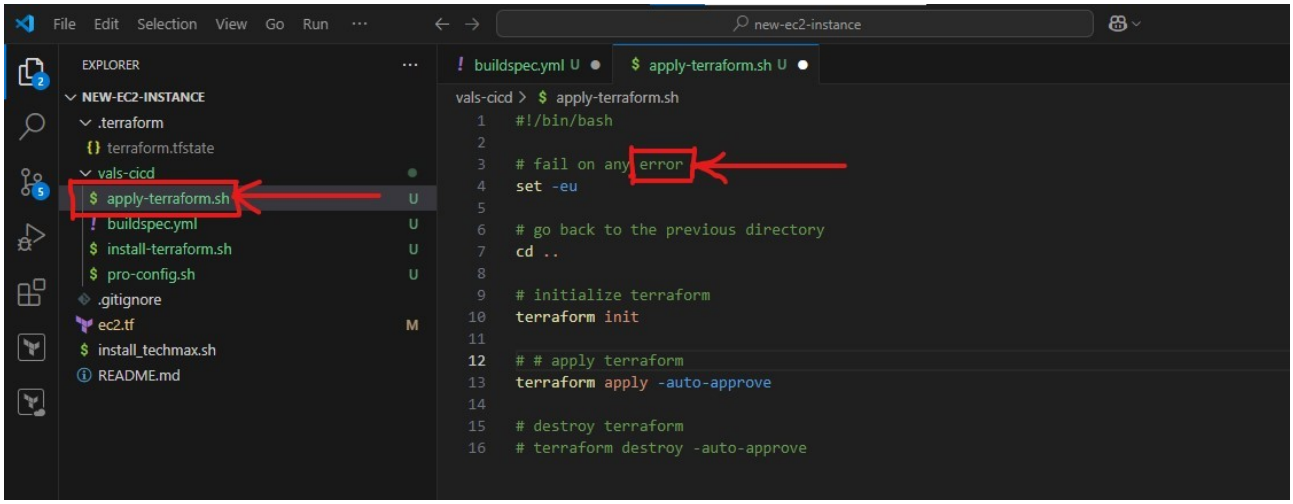

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh U
buildspec.yml U
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
buildspec.yml U
vals-cicd > buildspec.yml
1 version: 0.2
2
3 phases:
4 install:
5 runtime-versions:
6 python: 3.x
7
8 pre_build:
9 commands:
10 - cd cicd # change directory
11 - chmod +x install-terraform.sh configure-named-profile.sh apply-terraform.sh # make files execut
12 - ./install-terraform.sh # install terraform
13 - ./configure-named-profile.sh # configure named profile
14
15 build:
16 commands:
17 - ./apply-terraform.sh
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\Val's\Desktop\new-ec2-instance>

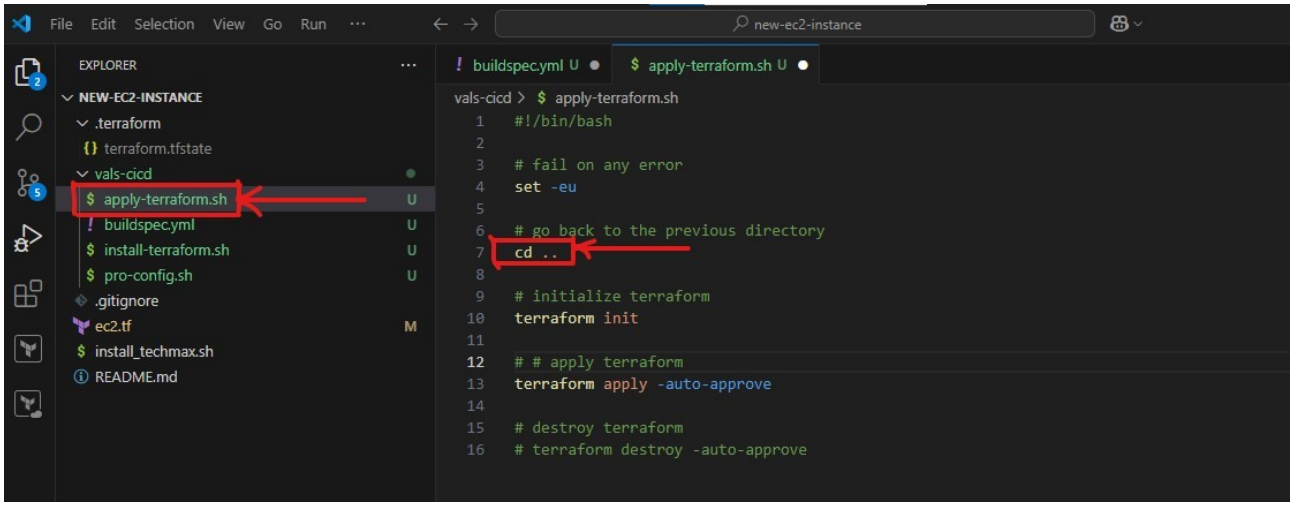

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh U
buildspec.yml U
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
buildspec.yml U
vals-cicd > buildspec.yml
1 version: 0.2
2
3 phases:
4 install:
5 runtime-versions:
6 python: 3.x
7
8 pre_build:
9 commands:
10 - cd cicd # change directory
11 - chmod +x install-terraform.sh configure-named-profile.sh apply-terraform.sh # make files execut
12 - ./install-terraform.sh # install terraform
13 - ./configure-named-profile.sh # configure named profile
14
15 build:
16 commands:
17 - ./apply-terraform.sh

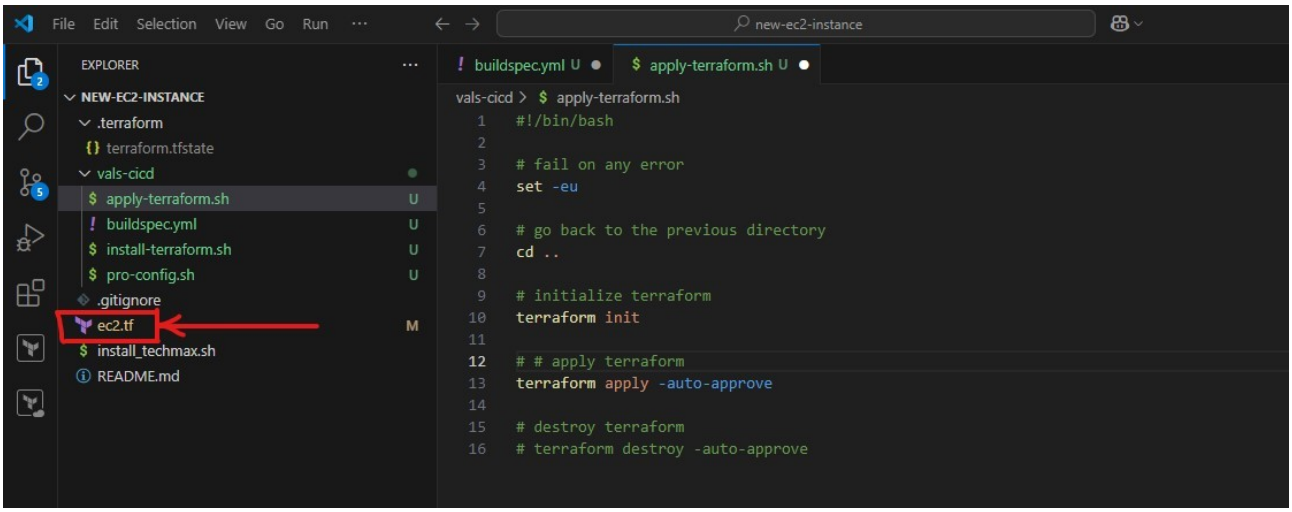

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
ec2.tf
install_techmax.sh
README.md
buildspec.yml U
vals-cicd > buildspec.yml
phases:
install:
runtime-versions:
pre_build:
commands:
- cd cicd # change directory
- chmod +x install-terraform.sh pro-config.sh apply-terraform.sh # make files executable
- ./install-terraform.sh # install terraform
- ./configure-named-profile.sh # configure named profile
build:
commands:
- ./apply-terraform.sh

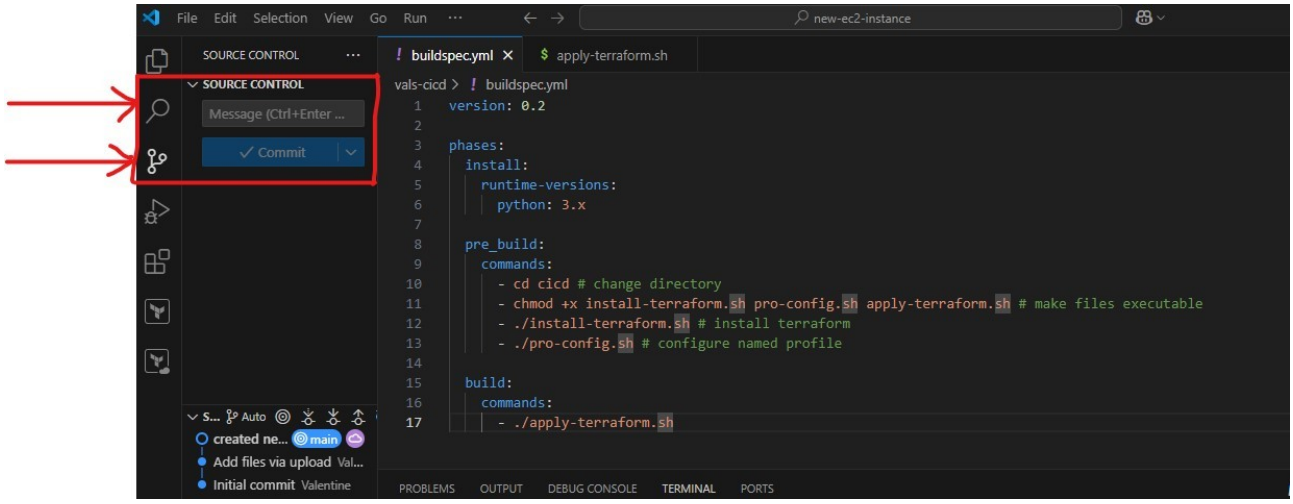

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
ec2.tf
install_techmax.sh
README.md
buildspec.yml U
install-terraform.sh U
vals-cicd > install-terraform.sh
#!/bin/bash
fail on any error
set -eu
# install yum-config-manager to manage your repositories
sudo yum install -y yum-utils
# use yum-config-manager to add the official HashiCorp Linux repository
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
# install terraform
sudo yum -y install terraform
# verify terraform is installed
terraform --version

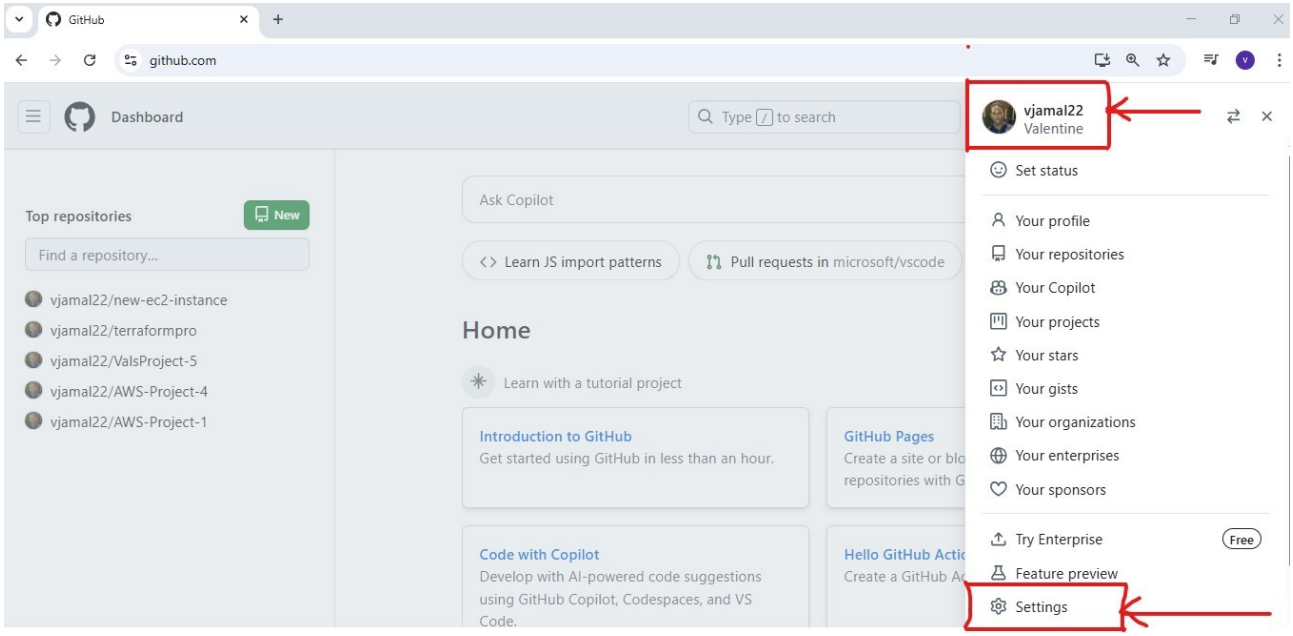

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
ec2.tf
install_techmax.sh
README.md
buildspec.yml U
pro-config.sh U
vals-cicd > pro-config.sh
#!/bin/bash
fail on any error
set -eu
#configure named profile
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profile $PROFILE_NAME
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --profile $PROFILE_NAME
aws configure set region $AWS_REGION --profile $PROFILE_NAME
verify that profile is configured
aws configure list --profile $PROFILE_NAME

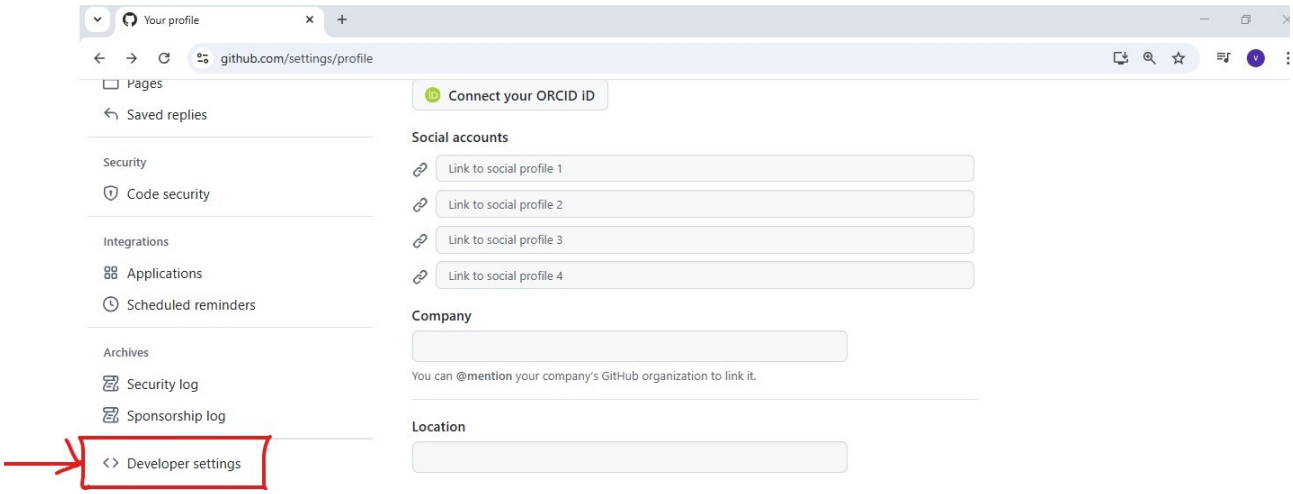

File Edit Selection View Go Run ···
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh U
buildspec.yml U
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
buildspec.yml U
apply-terraform.sh U
vals-cicd > apply-terraform.sh
#!/bin/bash
fail on any error
set -eu
go back to the previous directory
cd ..
initialize terraform
terraform init
apply terraform
terraform apply -auto-approve
destroy terraform
terraform destroy -auto-approve

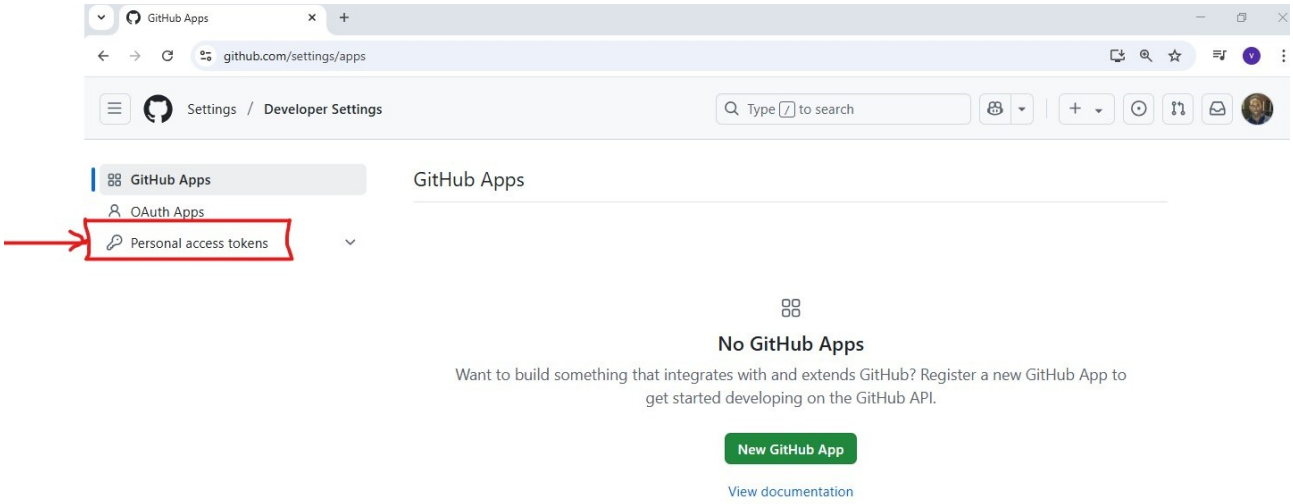

File Edit Selection View Go Run ···
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh U
buildspec.yml U
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
buildspec.yml U
apply-terraform.sh U
vals-cicd > apply-terraform.sh
#!/bin/bash
fail on any error
set -eu
go back to the previous directory
cd ..
initialize terraform
terraform init
apply terraform
terraform apply -auto-approve
destroy terraform
terraform destroy -auto-approve

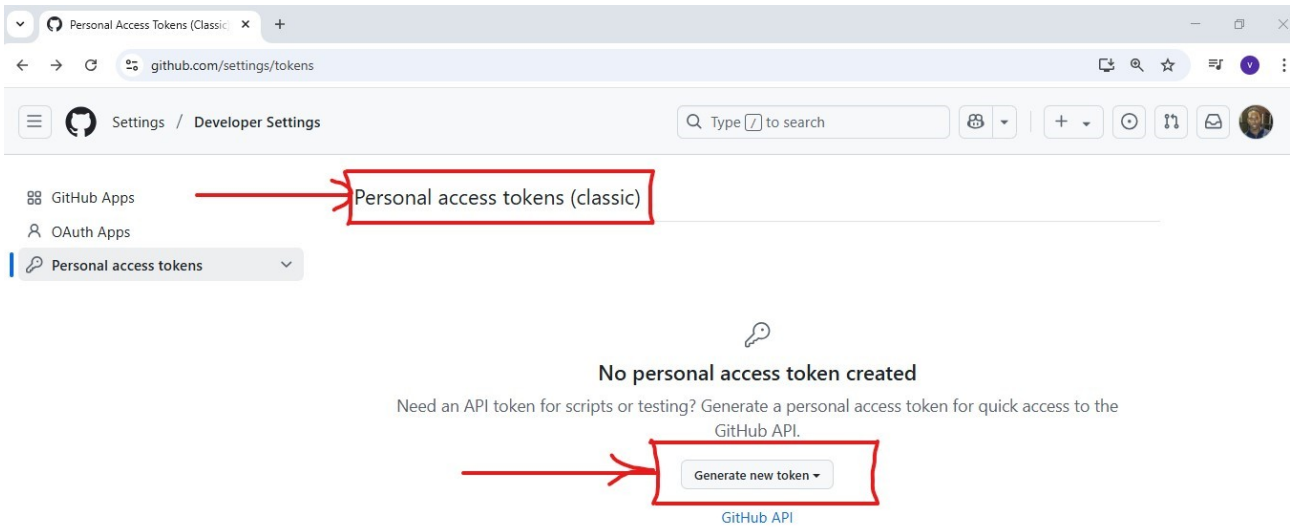

File Edit Selection View Go Run ···
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
terraform.tfstate
vals-cicd
apply-terraform.sh U
buildspec.yml U
install-terraform.sh U
pro-config.sh U
.gitignore
ec2.tf M
install_techmax.sh
README.md
buildspec.yml U
apply-terraform.sh U
vals-cicd > apply-terraform.sh
#!/bin/bash
fail on any error
set -eu
go back to the previous directory
cd ..
initialize terraform
terraform init
apply terraform
terraform apply -auto-approve
destroy terraform
terraform destroy -auto-approve

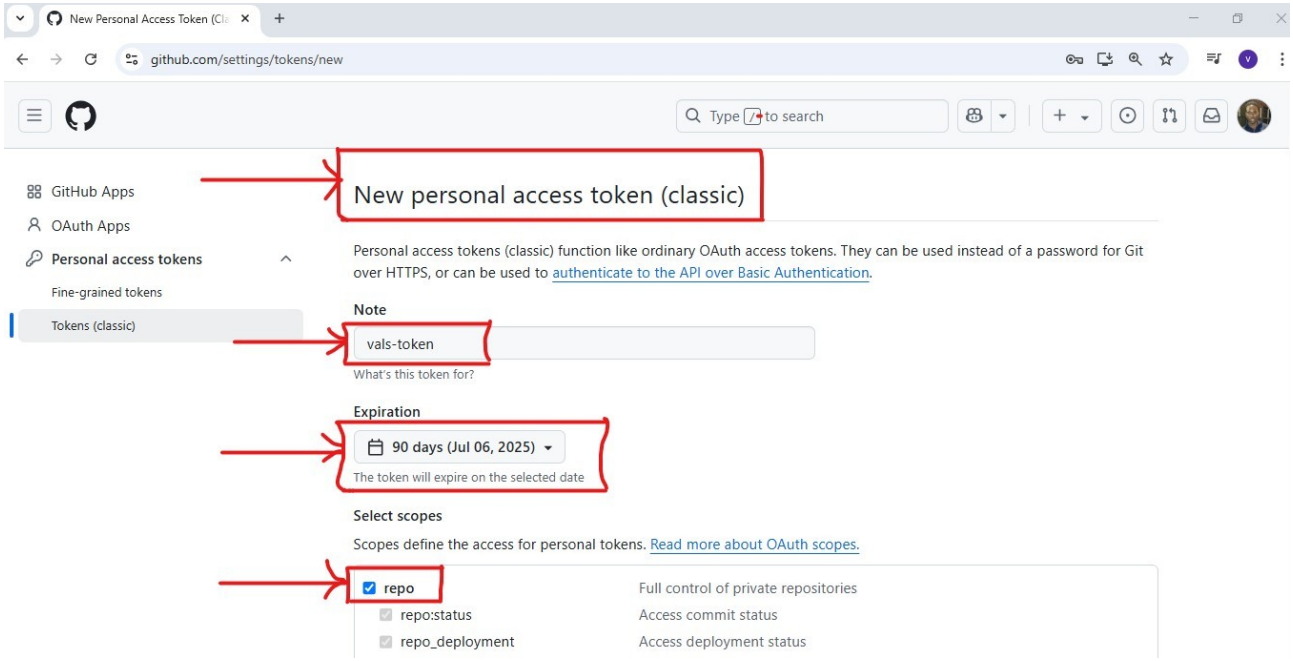

File Edit Selection View Go Run
new-ec2-instance
SOURCE CONTROL
buildspec.yml
apply-terraform.sh
vals-cicd > buildspec.yml
Message (Ctrl+Enter ...
Commit
version: 0.2
phases:
install:
runtime-versions:
python: 3.x
pre_build:
commands:
- cd cicd # change directory
- chmod +x install-terraform.sh pro-config.sh apply-terraform.sh # make files executable
- ./install-terraform.sh # install terraform
- ./pro-config.sh # configure named profile
build:
commands:
- ./apply-terraform.sh
created ne... main
Add files via upload
Initial commit Valentine
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

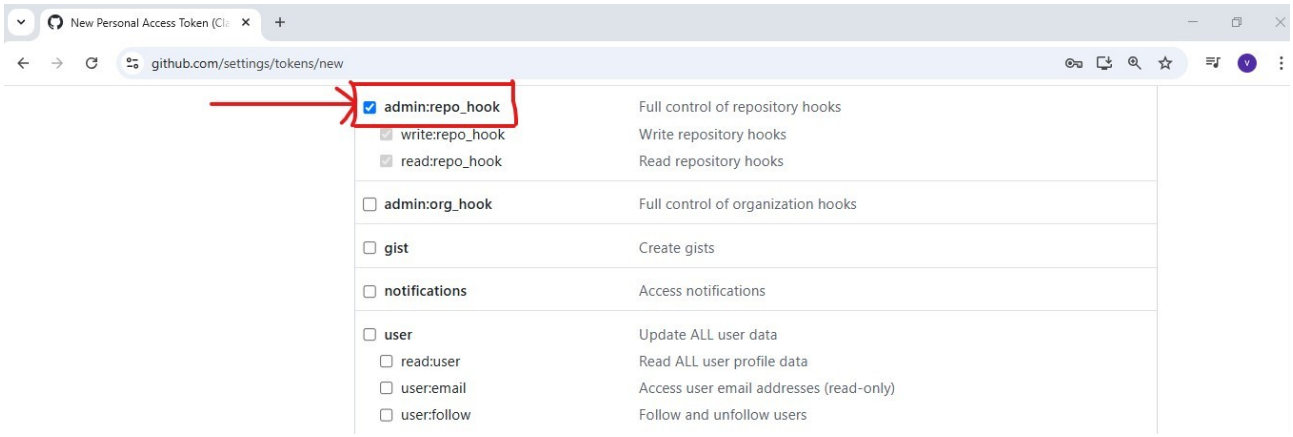

GitHub
github.com
Dashboard
Type / to search
vjamal22
Valentine
Set status
Your profile
Your repositories
Your Copilot
Your projects
Your stars
Your gists
Your organizations
Your enterprises
Your sponsors
Try Enterprise
Free
Feature preview
Settings
Top repositories
New
Find a repository...
vjamal22/new-ec2-instance
vjamal22/terraformpro
vjamal22/ValsProject-5
vjamal22/AWS-Project-4
vjamal22/AWS-Project-1
Ask Copilot
Learn JS import patterns
Pull requests in microsoft/vscode
Home
Learn with a tutorial project
Introduction to GitHub
Get started using GitHub in less than an hour.
GitHub Pages
Code with Copilot
Develop with AI-powered code suggestions using GitHub Copilot, Codespaces, and VS Code.

