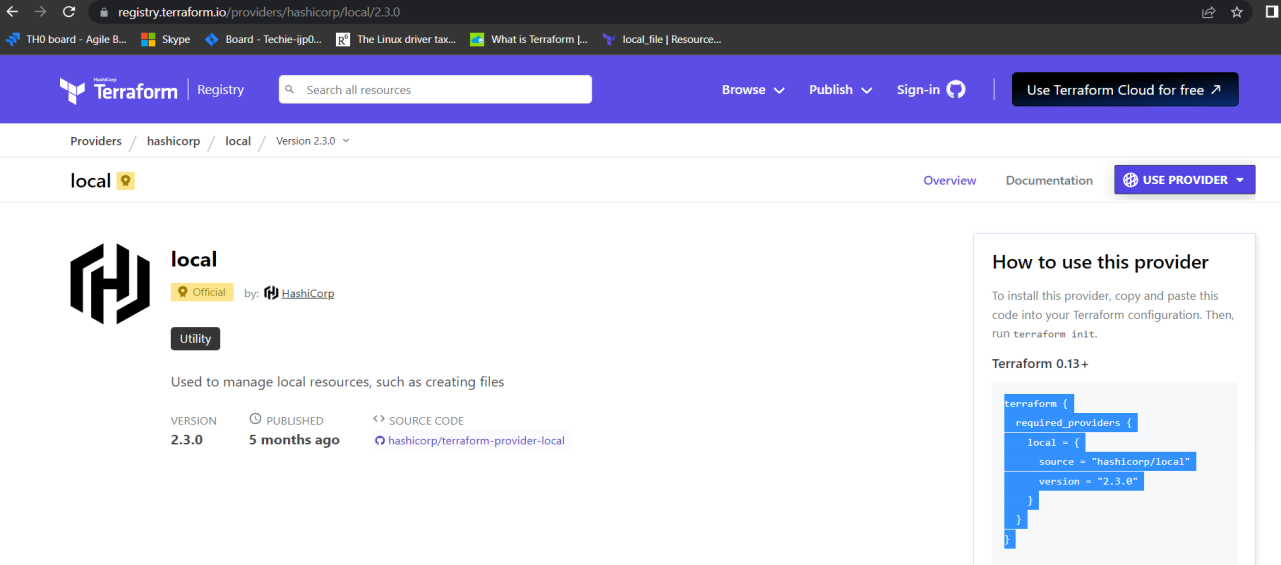
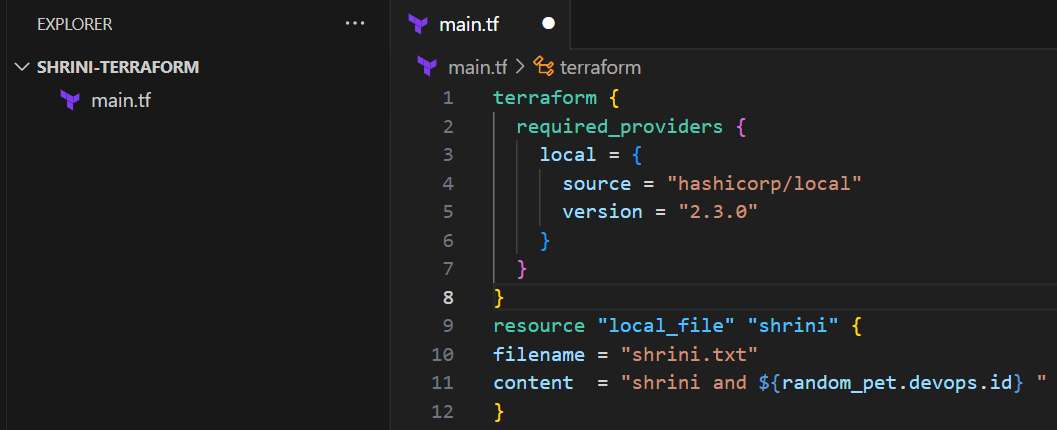
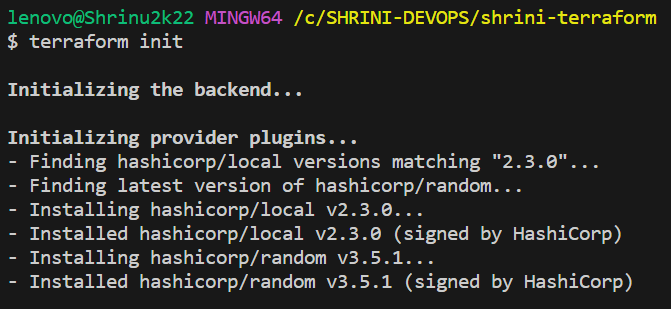
**Terraform-04**

