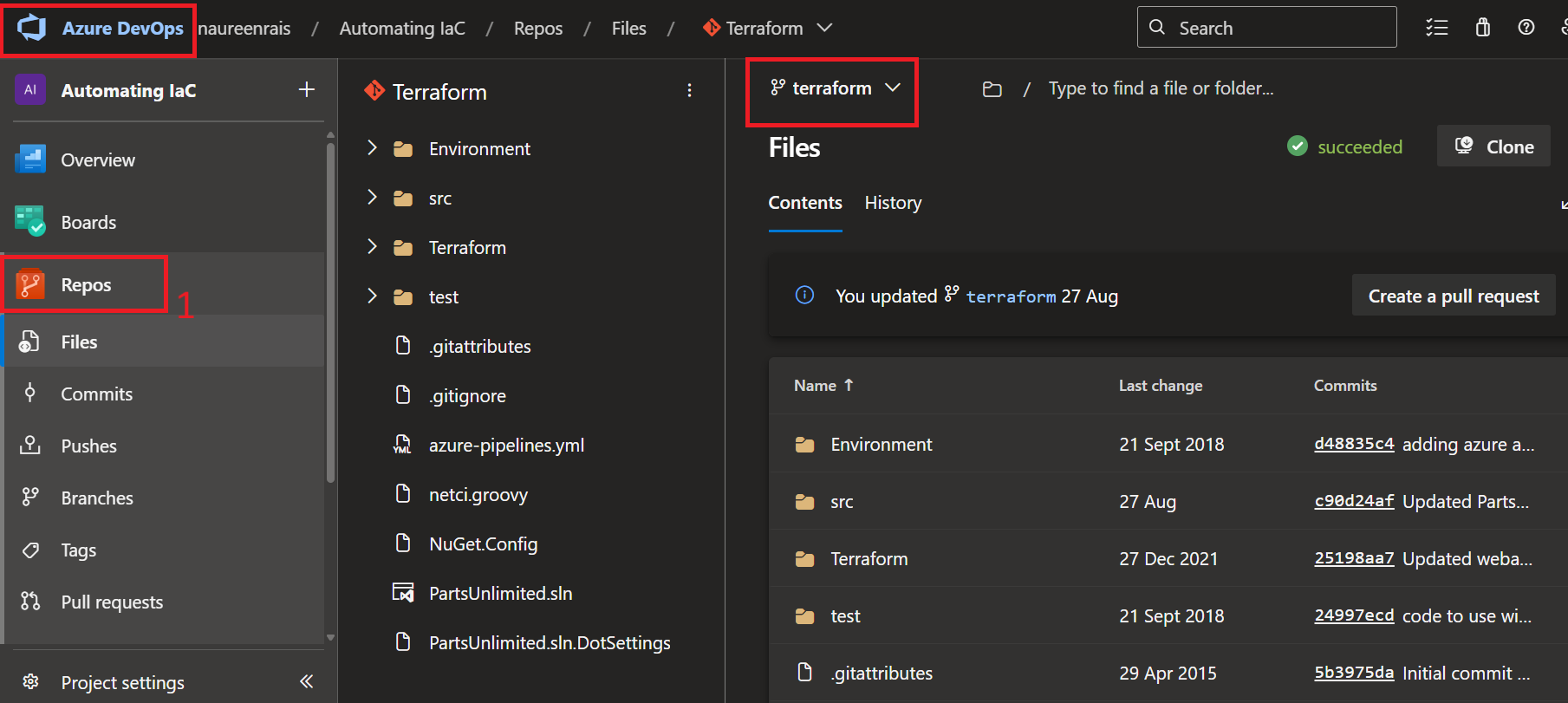
**Automating Infrasture as Code with terraform and azure pipeline**

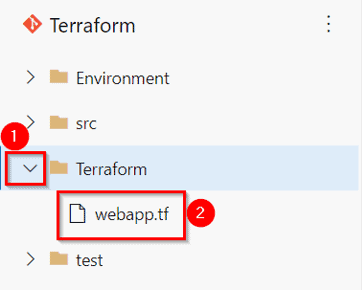
**1. Examine the Terraform configuration files**

**In this task,** we will examine the use of Terraform in provisioning [Azure Resources](https://k21academy.com/microsoft-azure/architect/azure-cli-commands/) required to deploy PartsUnlimited website

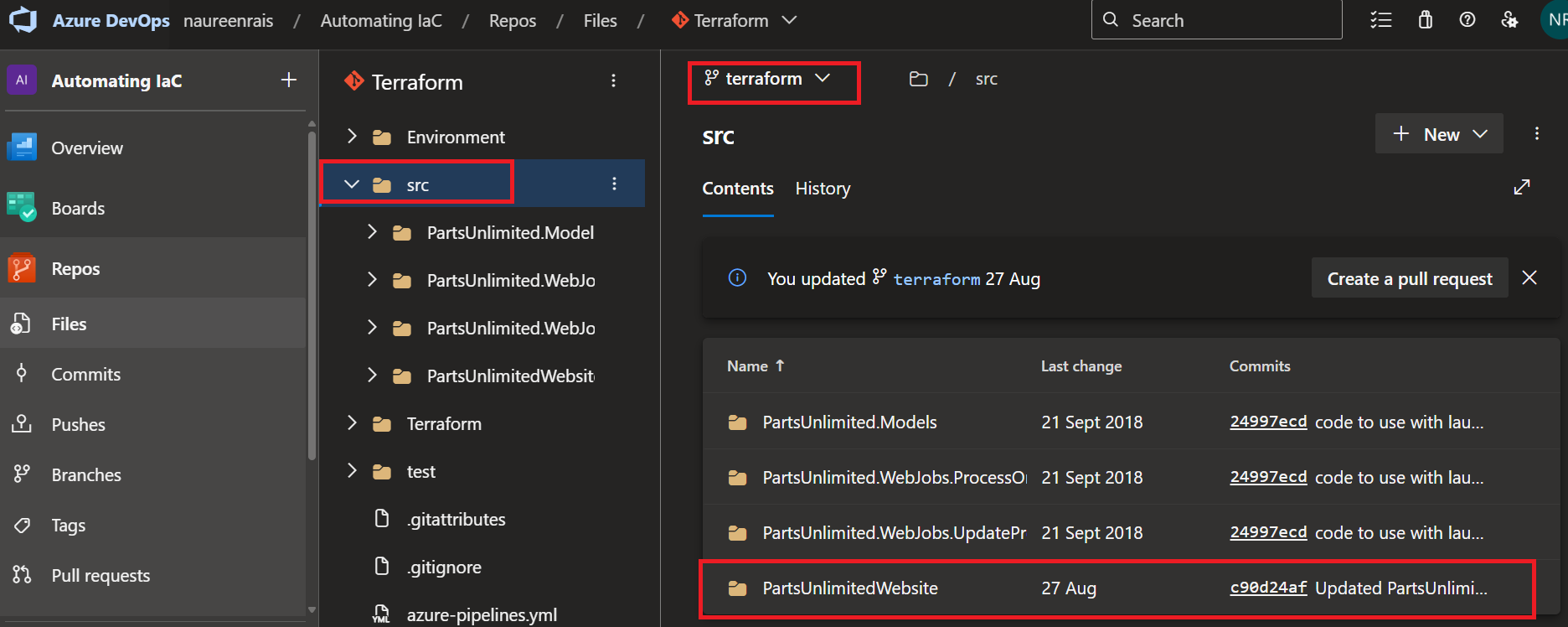
* 1. On the web browser window displaying the Azure DevOps portal with the project name here we are using **Automating IaC** project open, in the vertical menu bar at the far left of the Azure DevOps portal, click **Repos.**



