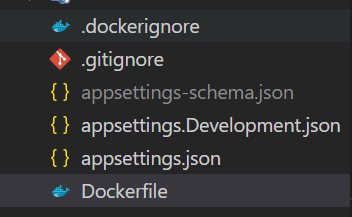
**Umbraco CMS – PureTv**

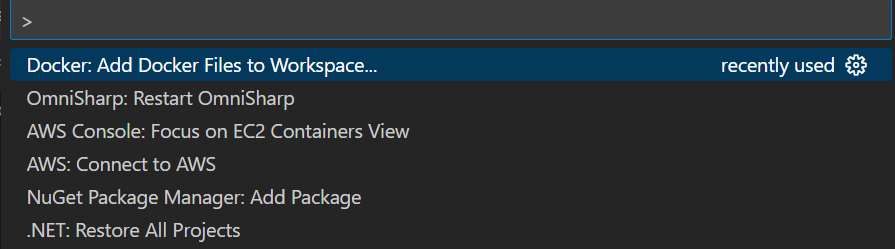
**ECS Deployment Steps:**

1. Download the "**Docker**" extension in the vscode.



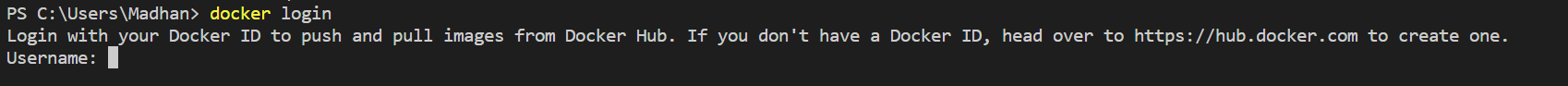
1. To add docker file and dockerignore file into our project. Press **Ctrl+Shift+P** and choose “**add docker file to workspace**”