**(versions, data source, meta arguments, Aws terraform)**

**Changing the version of provider:**

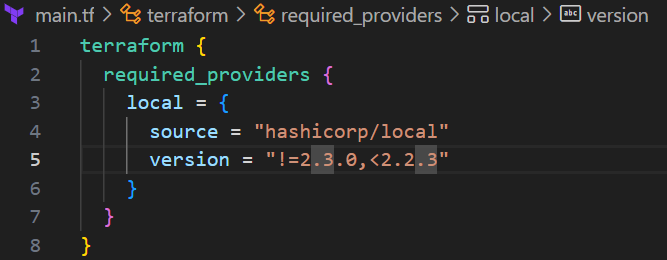
**Reason for changing the provider version:**

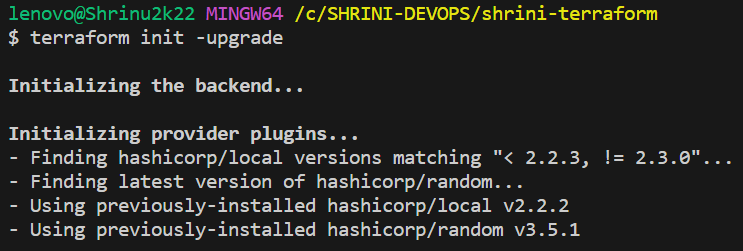
**If any application has some version dependencies like tomcat have specific version required then we have to manage the compatibility then we change the provider version like below:**

* **When-ever we execute “terraform apply” command then it will download the latest provider versions**
* **Then we have search for the provider and select the required version of provider  
  **
* **Copied the provider version block and paste it to the main.tf   
  **
* **And run the “terraform apply” command it will download the mentioned version in main.tf  
  **

**Version constraints: (terraform init –upgrade)**

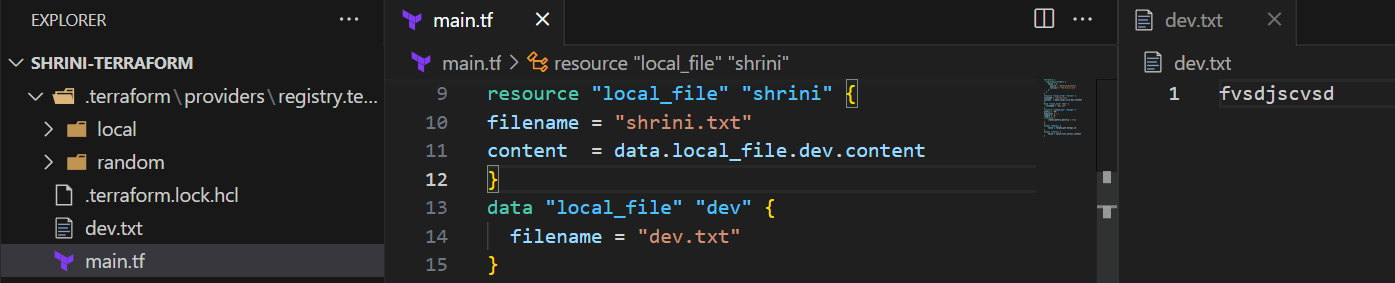
1. **version = "2.3.0" --> download the exact version**
2. **version = "!=2.3.0" --> will not use the mentioned version**
3. **version = "< 2.3.0" --> lesser than the mention version**
4. **version = "> 2.3.0" --> greater than the given version**
5. **version = "~> 2.3.0" --> specific version or higher version**