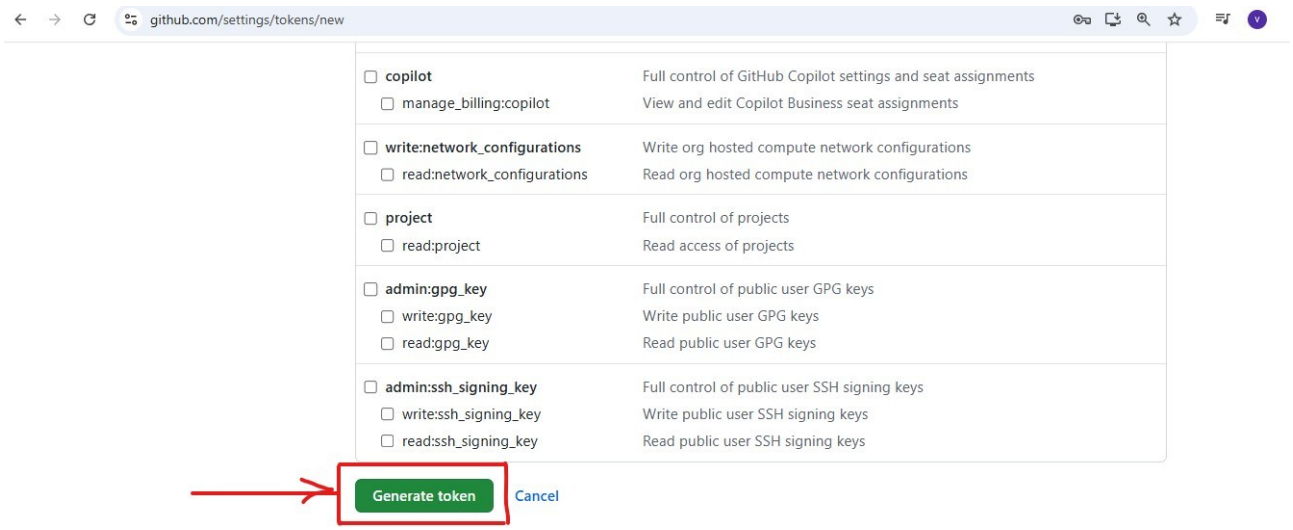

Your profile
github.com/settings/profile
Pages
Saved replies
Security
Code security
Integrations
Applications
Scheduled reminders
Archives
Security log
Sponsorship log
Developer settings
Connect your ORCID iD
Social accounts
Link to social profile 1
Link to social profile 2
Link to social profile 3
Link to social profile 4
Company
You can @mention your company's GitHub organization to link it.
Location

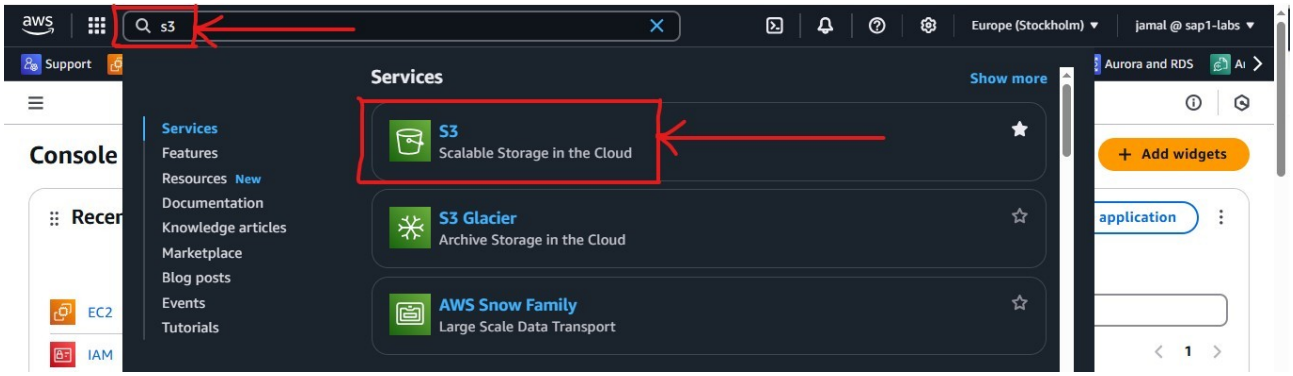

GitHub Apps
github.com/settings/apps
Settings / Developer Settings
Type / to search
GitHub Apps
OAuth Apps
Personal access tokens
GitHub Apps
No GitHub Apps
Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API.
New GitHub App
View documentation

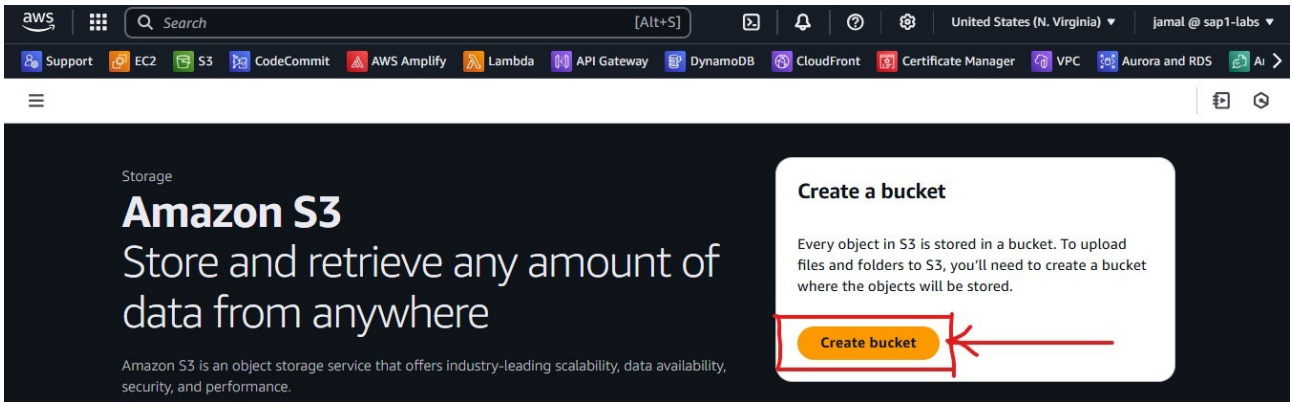

Personal Access Tokens (Classic
github.com/settings/tokens
Settings / Developer Settings
Type / to search
GitHub Apps
OAuth Apps
Personal access tokens
Personal access tokens (classic)
No personal access token created
Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.
Generate new token
GitHub API

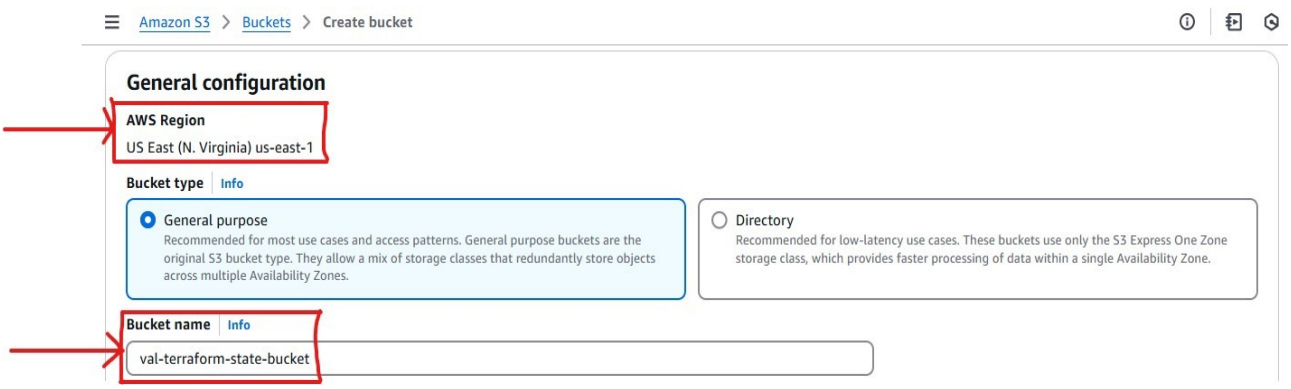

New Personal Access Token (Cl
github.com/settings/tokens/new
Type / to search
GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens
Tokens (classic)
New personal access token (classic)
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.
Note
vals-token
What's this token for?
Expiration
90 days (Jul 06, 2025)
The token will expire on the selected date
Select scopes
Scopes define the access for personal tokens. Read more about OAuth scopes.
repo
Full control of private repositories
repo:status
Access commit status
repo_deployment
Access deployment status

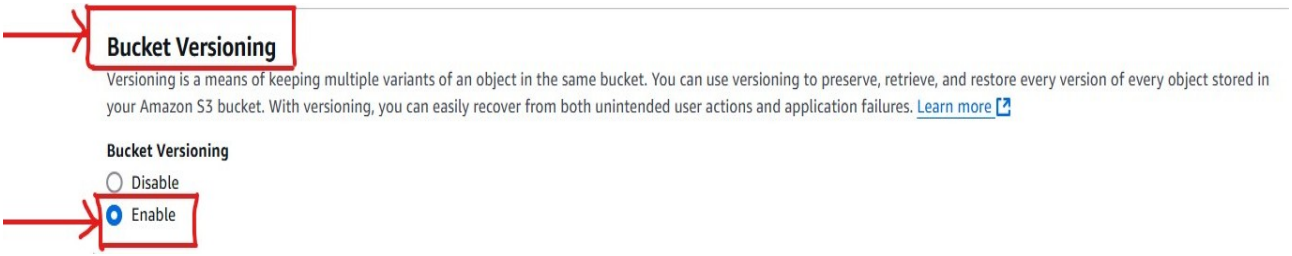

New Personal Access Token (Cl
github.com/settings/tokens/new
admin:repo_hook
Full control of repository hooks
write:repo_hook
Write repository hooks
read:repo_hook
Read repository hooks
admin:org_hook
Full control of organization hooks
gist
Create gists
notifications
Access notifications
user
Update ALL user data
read:user
Read ALL user profile data
user:email
Access user email addresses (read-only)
user:follow
Follow and unfollow users

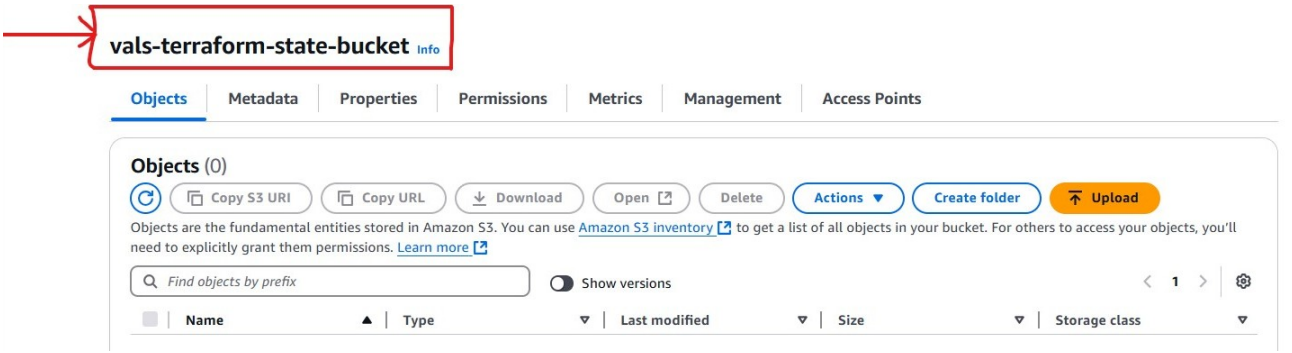

github.com/settings/tokens/new
copilot
Full control of GitHub Copilot settings and seat assignments
manage_billing:copilot
View and edit Copilot Business seat assignments
write:network_configurations
Write org hosted compute network configurations
read:network_configurations
Read org hosted compute network configurations
project
Full control of projects
read:project
Read access of projects
admin:gpg_key
Full control of public user GPG keys
write:gpg_key
Write public user GPG keys
read:gpg_key
Read public user GPG keys
admin:ssh_signing_key
Full control of public user SSH signing keys
write:ssh_signing_key
Write public user SSH signing keys
read:ssh_signing_key
Read public user SSH signing keys
Generate token
Cancel

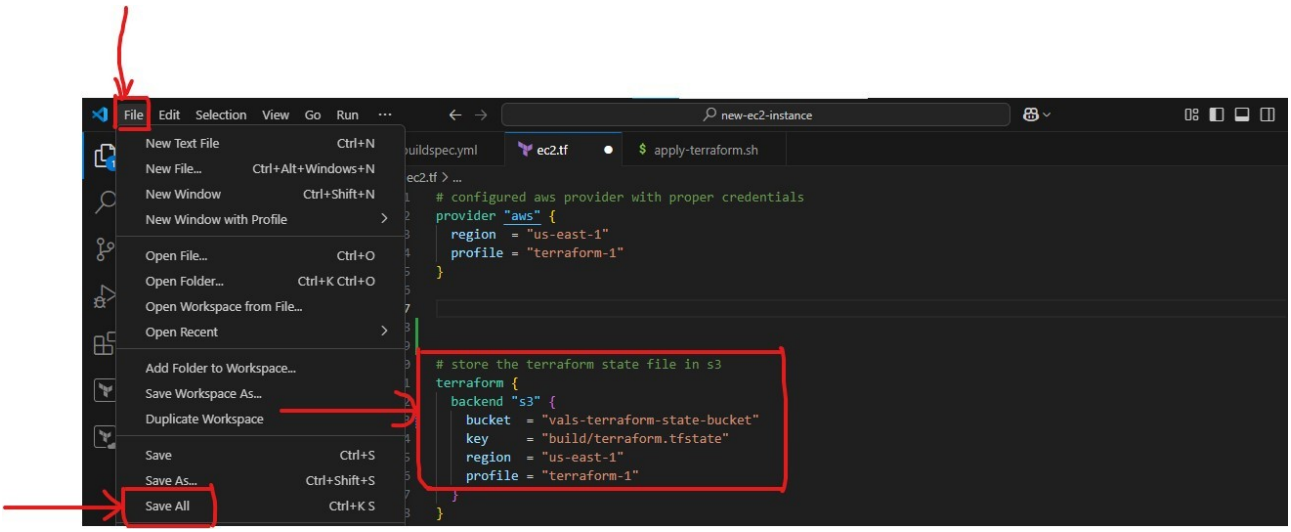

s3
Europe (Stockholm)
jamal @ sap1-labs
Support
Aurora and RDS
Services
Show more
Services
Features
Resources New
Documentation
Knowledge articles
Marketplace
Blog posts
Events
Tutorials
S3
Scalable Storage in the Cloud
S3 Glacier
Archive Storage in the Cloud
AWS Snow Family
Large Scale Data Transport
Console
Add widgets
application
EC2
IAM

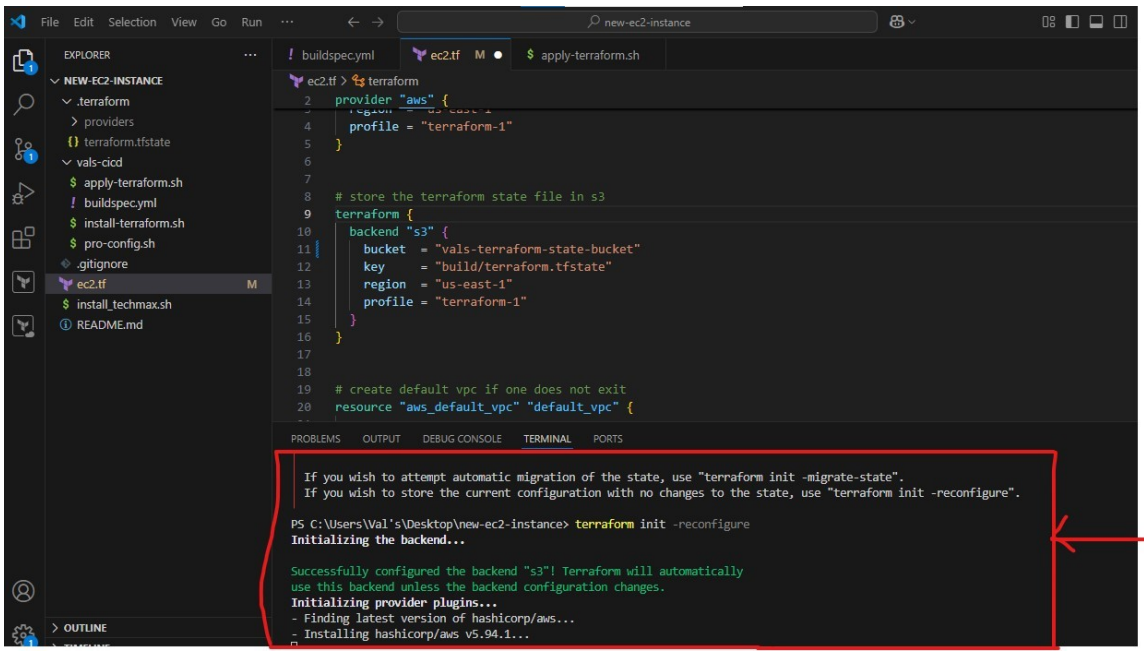

Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and RDS
Storage
Amazon S3
Store and retrieve any amount of data from anywhere
Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.
Create a bucket
Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.
Create bucket

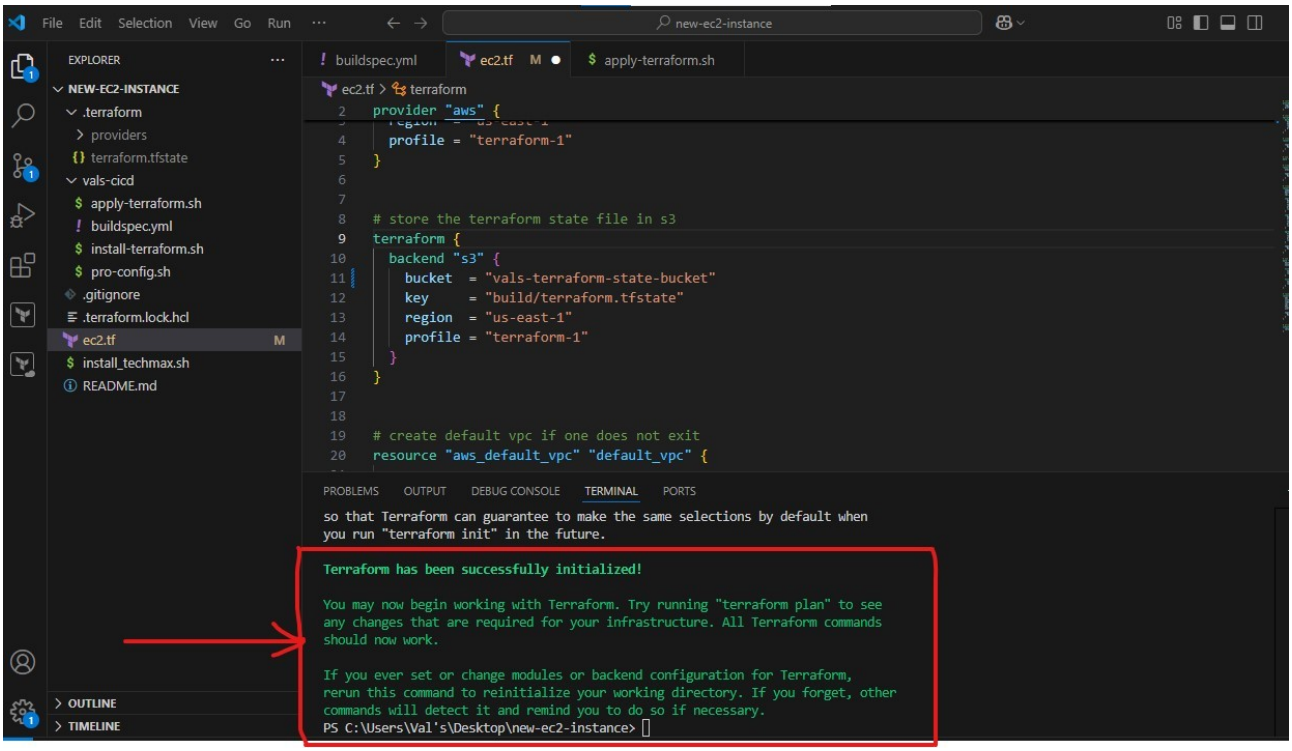

Amazon S3 > Buckets > Create bucket
General configuration
AWS Region
US East (N. Virginia) us-east-1
Bucket type Info
General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
val-terraform-state-bucket

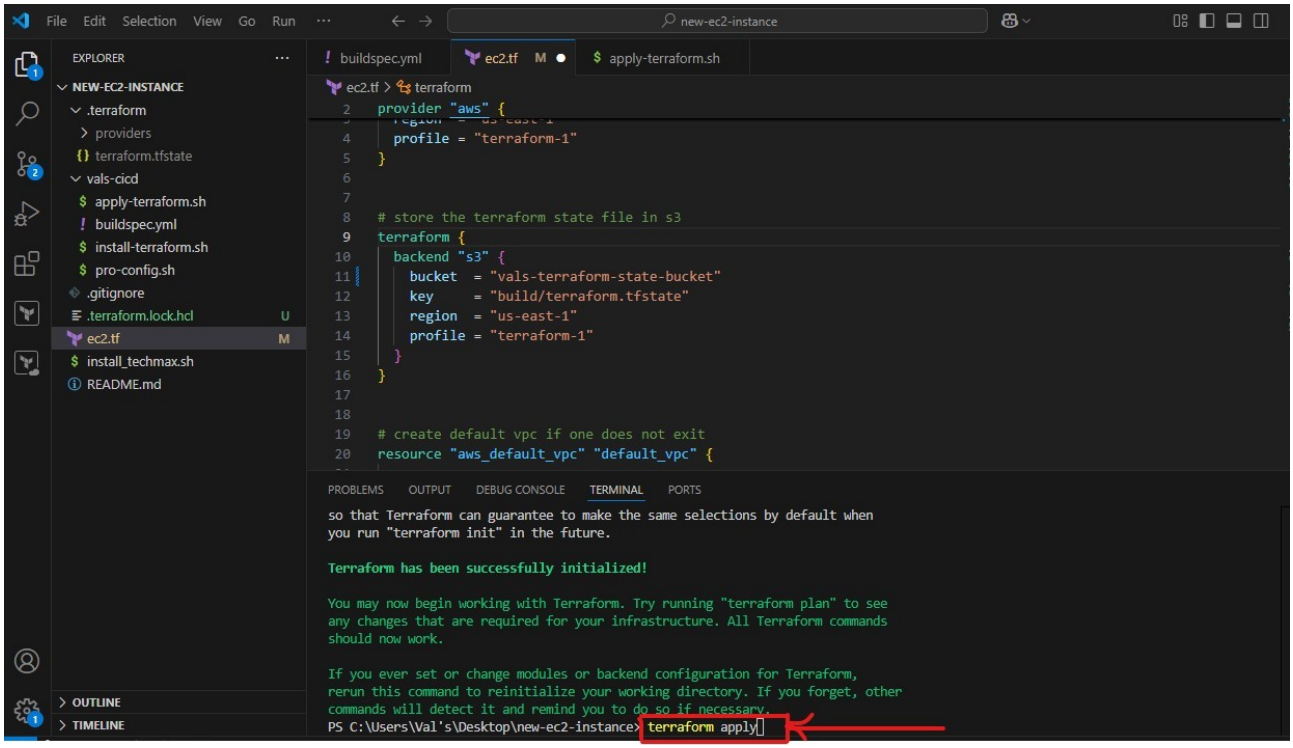

Bucket Versioning
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more
Bucket Versioning
Disable
Enable

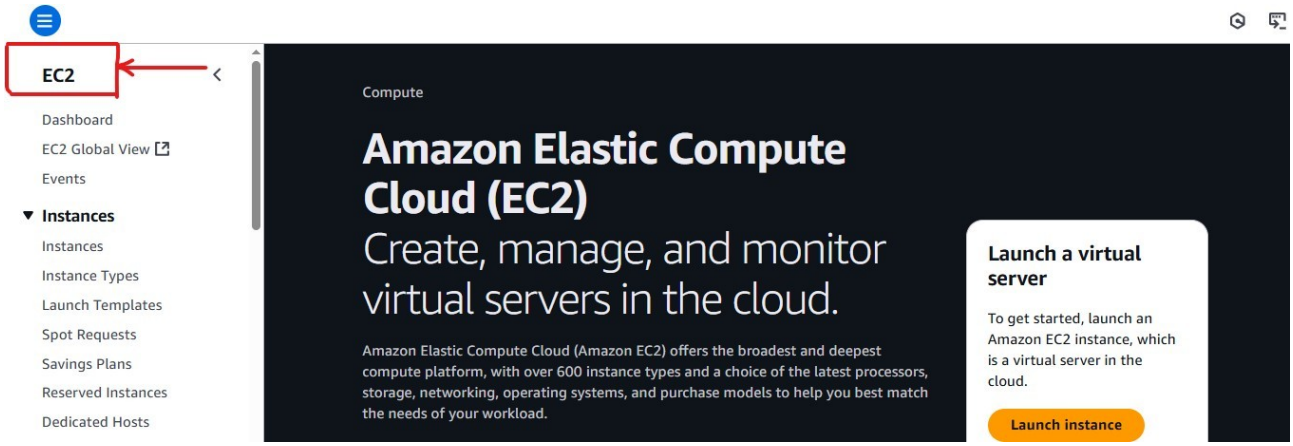

vals-terraform-state-bucket Info
Objects
Metadata
Properties
Permissions
Metrics
Management
Access Points
Objects (0)
Copy S3 URI
Copy URL
Download
Open
Delete
Actions
Create folder
Upload
Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects 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
Name
Type
Last modified
Size
Storage class


File
Edit
Selection
View
Go
Run
new-ec2-instance
New Text File Ctrl+N
New File... Ctrl+Alt+Windows+N
New Window Ctrl+Shift+N
New Window with Profile
Open File... Ctrl+O
Open Folder... Ctrl+K Ctrl+O
Open Workspace from File...
Open Recent
Add Folder to Workspace...
Save Workspace As...
Duplicate Workspace
Save Ctrl+S
Save As... Ctrl+Shift+S
Save All Ctrl+K S
ec2.tf
apply-terraform.sh
# configured aws provider with proper credentials
provider "aws" {
region = "us-east-1"
profile = "terraform-1"
}
# store the terraform state file in s3
terraform {
backend "s3" {
bucket = "vals-terraform-state-bucket"
key = "build/terraform.tfstate"
region = "us-east-1"
profile = "terraform-1"
}
}

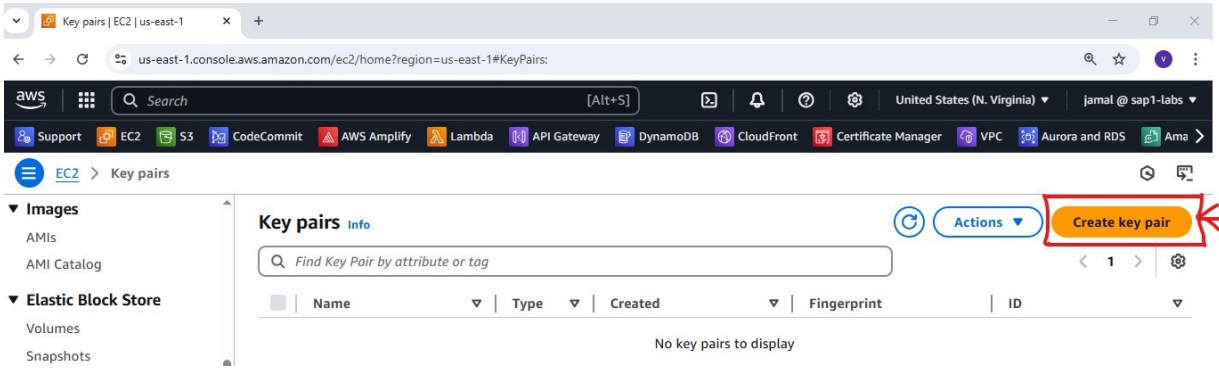

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
ec2.tf
install_techmax.sh
README.md
OUTLINE
buildspec.yml
ec2.tf M
apply-terraform.sh
ec2.tf > terraform
provider "aws" {
profile = "terraform-1"
}
# store the terraform state file in s3
terraform {
backend "s3" {
bucket = "vals-terraform-state-bucket"
key = "build/terraform.tfstate"
region = "us-east-1"
profile = "terraform-1"
}
}
# create default vpc if one does not exit
resource "aws_default_vpc" "default_vpc" {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
If you wish to attempt automatic migration of the state, use "terraform init -migrate-state".
If you wish to store the current configuration with no changes to the state, use "terraform init -reconfigure".
PS C:\Users\Val's\Desktop\new-ec2-instance> terraform init -reconfigure
Initializing the backend...
Successfully configured the backend "s3"! Terraform will automatically
use this backend unless the backend configuration changes.
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.94.1...