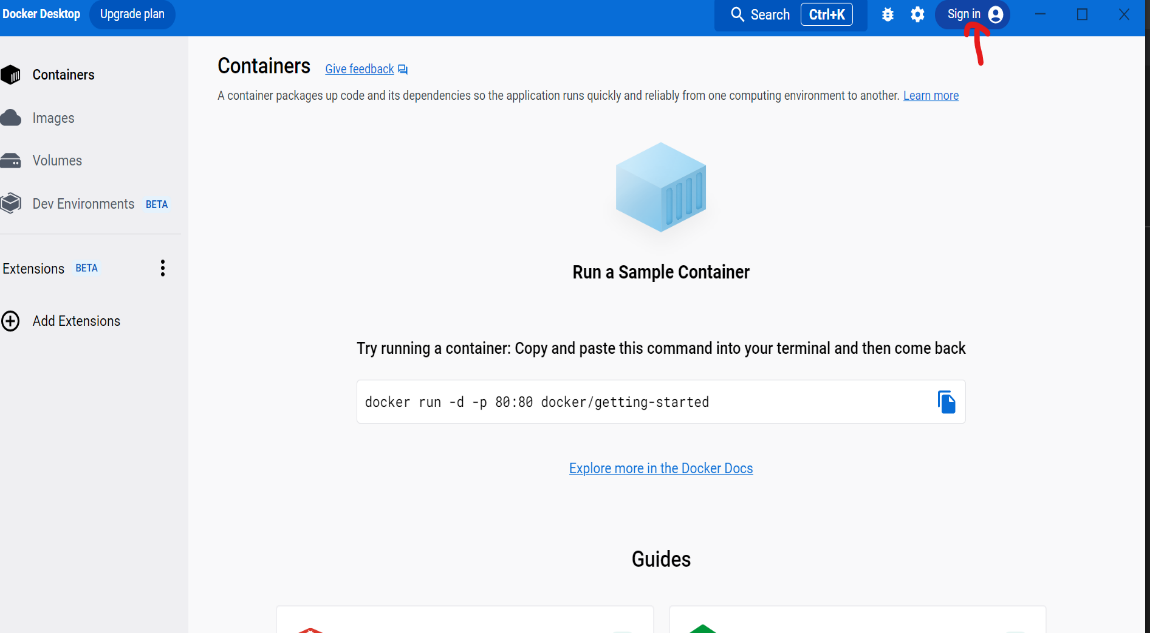
d

1. In the VSCode terminal, enter 🡪 **docker login**. It will prompt you to enter the docker username and password.

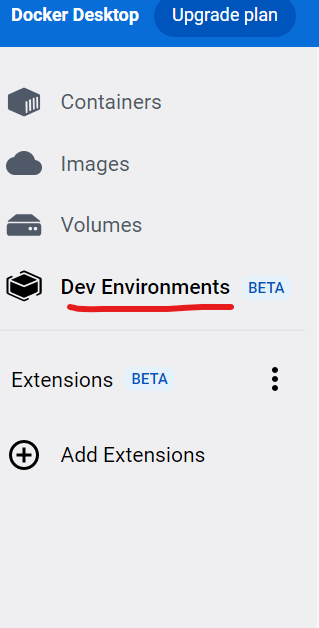


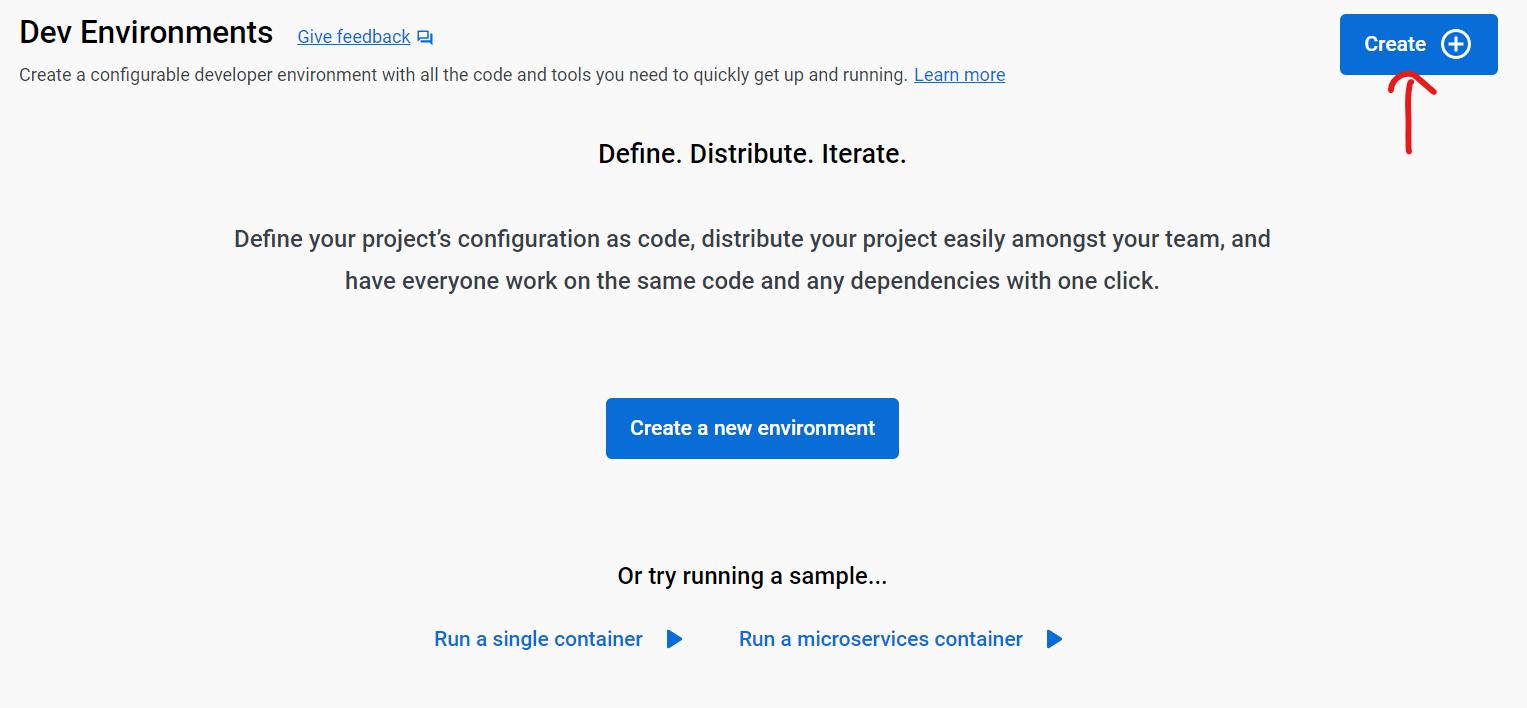
1. Install the “Docker Desktop” in the system.

