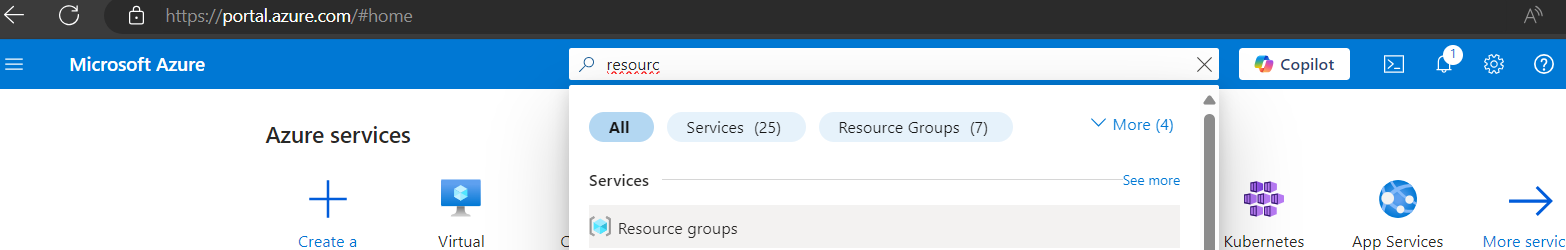
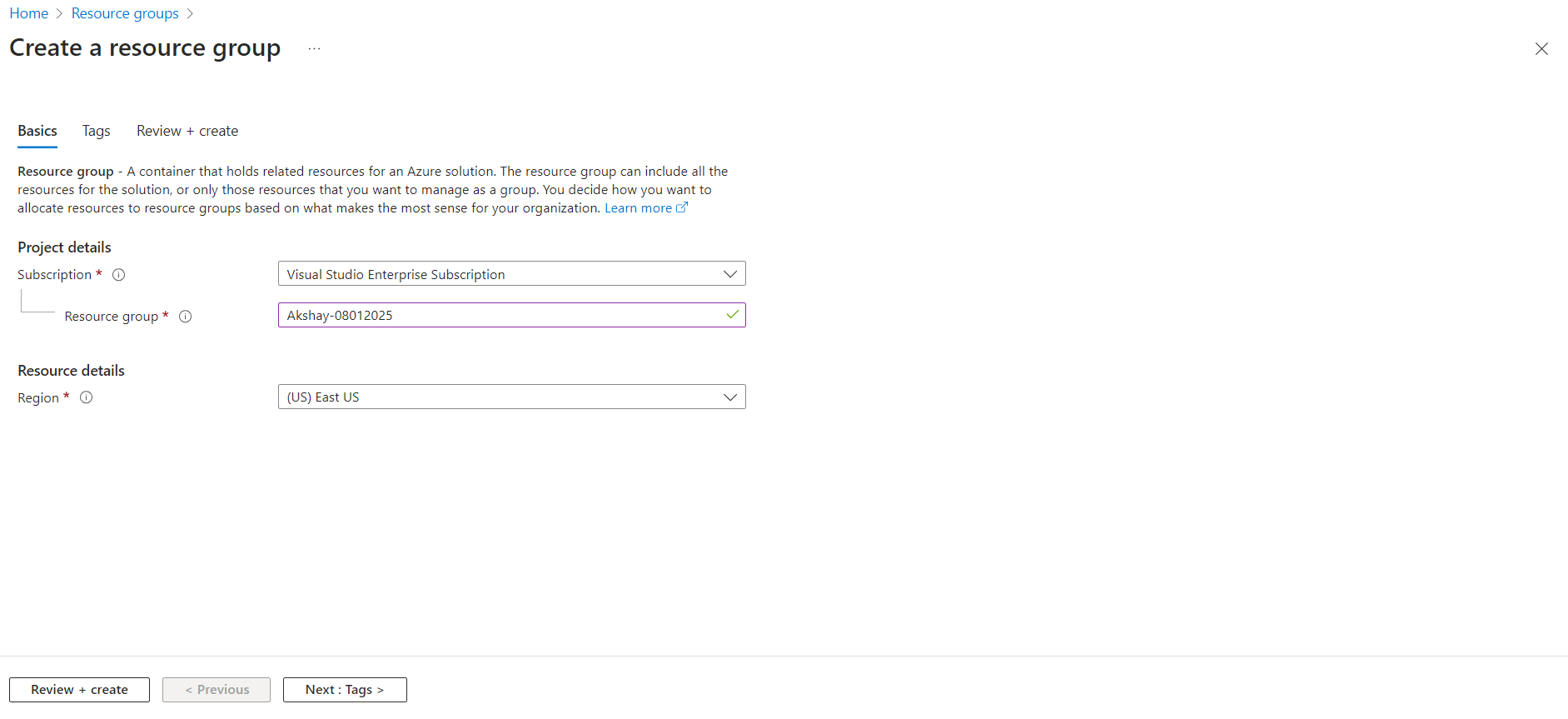
Backend CI CD using Azure DevOps (YAML approach)