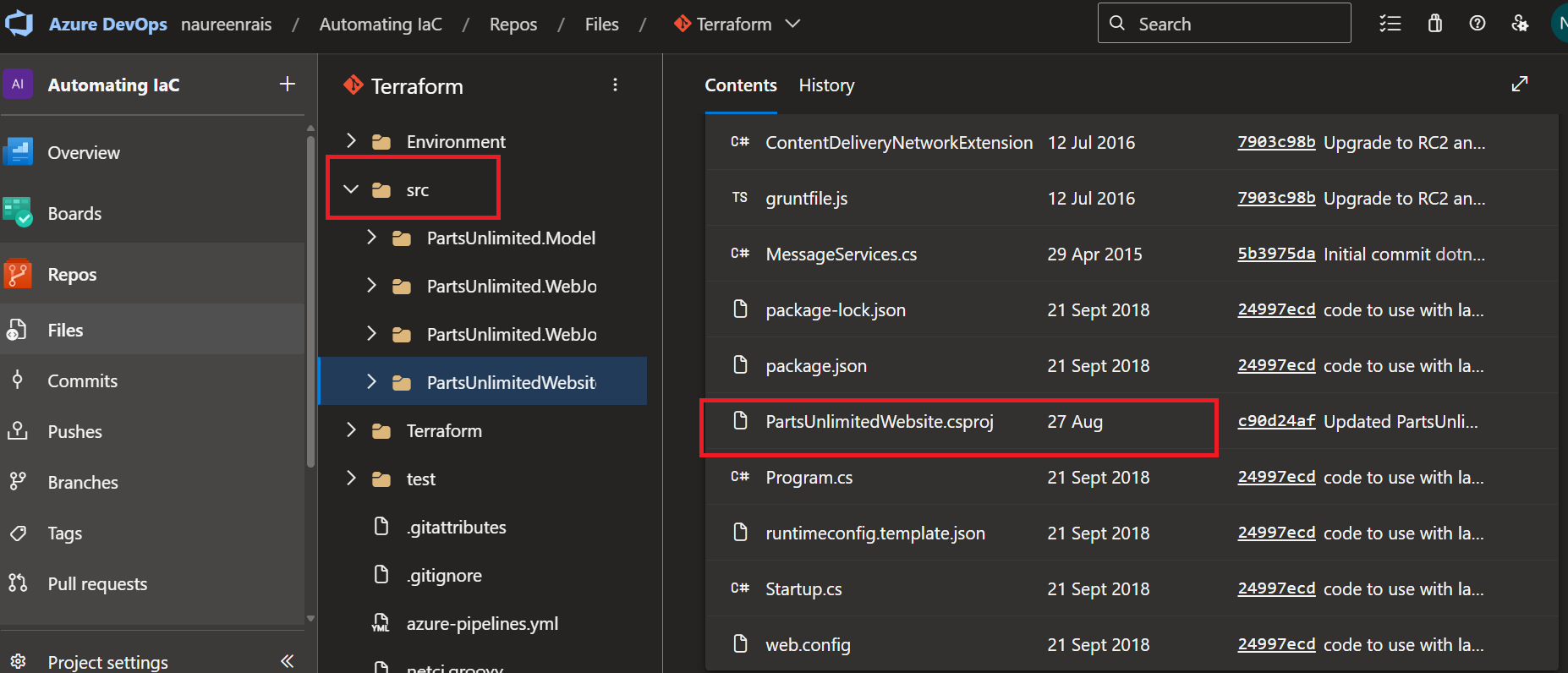
1. On the **Files** pane, click the facing-down caret next to the **master**entry at the top and, in the dropdown list of branches, click the entry representing the **terraform**branch.
2. Under **terraform**branch go under following file to **review** terraform file: **Terraform** folder **->** **webapp.tf .**You can edit the content inside it by clicking **edit.**



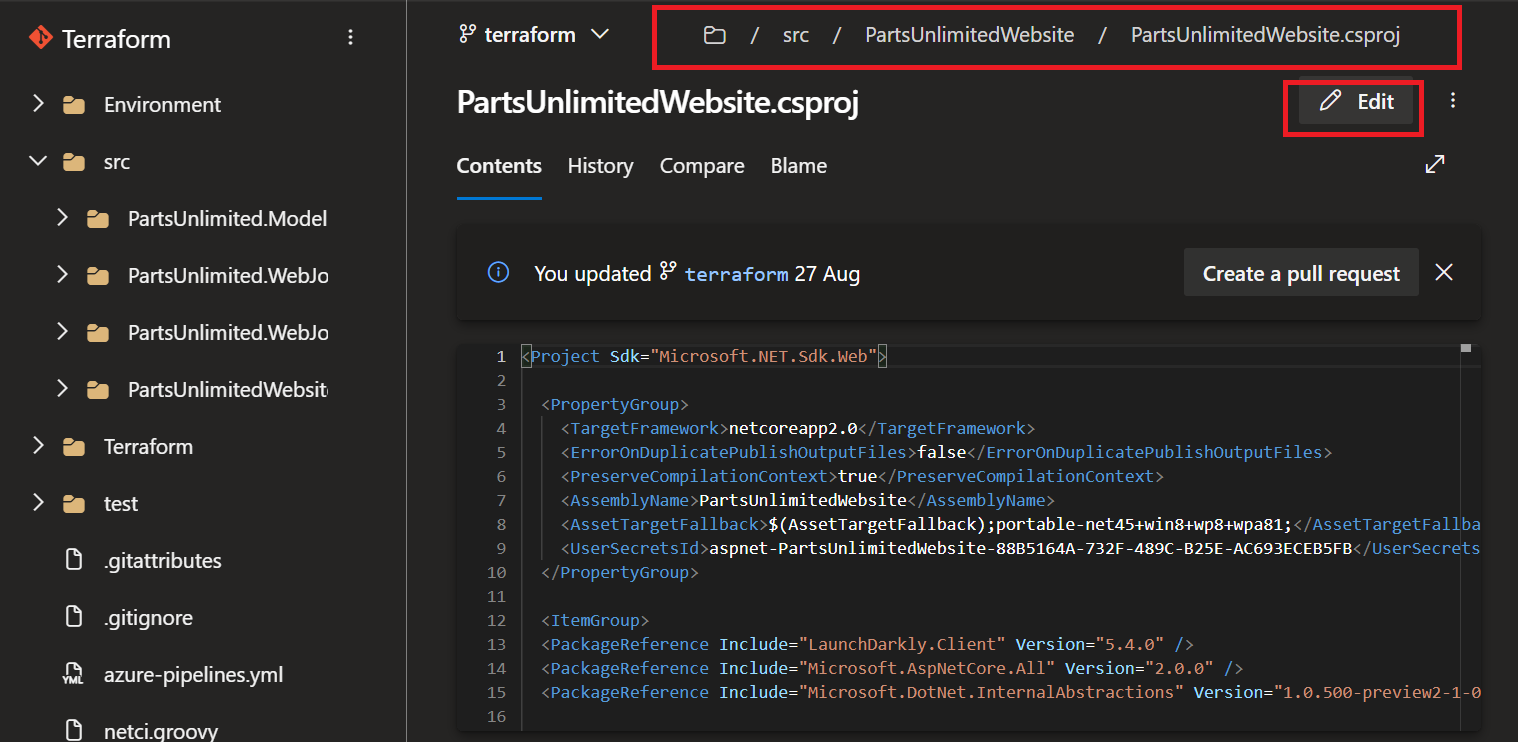
1. Now on left pane of folder click on **src**, then click on **PartsUnlimitedWebsite** on right pane