* Open Docker Desktop and Login with the credentials.

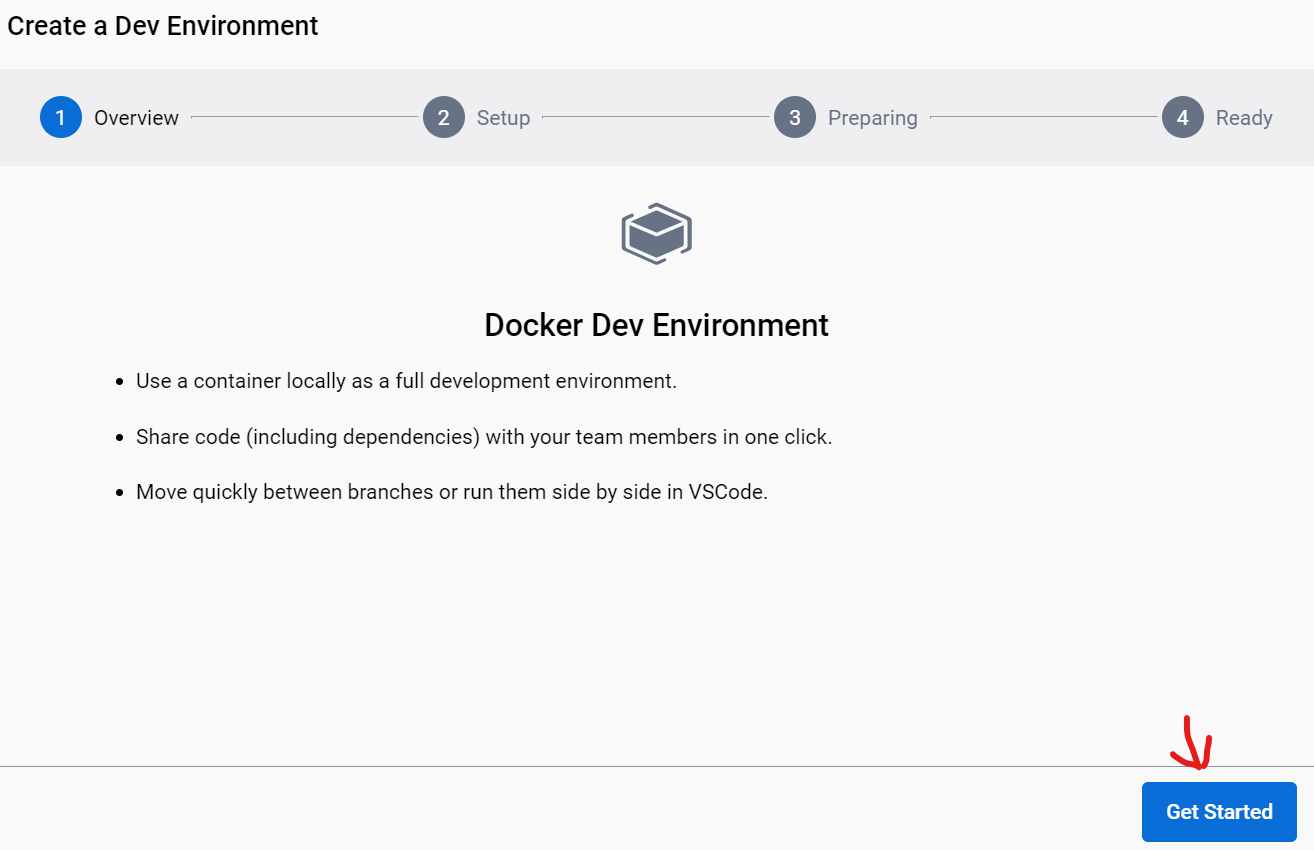


* Click the Dev Environments and click “**Create**” button.