step 1: Create a resource group which we will be using throught our Practice

* Go to Azure portal and search for 'Resource Group'



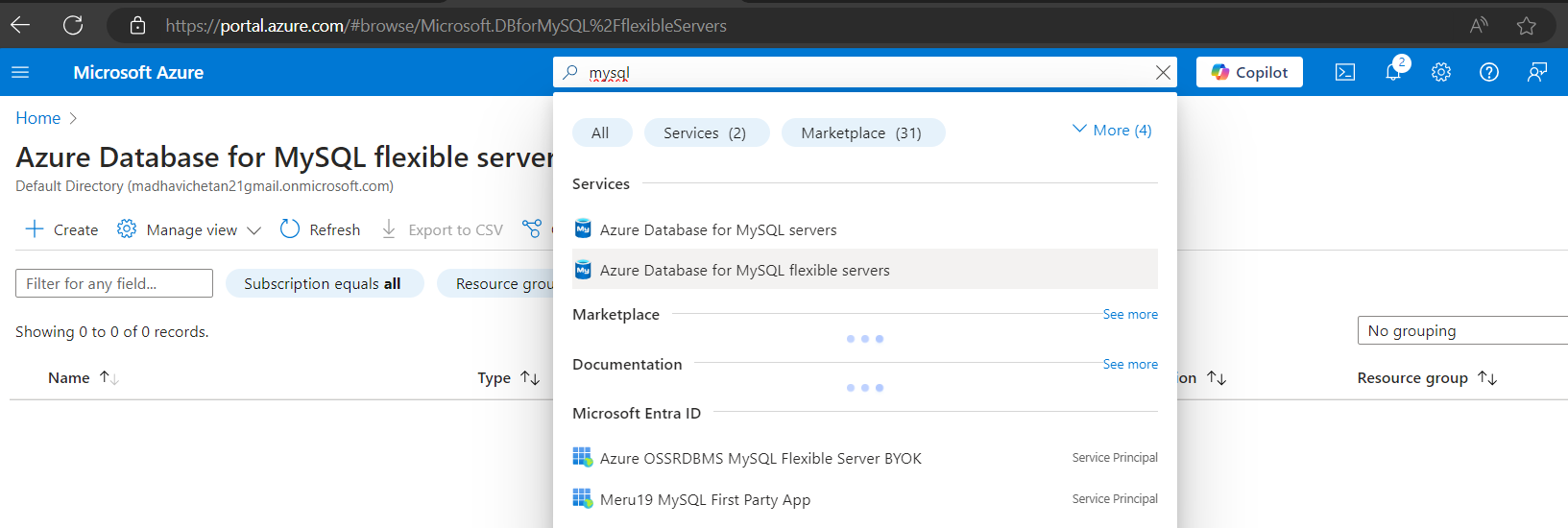
* Click on Create
* Follow below pattern for naming convention and make sure you keep region as East US Firstname - DDMMYYYY



* Click on Review + Create
* Click on Create.

step 2: Create a Azure Database for MySQL flexible servers for our strapi backend

* Go to Azure portal and search for 'MySQL'