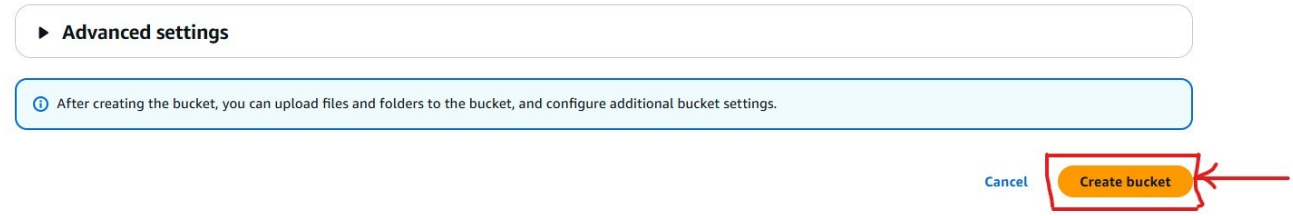

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
OUTLINE
TIMELINE
buildspec.yml
ec2.tf M
apply-terraform.sh
ec2.tf > terraform
provider "aws" {
profile = "terraform-1"
}
# store the terraform state file in s3
terraform {
backend "s3" {
bucket = "vals-terraform-state-bucket"
key = "build/terraform.tfstate"
region = "us-east-1"
profile = "terraform-1"
}
}
# create default vpc if one does not exit
resource "aws_default_vpc" "default_vpc" {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\Val's\Desktop\new-ec2-instance>

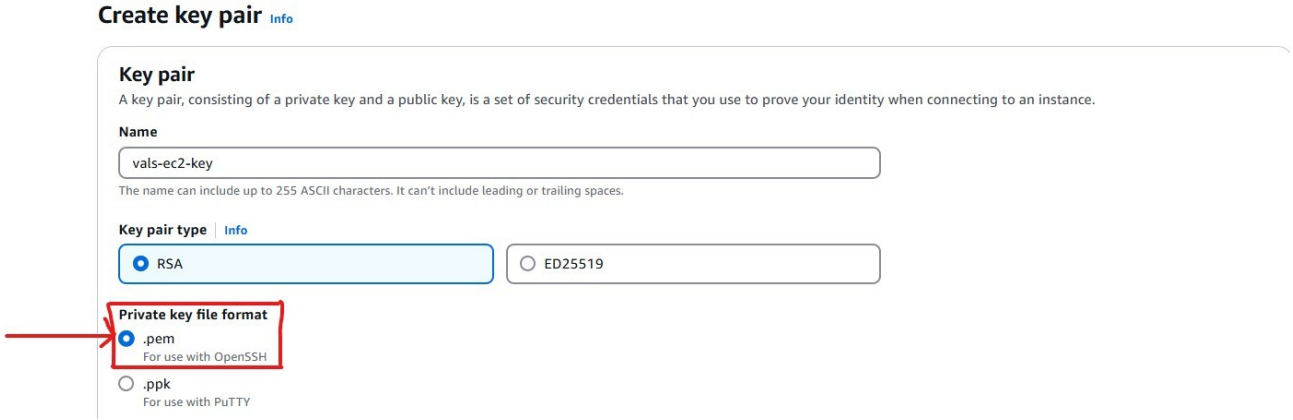

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
OUTLINE
TIMELINE
buildspec.yml
ec2.tf M
apply-terraform.sh
ec2.tf > terraform
provider "aws" {
profile = "terraform-1"
}
# store the terraform state file in s3
terraform {
backend "s3" {
bucket = "vals-terraform-state-bucket"
key = "build/terraform.tfstate"
region = "us-east-1"
profile = "terraform-1"
}
}
# create default vpc if one does not exit
resource "aws_default_vpc" "default_vpc" {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
PS C:\Users\Val's\Desktop\new-ec2-instance> terraform apply

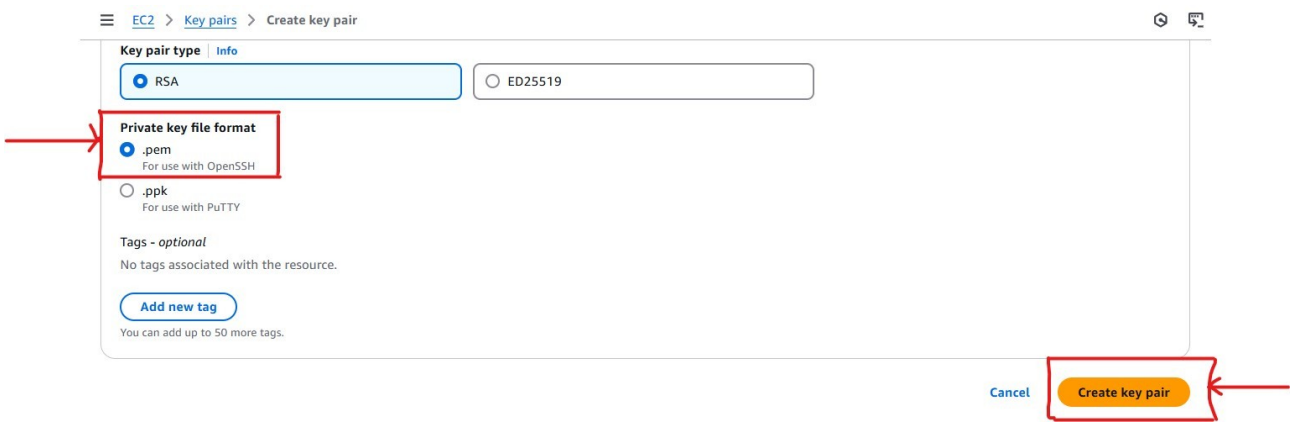

EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Compute
Amazon Elastic Compute Cloud (EC2)
Create, manage, and monitor virtual servers in the cloud.
Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.
Launch a virtual server
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance

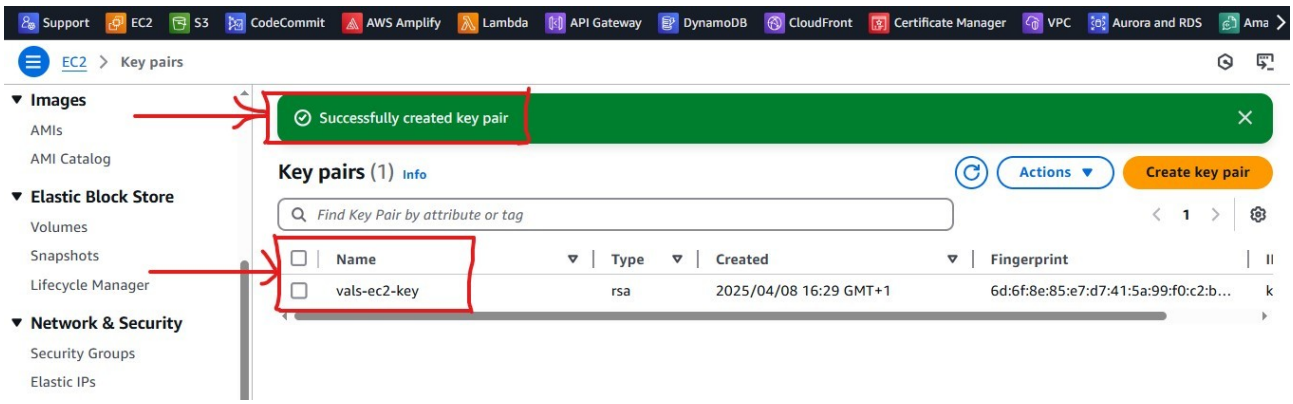

Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
Compute
Amazon Elastic Compute Cloud (EC2)
Create, manage, and monitor virtual servers in the cloud.
Amazon Elastic Compute Cloud (Amazon EC2) offers the broadest and deepest compute platform, with over 600 instance types and a choice of the latest processors, storage, networking, operating systems, and purchase models to help you best match the needs of your workload.
Launch a virtual server
To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.
Launch instance

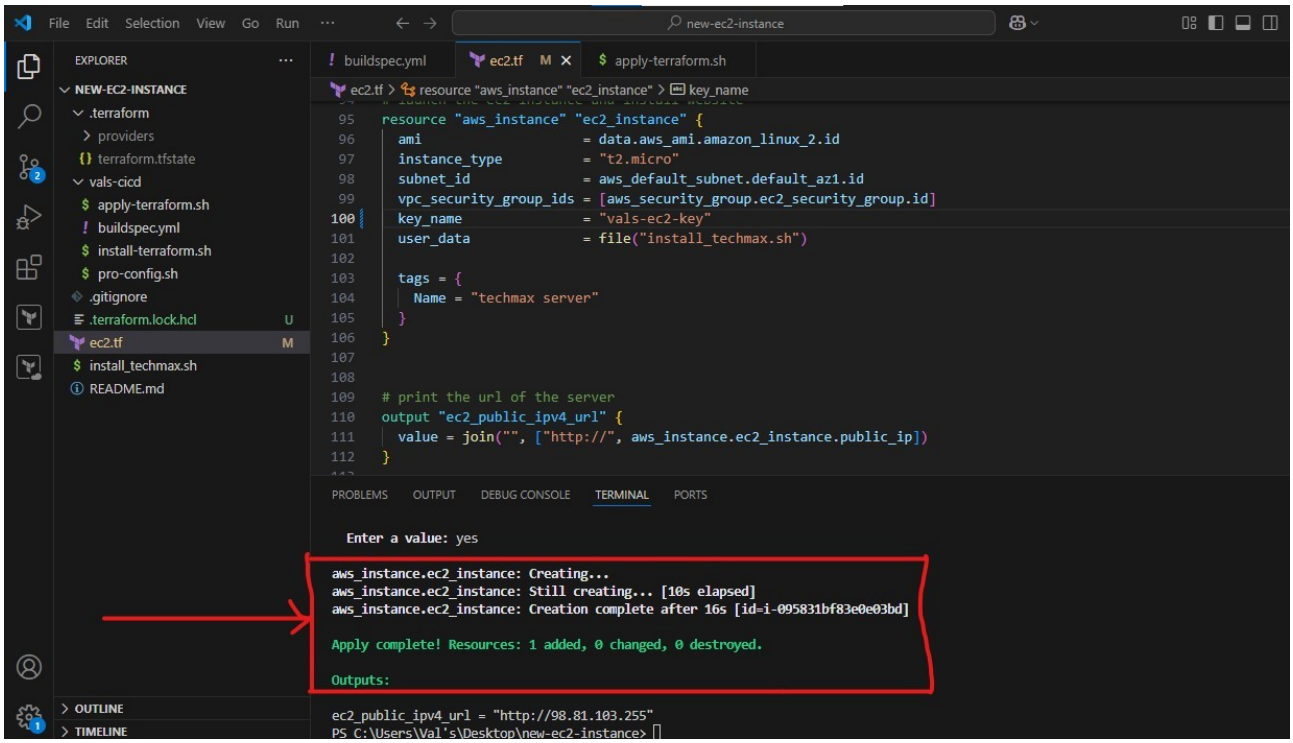

Key pairs | EC2 | us-east-1
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#KeyPairs:
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and RDS
EC2 > Key pairs
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Key pairs Info
Actions
Create key pair
Find Key Pair by attribute or tag
Name
Type
Created
Fingerprint
ID
No key pairs to display

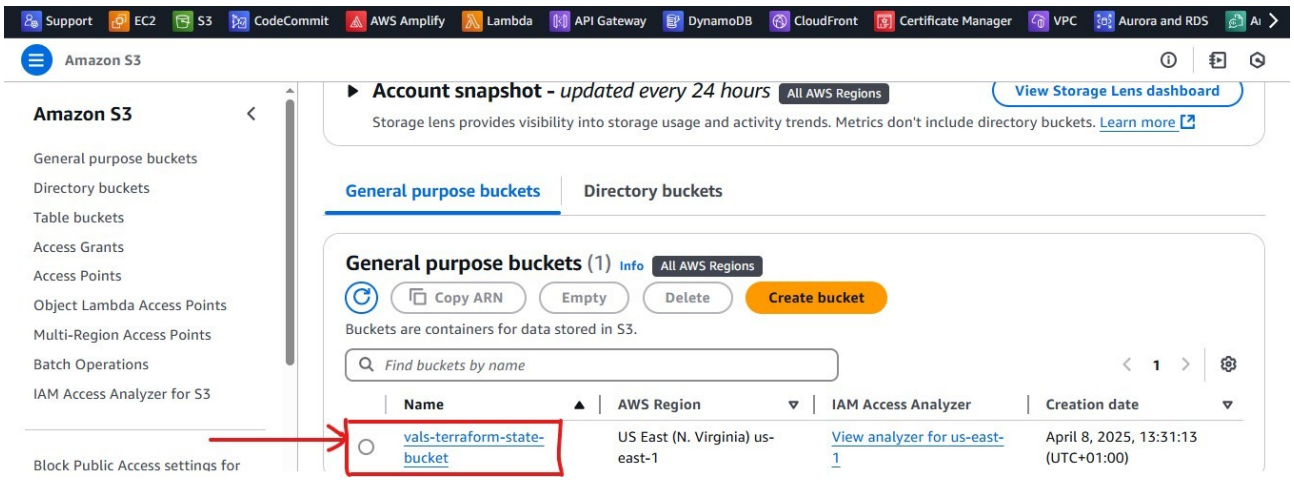

Advanced settings
After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.
Cancel
Create bucket

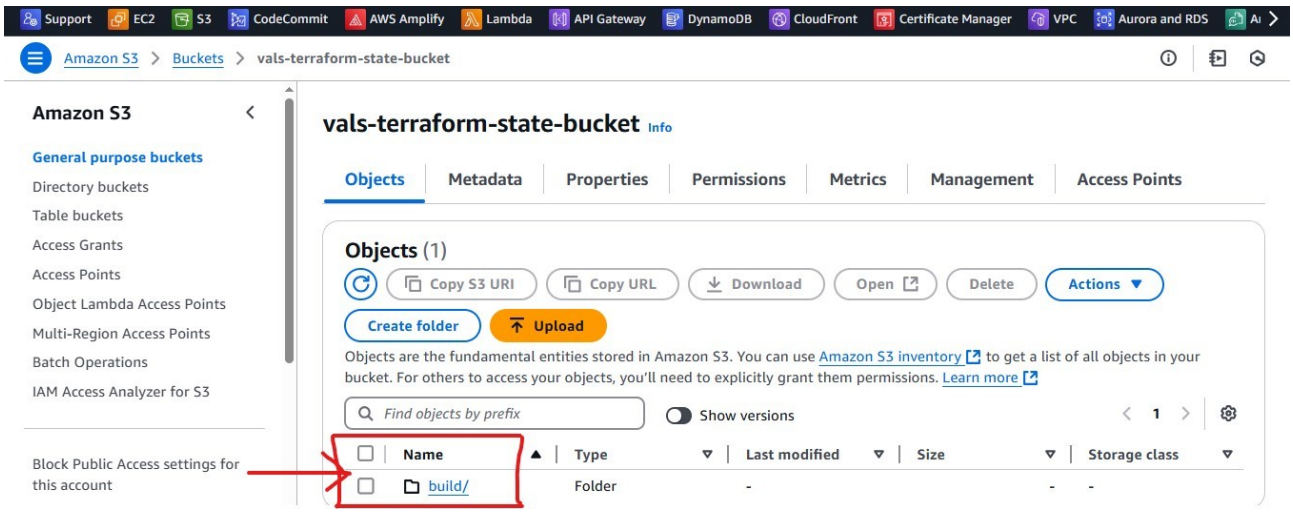

Create key pair Info
Key pair
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.
Name
vals-ec2-key
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.
Key pair type Info
RSA
ED25519
Private key file format
.pem
For use with OpenSSH
.ppk
For use with PuTTY

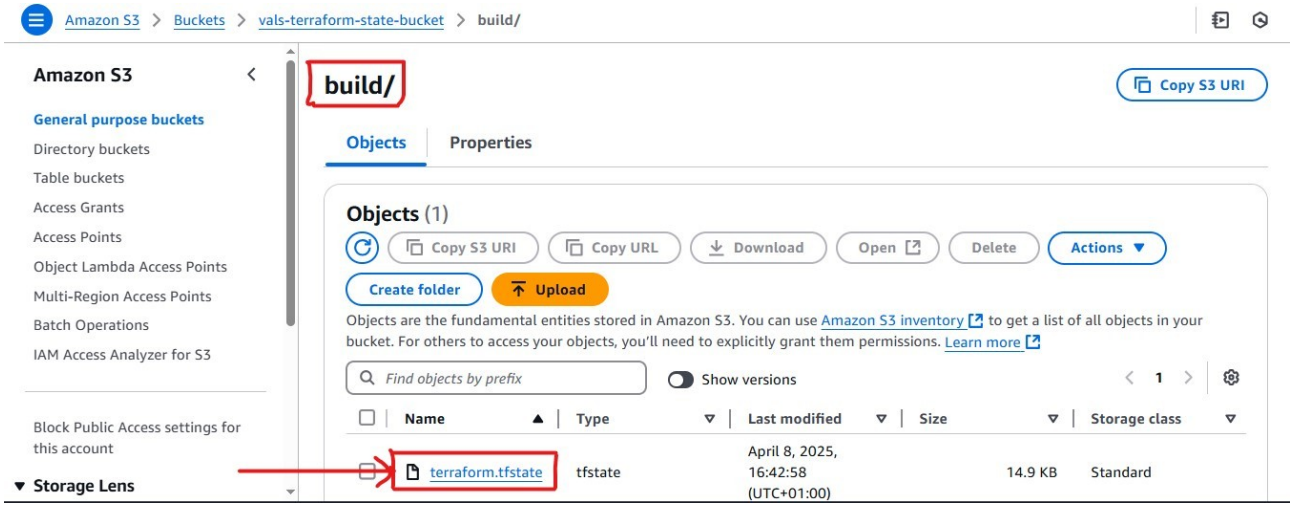

EC2 > Key pairs > Create key pair
Key pair type Info
RSA
ED25519
Private key file format
.pem
For use with OpenSSH
.ppk
For use with PuTTY
Tags - optional
No tags associated with the resource.
Add new tag
You can add up to 50 more tags.
Cancel
Create key pair

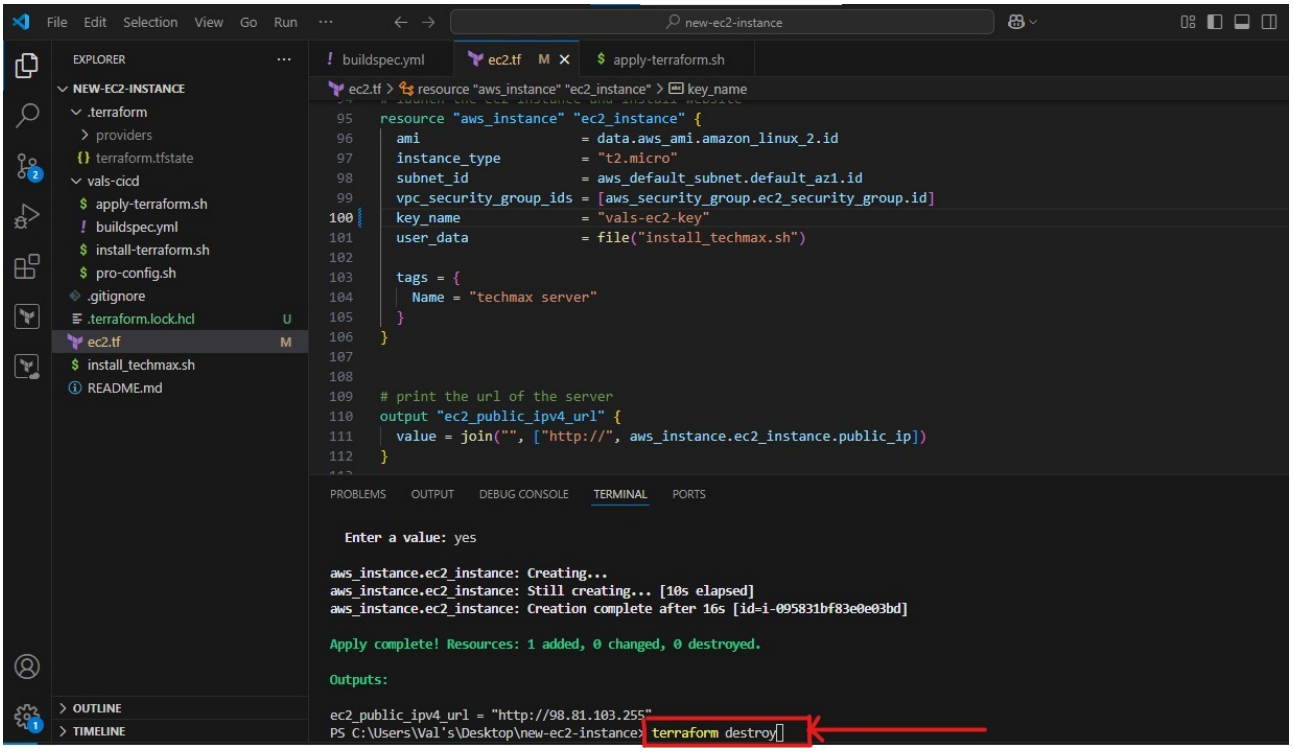

Support EC2 S3 CodeCommit AWS Amplify Lambda API Gateway DynamoDB CloudFront Certificate Manager VPC Aurora and RDS
EC2 > Key pairs
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Successfully created key pair
Key pairs (1) Info
Actions
Create key pair
Find Key Pair by attribute or tag
Name
Type
Created
Fingerprint
vals-ec2-key
rsa
2025/04/08 16:29 GMT+1
6d:6f:8e:85:e7:d7:41:5a:99:f0:c2:b...

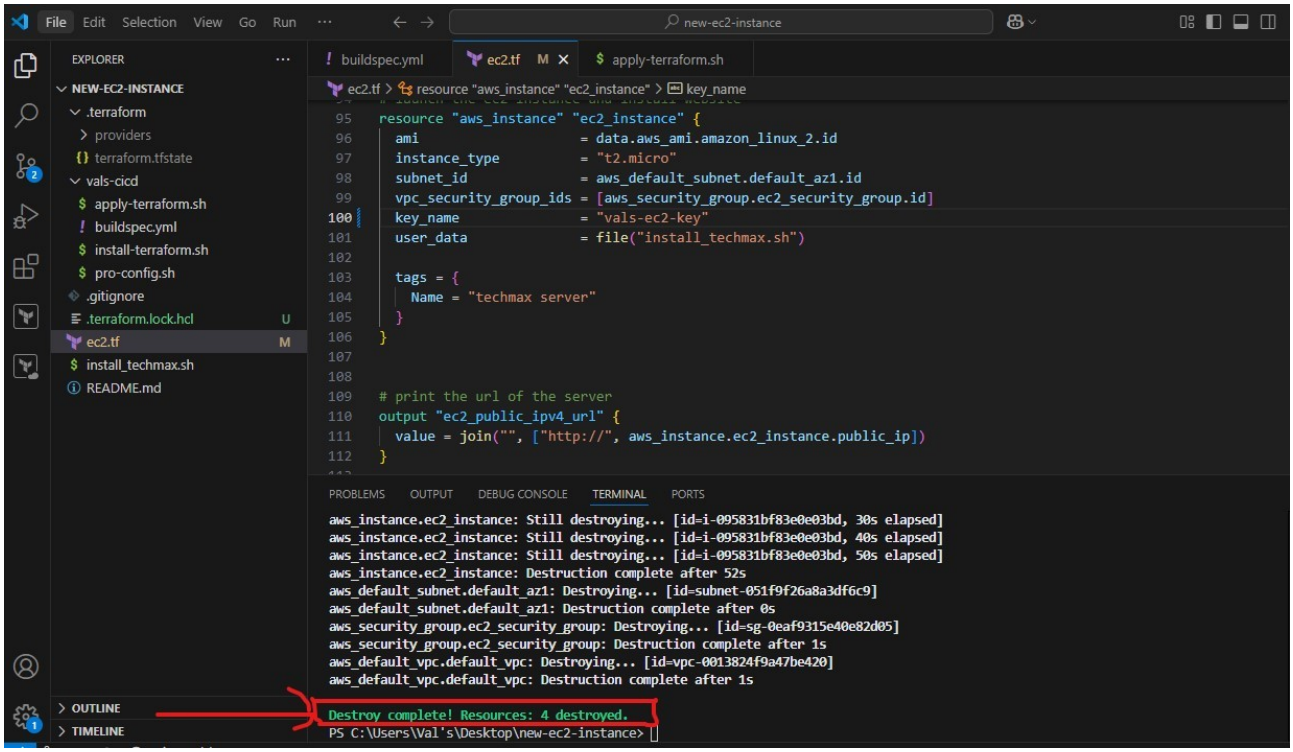

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
apply-terraform.sh
resource "aws_instance" "ec2_instance" {
ami = data.aws_ami.amazon_linux_2.id
instance_type = "t2.micro"
subnet_id = aws_default_subnet.default_az1.id
vpc_security_group_ids = [aws_security_group.ec2_security_group.id]
key_name = "vals-ec2-key"
user_data = file("install_techmax.sh")
tags = {
Name = "techmax server"
}
}
# print the url of the server
output "ec2_public_ipv4_url" {
value = join("", ["http://", aws_instance.ec2_instance.public_ip])
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter a value: yes
aws_instance.ec2_instance: Creating...
aws_instance.ec2_instance: Still creating... [10s elapsed]
aws_instance.ec2_instance: Creation complete after 16s [id=i-095831bf83e0e03bd]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
Outputs:
ec2_public_ipv4_url = "http://98.81.103.255"
PS C:\Users\Val's\Desktop\new-ec2-instance>
OUTLINE
TIMELINE

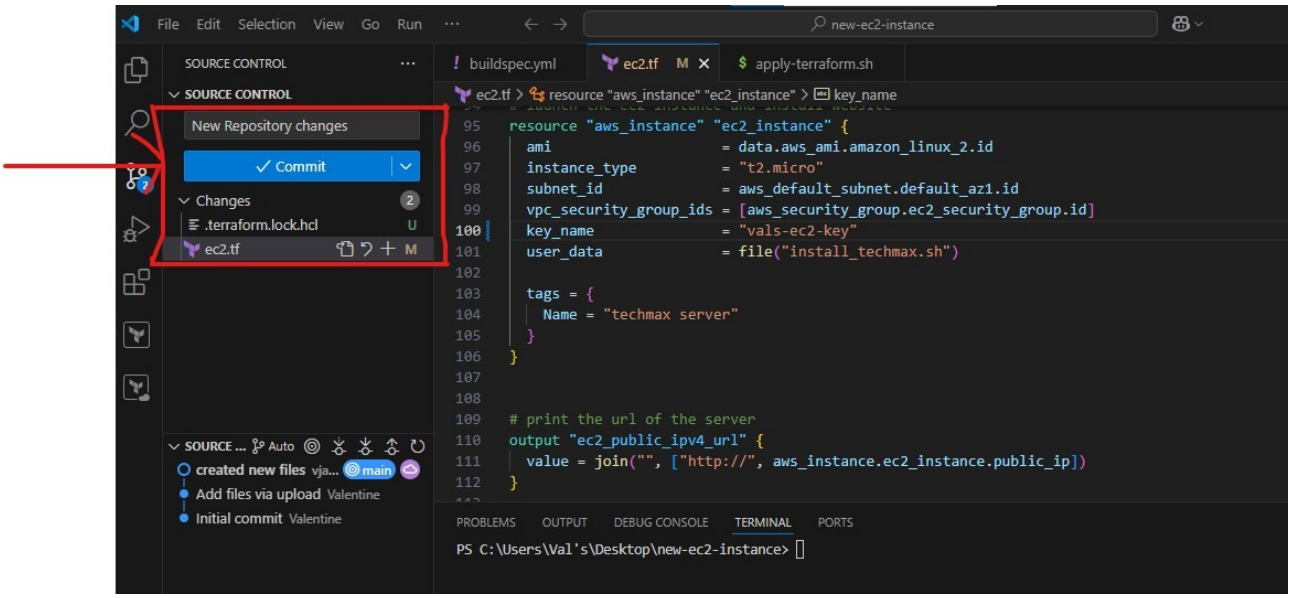

Support EC2 S3 CodeCommit AWS Amplify Lambda API Gateway DynamoDB CloudFront Certificate Manager VPC Aurora and RDS
Amazon S3
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for
Account snapshot - updated every 24 hours
All AWS Regions
View Storage Lens dashboard
Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more
General purpose buckets
Directory buckets
General purpose buckets (1) Info
All AWS Regions
Copy ARN
Empty
Delete
Create bucket
Buckets are containers for data stored in S3.
Find buckets by name
Name
AWS Region
IAM Access Analyzer
Creation date
vals-terraform-state-bucket
US East (N. Virginia) us-east-1
View analyzer for us-east-1
April 8, 2025, 13:31:13 (UTC+01:00)