1. Now scroll down and search for the file **PartsunlimitedWebsite.csproj**. Click on it to open

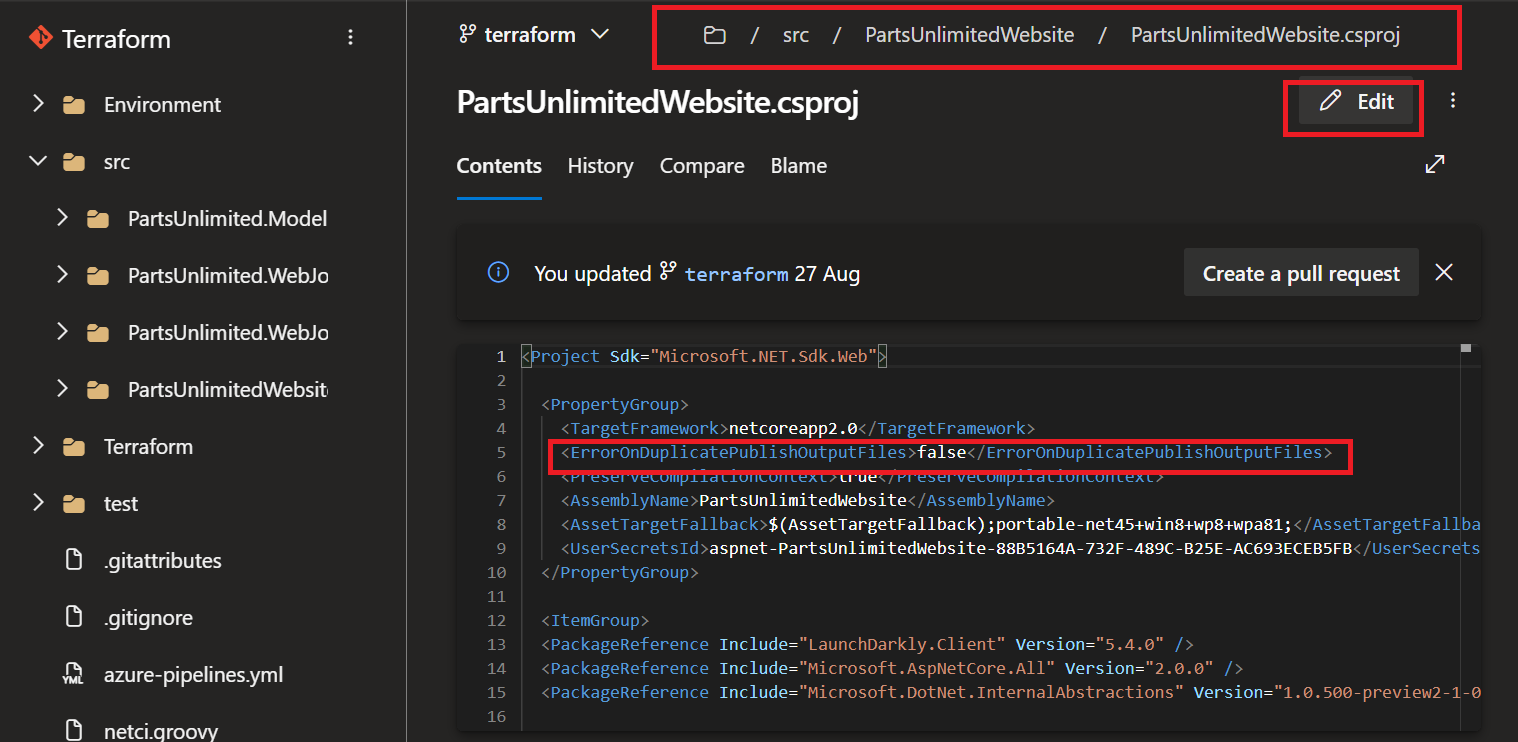


1. Now click on edit to edit the file. Here we will make some changes in **PartsUnlimitedWebsite.csproj** file.



1. Add the code inside <propertygroup> which is given below, then click **commit.**

<ErrorOnDuplicatePublishOutputFiles>**false**</ErrorOnDuplicatePublishOutputFiles>

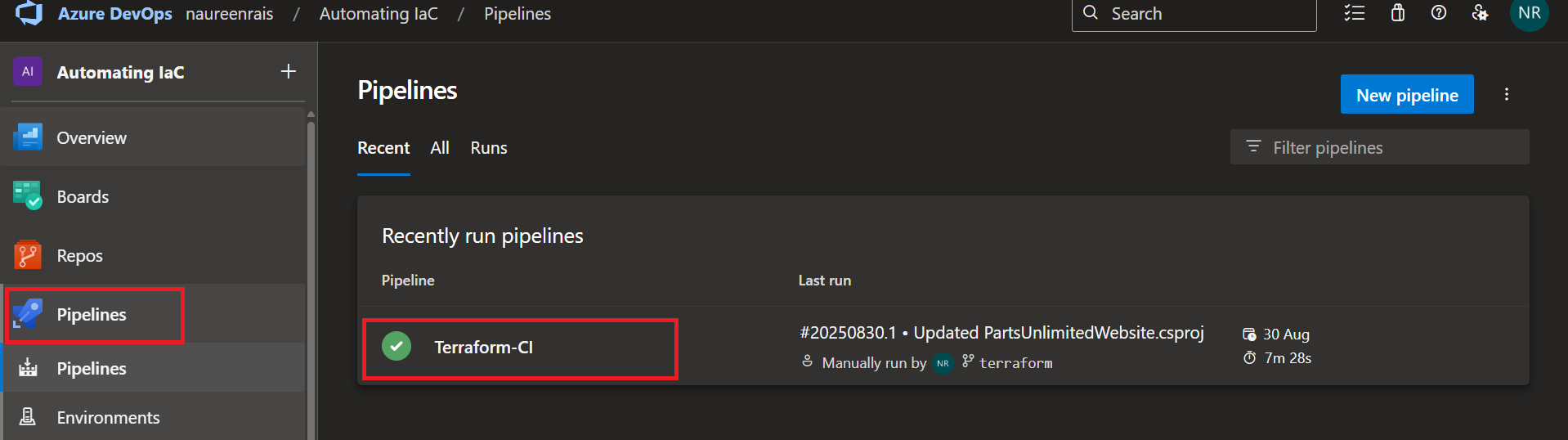


1. Click **Commit** again on commit pane.

**2. Build your application using Azure CI Pipeline**

**In this task,** we will build your application and publish the required files as an artifact called drop

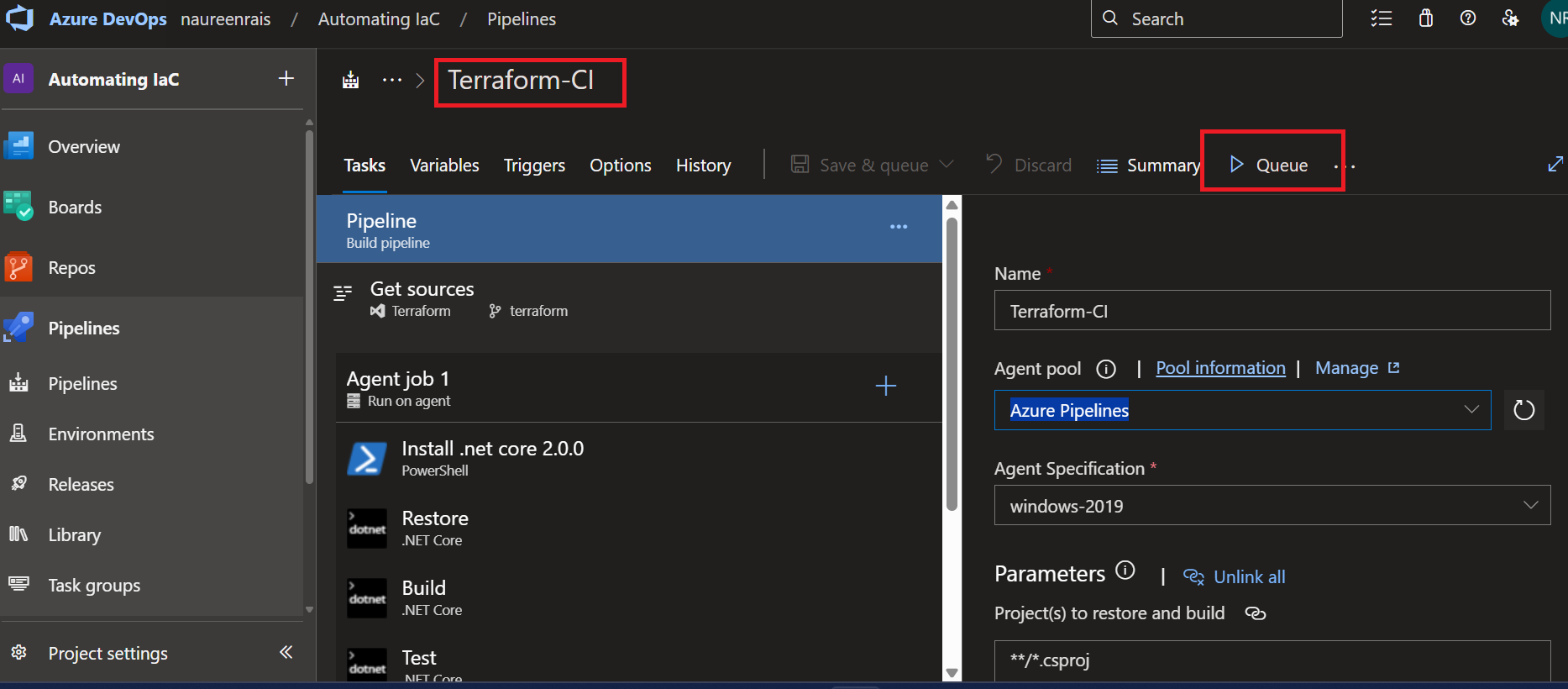
1. In the Azure DevOps portal, in the menu bar at the left of the Azure DevOps portal, click **Pipelines.** Then, select **Pipelines.**