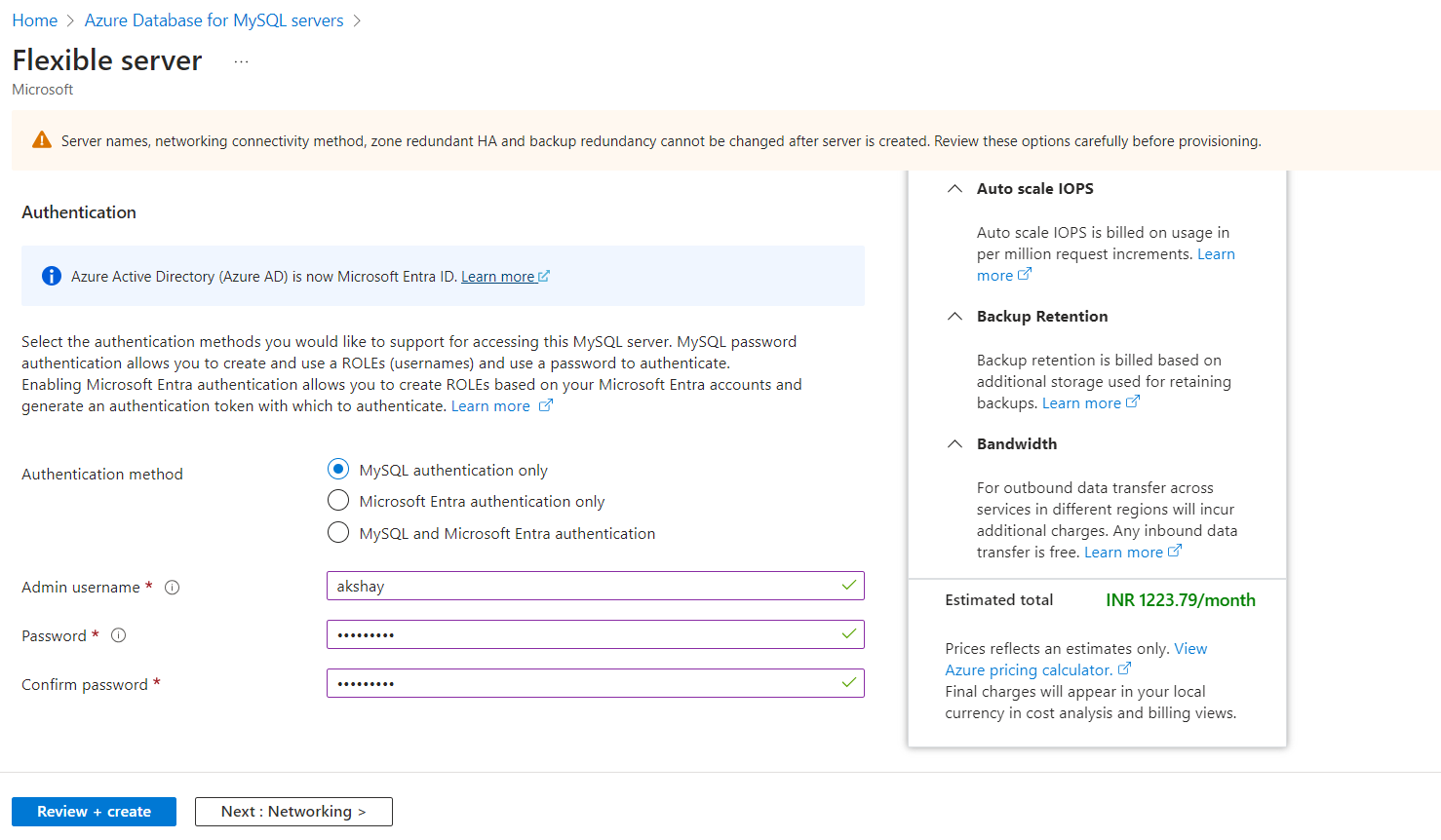
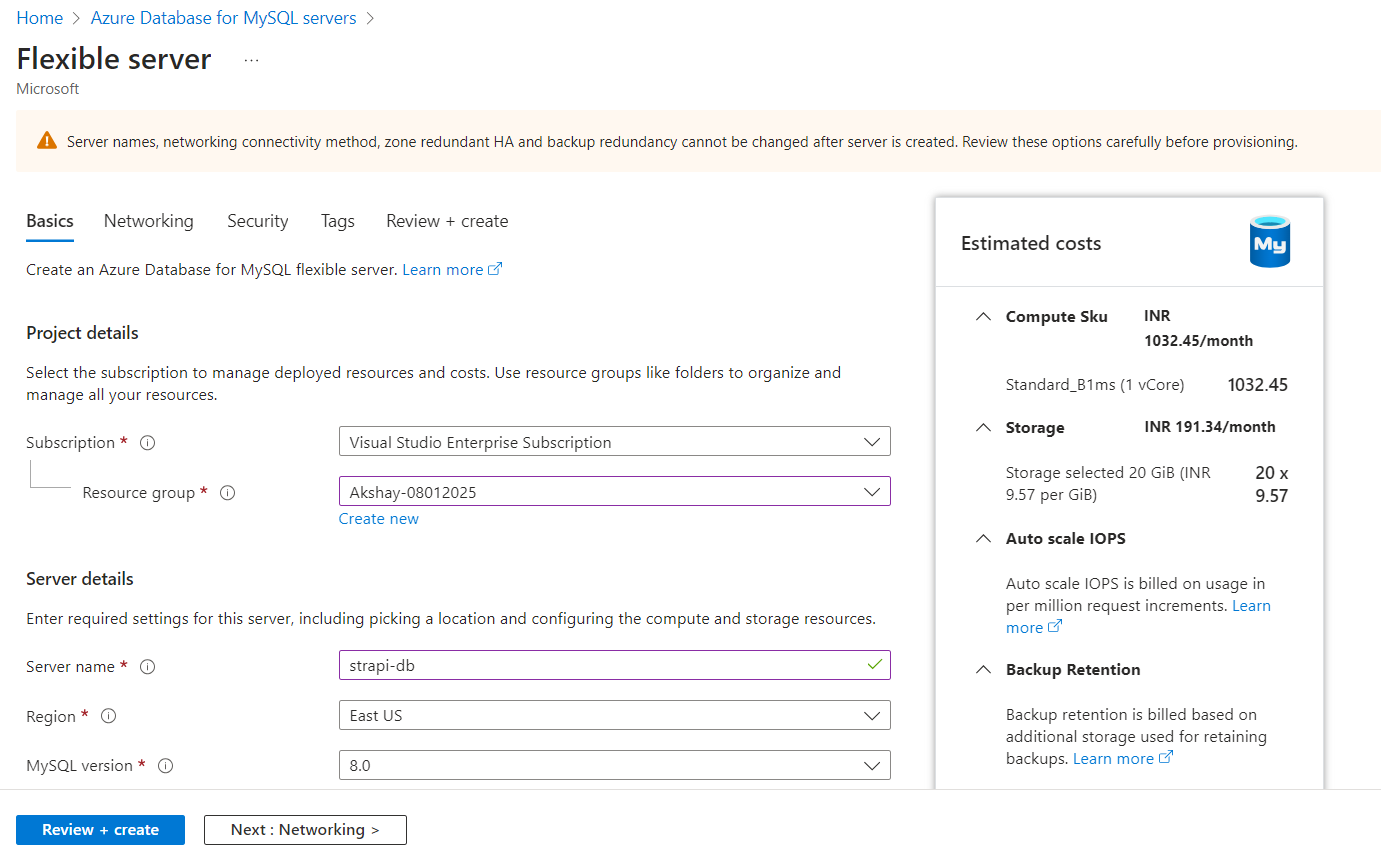
* Click on Create, choose Flexible server and choose 'Advance'
* now provide 4 fields