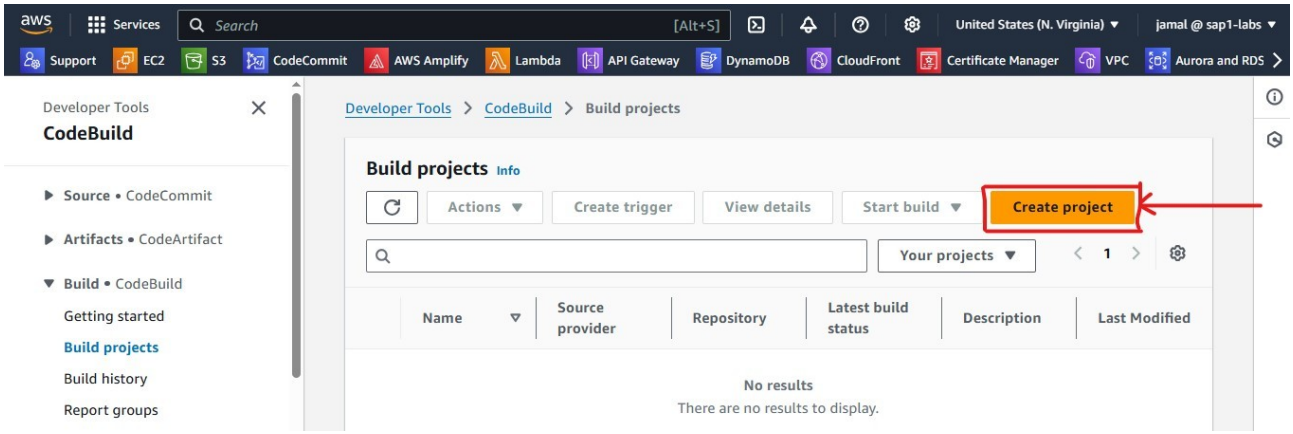

Support EC2 S3 CodeCommit AWS Amplify Lambda API Gateway DynamoDB CloudFront Certificate Manager VPC Aurora and RDS
Amazon S3 > Buckets > vals-terraform-state-bucket
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
vals-terraform-state-bucket 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
Name
Type
Last modified
Size
Storage class
build/
Folder

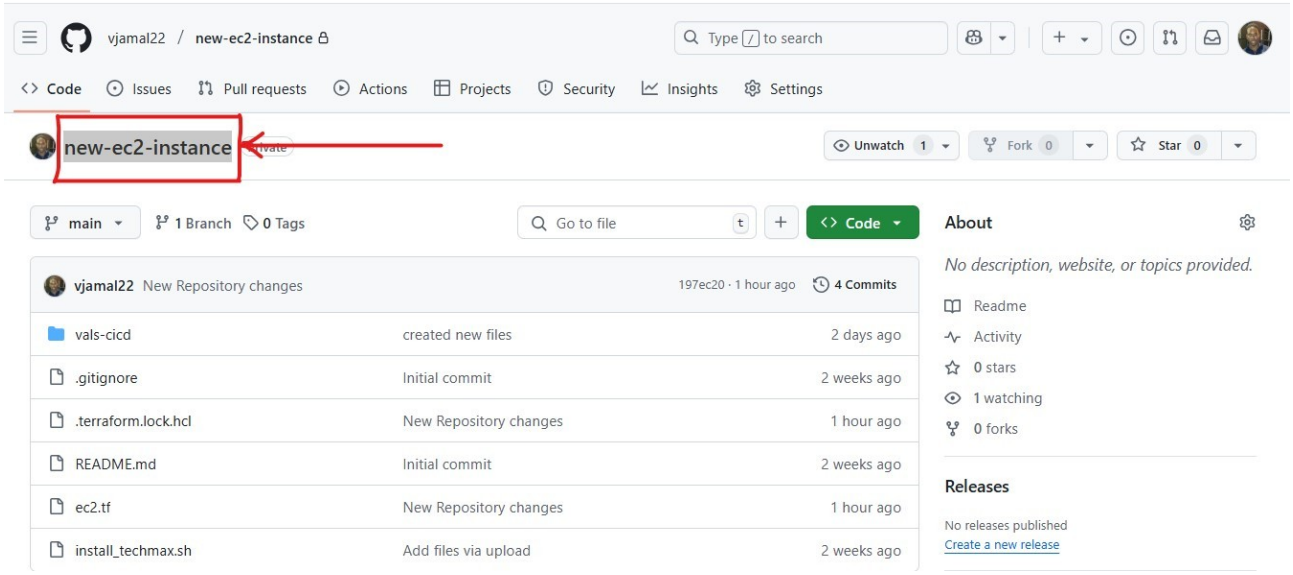

Amazon S3 > Buckets > vals-terraform-state-bucket > build/
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
build/
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
Name
Type
Last modified
Size
Storage class
terraform.tfstate
tfstate
April 8, 2025, 16:42:58 (UTC+01:00)
14.9 KB
Standard

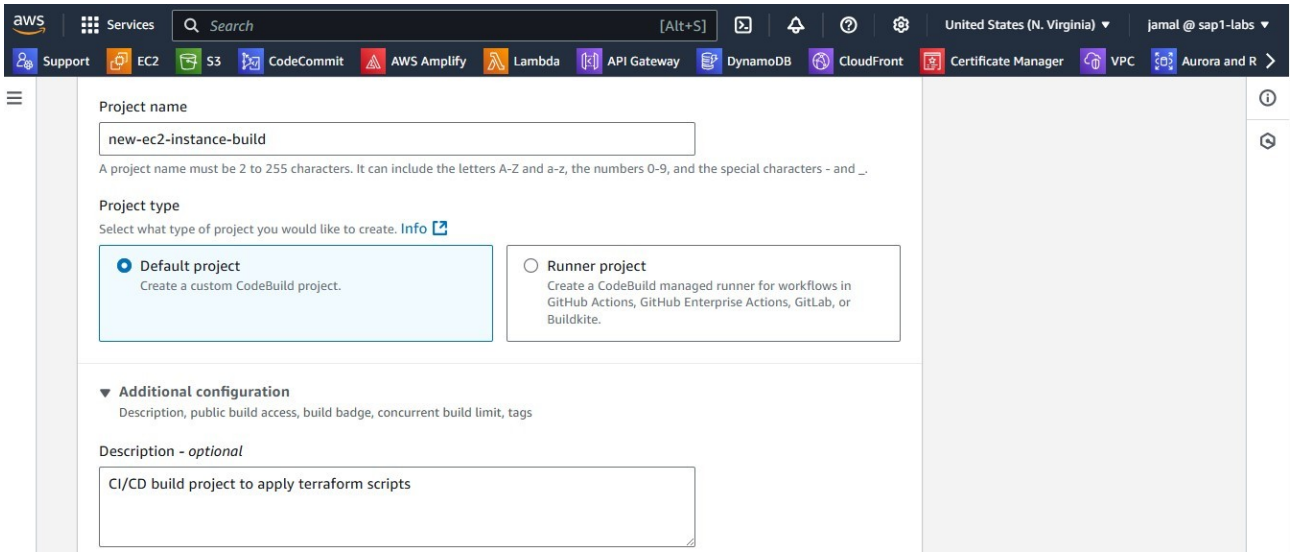

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
OUTLINE
TIMELINE
buildspec.yml
ec2.tf M
apply-terraform.sh
ec2.tf > resource "aws_instance" "ec2_instance" > key_name
95 resource "aws_instance" "ec2_instance" {
96 ami = data.aws_ami.amazon_linux_2.id
97 instance_type = "t2.micro"
98 subnet_id = aws_default_subnet.default_az1.id
99 vpc_security_group_ids = [aws_security_group.ec2_security_group.id]
100 key_name = "vals-ec2-key"
101 user_data = file("install_techmax.sh")
102
103 tags = {
104 Name = "techmax server"
105 }
106 }
107
108
109 # print the url of the server
110 output "ec2_public_ipv4_url" {
111 value = join("", ["http://", aws_instance.ec2_instance.public_ip])
112 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter a value: yes
aws_instance.ec2_instance: Creating...
aws_instance.ec2_instance: Still creating... [10s elapsed]
aws_instance.ec2_instance: Creation complete after 16s [id=i-095831bf83e0e03bd]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
Outputs:
ec2_public_ipv4_url = "http://98.81.103.255"
PS C:\Users\Val's\Desktop\new-ec2-instance> terraform destroy

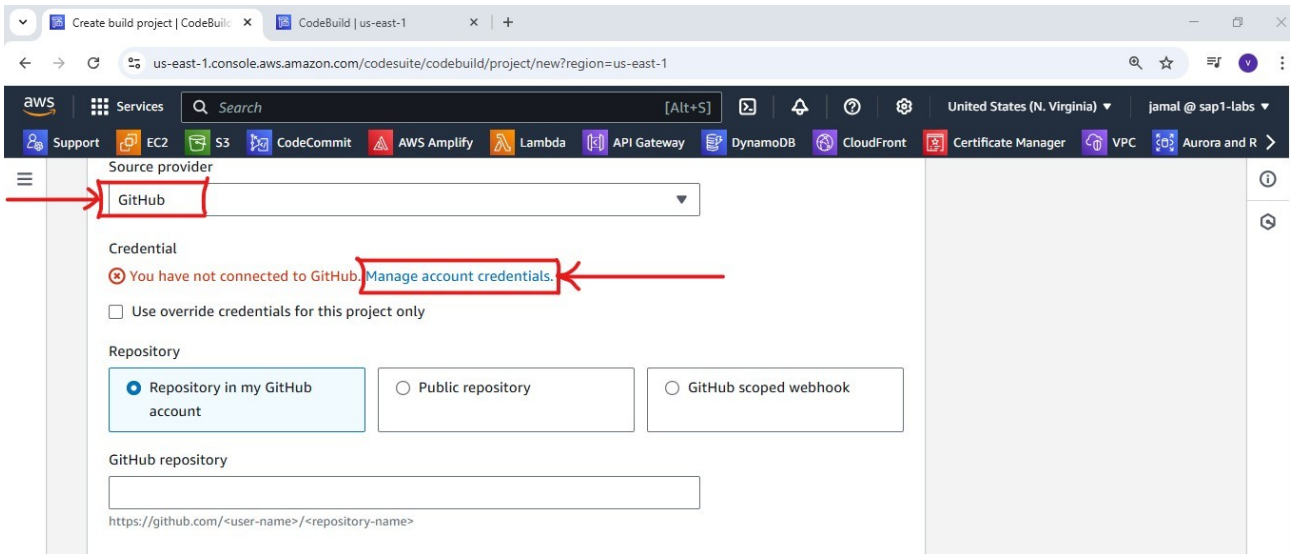

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl U
ec2.tf M
install_techmax.sh
README.md
buildspec.yml
ec2.tf M
apply-terraform.sh
ec2.tf > resource "aws_instance" "ec2_instance" > key_name
95 resource "aws_instance" "ec2_instance" {
96 ami = data.aws_ami.amazon_linux_2.id
97 instance_type = "t2.micro"
98 subnet_id = aws_default_subnet.default_az1.id
99 vpc_security_group_ids = [aws_security_group.ec2_security_group.id]
100 key_name = "vals-ec2-key"
101 user_data = file("install_techmax.sh")
102
103 tags = {
104 Name = "techmax server"
105 }
106 }
107
108
109 # print the url of the server
110 output "ec2_public_ipv4_url" {
111 value = join("", ["http://", aws_instance.ec2_instance.public_ip])
112 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
aws_instance.ec2_instance: Still destroying... [id=i-095831bf83e0e03bd, 30s elapsed]
aws_instance.ec2_instance: Still destroying... [id=i-095831bf83e0e03bd, 40s elapsed]
aws_instance.ec2_instance: Still destroying... [id=i-095831bf83e0e03bd, 50s elapsed]
aws_instance.ec2_instance: Destruction complete after 52s
aws_default_subnet.default_az1: Destroying... [id=subnet-051f9f26a8a3df6c9]
aws_default_subnet.default_az1: Destruction complete after 0s
aws_security_group.ec2_security_group: Destroying... [id=sg-0eaf9315e40e82d05]
aws_security_group.ec2_security_group: Destruction complete after 1s
aws_default_vpc.default_vpc: Destroying... [id=vpc-0013824f9a47be420]
aws_default_vpc.default_vpc: Destruction complete after 1s
Destroy complete! Resources: 4 destroyed.
PS C:\Users\Val's\Desktop\new-ec2-instance>
OUTLINE
TIMELINE

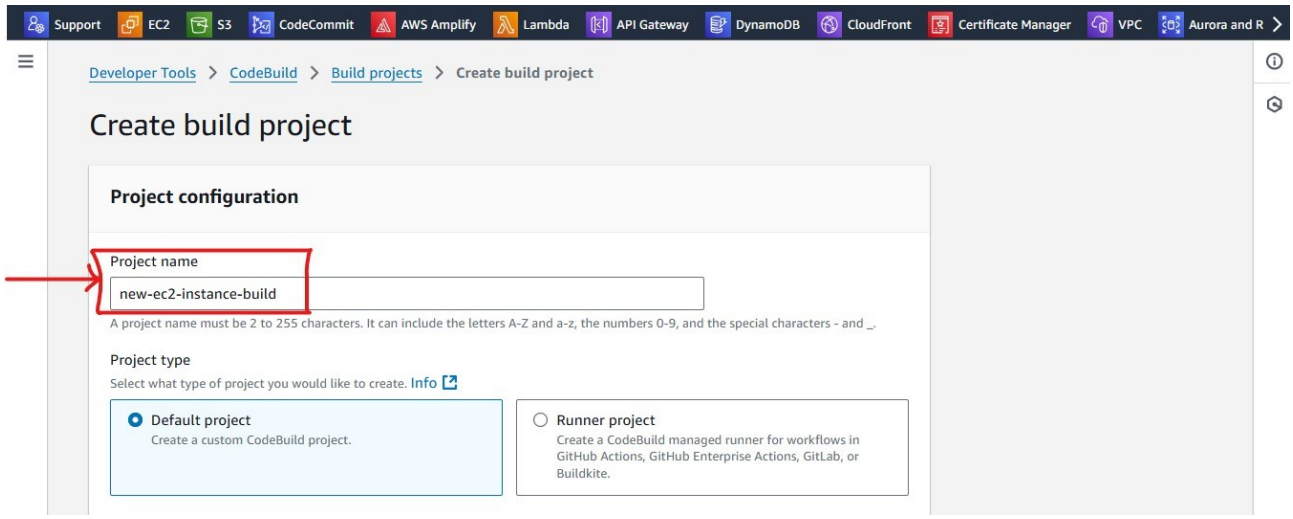

File Edit Selection View Go Run
new-ec2-instance
SOURCE CONTROL
SOURCE CONTROL
New Repository changes
Commit
Changes 2
.terraform.lock.hcl U
ec2.tf M
SOURCE ... Auto
created new files vja... main
Add files via upload Valentine
Initial commit Valentine
buildspec.yml
ec2.tf M
apply-terraform.sh
ec2.tf > resource "aws_instance" "ec2_instance" > key_name
95 resource "aws_instance" "ec2_instance" {
96 ami = data.aws_ami.amazon_linux_2.id
97 instance_type = "t2.micro"
98 subnet_id = aws_default_subnet.default_az1.id
99 vpc_security_group_ids = [aws_security_group.ec2_security_group.id]
100 key_name = "vals-ec2-key"
101 user_data = file("install_techmax.sh")
102
103 tags = {
104 Name = "techmax server"
105 }
106 }
107
108
109 # print the url of the server
110 output "ec2_public_ipv4_url" {
111 value = join("", ["http://", aws_instance.ec2_instance.public_ip])
112 }
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Val's\Desktop\new-ec2-instance>

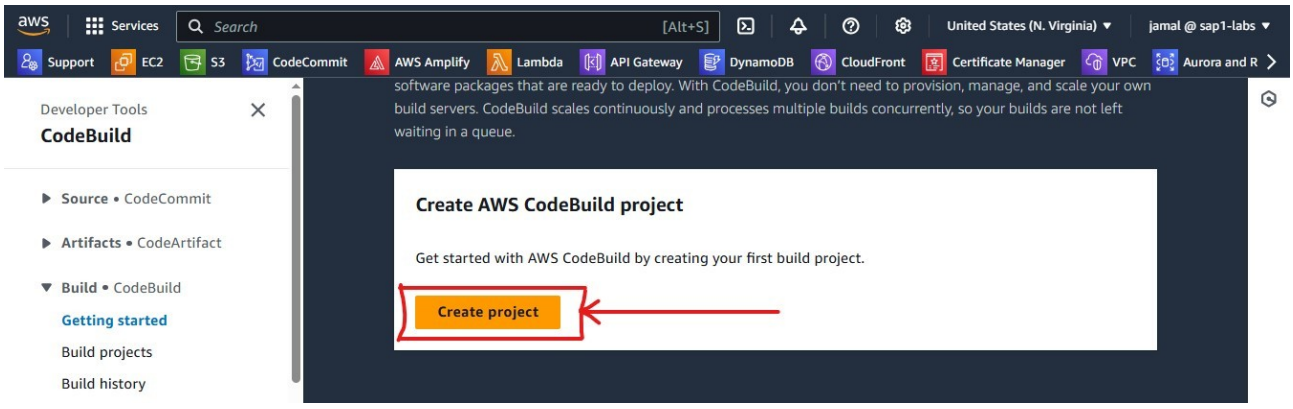

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build history
Report groups
Developer Tools > CodeBuild > Build projects
Build projects Info
Actions
Create trigger
View details
Start build
Create project
Your projects
Name
Source provider
Repository
Latest build status
Description
Last Modified
No results
There are no results to display.

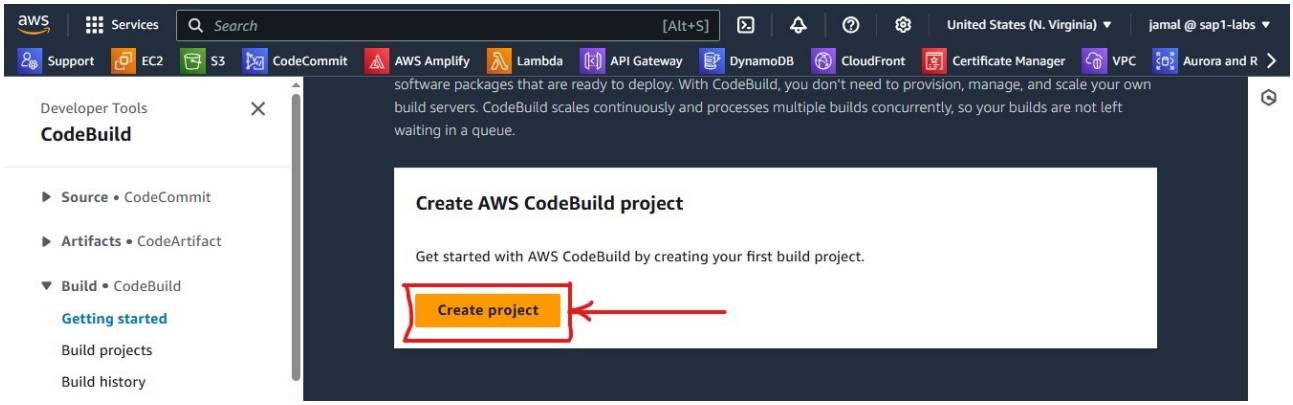

vjamal22 / new-ec2-instance
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
new-ec2-instance
Unwatch 1
Fork 0
Star 0
main
1 Branch
0 Tags
Go to file
Code
vjamal22 New Repository changes
197ec20 · 1 hour ago
4 Commits
vals-cicd created new files 2 days ago
.gitignore Initial commit 2 weeks ago
.terraform.lock.hcl New Repository changes 1 hour ago
README.md Initial commit 2 weeks ago
ec2.tf New Repository changes 1 hour ago
install_techmax.sh Add files via upload 2 weeks ago
About
No description, website, or topics provided.
Readme
Activity
0 stars
1 watching
0 forks
Releases
No releases published
Create a new release

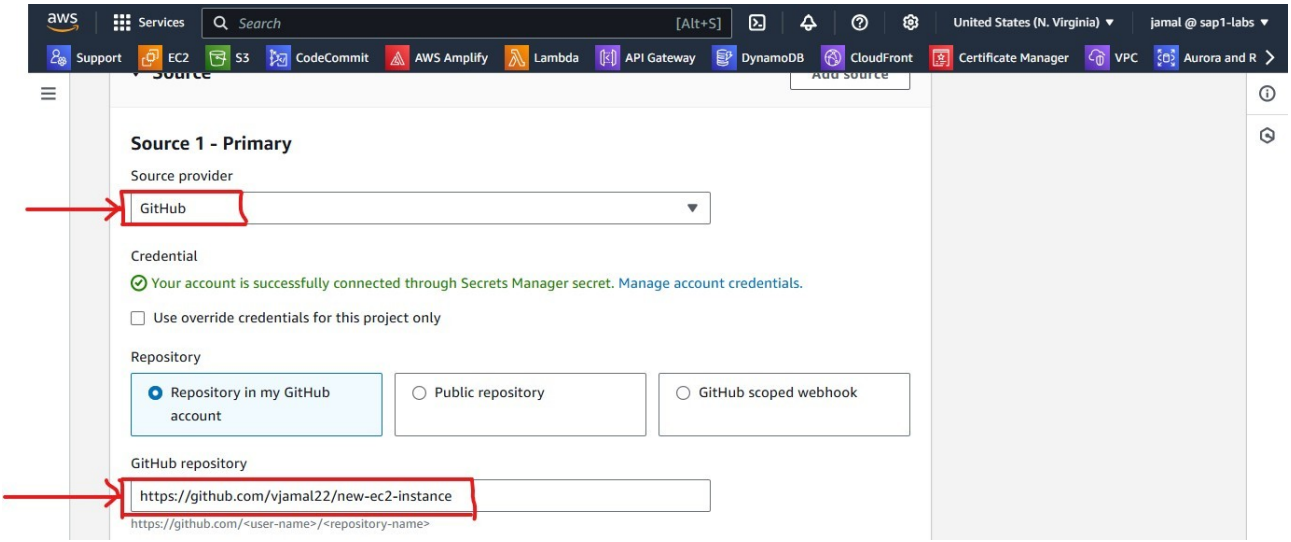

Project name
new-ec2-instance-build
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.
Project type
Select what type of project you would like to create. Info
Default project
Create a custom CodeBuild project.
Runner project
Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.
Additional configuration
Description, public build access, build badge, concurrent build limit, tags
Description - optional
CI/CD build project to apply terraform scripts

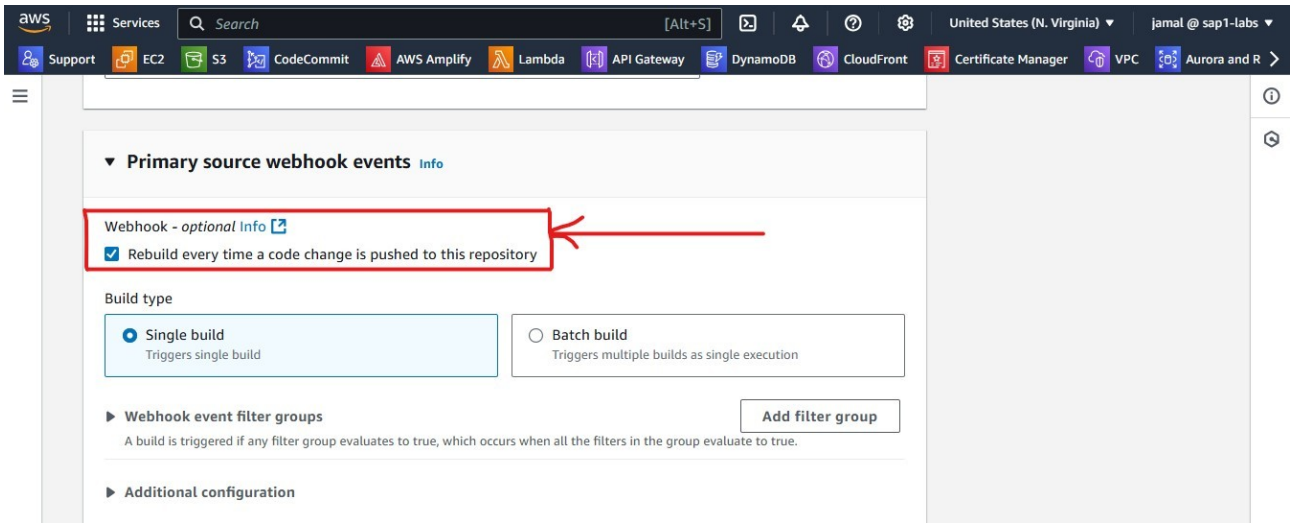

Create build project | CodeBuild
CodeBuild | us-east-1
us-east-1.console.aws.amazon.com/codesuite/codebuild/project/new?region=us-east-1
Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Source provider
GitHub
Credential
You have not connected to GitHub. Manage account credentials.
Use override credentials for this project only
Repository
Repository in my GitHub account
Public repository
GitHub scoped webhook
GitHub repository
https://github.com/<user-name>/<repository-name>

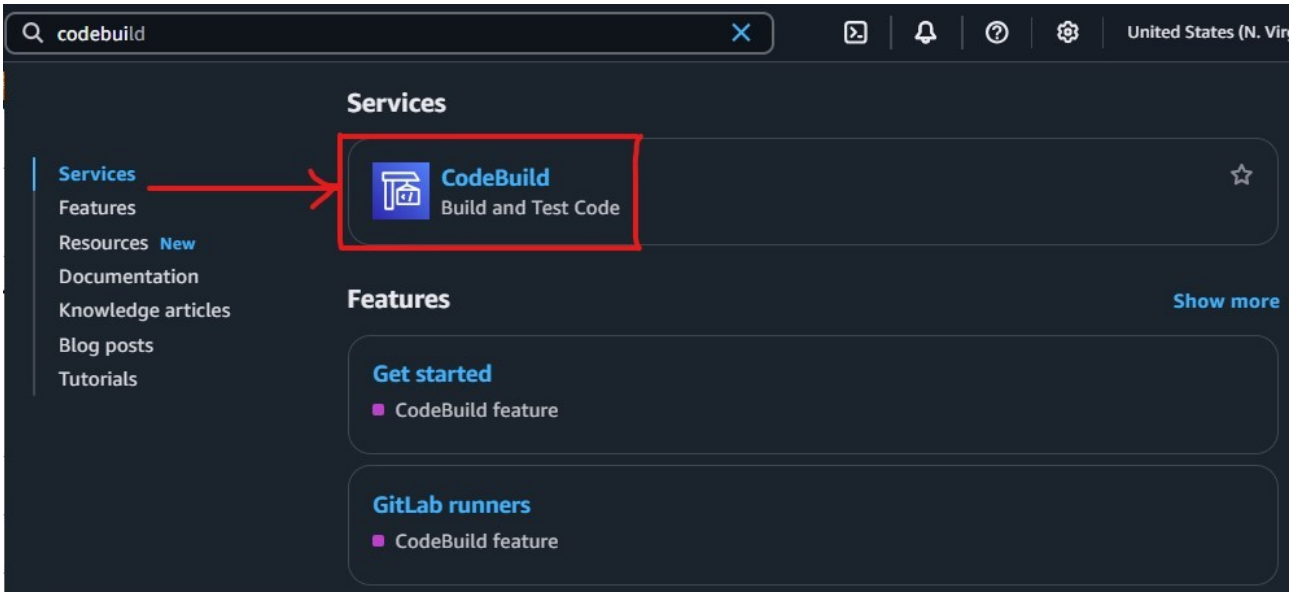

Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Developer Tools
CodeBuild
Build projects
Create build project
Create build project
Project configuration
Project name
new-ec2-instance-build
A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.
Project type
Select what type of project you would like to create. Info
Default project
Create a custom CodeBuild project.
Runner project
Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.