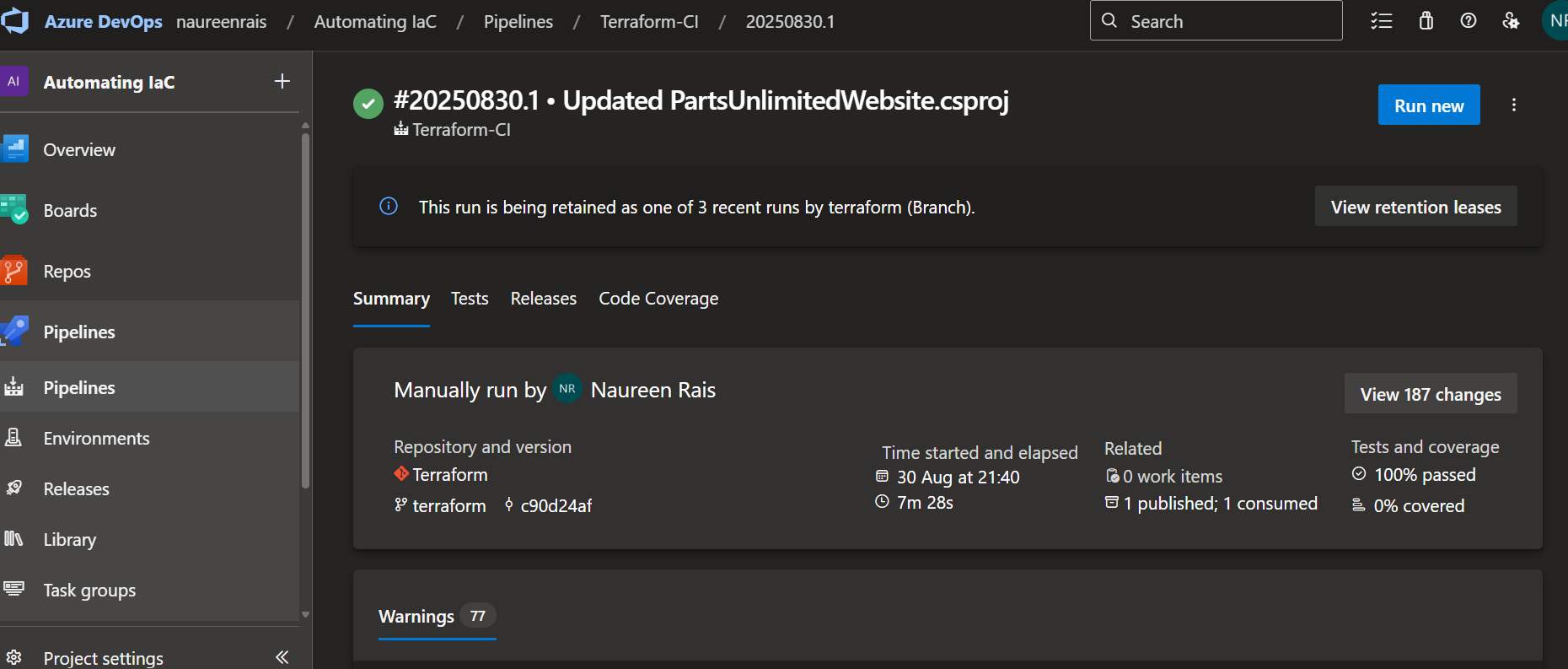
1. On the **Pipelines** pane, click **Terraform-CI** to select it and, on the Terraform-CI pane, click **Edit.**



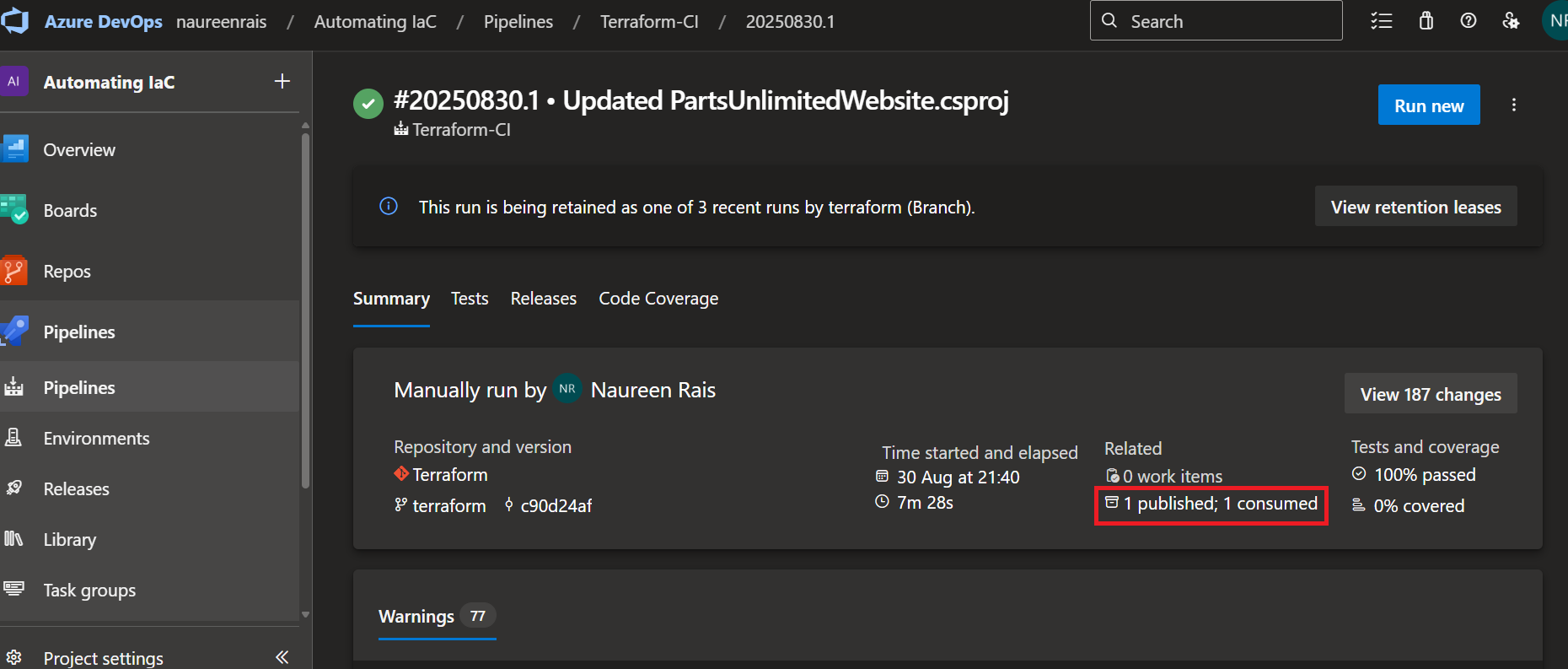
1. Once you review the**Tasks** tab of the **Terraform-CI** pane, click**Queue.**



1. On the **Run pipeline** pane, click **Run** to initiate the build.
2. On the **Summary**tab of the build run pane, in the **Jobs** section, click **Agent** **job 1** and monitor the progress of the build process.