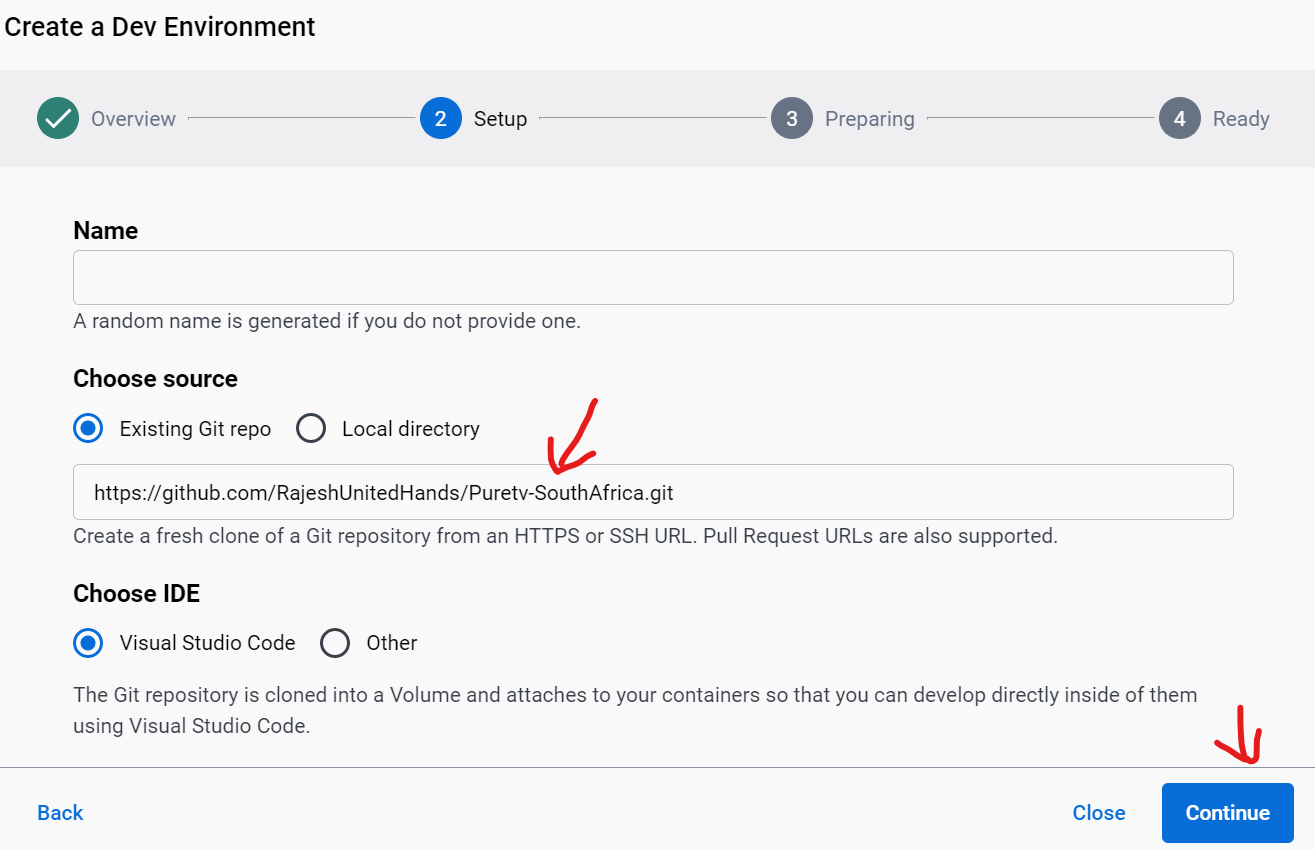




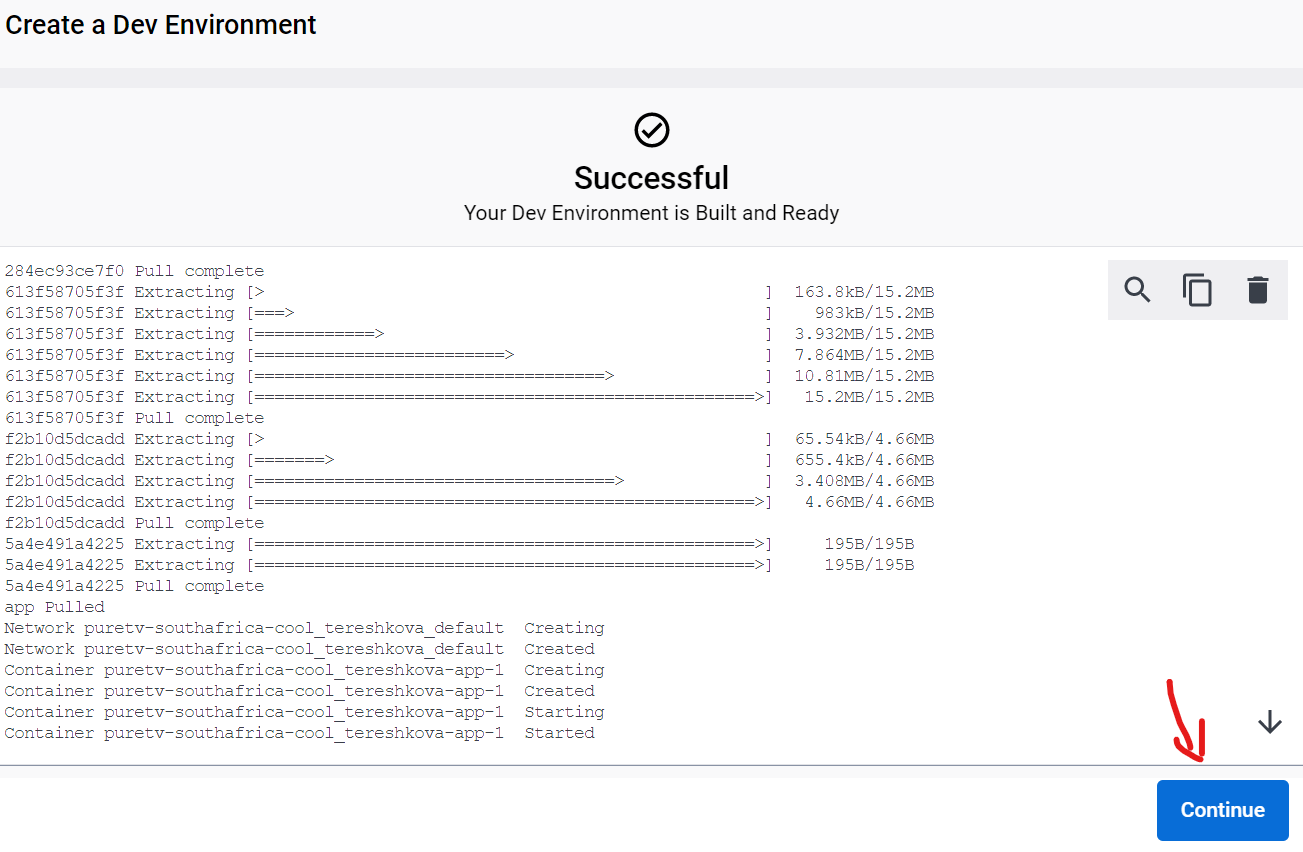
* It will open one window, in that click “**Get Started**”.