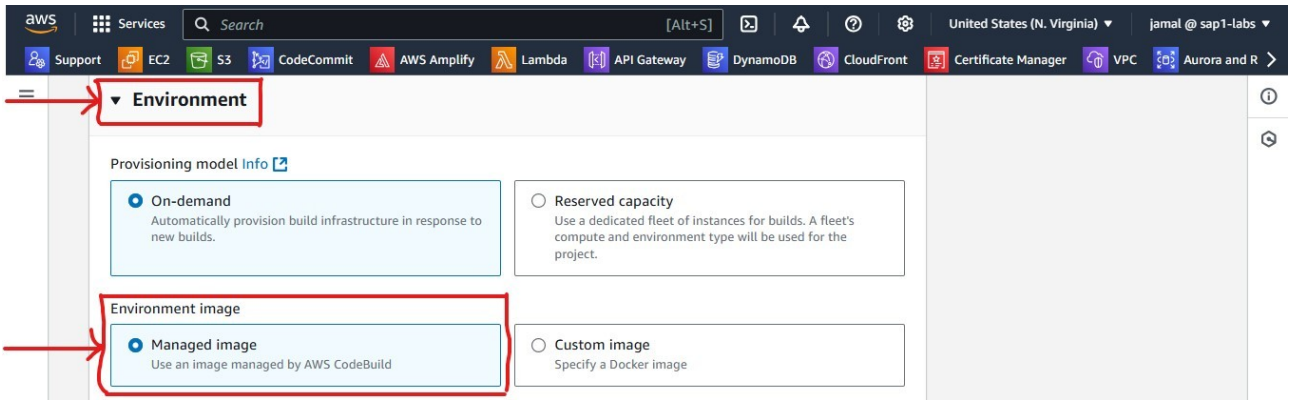

Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build history
software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue.
Create AWS CodeBuild project
Get started with AWS CodeBuild by creating your first build project.
Create project

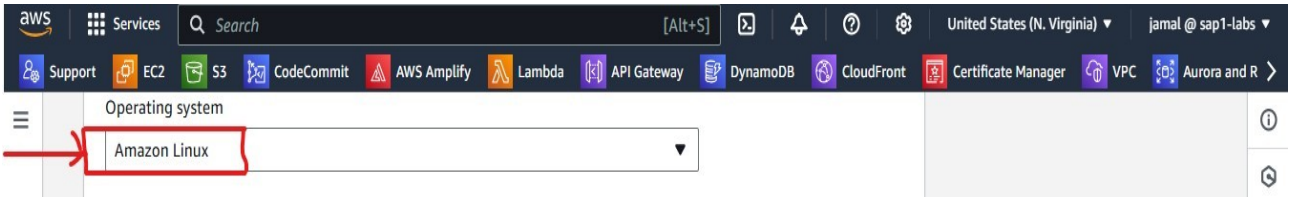

Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build history
software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers. CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue.
Create AWS CodeBuild project
Get started with AWS CodeBuild by creating your first build project.
Create project

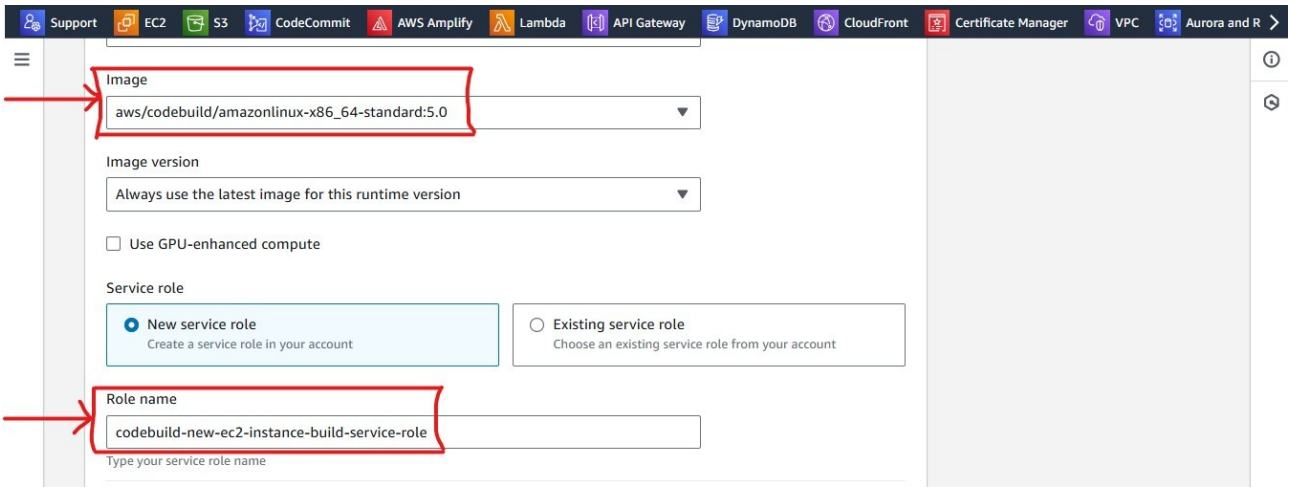

Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Source 1 - Primary
Source provider
GitHub
Credential
Your account is successfully connected through Secrets Manager secret. Manage account credentials.
Use override credentials for this project only
Repository
Repository in my GitHub account
Public repository
GitHub scoped webhook
GitHub repository
https://github.com/vjamal22/new-ec2-instance
https://github.com/<user-name>/<repository-name>

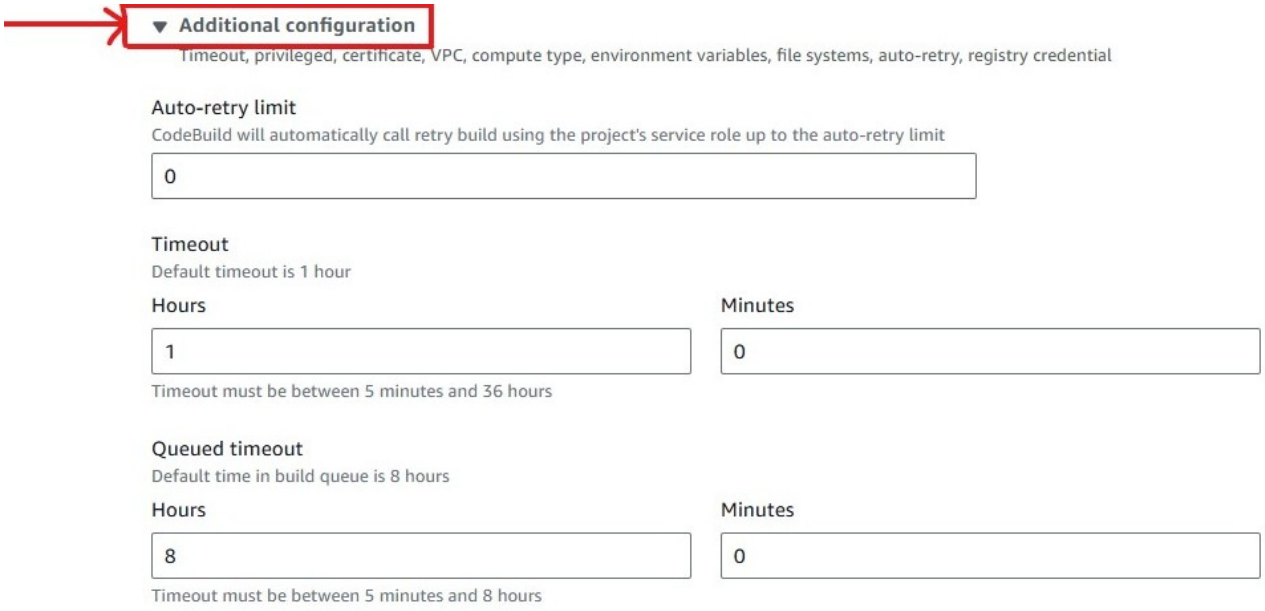

Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Primary source webhook events
Info
Webhook - optional Info
Rebuild every time a code change is pushed to this repository
Build type
Single build
Triggers single build
Batch build
Triggers multiple builds as single execution
Webhook event filter groups
Add filter group
A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.
Additional configuration

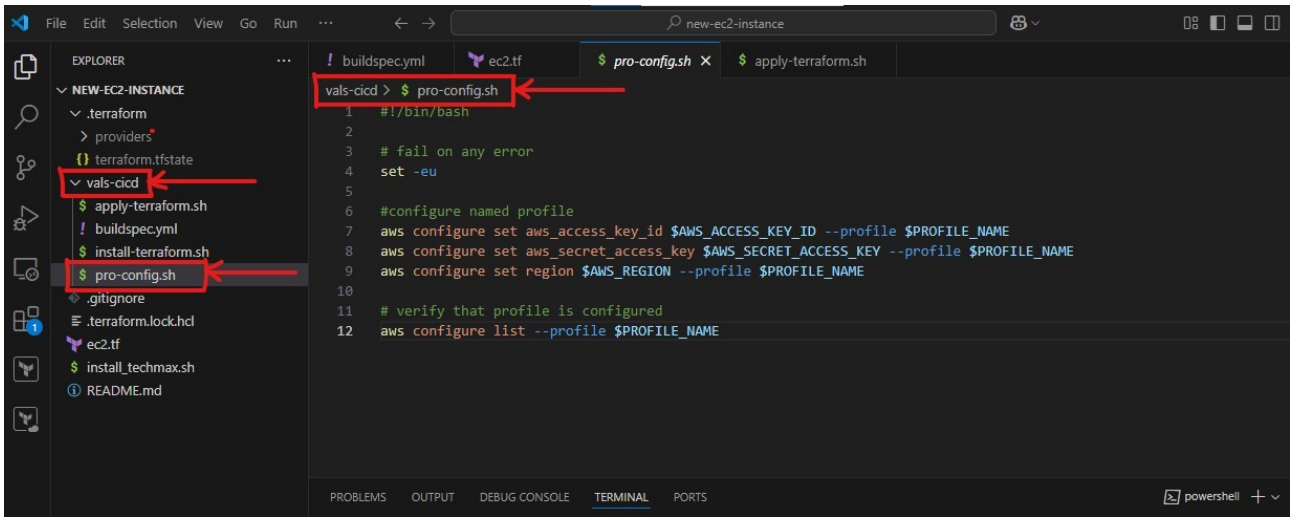

codebuild
United States (N. Vir
Services
Services
Features
Resources New
Documentation
Knowledge articles
Blog posts
Tutorials
CodeBuild
Build and Test Code
Features
Show more
Get started
CodeBuild feature
GitLab runners
CodeBuild feature

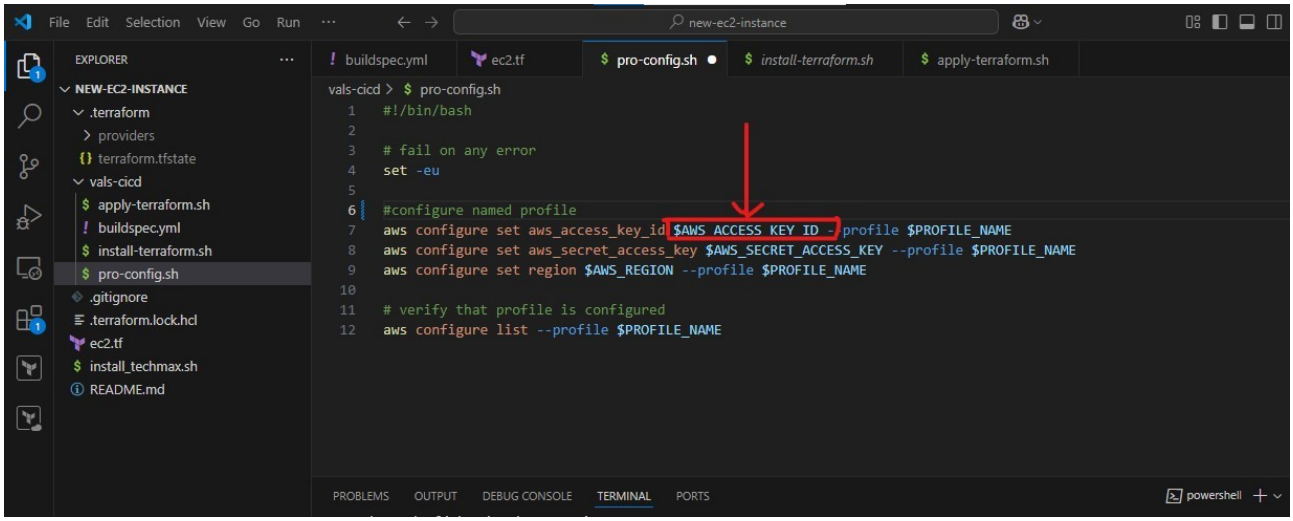

aws
Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Environment
Provisioning model Info
On-demand
Automatically provision build infrastructure in response to new builds.
Reserved capacity
Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.
Environment image
Managed image
Use an image managed by AWS CodeBuild
Custom image
Specify a Docker image

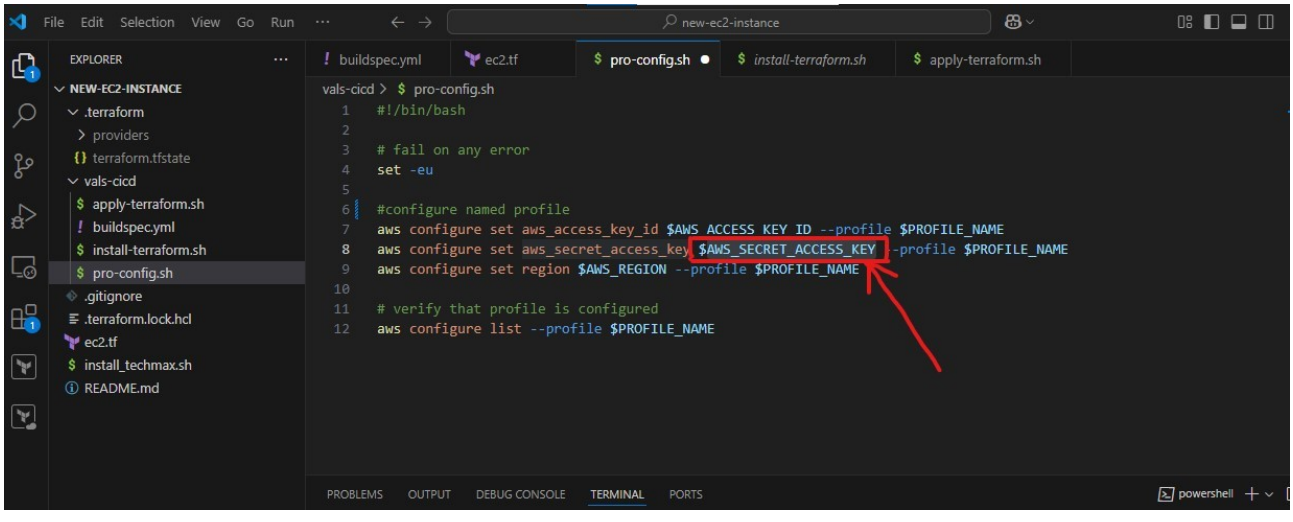

aws
Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Operating system
Amazon Linux

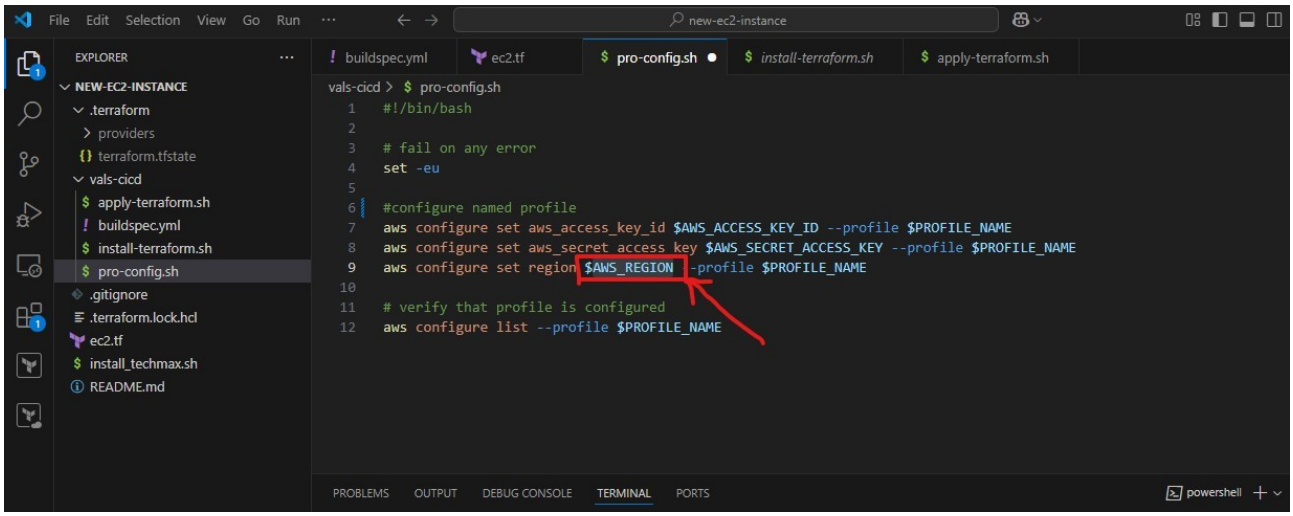

Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Image
aws/codebuild/amazonlinux-x86_64-standard:5.0
Image version
Always use the latest image for this runtime version
Use GPU-enhanced compute
Service role
New service role
Create a service role in your account
Existing service role
Choose an existing service role from your account
Role name
codebuild-new-ec2-instance-build-service-role
Type your service role name

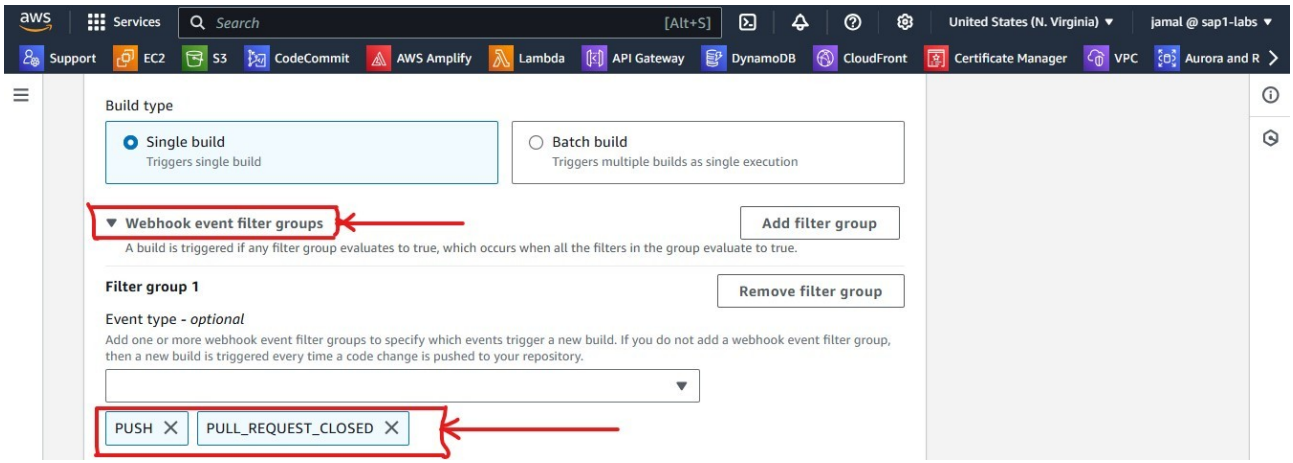

Additional configuration
Timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry, registry credential
Auto-retry limit
CodeBuild will automatically call retry build using the project's service role up to the auto-retry limit
0
Timeout
Default timeout is 1 hour
Hours
1
Minutes
0
Timeout must be between 5 minutes and 36 hours
Queued timeout
Default time in build queue is 8 hours
Hours
8
Minutes
0
Timeout must be between 5 minutes and 8 hours

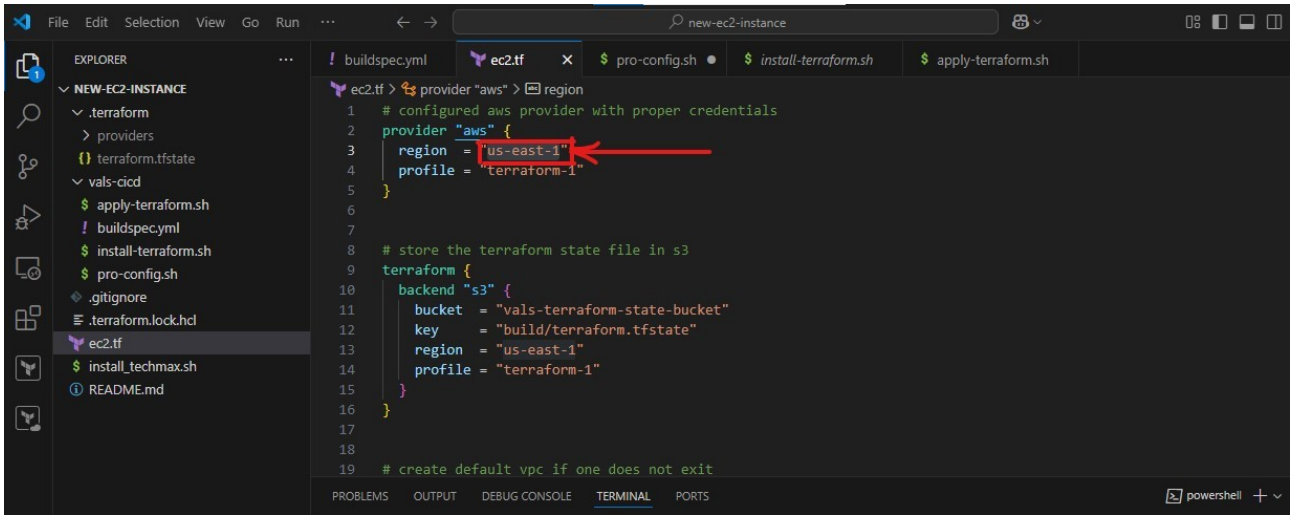

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
pro-config.sh
apply-terraform.sh
vals-cicd > pro-config.sh
#!/bin/bash
fail on any error
set -eu
#configure named profile
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profile $PROFILE_NAME
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --profile $PROFILE_NAME
aws configure set region $AWS_REGION --profile $PROFILE_NAME
verify that profile is configured
aws configure list --profile $PROFILE_NAME
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

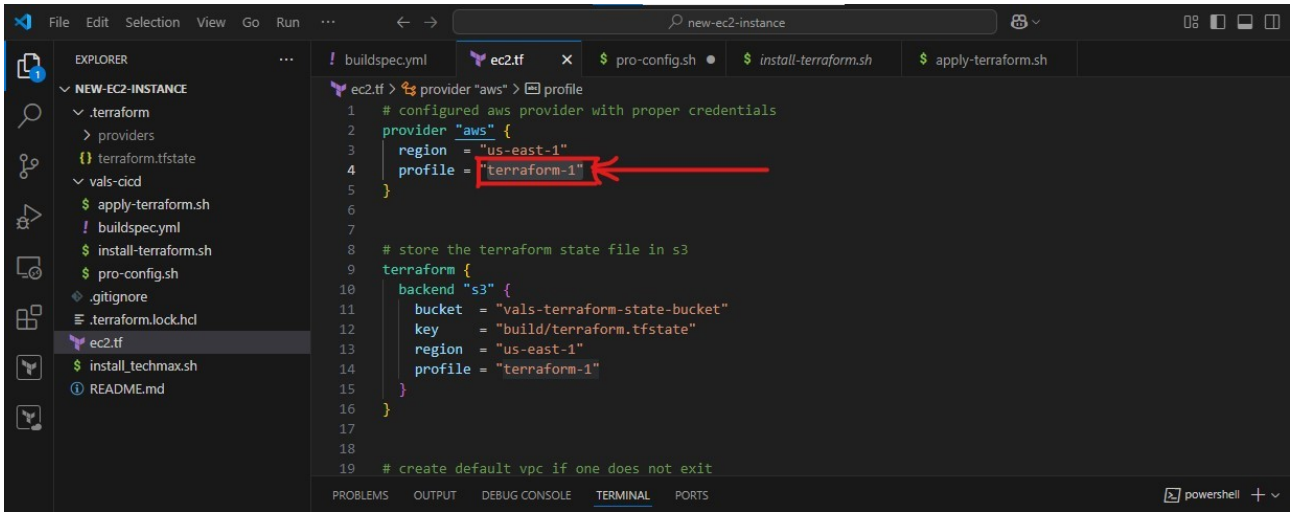

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
pro-config.sh
install-terraform.sh
apply-terraform.sh
vals-cicd > pro-config.sh
#!/bin/bash
fail on any error
set -eu
#configure named profile
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profile $PROFILE_NAME
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --profile $PROFILE_NAME
aws configure set region $AWS_REGION --profile $PROFILE_NAME
verify that profile is configured
aws configure list --profile $PROFILE_NAME
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

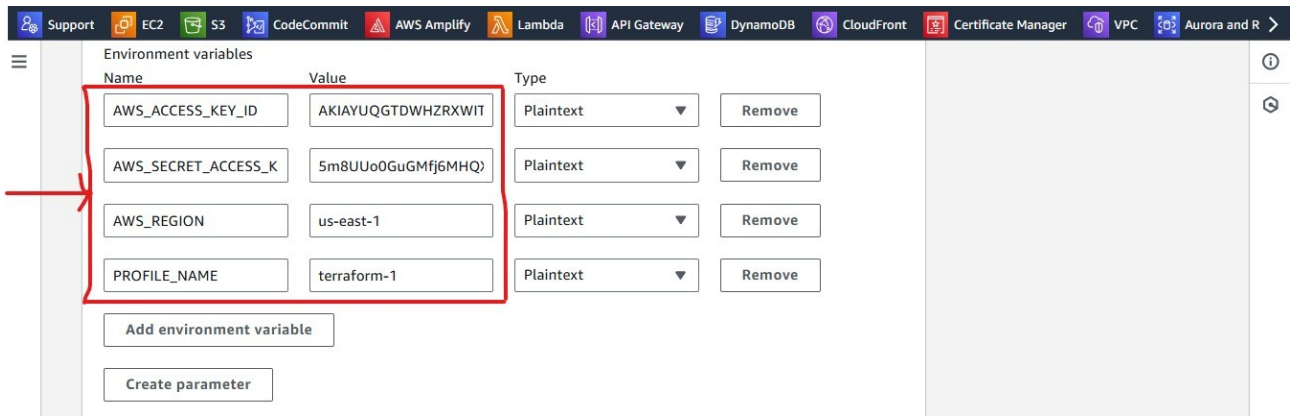

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
pro-config.sh
install-terraform.sh
apply-terraform.sh
vals-cicd > pro-config.sh
#!/bin/bash
fail on any error
set -eu
#configure named profile
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profile $PROFILE_NAME
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --profile $PROFILE_NAME
aws configure set region $AWS_REGION --profile $PROFILE_NAME
verify that profile is configured
aws configure list --profile $PROFILE_NAME
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

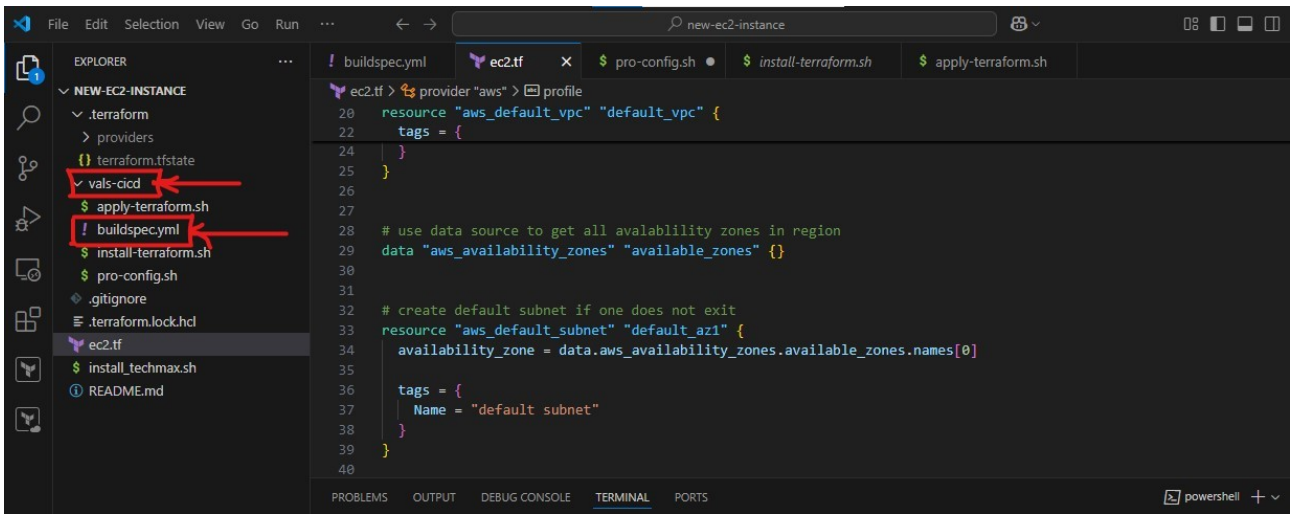

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
pro-config.sh
install-terraform.sh
apply-terraform.sh
vals-cicd > pro-config.sh
#!/bin/bash
fail on any error
set -eu
#configure named profile
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profile $PROFILE_NAME
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --profile $PROFILE_NAME
aws configure set region $AWS_REGION -profile $PROFILE_NAME
verify that profile is configured
aws configure list --profile $PROFILE_NAME
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