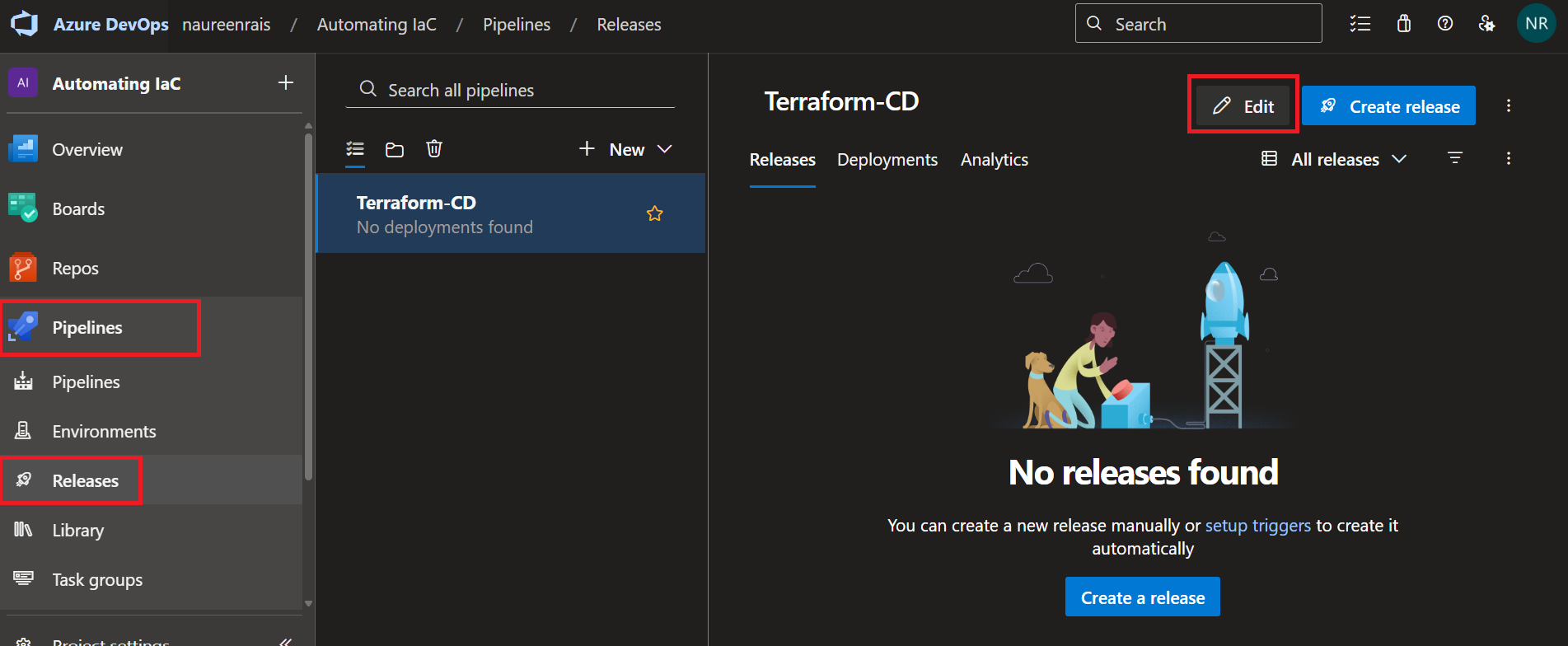
1. Once the build succeeds, switch back to the **Summary** tab of the build run pane, in the **Related** section, click the **1 published**, **1 consumed** link. This will display the Artifacts pane



**3. Deploy resources using Terraform (IaC) in Azure CD pipeline**

**In this task**, we will create Azure resources using Terraform as part of your deployment pipeline and then deploy the PartsUnlimited application to an Azure app service web app provisioned by Terraform.

1. In the Azure DevOps portal, in the vertical menu bar at the left of the Azure DevOps portal, in the **Pipelines** section, click **Releases**, ensure that the **Terraform-CD** entry is selected, and click **Edit.**

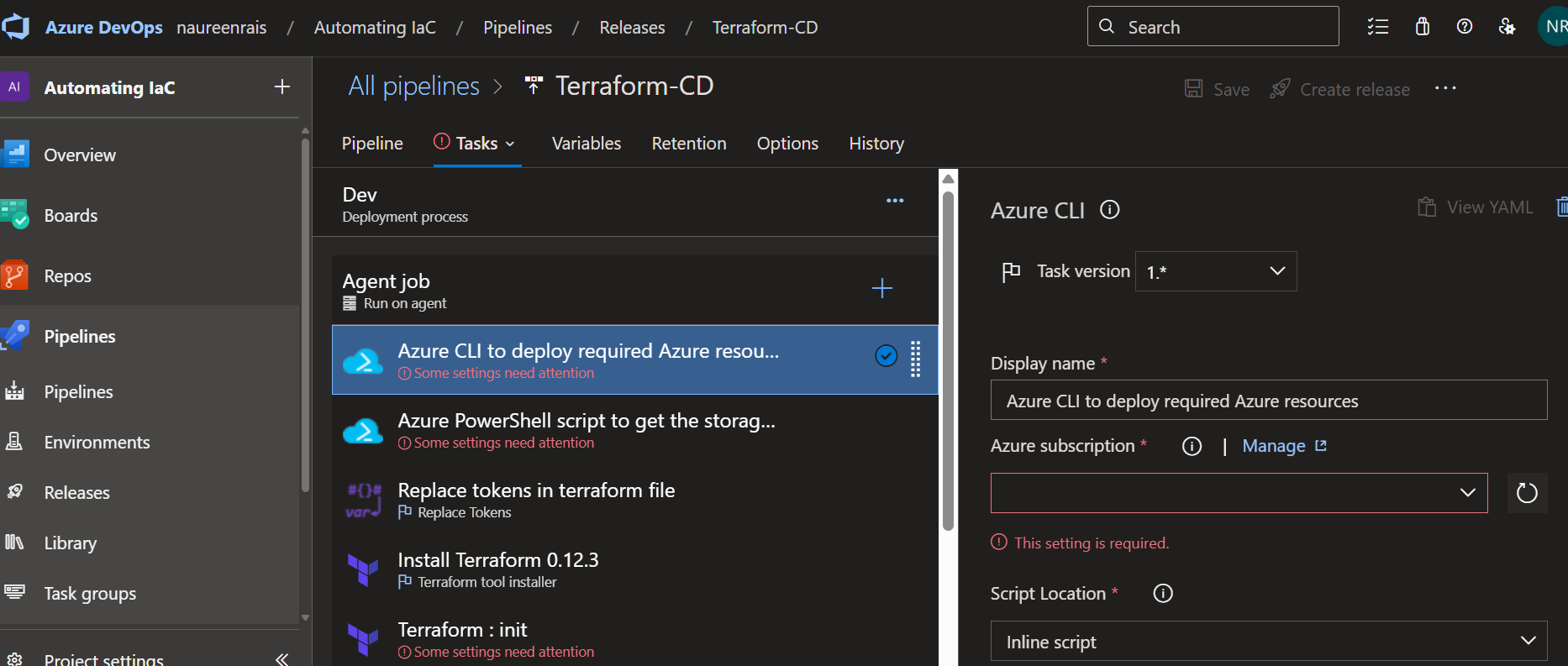


1. On the **All pipelines > Terraform-CD** pane, in the rectangle representing the **Dev**stage, click the **1 job, 8 tasks** link