Server-name : 'strapi-db'

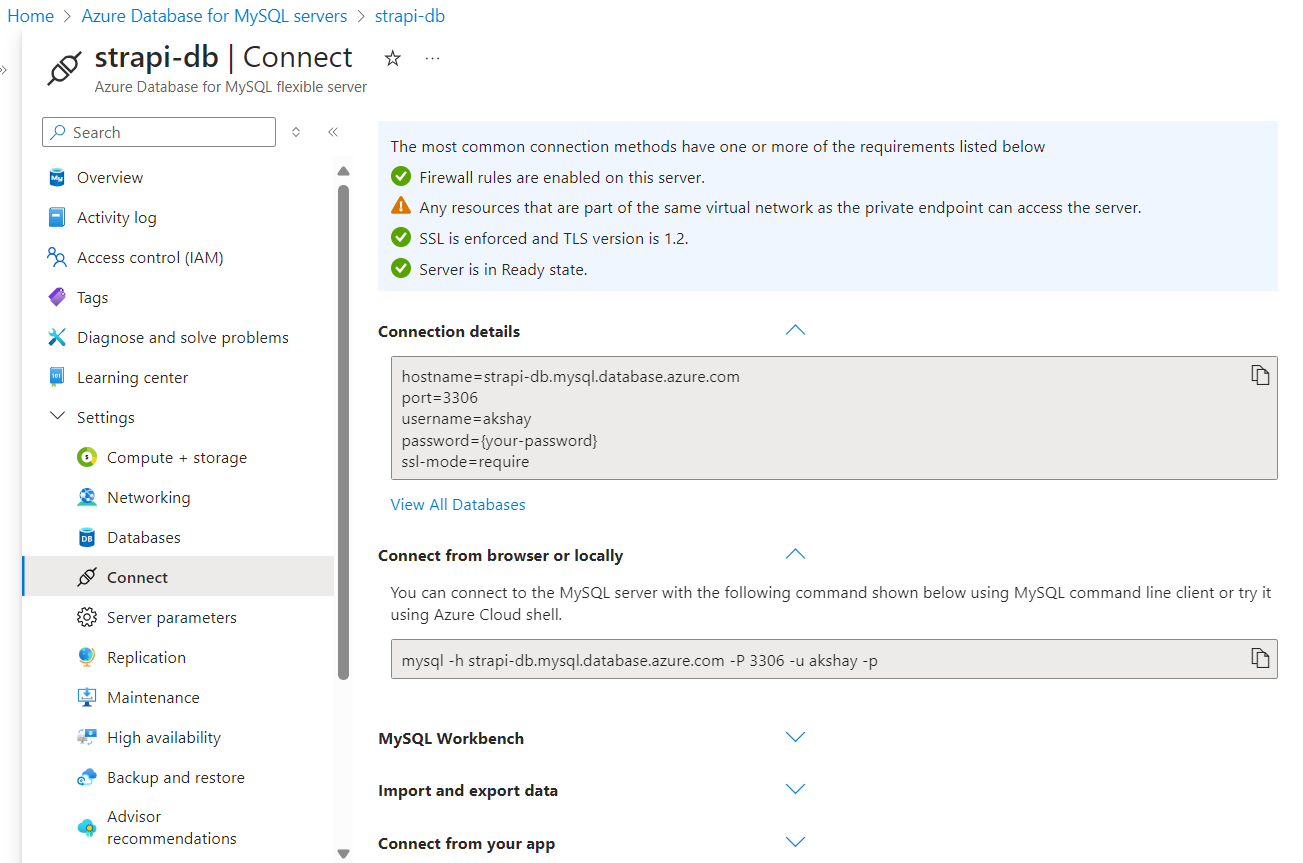
Admin username: 'yourfirstname'

password: 'Admin@123'

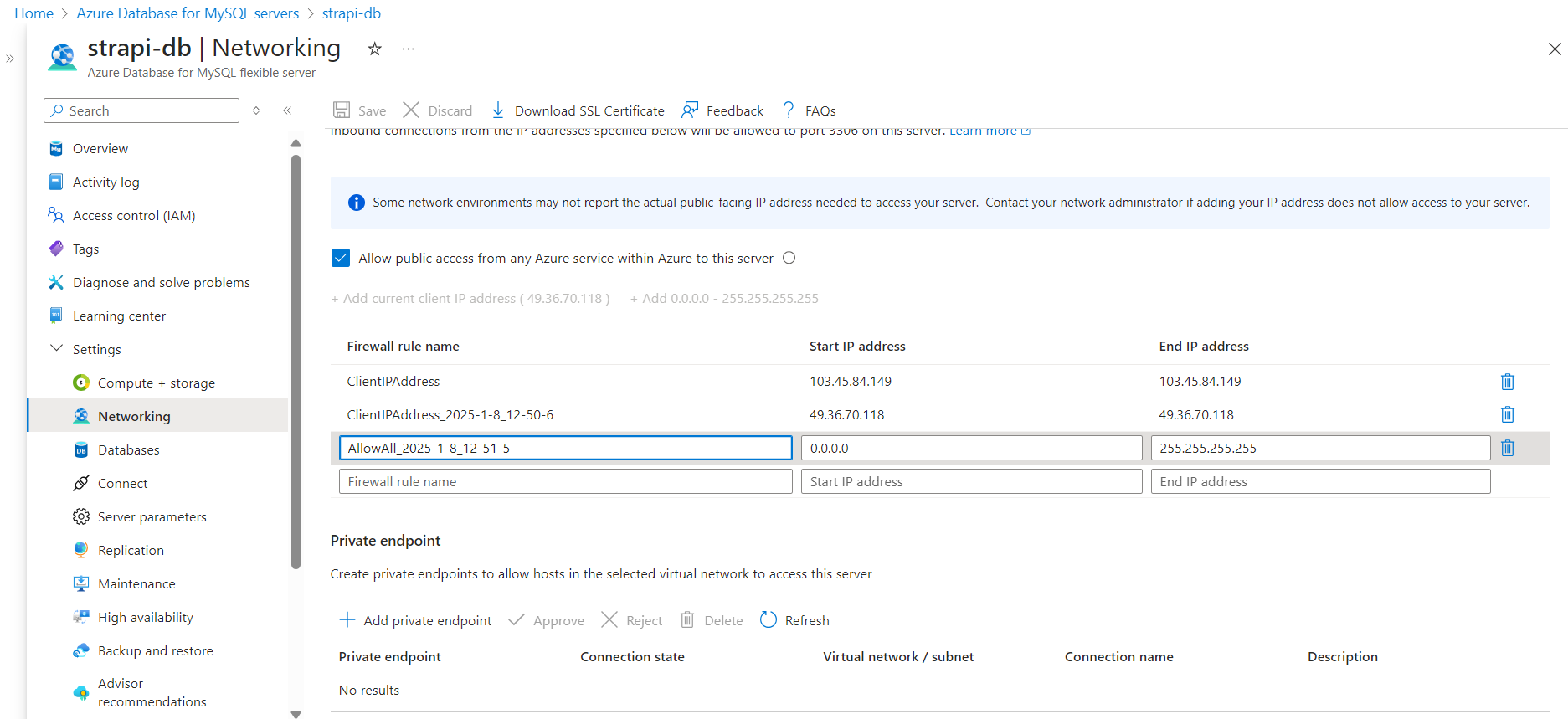
confirm password: 'Admin@123'