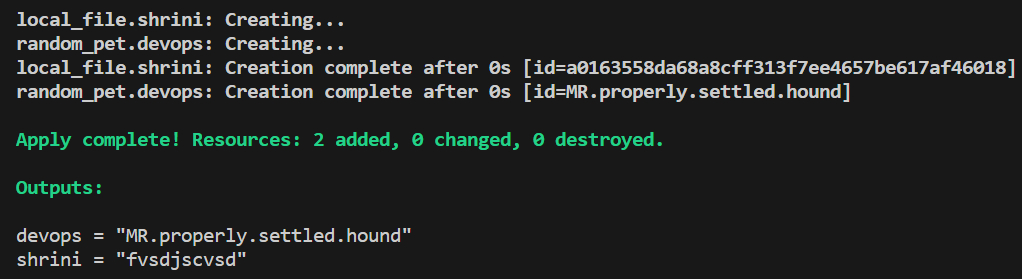
****

****

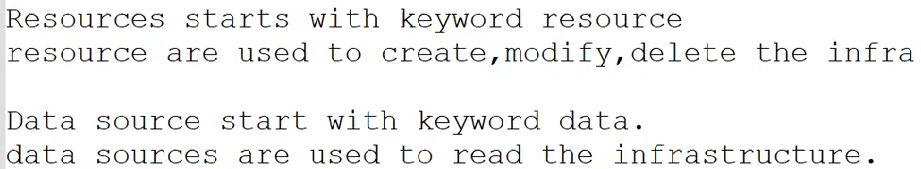
**DATA-SOURCES:**

**It is used to read the content of the infrastructure  
example: If any file created in terraform directory and want to use that file with the help of data source we can do that**

****

****

**Difference between data-source and resources:**

****

**Meta-arguments:**

**Meta arguments are used if we want to create multiple resources.**

**Meta arguments can be used within any resource block to change the behavior of the resources.**

**Examples for Meta arguments:**

**1) Depends on**

**2) Lifecycle rules**

**3) Count**

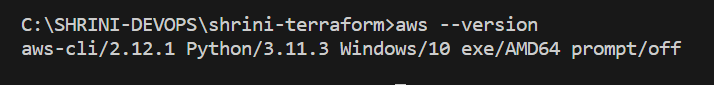
**4) For each**

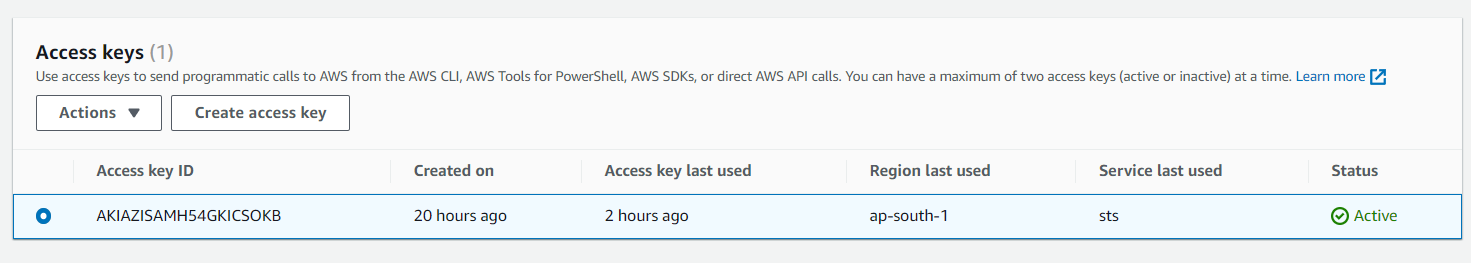
**AWS-TERRAFORM**

**Creating IAM-USER with the help of terraform:**

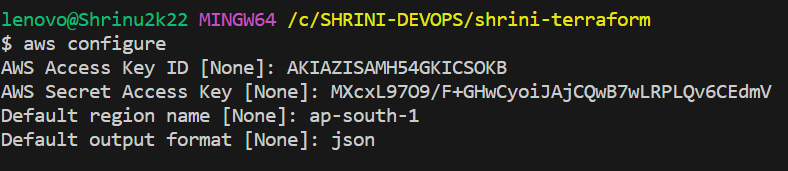
**For that we have to 1st install aws cli and then we can configure our aws accn to our terminal**