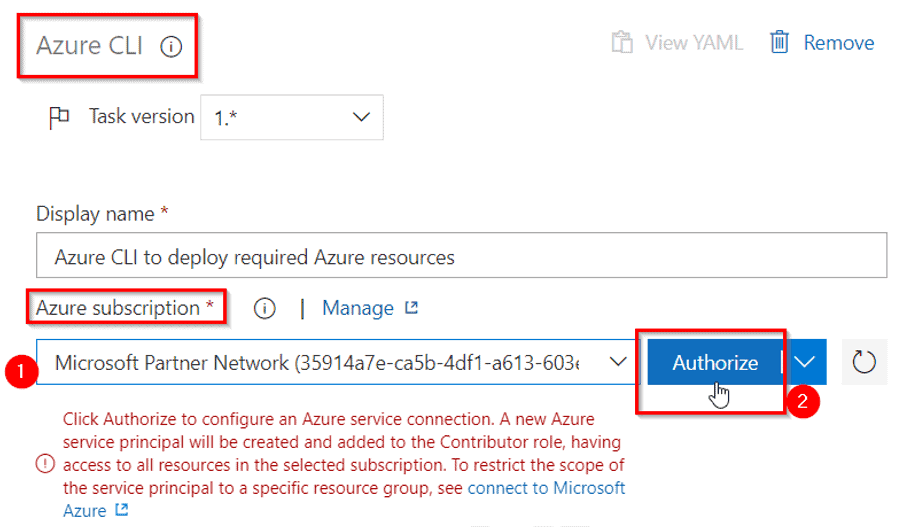
A screenshot of a computer

AI-generated content may be incorrect.

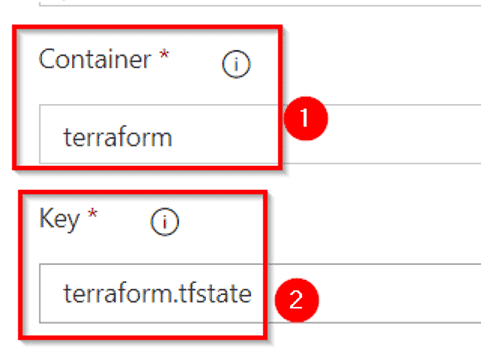
1. In the list of tasks of the **Dev** stage, select the **Azure CLI** **to deploy required Azure resources** task as it require some action.



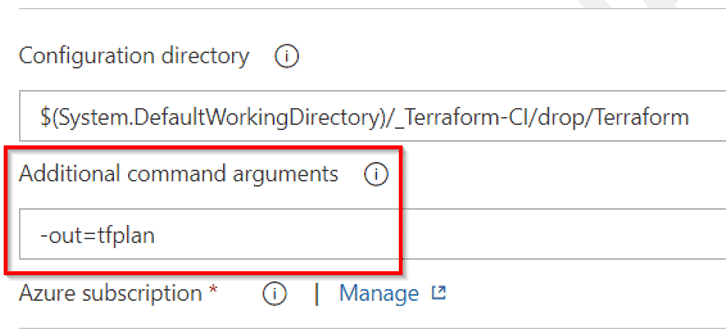
1. In **Azure subscription,**select the right one from the dropdown and click **authorize.**



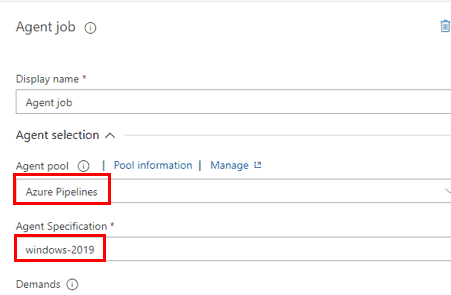
1. If asked, sign in using the account with the active Azure subscription.
2. Similarly, setup for other tasks showing **some settings needs attention** too but with **Azure** **service connection** from drop down.
3. In **terraform:init**pane after selecting Azure subscription as Service connection, go in **Container**drop down list type **terraform**and under **Key**parameter type **terraform.tfstate**



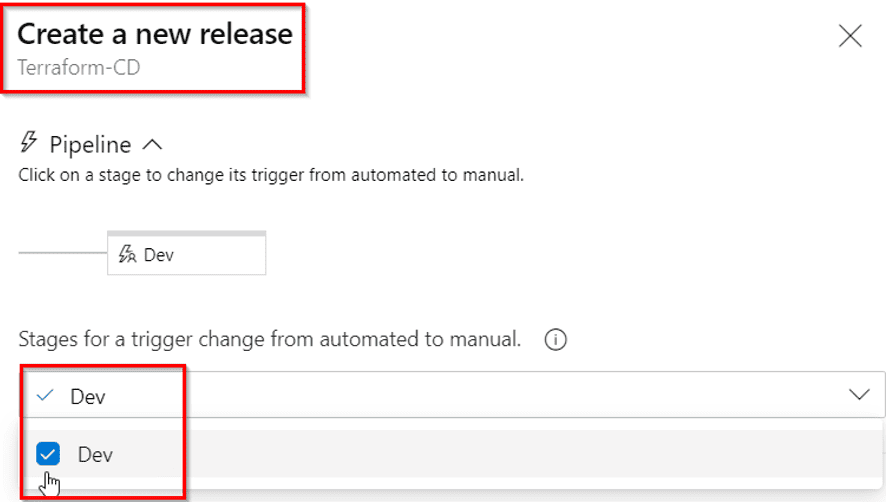
1. In **Terraform: plan,**select azure subscription as **Service connection**from drop down. In **Additional command arguments** text box,enter **-out=tfplan**



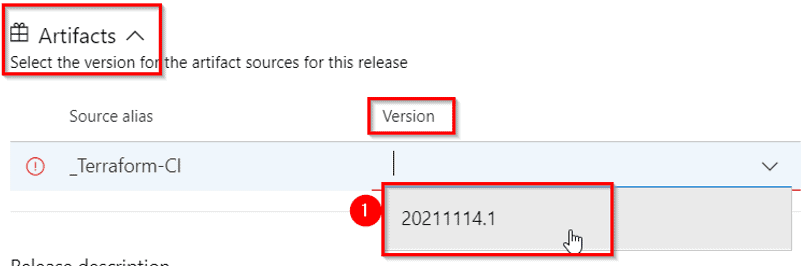
1. In **Terraform: apply -auto-approve** task, first select subscription as done in earlier task for **service connection**. Then under **Additional command arguements**enter **-auto-approve tfplan** by replacing **-auto-approve**
2. Under task **Azure app service deploy,** select the **Azure subscription**as **Service connection**available in drop down.
3. In **Agent job** available on top of tasks, select **Azure pipeline** as **Agent pool** and **windows-2019** as **Agent specification.**



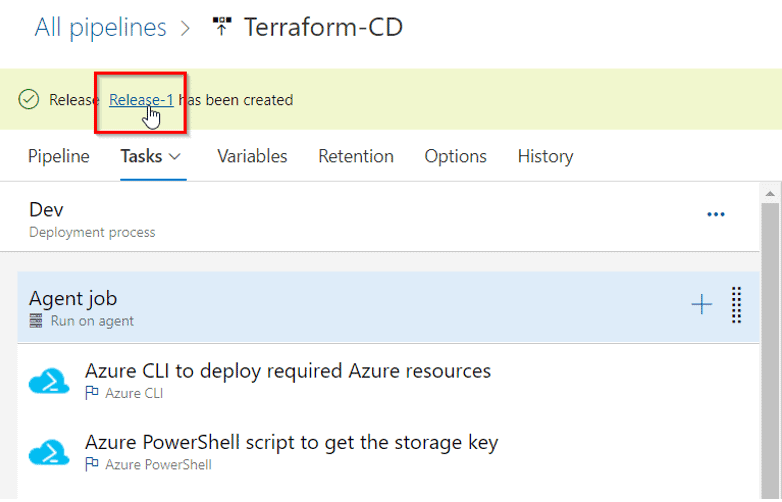
1. Now click on **Save**button available on top right corner to save the tasks. Then click on **Create release**available next to **Save** button.
2. Now a **Create new release**box will appear, under **Stages for a trigger change from automated to manual** dropdown list select **dev.**