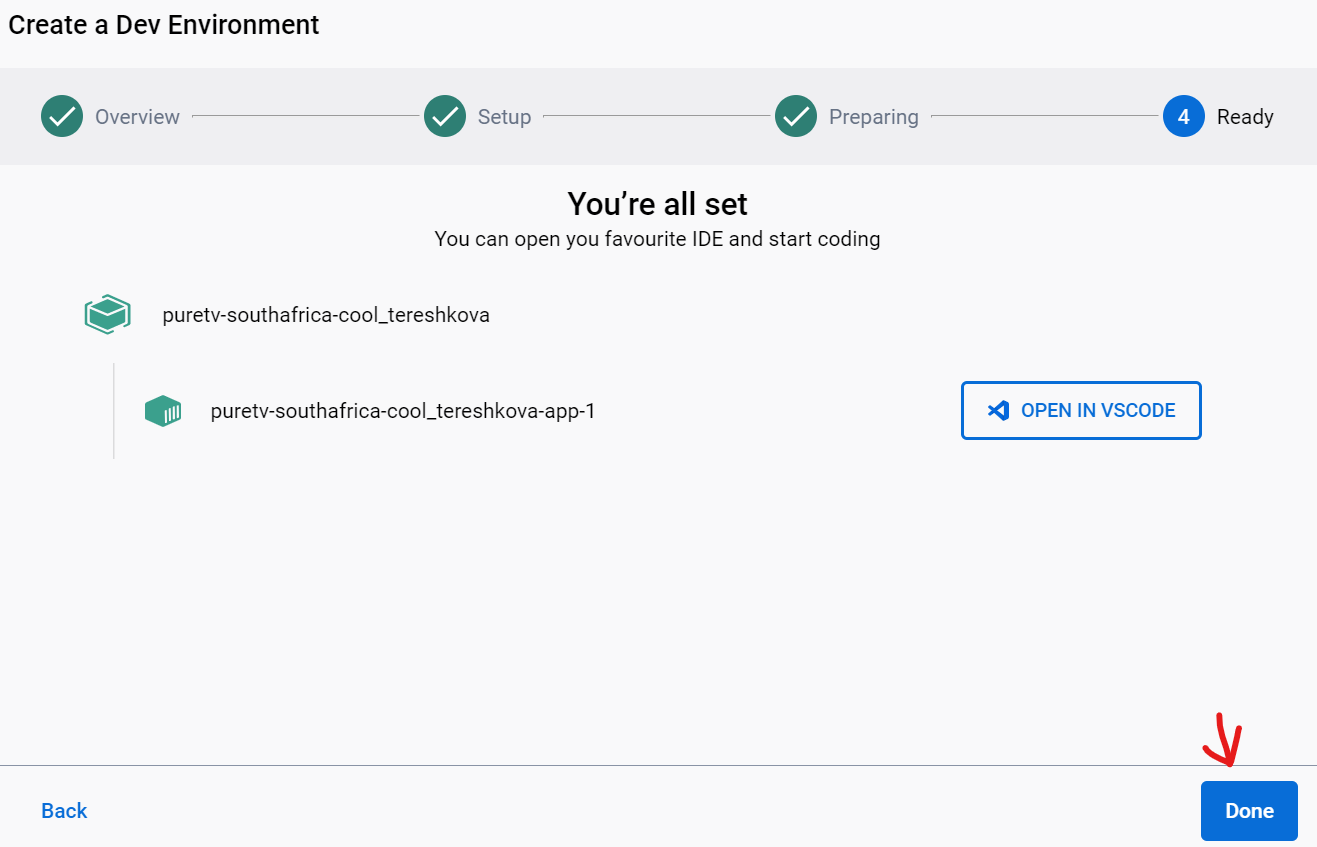
* Then Enter the Git Repository Link and click **continue**.