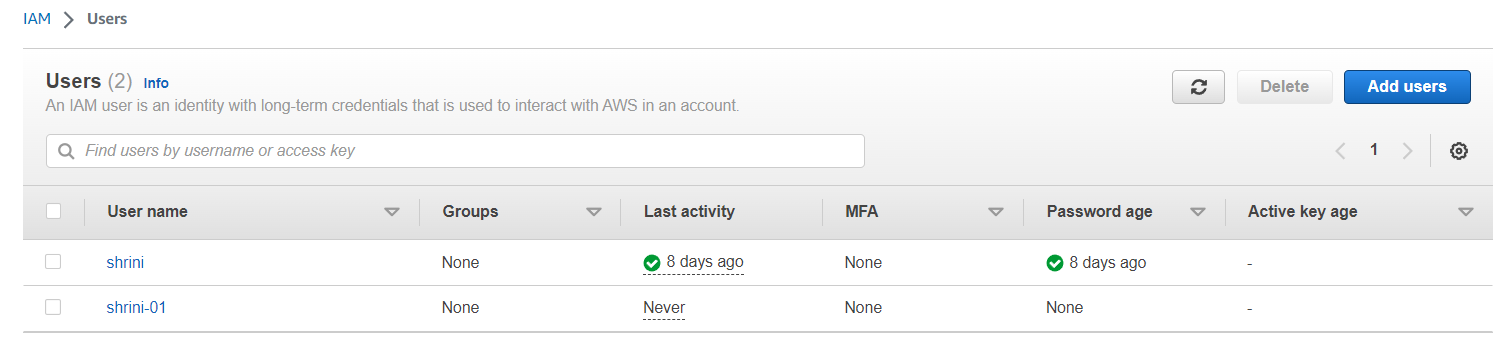
**We can follow AWS official link to download AWS CLI**[**https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html**](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html)