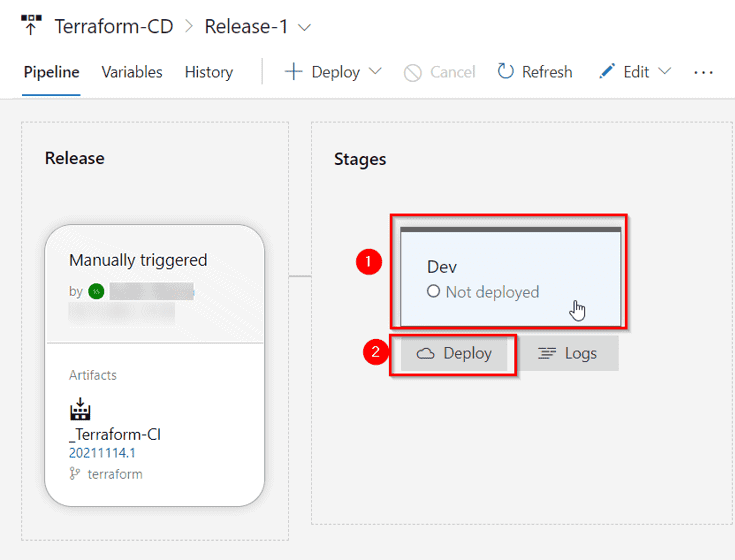
1. Now click on Save button available on top right corner to save the tasks. Then click on Create release available next to Save button.
2. Now a Create new release box will appear, under Stages for a trigger change from automated to manual dropdown list select dev.



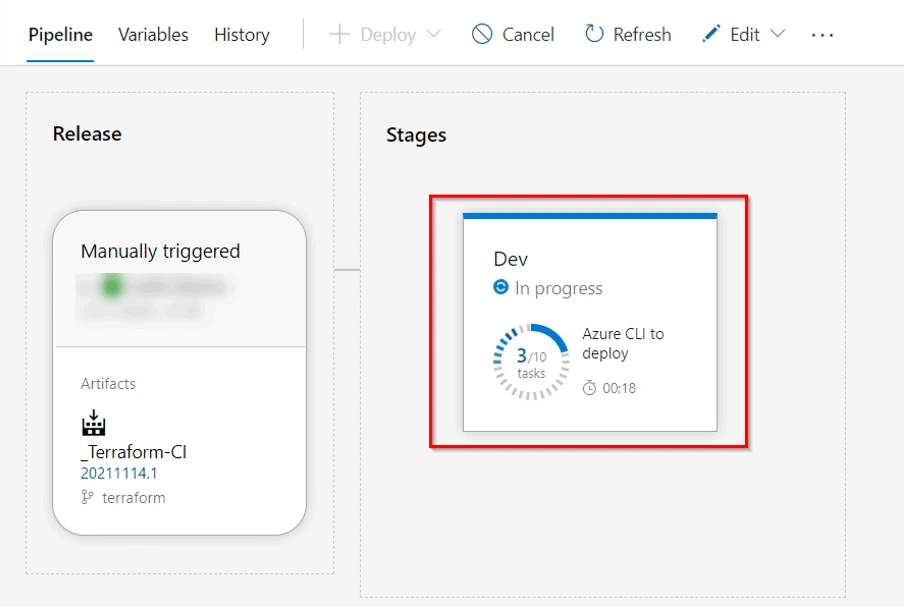
1. A **Release**entry will be shown on the same pane. Click on it to view the release



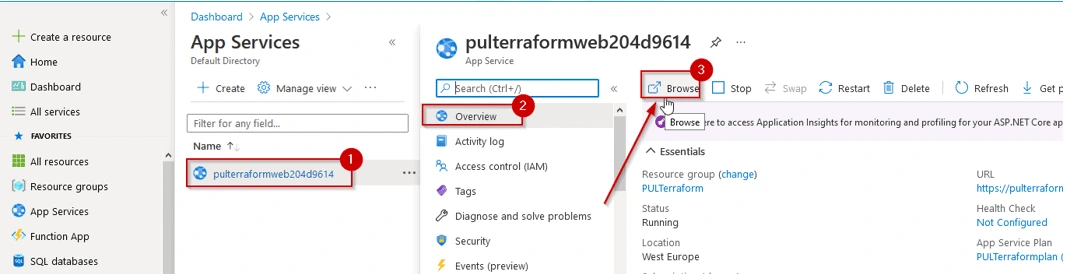
1. On the **Terraform-CD > Release-1** blade, click the rectangle representing the **Dev**stage, on the **Dev** pane, click **Deploy** and then click **Deploy**again