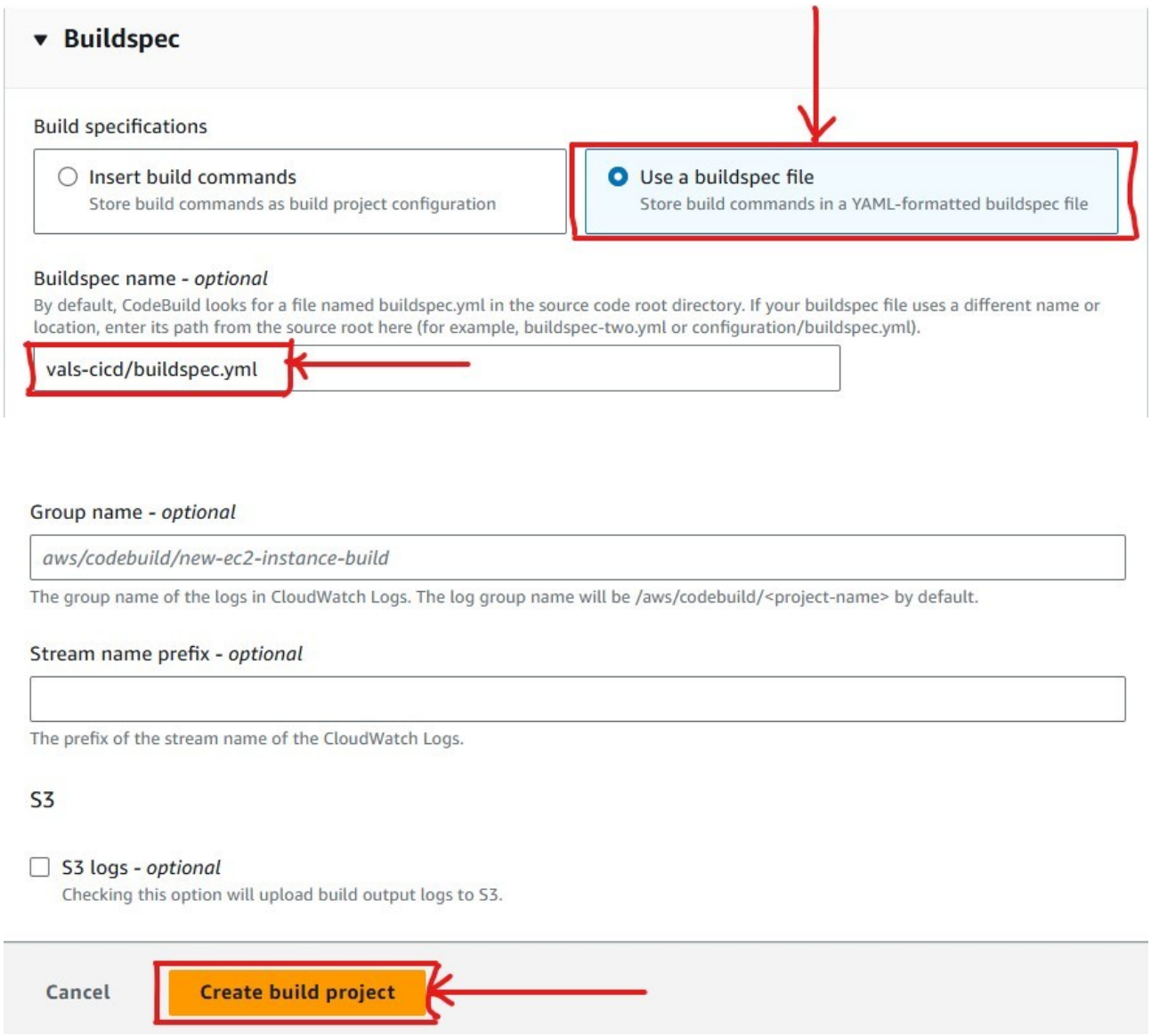

aws
Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and R
Build type
Single build
Triggers single build
Batch build
Triggers multiple builds as single execution
Webhook event filter groups
Add filter group
A build is triggered if any filter group evaluates to true, which occurs when all the filters in the group evaluate to true.
Filter group 1
Remove filter group
Event type - optional
Add one or more webhook event filter groups to specify which events trigger a new build. If you do not add a webhook event filter group, then a new build is triggered every time a code change is pushed to your repository.
PUSH
PULL_REQUEST_CLOSED

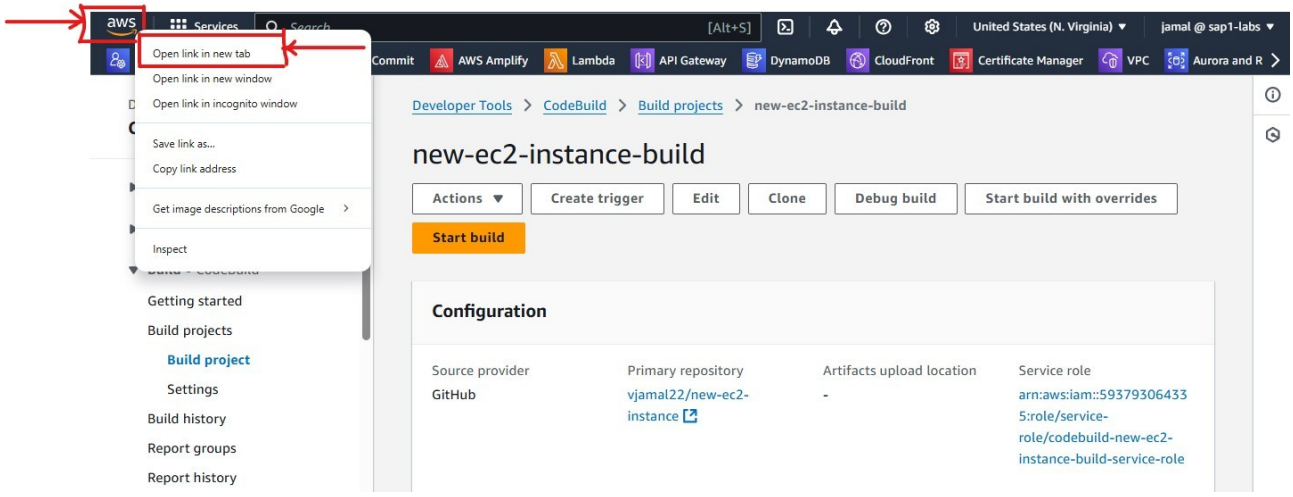

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
pro-config.sh
install-terraform.sh
apply-terraform.sh
ec2.tf > provider "aws" > region
# configured aws provider with proper credentials
provider "aws" {
region = "us-east-1"
profile = "terraform-1"
}
# store the terraform state file in s3
terraform {
backend "s3" {
bucket = "vals-terraform-state-bucket"
key = "build/terraform.tfstate"
region = "us-east-1"
profile = "terraform-1"
}
}
# create default vpc if one does not exit
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

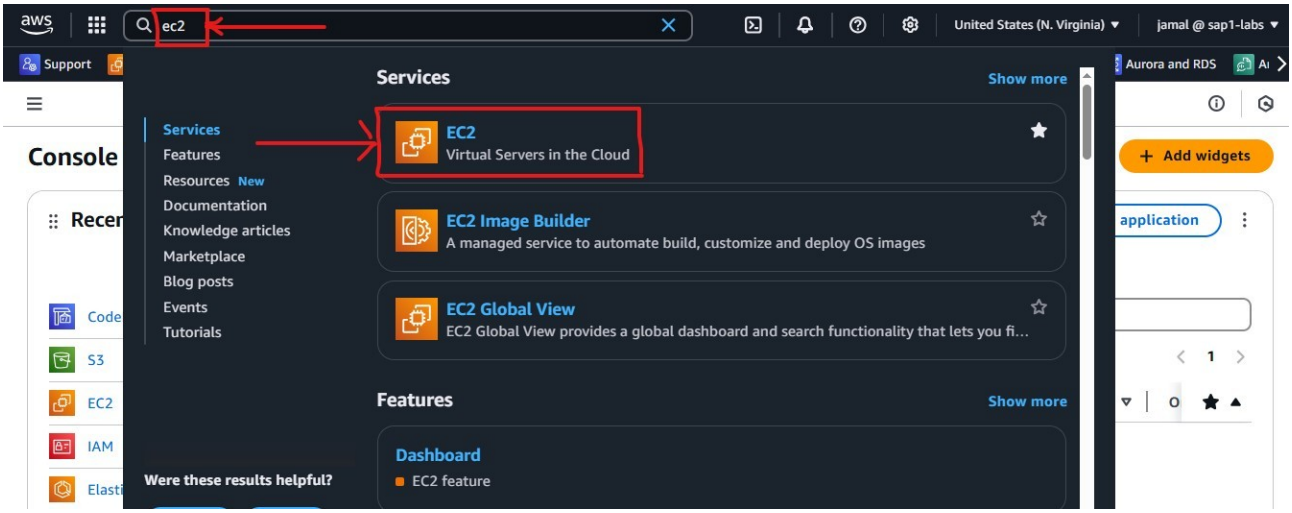

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
pro-config.sh
install-terraform.sh
apply-terraform.sh
ec2.tf > provider "aws" > profile
# configured aws provider with proper credentials
provider "aws" {
region = "us-east-1"
profile = "terraform-1"
}
# store the terraform state file in s3
terraform {
backend "s3" {
bucket = "vals-terraform-state-bucket"
key = "build/terraform.tfstate"
region = "us-east-1"
profile = "terraform-1"
}
}
# create default vpc if one does not exit
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

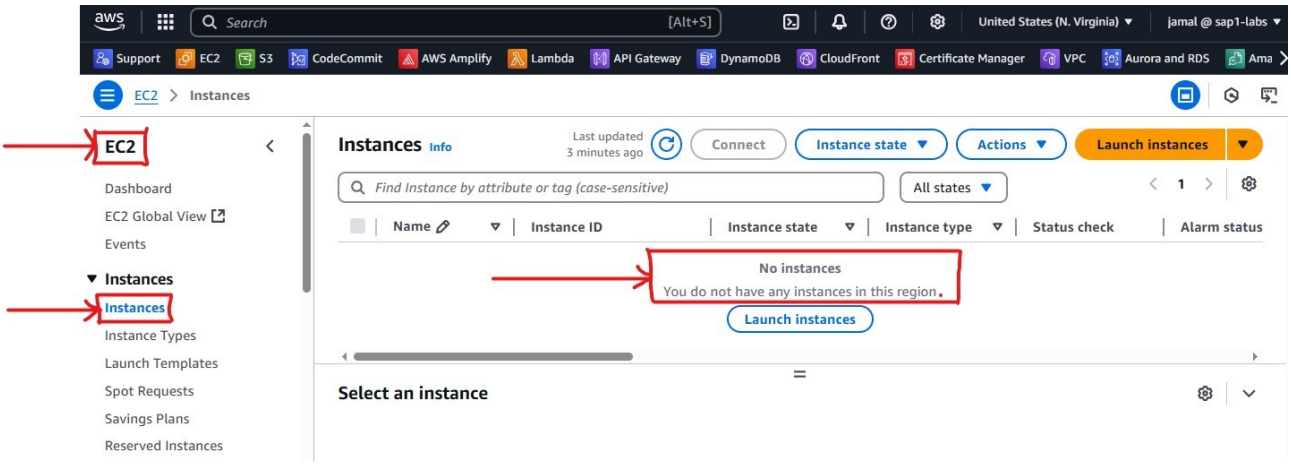

Support EC2 S3 CodeCommit AWS Amplify Lambda API Gateway DynamoDB CloudFront Certificate Manager VPC Aurora and R
Environment variables
Name
Value
Type
AWS_ACCESS_KEY_ID
AKIAYUQGTDWHZRXWIT
Plaintext
Remove
AWS_SECRET_ACCESS_K
5m8UUo0GuGMfj6MHQ
Plaintext
Remove
AWS_REGION
us-east-1
Plaintext
Remove
PROFILE_NAME
terraform-1
Plaintext
Remove
Add environment variable
Create parameter

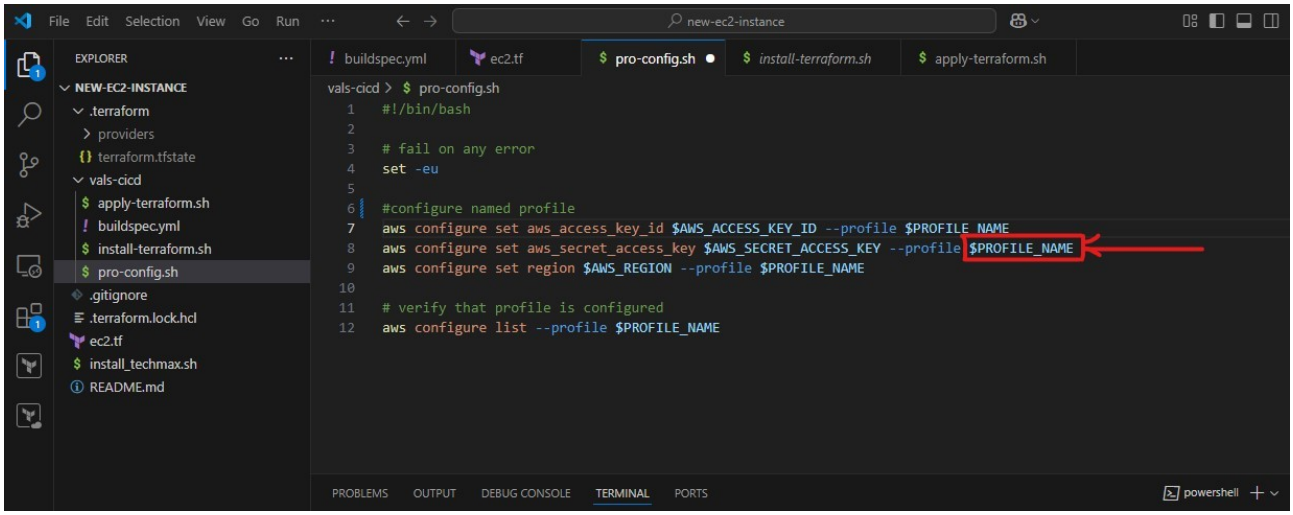

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
pro-config.sh
install-terraform.sh
apply-terraform.sh
ec2.tf > provider "aws" > profile
resource "aws_default_vpc" "default_vpc" {
tags = {
}
}
# use data source to get all avalablility zones in region
data "aws_availability_zones" "available_zones" {}
# create default subnet if one does not exit
resource "aws_default_subnet" "default_az1" {
availability_zone = data.aws_availability_zones.available_zones.names[0]
tags = {
Name = "default subnet"
}
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

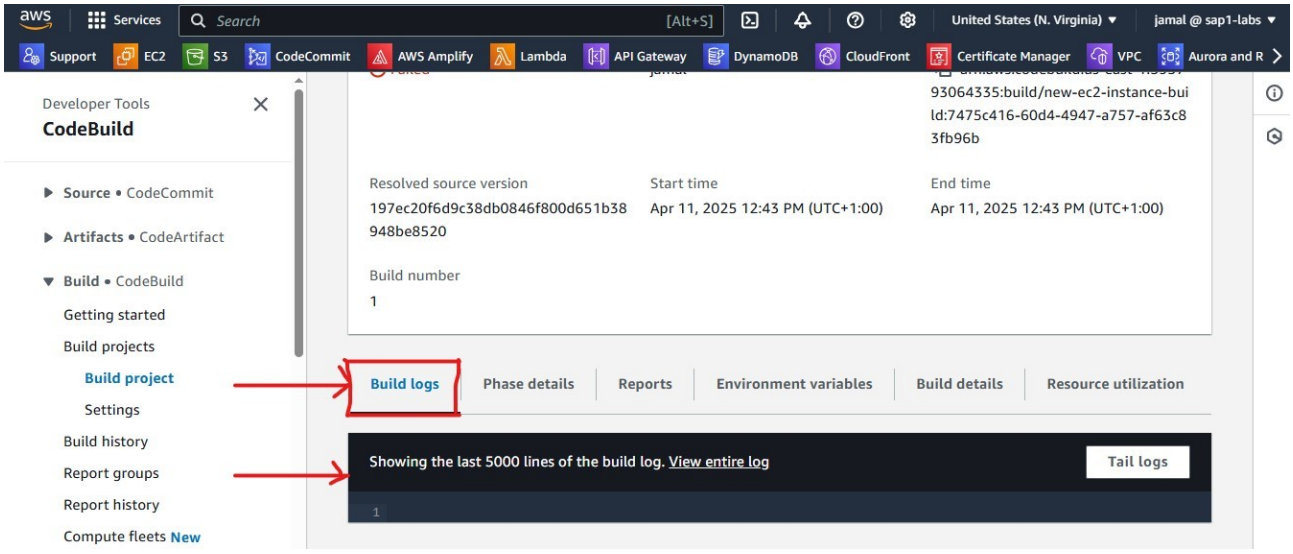

Buildspec
Build specifications
Insert build commands
Store build commands as build project configuration
Use a buildspec file
Store build commands in a YAML-formatted buildspec file
Buildspec name - optional
By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).
vals-cicd/buildspec.yml
Group name - optional
aws/codebuild/new-ec2-instance-build
The group name of the logs in CloudWatch Logs. The log group name will be /aws/codebuild/<project-name> by default.
Stream name prefix - optional
The prefix of the stream name of the CloudWatch Logs.
S3
S3 logs - optional
Checking this option will upload build output logs to S3.
Cancel
Create build project

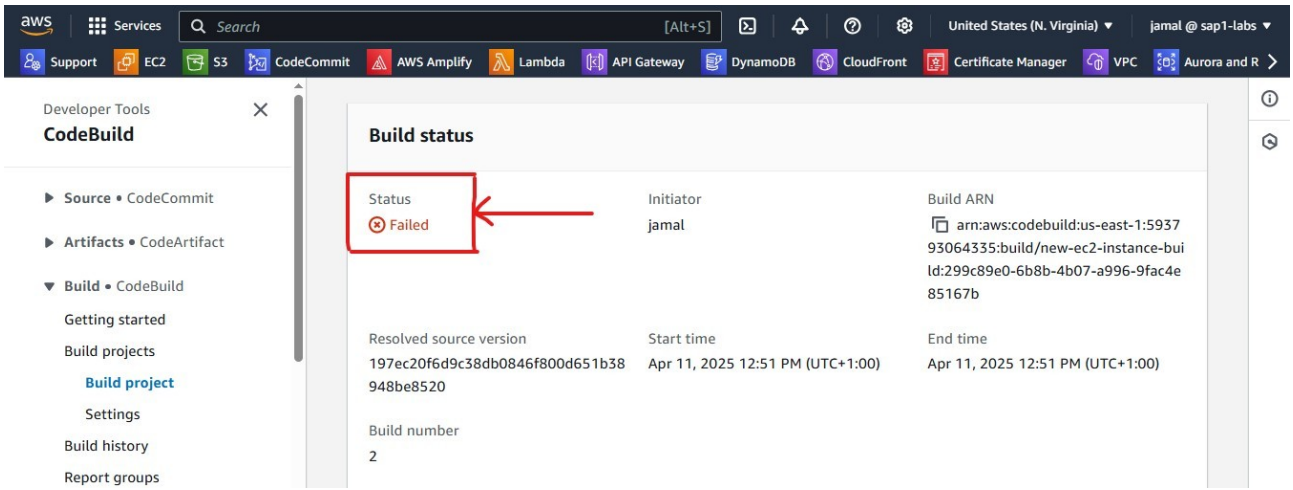

Open link in new tab
Open link in new window
Open link in incognito window
Save link as...
Copy link address
Get image descriptions from Google
Inspect
Developer Tools > CodeBuild > Build projects > new-ec2-instance-build
new-ec2-instance-build
Actions
Create trigger
Edit
Clone
Debug build
Start build with overrides
Start build
Configuration
Source provider
GitHub
Primary repository
vjamal22/new-ec2-instance
Artifacts upload location
-
Service role
arn:aws:iam::593793064335:role/service-role/codebuild-new-ec2-instance-build-service-role
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history

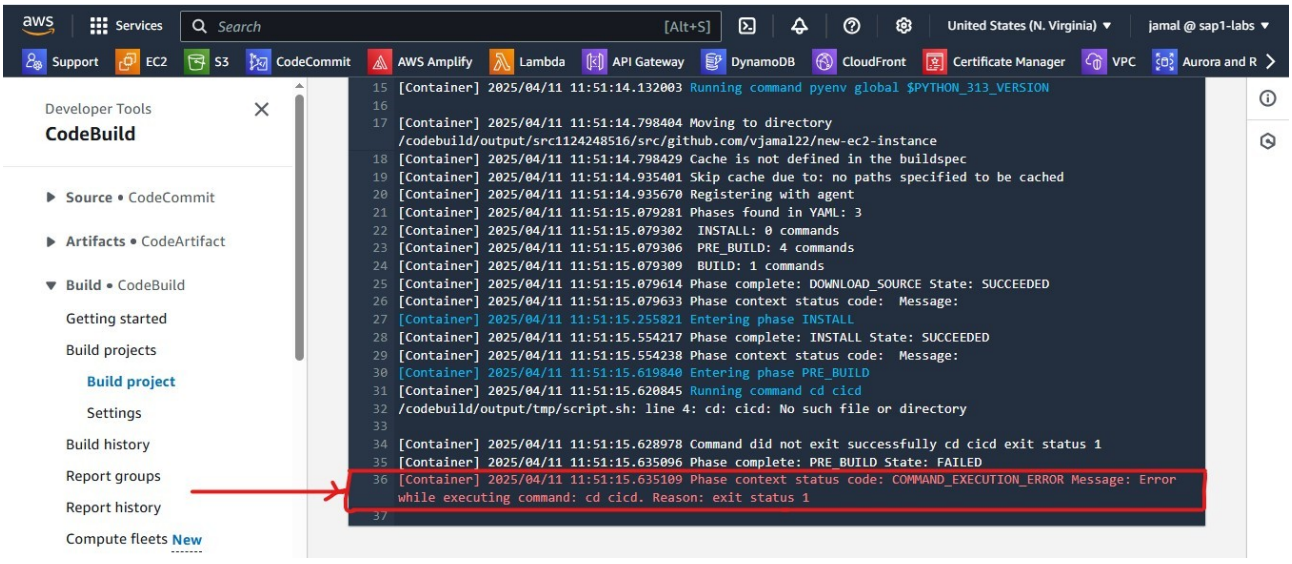

ec2
United States (N. Virginia)
jamal @ sap1-labs
Services
Show more
Services
Features
Resources New
Documentation
Knowledge articles
Marketplace
Blog posts
Events
Tutorials
EC2
Virtual Servers in the Cloud
EC2 Image Builder
A managed service to automate build, customize and deploy OS images
EC2 Global View
EC2 Global View provides a global dashboard and search functionality that lets you fi...
Features
Show more
Dashboard
EC2 feature
Were these results helpful?
Console
Add widgets
application
S3
EC2
IAM


Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Instances Info
Last updated 3 minutes ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
No instances
You do not have any instances in this region.
Launch instances
Select an instance


new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
ec2.tf
pro-config.sh
install-terraform.sh
apply-terraform.sh
vals-cicd > pro-config.sh
#!/bin/bash
fail on any error
set -eu
#configure named profile
aws configure set aws_access_key_id $AWS_ACCESS_KEY_ID --profile $PROFILE_NAME
aws configure set aws_secret_access_key $AWS_SECRET_ACCESS_KEY --profile $PROFILE_NAME
aws configure set region $AWS_REGION --profile $PROFILE_NAME
verify that profile is configured
aws configure list --profile $PROFILE_NAME
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
powershell


Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
93064335:build/new-ec2-instance-bui
ld:7475c416-60d4-4947-a757-af63c8
3fb96b
Resolved source version
197ec20f6d9c38db0846f800d651b38
948be8520
Start time
Apr 11, 2025 12:43 PM (UTC+1:00)
End time
Apr 11, 2025 12:43 PM (UTC+1:00)
Build number
1
Build logs
Phase details
Reports
Environment variables
Build details
Resource utilization
Showing the last 5000 lines of the build log. View entire log
Tail logs


Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Build status
Status
Failed
Initiator
jamal
Build ARN
arn:aws:codebuild:us-east-1:5937
93064335:build/new-ec2-instance-bui
ld:299c89e0-6b8b-4b07-a996-9fac4e
85167b
Resolved source version
197ec20f6d9c38db0846f800d651b38
948be8520
Start time
Apr 11, 2025 12:51 PM (UTC+1:00)
End time
Apr 11, 2025 12:51 PM (UTC+1:00)
Build number
2


Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
[Container] 2025/04/11 11:51:14.132003 Running command pyenv global $PYTHON_313_VERSION
[Container] 2025/04/11 11:51:14.798404 Moving to directory
/codebuild/output/src1124248516/src/github.com/vjamal22/new-ec2-instance
[Container] 2025/04/11 11:51:14.798429 Cache is not defined in the buildspec
[Container] 2025/04/11 11:51:14.935401 Skip cache due to: no paths specified to be cached
[Container] 2025/04/11 11:51:14.935670 Registering with agent
[Container] 2025/04/11 11:51:15.079281 Phases found in YAML: 3
[Container] 2025/04/11 11:51:15.079302 INSTALL: 0 commands
[Container] 2025/04/11 11:51:15.079306 PRE_BUILD: 4 commands
[Container] 2025/04/11 11:51:15.079309 BUILD: 1 commands
[Container] 2025/04/11 11:51:15.079614 Phase complete: DOWNLOAD_SOURCE State: SUCCEEDED
[Container] 2025/04/11 11:51:15.079633 Phase context status code: Message:
[Container] 2025/04/11 11:51:15.255821 Entering phase INSTALL
[Container] 2025/04/11 11:51:15.554217 Phase complete: INSTALL State: SUCCEEDED
[Container] 2025/04/11 11:51:15.554238 Phase context status code: Message:
[Container] 2025/04/11 11:51:15.619840 Entering phase PRE_BUILD
[Container] 2025/04/11 11:51:15.620845 Running command cd cicd
/codebuild/output/tmp/script.sh: line 4: cd: cicd: No such file or directory
[Container] 2025/04/11 11:51:15.628978 Command did not exit successfully cd cicd exit status 1
[Container] 2025/04/11 11:51:15.635096 Phase complete: PRE_BUILD State: FAILED
[Container] 2025/04/11 11:51:15.635109 Phase context status code: COMMAND_EXECUTION_ERROR Message: Error
while executing command: cd cicd. Reason: exit status 1