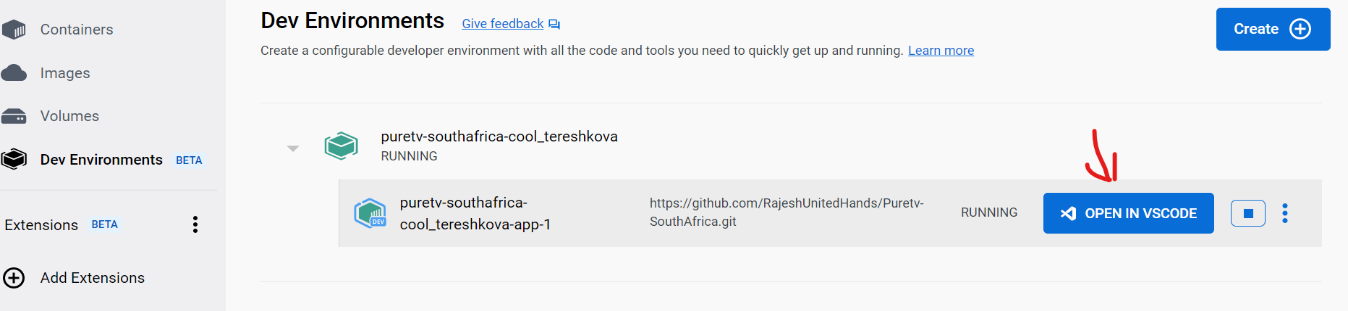
* It will clone the repository and click **continue**.



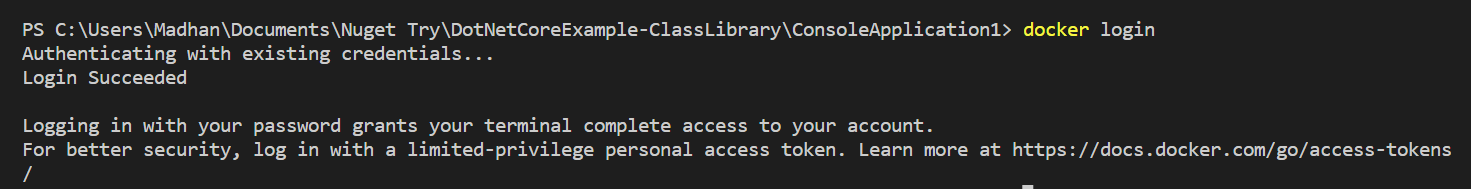
* Then the environment will be created. Click “**Done**”



1. New DevEnvironments is created now. Near the newly created DevEnvironment, there will be an option called "**open in VSCODE**." Click that. It will open the vscode.

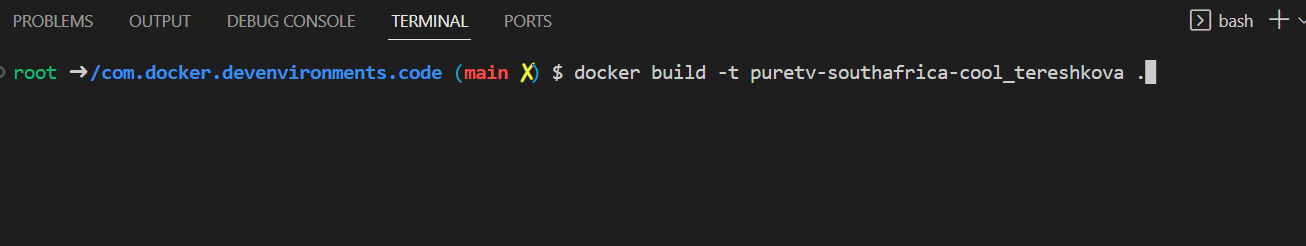


1. In that vscode terminal, type 🡪 **docker login**. It will prompt you to enter the docker username and password.



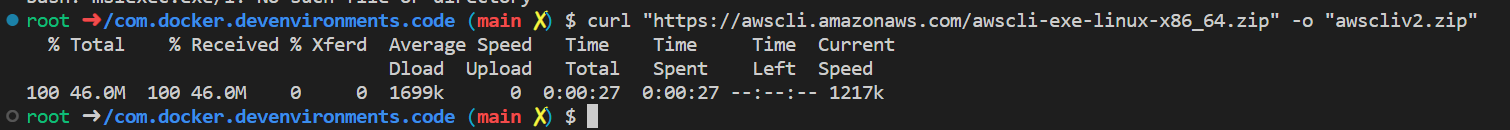
1. After the login, type 🡪 **docker build -t dazzling\_feistel .**