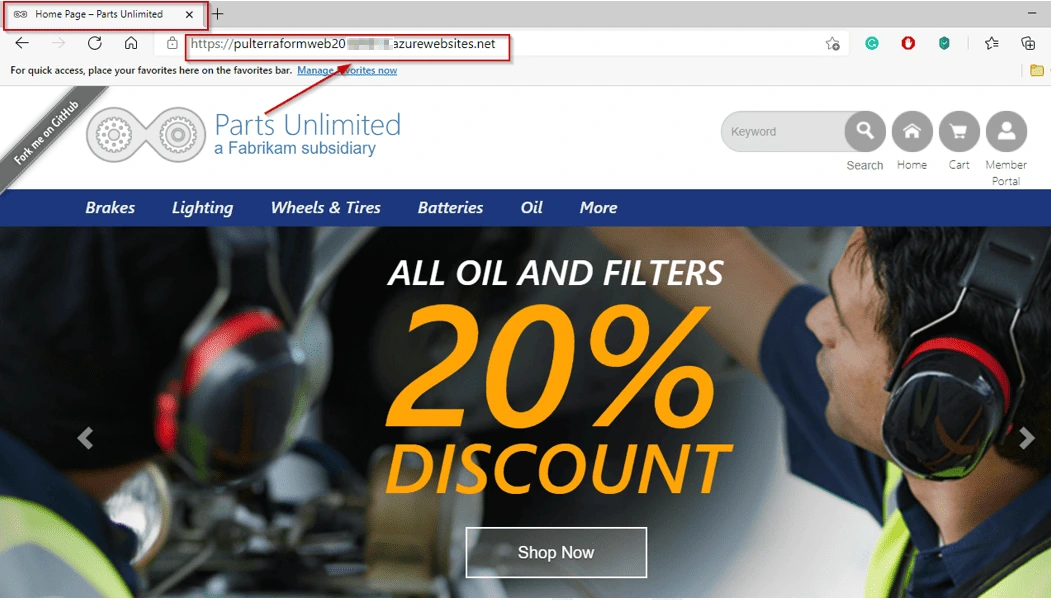


1. Now this will be deployed after process is completed. You can **Dev** to monitor stages

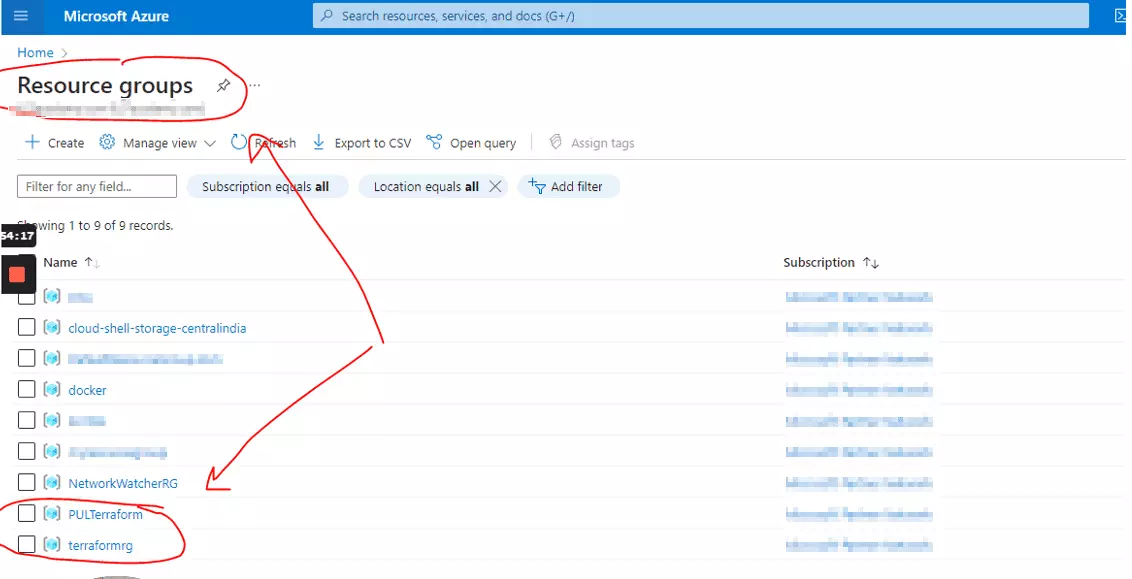
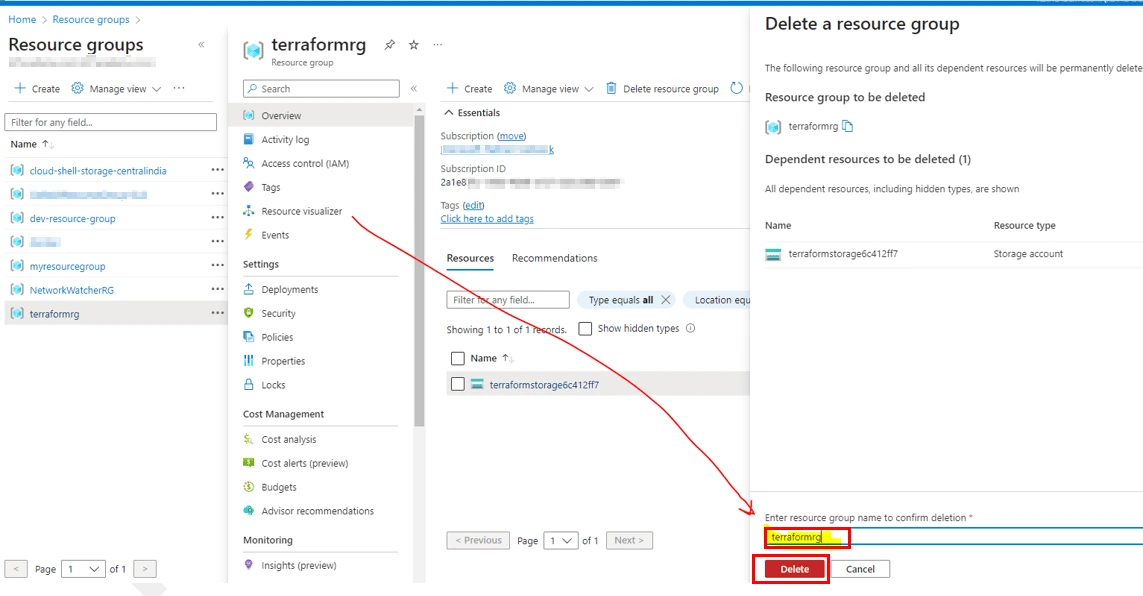
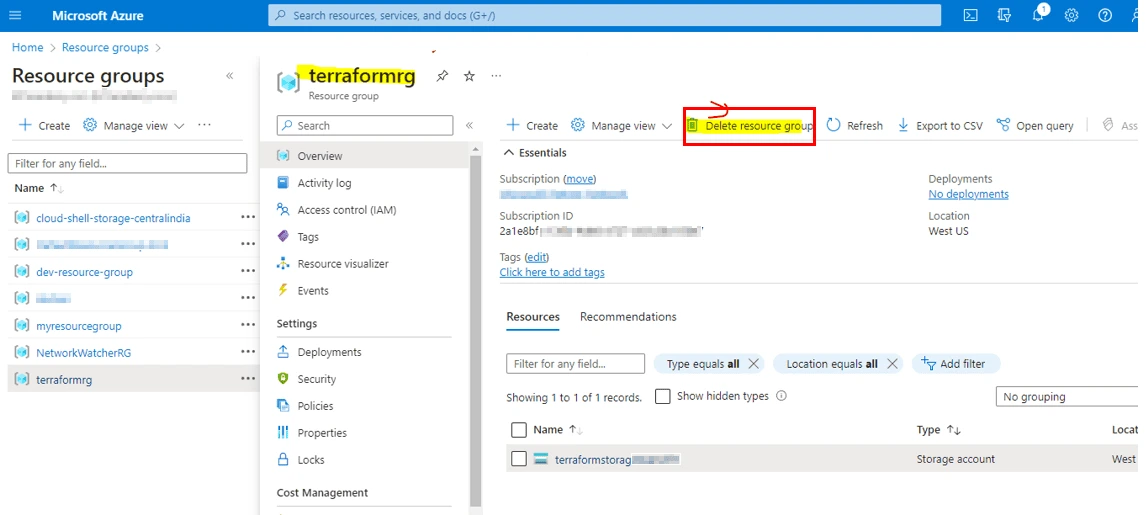


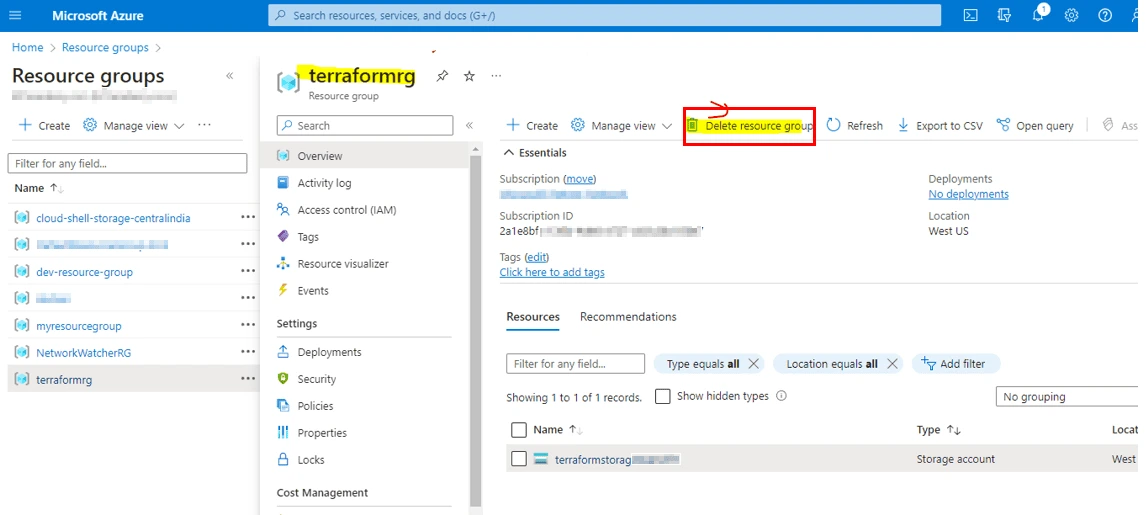
1. Now to check your deployment is successful or not, Simply go to [Azure portal](https://portal.azure.com/), in the search bar search for **App Services.** Navigate to app service with name **pulterraformweb.**Click on browse to view the web page.





**Cleaning the resources**

* 1. Since the resources are running so we need to clean them up so that it doesn’t eat up your credits. To clean the resources follow the steps below in your project:Search for Resource Group **terraformrg & PULLTerraform** and delete both of them.
  2. Select the resource groups, click on **Delete resource group->**Enter resource group ****name in the box->**Delete**

****

1. **Conclusion**

Automating infrastructure deployments with Terraform & Azure Pipelines can help to improve the reliability, efficiency, and visibility of your infrastructure deployment process. If you are looking for a way to automate your infrastructure deployments, I encourage you to check out Terraform & Azure Pipelines. So here we have learned how to deploy the infrastructure as a code by automating the process. The automation process took place with the help of Azure CI/CD process.

**Structural Diagram**

A computer screen shot of a diagram

AI-generated content may be incorrect.