****

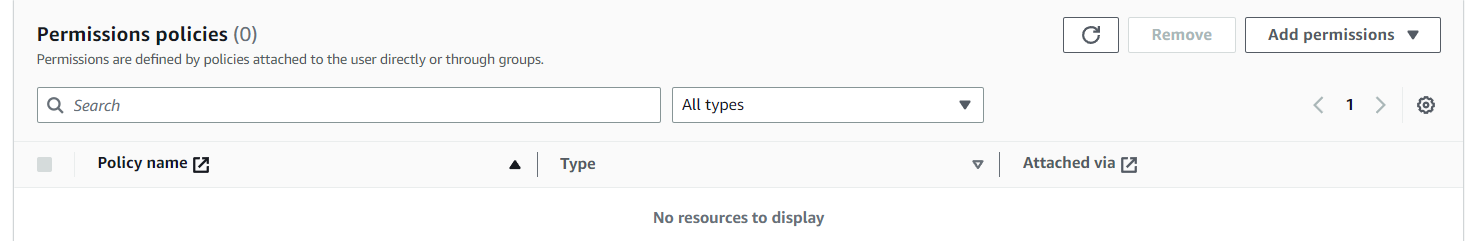
****

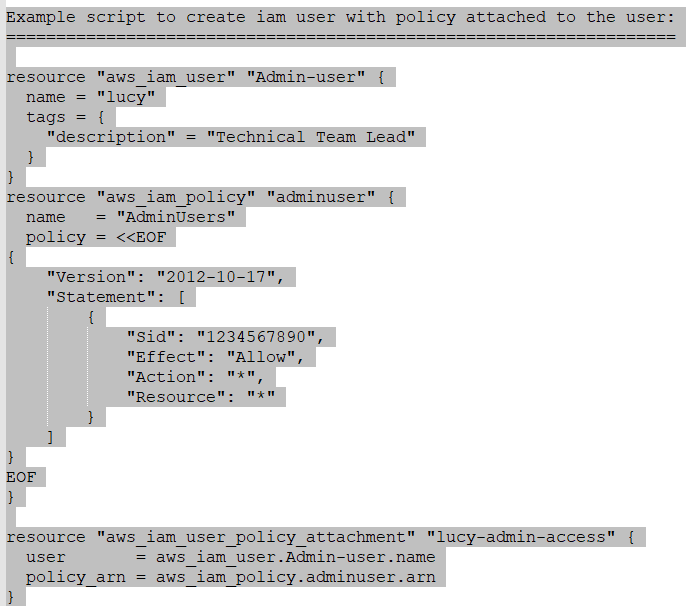
**“When above created access key get deactivated then aws configure will not work”**

****

****

**Now, we will attach some policies to our created AWS-IAM-user with the help of terraform:**

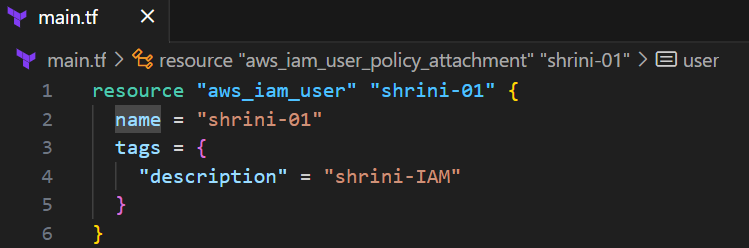
****

****

**{We have added the json policy using,  
 "heredoc syntax" and delimeters "EOF --> End of file" inside the main.tf.}**

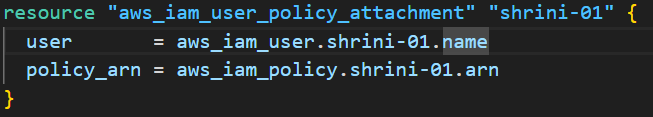
**\*\* Here I am giving three resources;**