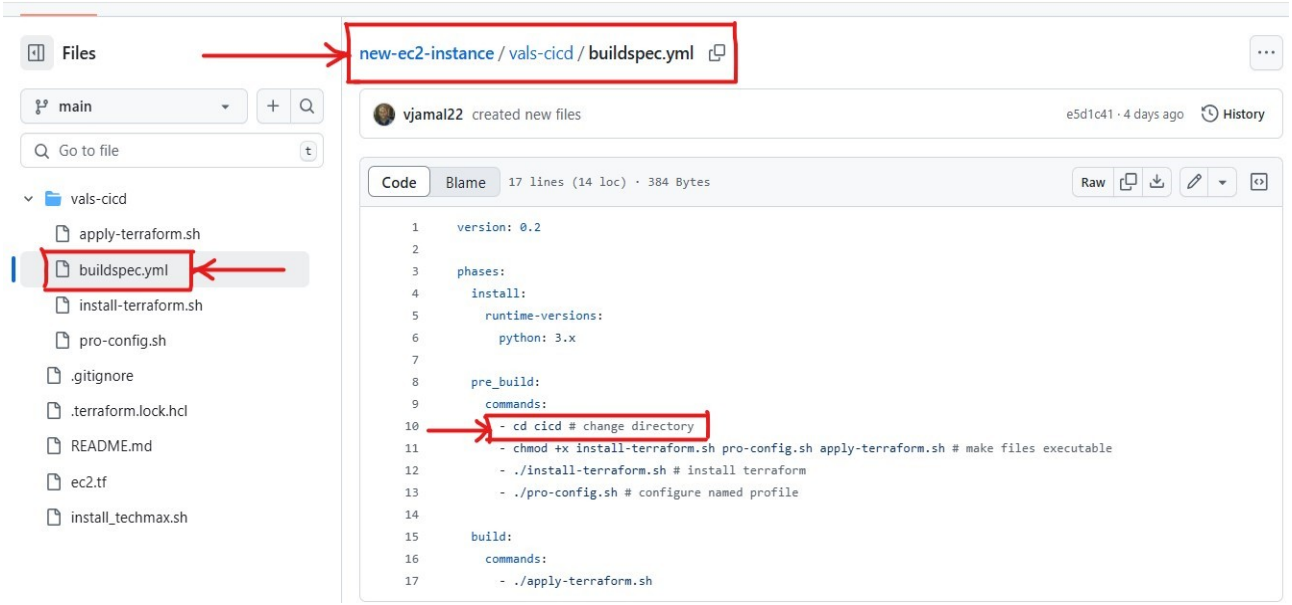
bash

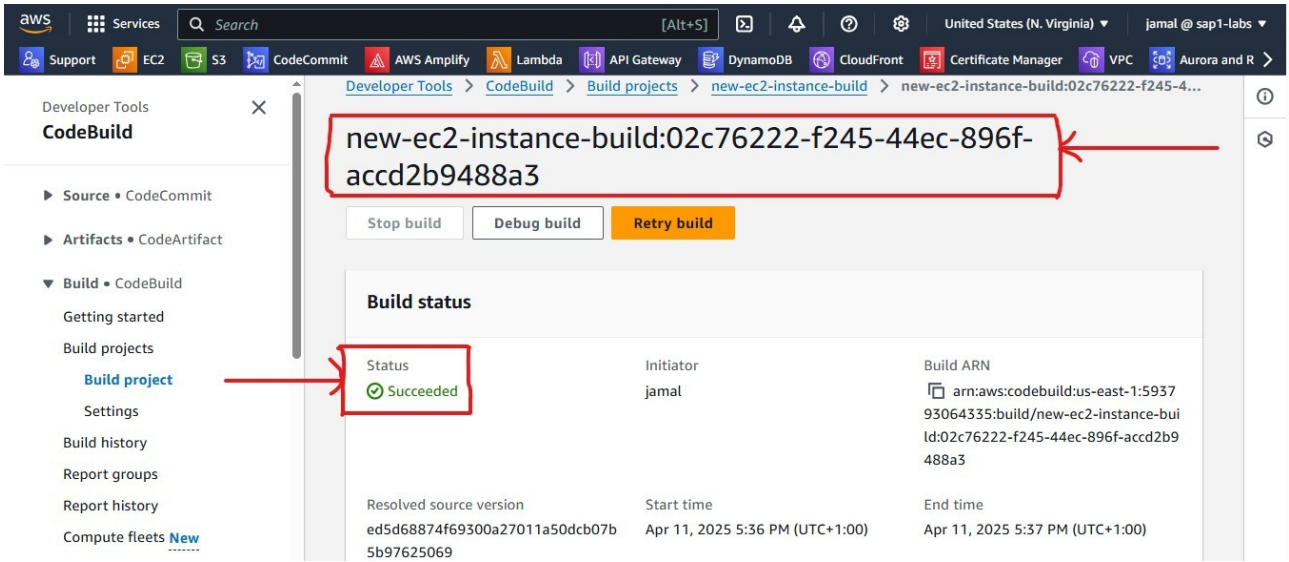
Copy Edit

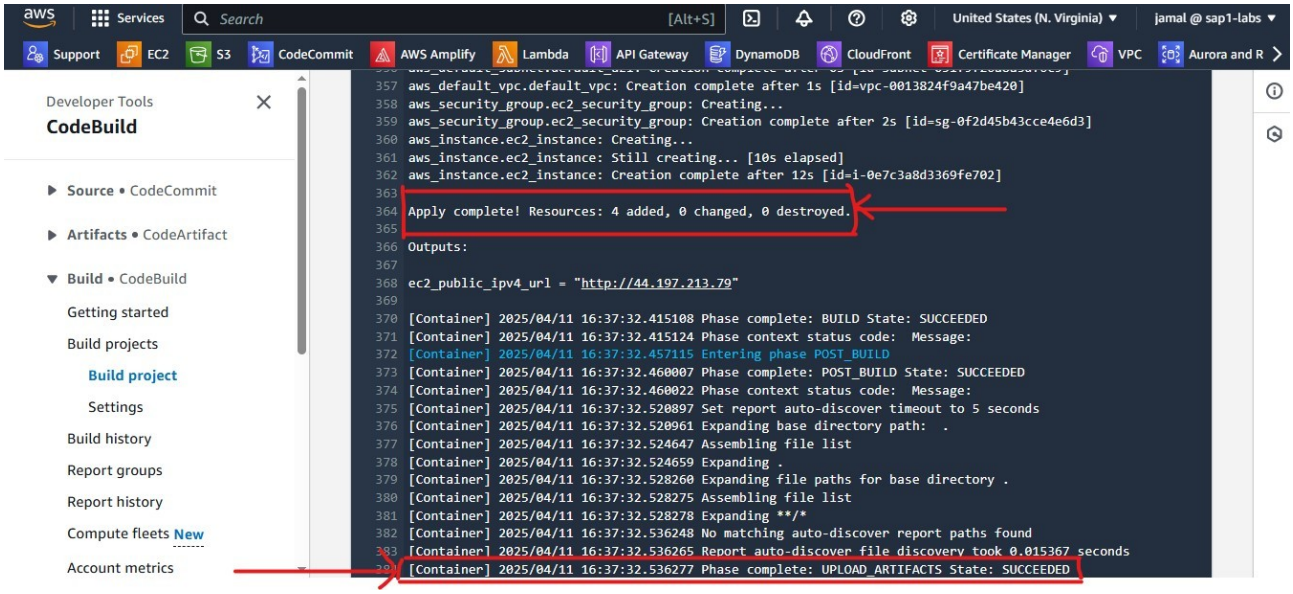
```
[Container] 2025/04/11 11:51:15.620845 Running command cd cicd
/codebuild/output/tmp/script.sh: line 4: cd: cicd: No such file or directory
```

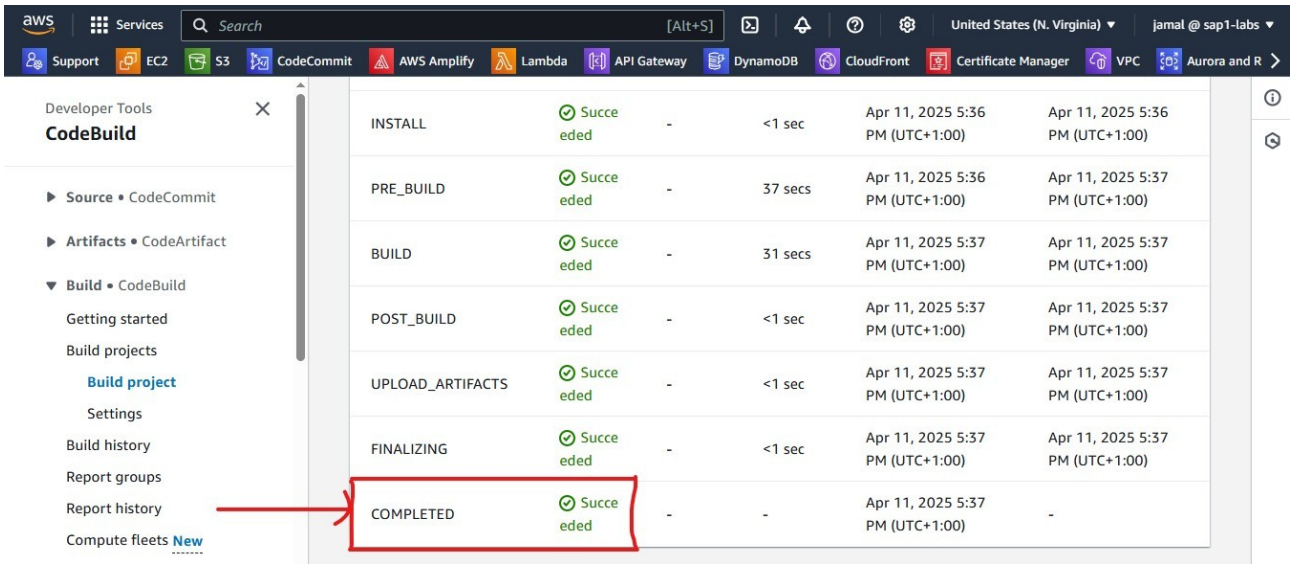
🔍 Root Cause:

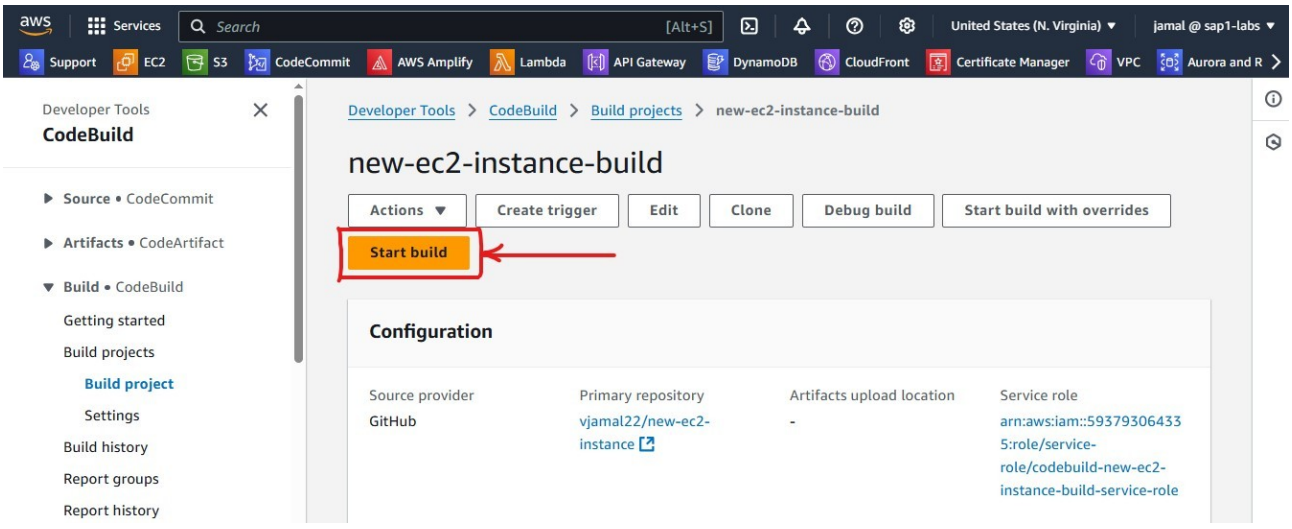
The buildspec.yml instructs CodeBuild to `cd cicd`, but there's no `cicd` folder in the working directory: `/codebuild/output/src1124248516/src/github.com/vjamal22/new-ec2-instance/`.

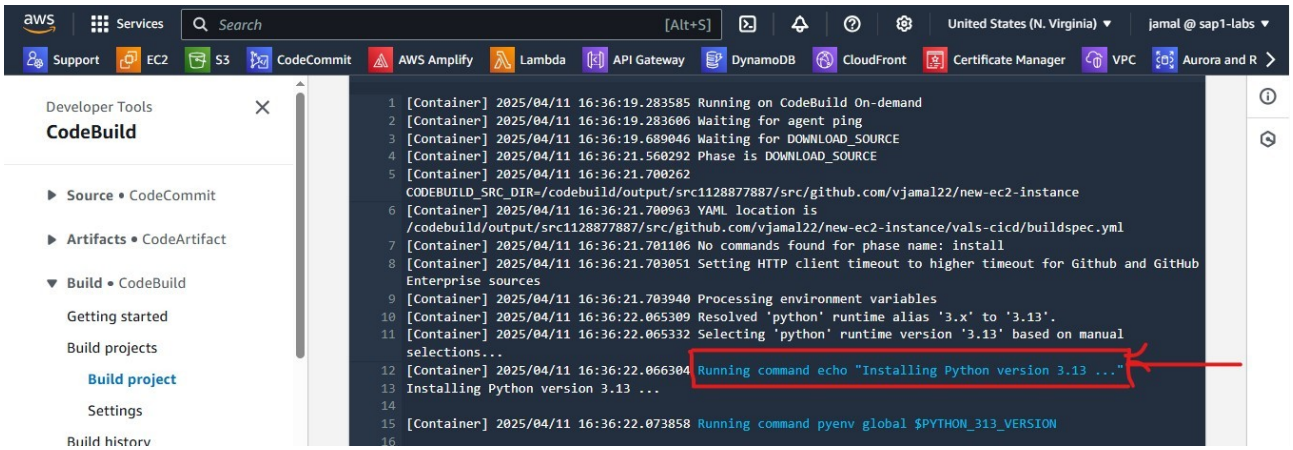


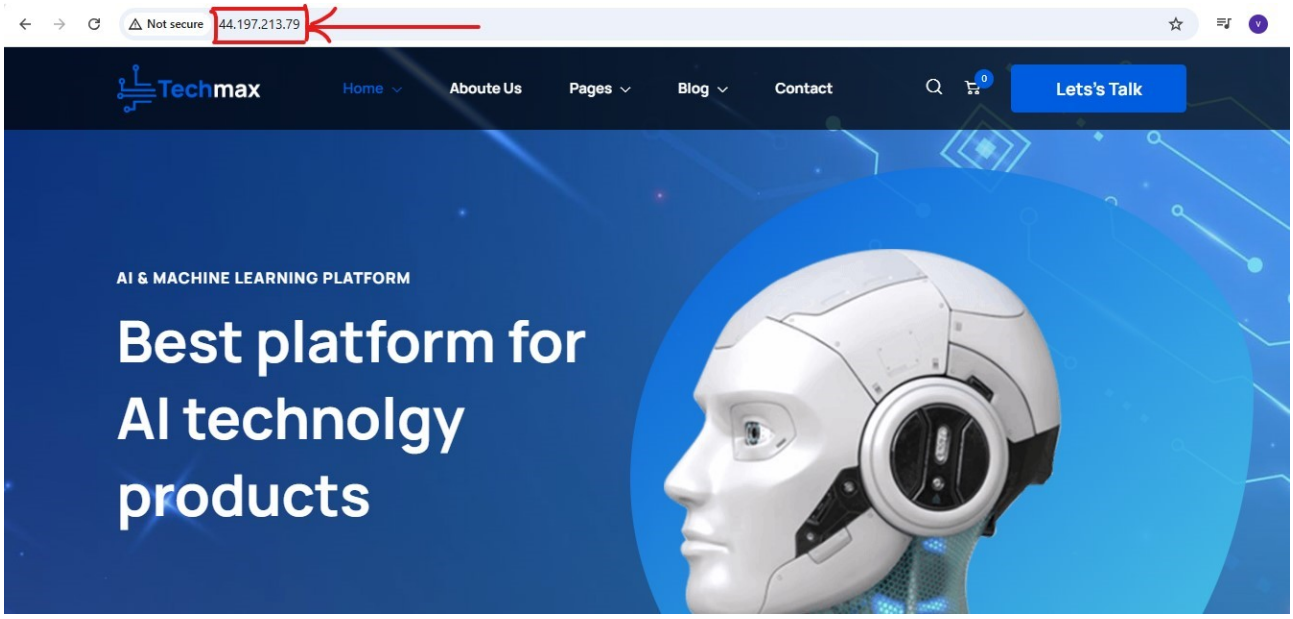


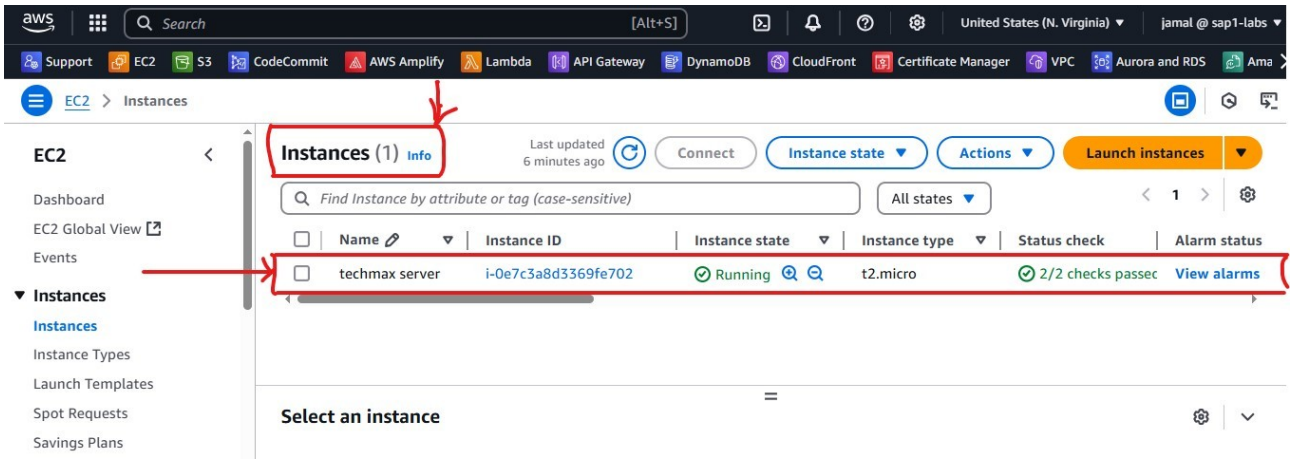

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
Account metrics
aws_default_vpc.default_vpc: Creation complete after 1s [id=vpc-0013824f9a47be420]
aws_security_group.ec2_security_group: Creating...
aws_security_group.ec2_security_group: Creation complete after 2s [id=sg-0f2d45b43cce4e6d3]
aws_instance.ec2_instance: Creating...
aws_instance.ec2_instance: Still creating... [10s elapsed]
aws_instance.ec2_instance: Creation complete after 12s [id=i-0e7c3a8d3369fe702]
Apply complete! Resources: 4 added, 0 changed, 0 destroyed.
Outputs:
ec2_public_ipv4_url = "http://44.197.213.79"
[Container] 2025/04/11 16:37:32.415108 Phase complete: BUILD State: SUCCEEDED
[Container] 2025/04/11 16:37:32.415124 Phase context status code: Message:
[Container] 2025/04/11 16:37:32.457115 Entering phase POST_BUILD
[Container] 2025/04/11 16:37:32.460007 Phase complete: POST_BUILD State: SUCCEEDED
[Container] 2025/04/11 16:37:32.460022 Phase context status code: Message:
[Container] 2025/04/11 16:37:32.520897 Set report auto-discover timeout to 5 seconds
[Container] 2025/04/11 16:37:32.520961 Expanding base directory path: .
[Container] 2025/04/11 16:37:32.524647 Assembling file list
[Container] 2025/04/11 16:37:32.524659 Expanding .
[Container] 2025/04/11 16:37:32.528260 Expanding file paths for base directory .
[Container] 2025/04/11 16:37:32.528275 Assembling file list
[Container] 2025/04/11 16:37:32.528278 Expanding **/*
[Container] 2025/04/11 16:37:32.536248 No matching auto-discover report paths found
[Container] 2025/04/11 16:37:32.536265 Report auto-discover file discovery took 0.015367 seconds
[Container] 2025/04/11 16:37:32.536277 Phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED

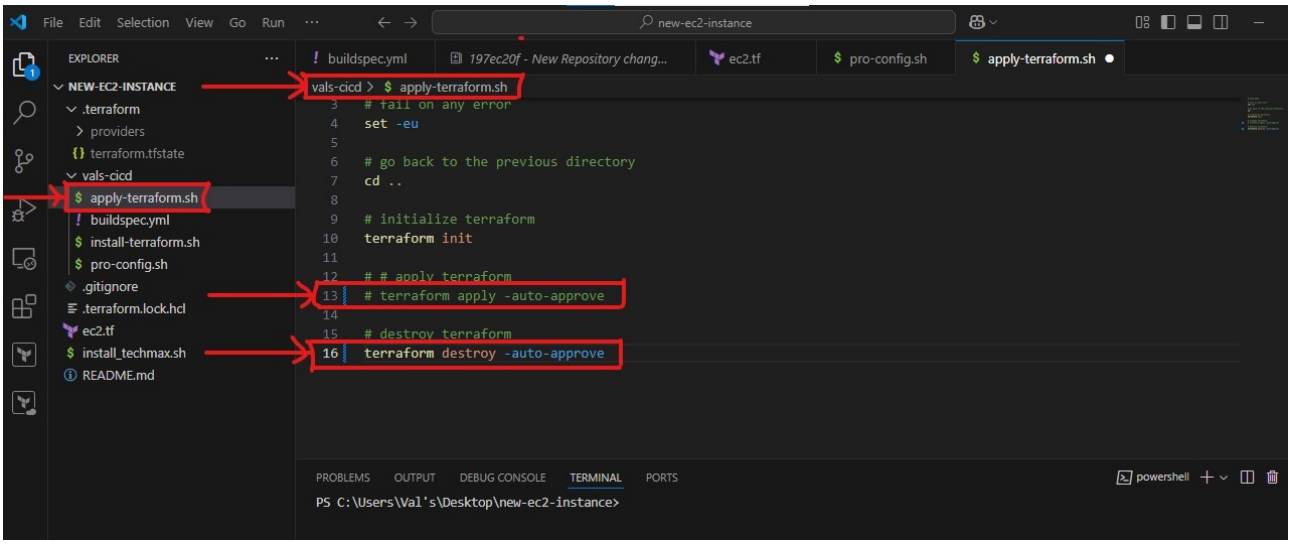

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
INSTALL Succeeded - <1 sec Apr 11, 2025 5:36 PM (UTC+1:00) Apr 11, 2025 5:36 PM (UTC+1:00)
PRE_BUILD Succeeded - 37 secs Apr 11, 2025 5:36 PM (UTC+1:00) Apr 11, 2025 5:37 PM (UTC+1:00)
BUILD Succeeded - 31 secs Apr 11, 2025 5:37 PM (UTC+1:00) Apr 11, 2025 5:37 PM (UTC+1:00)
POST_BUILD Succeeded - <1 sec Apr 11, 2025 5:37 PM (UTC+1:00) Apr 11, 2025 5:37 PM (UTC+1:00)
UPLOAD_ARTIFACTS Succeeded - <1 sec Apr 11, 2025 5:37 PM (UTC+1:00) Apr 11, 2025 5:37 PM (UTC+1:00)
FINALIZING Succeeded - <1 sec Apr 11, 2025 5:37 PM (UTC+1:00) Apr 11, 2025 5:37 PM (UTC+1:00)
COMPLETED Succeeded - - Apr 11, 2025 5:37 PM (UTC+1:00) -

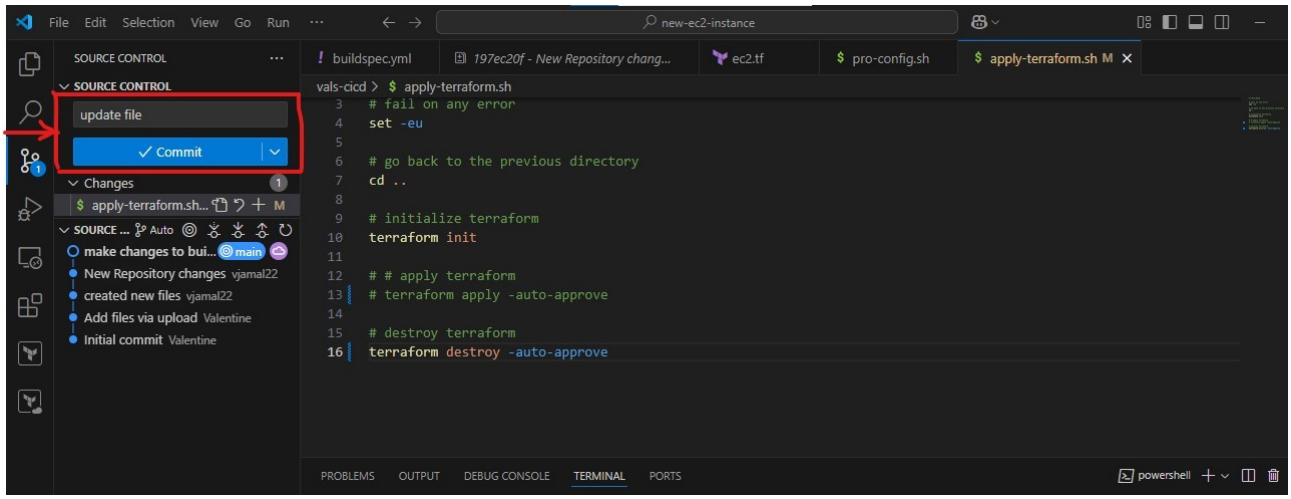

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Developer Tools > CodeBuild > Build projects > new-ec2-instance-build
new-ec2-instance-build
Actions
Create trigger
Edit
Clone
Debug build
Start build with overrides
Start build
Configuration
Source provider
GitHub
Primary repository
vjamal22/new-ec2-instance
Artifacts upload location
-
Service role
arn:aws:iam::593793064335:role/service-role/codebuild-new-ec2-instance-build-service-role

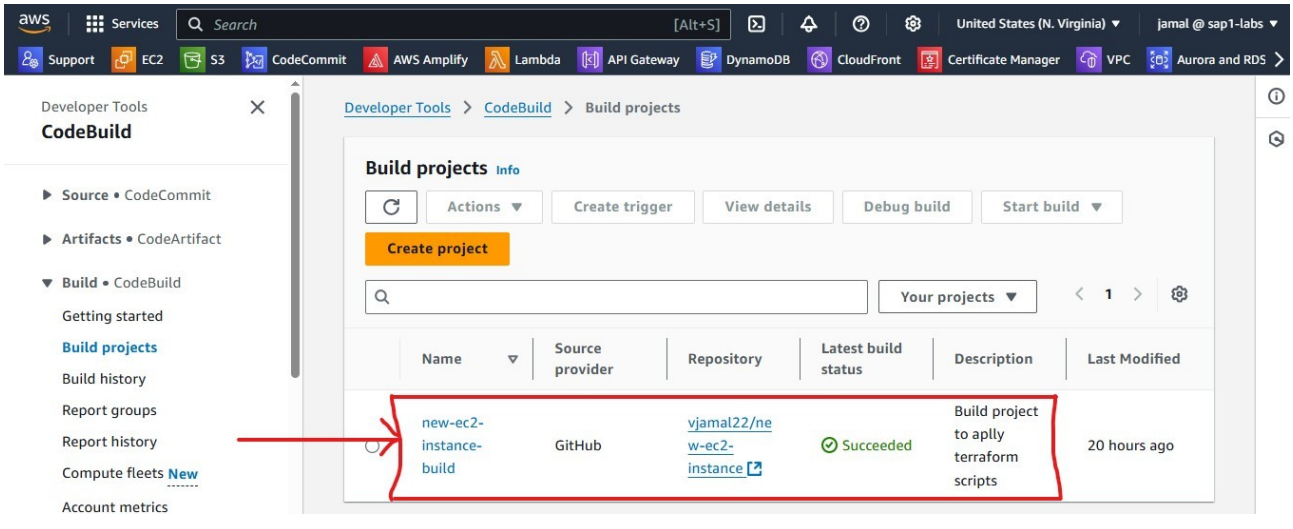

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
[Container] 2025/04/11 16:36:19.283585 Running on CodeBuild On-demand
[Container] 2025/04/11 16:36:19.283606 Waiting for agent ping
[Container] 2025/04/11 16:36:19.689046 Waiting for DOWNLOAD_SOURCE
[Container] 2025/04/11 16:36:21.560292 Phase is DOWNLOAD_SOURCE
[Container] 2025/04/11 16:36:21.700262
CODEBUILD_SRC_DIR=/codebuild/output/src1128877887/src/github.com/vjamal22/new-ec2-instance
[Container] 2025/04/11 16:36:21.700963 YAML location is
/codebuild/output/src1128877887/src/github.com/vjamal22/new-ec2-instance/vals-cicd/buildspec.yml
[Container] 2025/04/11 16:36:21.701106 No commands found for phase name: install
[Container] 2025/04/11 16:36:21.703051 Setting HTTP client timeout to higher timeout for Github and GitHub Enterprise sources
[Container] 2025/04/11 16:36:21.703940 Processing environment variables
[Container] 2025/04/11 16:36:22.065309 Resolved 'python' runtime alias '3.x' to '3.13'.
[Container] 2025/04/11 16:36:22.065332 Selecting 'python' runtime version '3.13' based on manual selections...
[Container] 2025/04/11 16:36:22.066304 Running command echo "Installing Python version 3.13 ..."
Installing Python version 3.13 ...
[Container] 2025/04/11 16:36:22.073858 Running command pyenv global $PYTHON_313_VERSION