(Here dazzling feistel is the name of the newly created DevEnvironment)

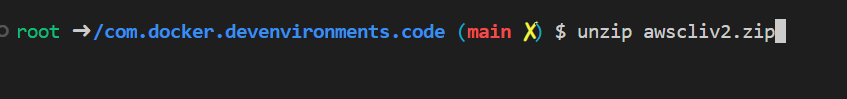


1. To install aws cli, Run this command in the terminal.

Type 🡪 curl <https://awscli.amazonaws.com/aescli-exe-linux-x86_64.zip> -o “awsxliv2.zip”



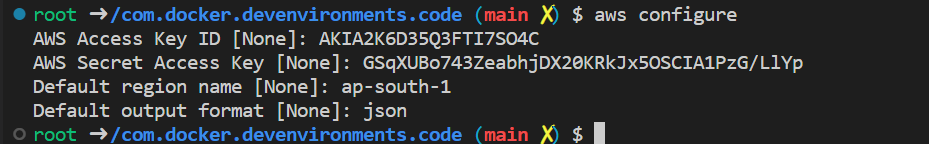
Type 🡪 **unzip awscliv2.zip**



Type 🡪 **sudo ./aws/install**

****

1. To configure the aws, Run this command. Type 🡪 **aws configure**. It will prompt you to enter the access key, secret key ,region and format.



1. Open the AWS console login. [AWS Management Console (amazon.com)](https://aws.amazon.com/console/). Enter the Credentials. Once signed in, type ECR in the Search box. Click **Repositories** under the ECR. It will open new page, in that click “**Create Repository**”. Enter the Repository name and click “ **Create Repository**”
2. Open the Created Respository and click “**View Push Commands**”. It will open a new page with a few commands, which will be executed one by one in the Vscode Terminal.

##### Retrieve an authentication token and authenticate your Docker client to your registry.

* + **aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 710690991158.dkr.ecr.ap-south-1.amazonaws.com**

##### Build your Docker image using the following command. You can skip this step if your image is already built:

* + **docker build -t puretv\_sa .**

##### After the build completes, tag your image so you can push the image to this repository:

* + **docker tag puretv\_sa:latest 710690991158.dkr.ecr.ap-south-1.amazonaws.com/puretv\_sa:latest**

##### Run the following command to push this image to your newly created AWS repository:

* + **docker push 710690991158.dkr.ecr.ap-south-1.amazonaws.com/puretv\_sa:latest**

1. Search for ECS on the AWS console page.

* Click "Task definitions" on the left. On that page click "Create New Task Definition".
* It will open a new page, enter “**Task definition** **family Name**”, “**Name**” and “**Image Uri**” (Copy the image URI from the ECR) and click “**Next**".
* Then choose the running Environment, CPU and RAM size and click “**Next**”. Once you click next, it will move to the preview page and then click “**Create**”. Now the task definition has been created.

1. Click "**Clusters**" on the left side of the ECS page.

* Click the “**Create Clusters**” button. It will open a new page, enter the **“Cluster name**”.
* In the Infrastructure section, tick the "External Instances using ECS anywhere" check box and click the "Create" button.
* Open the created clusters, Scroll down the screen and click the **Services** tab. Press the "Create" button. It will open a new page, under the Deployment Configuration section, Choose the “**family name”** that you specified in the task definition and enter the “ **Service Name**” and then click “**Create**”
* In the **Task** tab, Click the running task, it will open a new page. Under the **Configuration** section, there will be **Public IP** option. In that option, click “**Open address**”**,** it will open our deployed website.