**1st resources: we are creating user and with tag:**

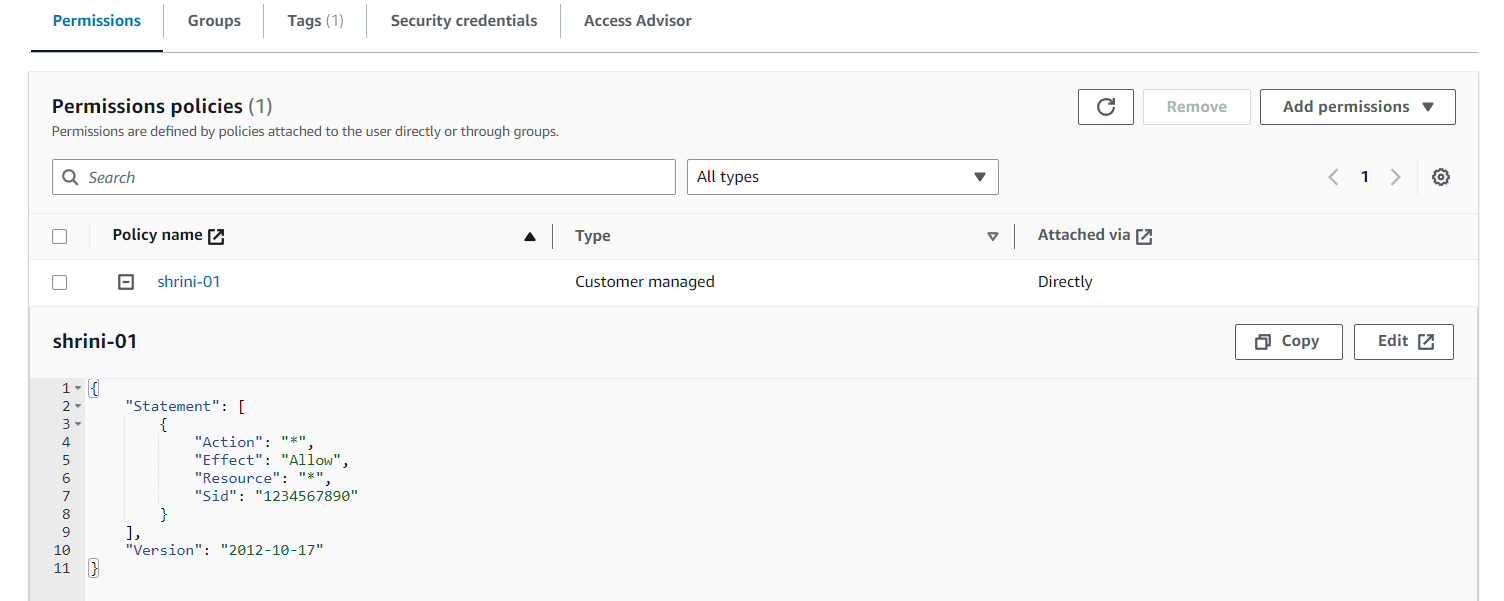
****

**2nd resources: we are creating a policy with the help of delimeter and “Heredoc-syntax”**

**3rd resources: we are attaching the policy to user with the help of arn**

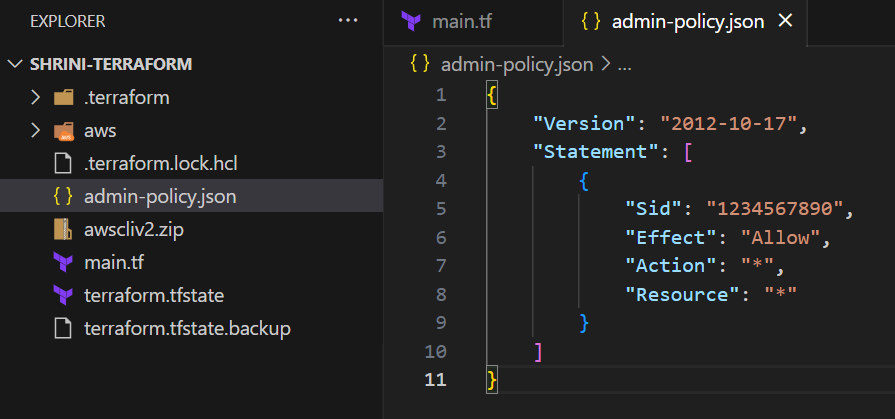
****

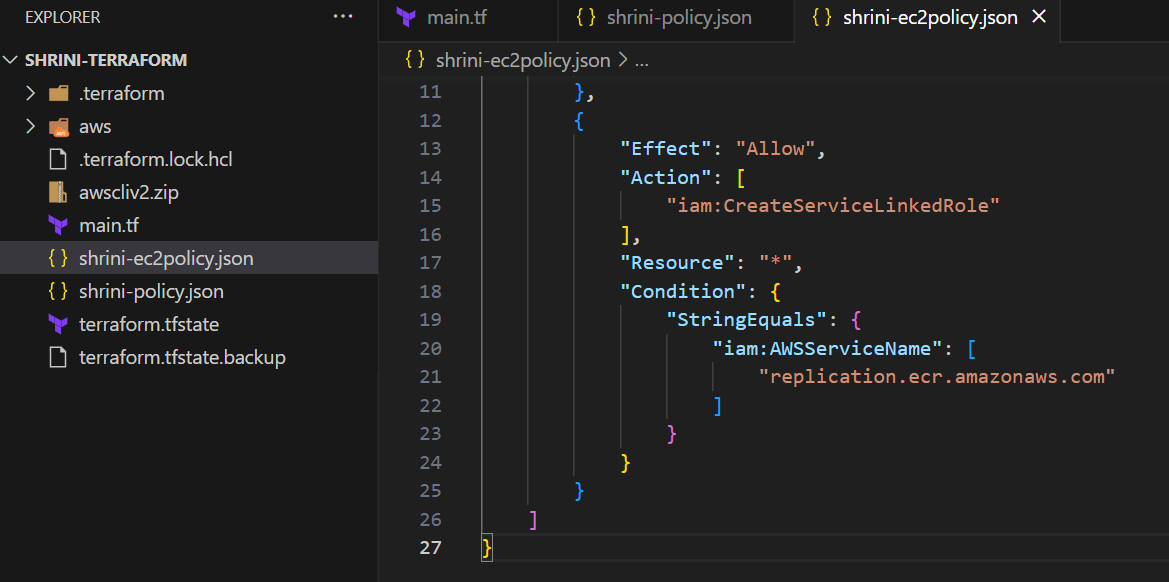
**So, the policy is get created and attached to user with the help of terraform :**