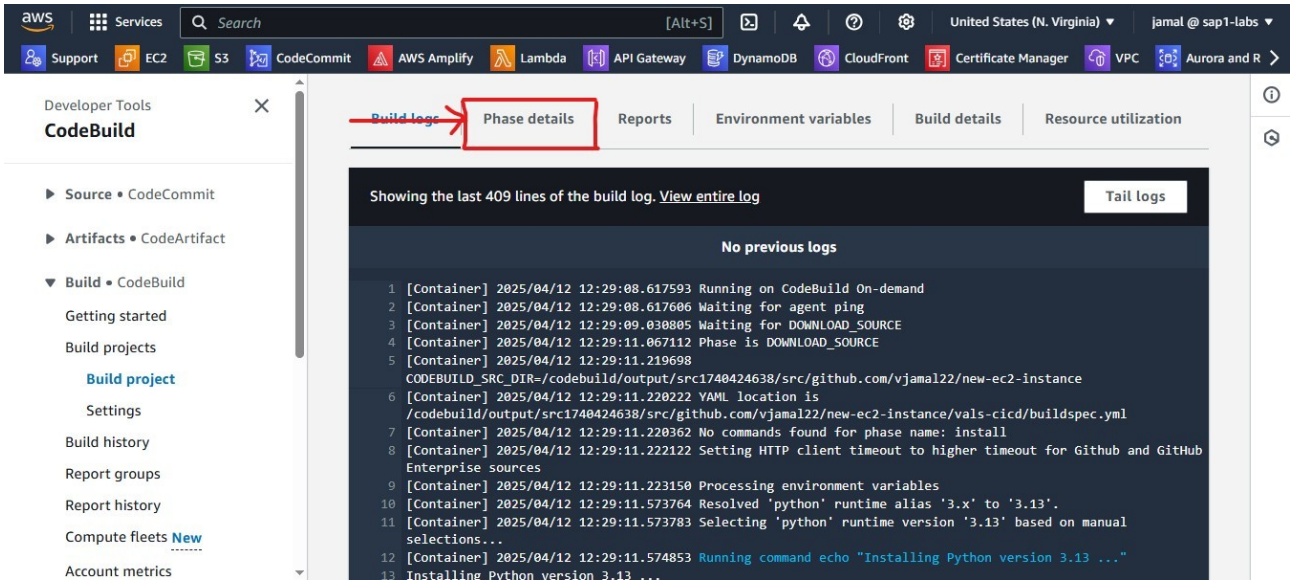

Not secure
44.197.213.79
Techmax
Home
Aboute Us
Pages
Blog
Contact
Lets's Talk
AI & MACHINE LEARNING PLATFORM
Best platform for AI technolgy products

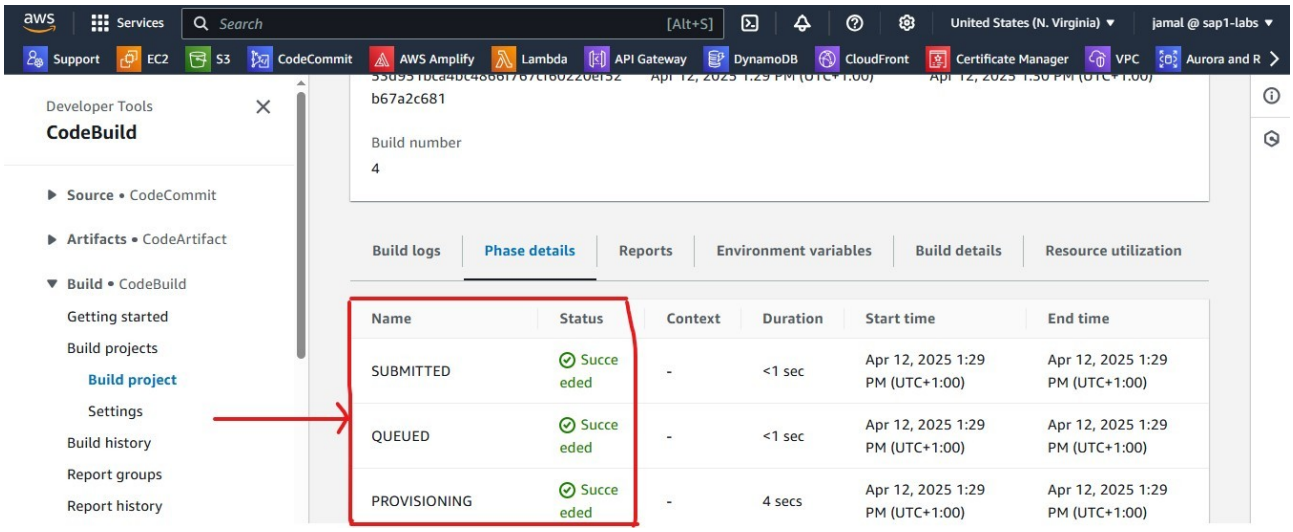

EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Instances (1) Info
Last updated 6 minutes ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
techmax server
i-0e7c3a8d3369fe702
Running
t2.micro
2/2 checks passec
View alarms
Select an instance

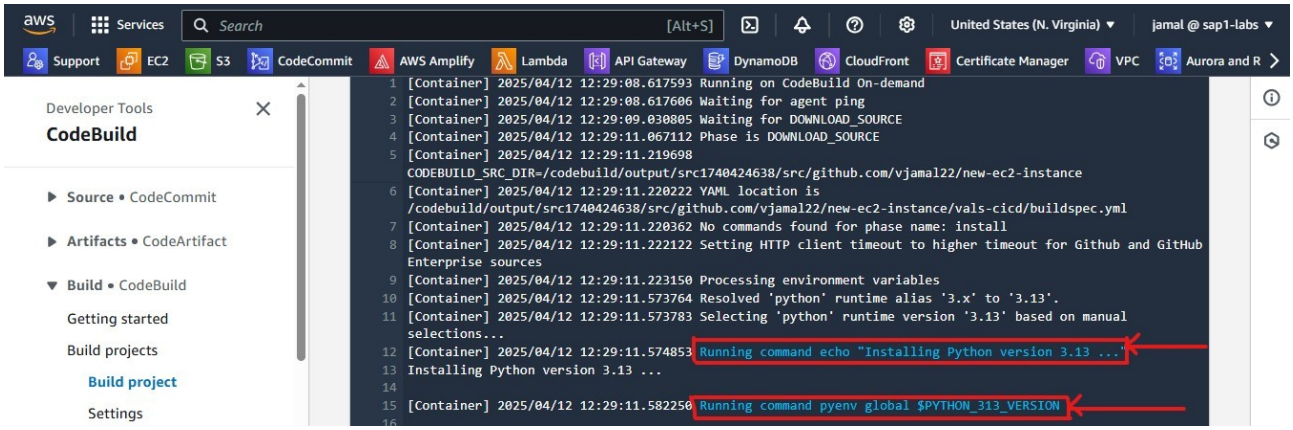

File Edit Selection View Go Run
new-ec2-instance
EXPLORER
NEW-EC2-INSTANCE
.terraform
providers
terraform.tfstate
vals-cicd
apply-terraform.sh
buildspec.yml
install-terraform.sh
pro-config.sh
.gitignore
.terraform.lock.hcl
ec2.tf
install_techmax.sh
README.md
buildspec.yml
197ec20f - New Repository chang...
ec2.tf
pro-config.sh
apply-terraform.sh
vals-cicd > $ apply-terraform.sh
fail on any error
set -eu
go back to the previous directory
cd ..
initialize terraform
terraform init
apply terraform
terraform apply -auto-approve
destroy terraform
terraform destroy -auto-approve
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
PS C:\Users\Val's\Desktop\new-ec2-instance>

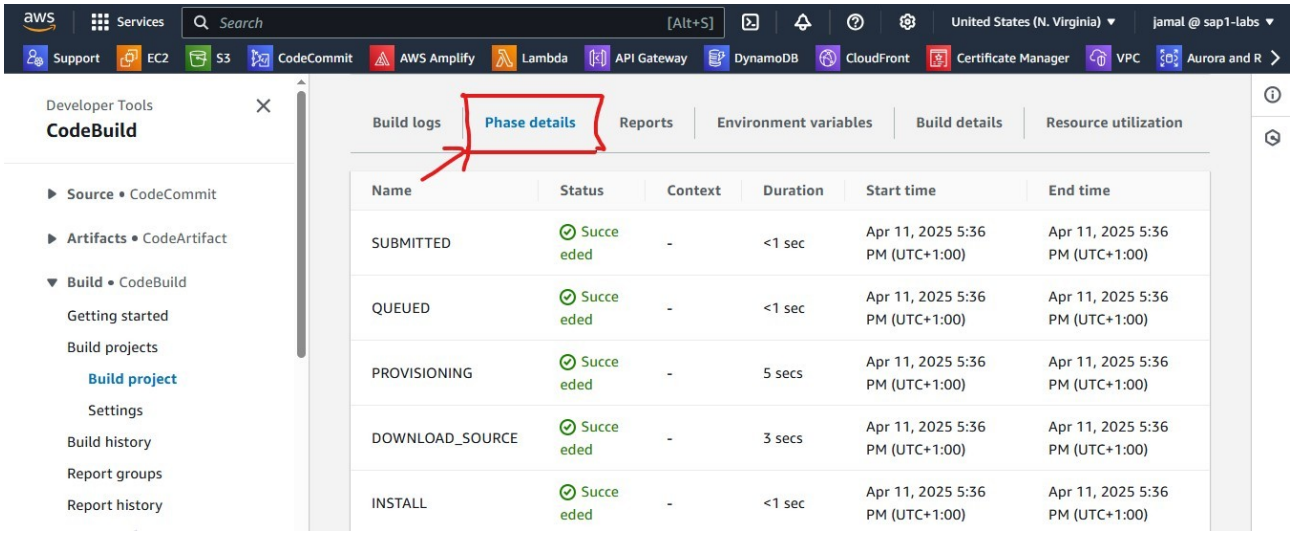

File Edit Selection View Go Run
new-ec2-instance
SOURCE CONTROL
SOURCE CONTROL
update file
Commit
Changes
apply-terraform.sh...
SOURCE ... Auto
make changes to bui... main
New Repository changes vjamal22
created new files vjamal22
Add files via upload Valentine
Initial commit Valentine
buildspec.yml
197ec20f - New Repository chang...
ec2.tf
pro-config.sh
apply-terraform.sh M
vals-cicd > $ apply-terraform.sh
fail on any error
set -eu
go back to the previous directory
cd ..
initialize terraform
terraform init
apply terraform
terraform apply -auto-approve
destroy terraform
terraform destroy -auto-approve
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell

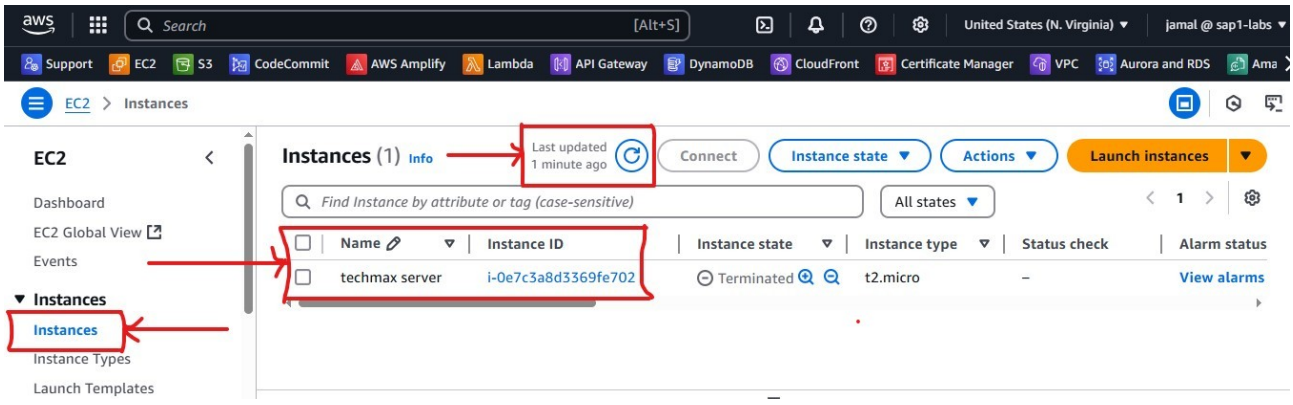

aws
Services
Search
[Alt+S]
United States (N. Virginia)
jamal @ sap1-labs
Support
EC2
S3
CodeCommit
AWS Amplify
Lambda
API Gateway
DynamoDB
CloudFront
Certificate Manager
VPC
Aurora and RDS
Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build history
Report groups
Report history
Compute fleets New
Account metrics
Developer Tools > CodeBuild > Build projects
Build projects Info
Actions
Create trigger
View details
Debug build
Start build
Create project
Your projects
1
Name
Source provider
Repository
Latest build status
Description
Last Modified
new-ec2-instance-build
GitHub
vjamal22/new-ec2-instance
Succeeded
Build project to aplly terraform scripts
20 hours ago

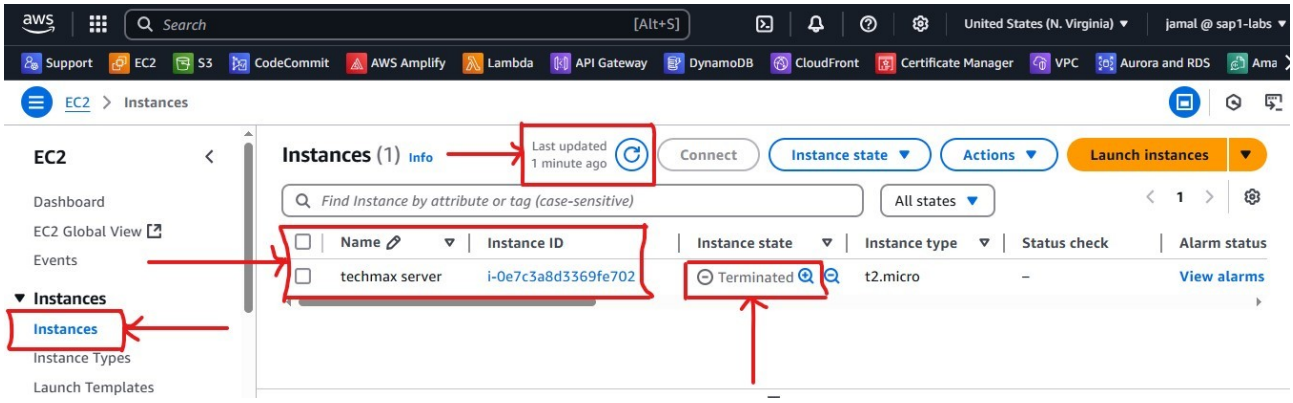

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Compute fleets New
Account metrics
Build logs
Phase details
Reports
Environment variables
Build details
Resource utilization
Showing the last 409 lines of the build log. View entire log
Tail logs
No previous logs
[Container] 2025/04/12 12:29:08.617593 Running on CodeBuild On-demand
[Container] 2025/04/12 12:29:08.617606 Waiting for agent ping
[Container] 2025/04/12 12:29:09.030805 Waiting for DOWNLOAD_SOURCE
[Container] 2025/04/12 12:29:11.067112 Phase is DOWNLOAD_SOURCE
[Container] 2025/04/12 12:29:11.219698 CODEBUILD_SRC_DIR=/codebuild/output/src1740424638/src/github.com/vjamal22/new-ec2-instance
[Container] 2025/04/12 12:29:11.220222 YAML location is /codebuild/output/src1740424638/src/github.com/vjamal22/new-ec2-instance/vals-cicd/buildspec.yml
[Container] 2025/04/12 12:29:11.220362 No commands found for phase name: install
[Container] 2025/04/12 12:29:11.222122 Setting HTTP client timeout to higher timeout for Github and GitHub Enterprise sources
[Container] 2025/04/12 12:29:11.223150 Processing environment variables
[Container] 2025/04/12 12:29:11.573764 Resolved 'python' runtime alias '3.x' to '3.13'.
[Container] 2025/04/12 12:29:11.573783 Selecting 'python' runtime version '3.13' based on manual selections...
[Container] 2025/04/12 12:29:11.574853 Running command echo "Installing Python version 3.13 ..."
Installing Python version 3.13 ...

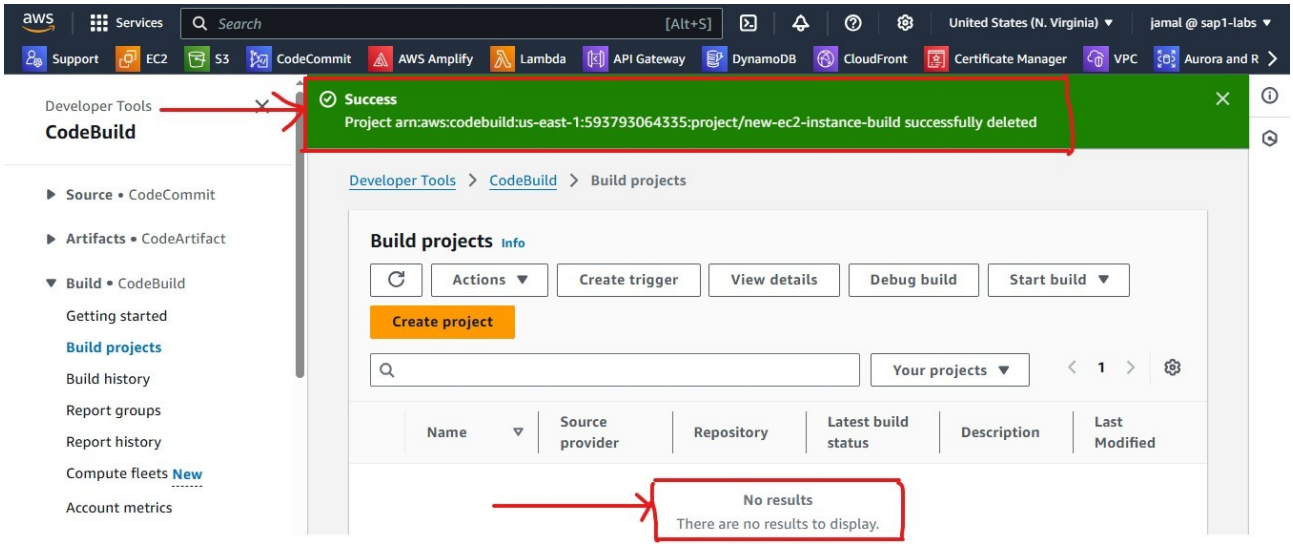

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
b67a2c681
Build number
4
Build logs
Phase details
Reports
Environment variables
Build details
Resource utilization
Name
Status
Context
Duration
Start time
End time
SUBMITTED
Succeeded
-
<1 sec
Apr 12, 2025 1:29 PM (UTC+1:00)
Apr 12, 2025 1:29 PM (UTC+1:00)
QUEUED
Succeeded
-
<1 sec
Apr 12, 2025 1:29 PM (UTC+1:00)
Apr 12, 2025 1:29 PM (UTC+1:00)
PROVISIONING
Succeeded
-
4 secs
Apr 12, 2025 1:29 PM (UTC+1:00)
Apr 12, 2025 1:29 PM (UTC+1:00)

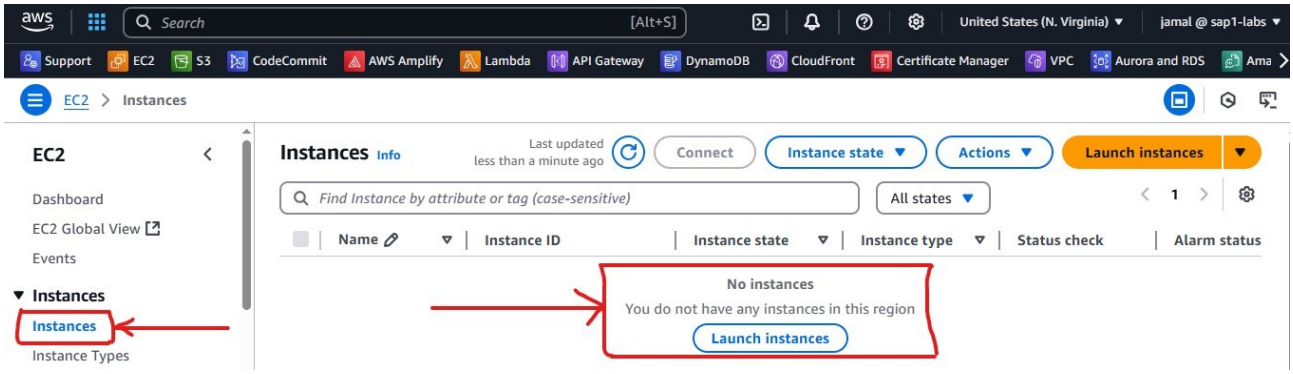

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
[Container] 2025/04/12 12:29:08.617593 Running on CodeBuild On-demand
[Container] 2025/04/12 12:29:08.617606 Waiting for agent ping
[Container] 2025/04/12 12:29:09.030805 Waiting for DOWNLOAD_SOURCE
[Container] 2025/04/12 12:29:11.067112 Phase is DOWNLOAD_SOURCE
[Container] 2025/04/12 12:29:11.219698 CODEBUILD_SRC_DIR=/codebuild/output/src1740424638/src/github.com/vjamal22/new-ec2-instance
[Container] 2025/04/12 12:29:11.220222 YAML location is /codebuild/output/src1740424638/src/github.com/vjamal22/new-ec2-instance/vals-cicd/buildspec.yml
[Container] 2025/04/12 12:29:11.220362 No commands found for phase name: install
[Container] 2025/04/12 12:29:11.222122 Setting HTTP client timeout to higher timeout for Github and GitHub Enterprise sources
[Container] 2025/04/12 12:29:11.223150 Processing environment variables
[Container] 2025/04/12 12:29:11.573764 Resolved 'python' runtime alias '3.x' to '3.13'.
[Container] 2025/04/12 12:29:11.573783 Selecting 'python' runtime version '3.13' based on manual selections...
[Container] 2025/04/12 12:29:11.574853 Running command echo "Installing Python version 3.13 ..."
Installing Python version 3.13 ...
[Container] 2025/04/12 12:29:11.582250 Running command pyenv global $PYTHON_313_VERSION

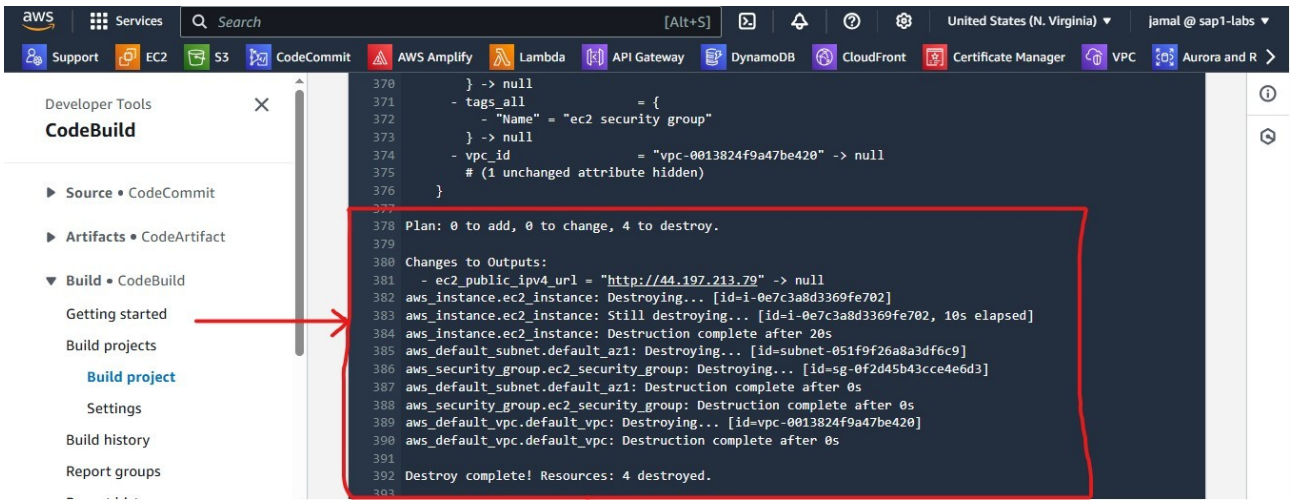

Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
Report history
Build logs
Phase details
Reports
Environment variables
Build details
Resource utilization
Name
Status
Context
Duration
Start time
End time
SUBMITTED
Succeeded
-
<1 sec
Apr 11, 2025 5:36 PM (UTC+1:00)
Apr 11, 2025 5:36 PM (UTC+1:00)
QUEUED
Succeeded
-
<1 sec
Apr 11, 2025 5:36 PM (UTC+1:00)
Apr 11, 2025 5:36 PM (UTC+1:00)
PROVISIONING
Succeeded
-
5 secs
Apr 11, 2025 5:36 PM (UTC+1:00)
Apr 11, 2025 5:36 PM (UTC+1:00)
DOWNLOAD_SOURCE
Succeeded
-
3 secs
Apr 11, 2025 5:36 PM (UTC+1:00)
Apr 11, 2025 5:36 PM (UTC+1:00)
INSTALL
Succeeded
-
<1 sec
Apr 11, 2025 5:36 PM (UTC+1:00)
Apr 11, 2025 5:36 PM (UTC+1:00)


EC2
Instances
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Instances (1) Info
Last updated 1 minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
techmax server
i-0e7c3a8d3369fe702
Terminated
t2.micro
-
View alarms


EC2
Instances
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Instances (1) Info
Last updated 1 minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
techmax server
i-0e7c3a8d3369fe702
Terminated
t2.micro
-
View alarms

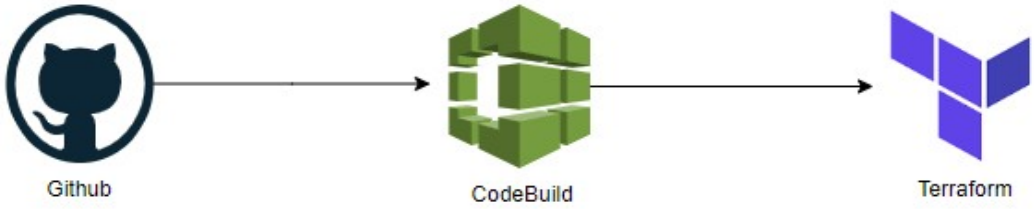

Developer Tools
CodeBuild
Success
Project arn:aws:codebuild:us-east-1:593793064335:project/new-ec2-instance-build successfully deleted
Developer Tools > CodeBuild > Build projects
Build projects Info
Actions
Create trigger
View details
Debug build
Start build
Create project
Your projects
Name
Source provider
Repository
Latest build status
Description
Last Modified
No results
There are no results to display.
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build history
Report groups
Report history
Compute fleets New
Account metrics


EC2 > Instances
EC2
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Instances Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
No instances
You do not have any instances in this region
Launch instances


Developer Tools
CodeBuild
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Getting started
Build projects
Build project
Settings
Build history
Report groups
} -> null
- tags_all = {
- "Name" = "ec2 security group"
} -> null
- vpc_id = "vpc-0013824f9a47be420" -> null
# (1 unchanged attribute hidden)
}
Plan: 0 to add, 0 to change, 4 to destroy.
Changes to Outputs:
- ec2_public_ipv4_url = "http://44.197.213.79" -> null
aws_instance.ec2_instance: Destroying... [id=i-0e7c3a8d3369fe702]
aws_instance.ec2_instance: Still destroying... [id=i-0e7c3a8d3369fe702, 10s elapsed]
aws_instance.ec2_instance: Destruction complete after 20s
aws_default_subnet.default_az1: Destroying... [id=subnet-051f9f26a8a3df6c9]
aws_security_group.ec2_security_group: Destroying... [id=sg-0f2d45b43cce4e6d3]
aws_default_subnet.default_az1: Destruction complete after 0s
aws_security_group.ec2_security_group: Destruction complete after 0s
aws_default_vpc.default_vpc: Destroying... [id=vpc-0013824f9a47be420]
aws_default_vpc.default_vpc: Destruction complete after 0s
Destroy complete! Resources: 4 destroyed.

Architecture diagram for this project

1



2

3