****

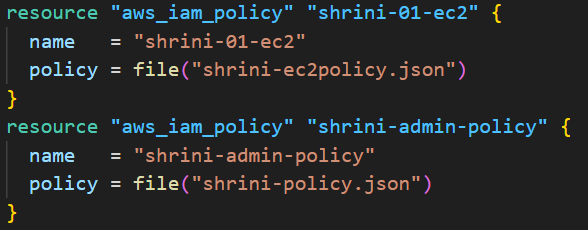
**“We can also use them by saving the template in separate file and call that file in our main.tf”**

* **First we have to create a file with logical-name and json extensions:**

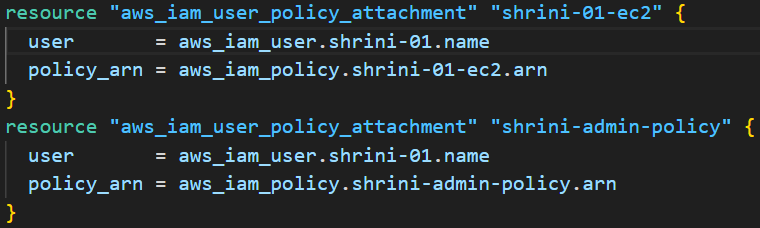
****

****

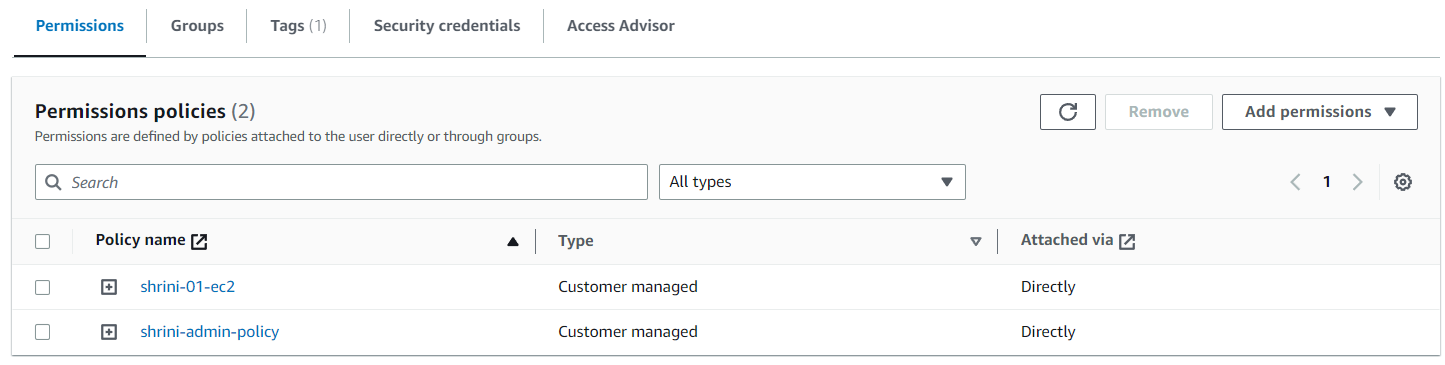
* **Then we will attach these polices with the IAM-user:**

****

* **And in last we are creating “arn” attribute for attaching the policy, For each policy we have to attach the policy by the help of arn:**

****

* **IN AWS:**

****