* now click on 'Review + create' and then 'create'
* keep all the fields as default
* once database is created go to settings -> connect option and copy all the details



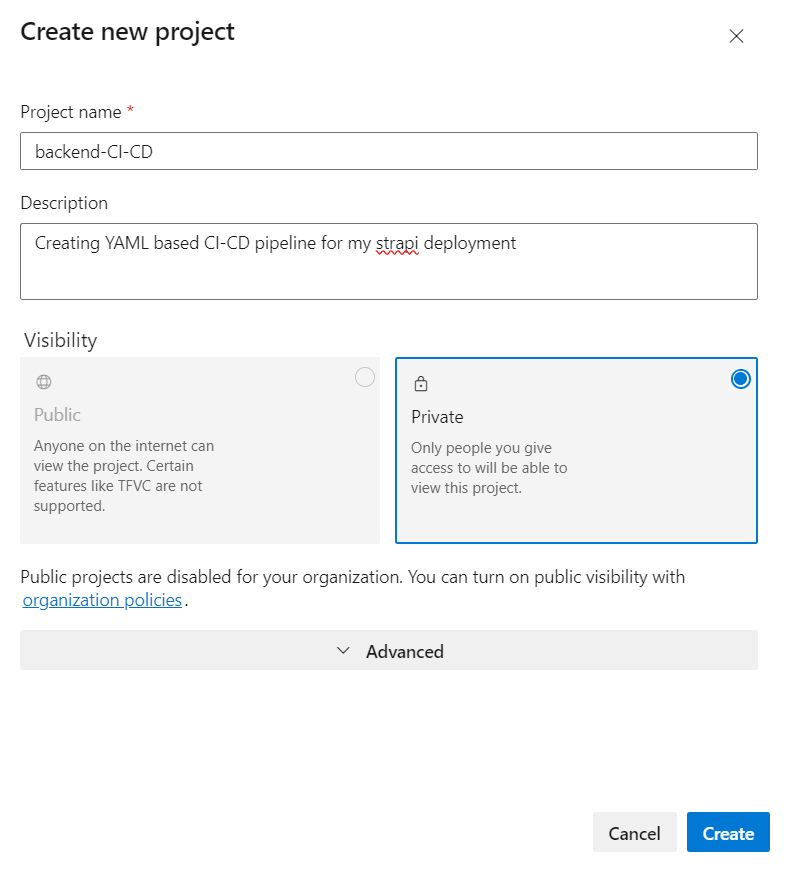
* make sure your public access is given to your database. (In production provide access to single instance only)



step 3: Setting up DevOps project and configuring self host VM

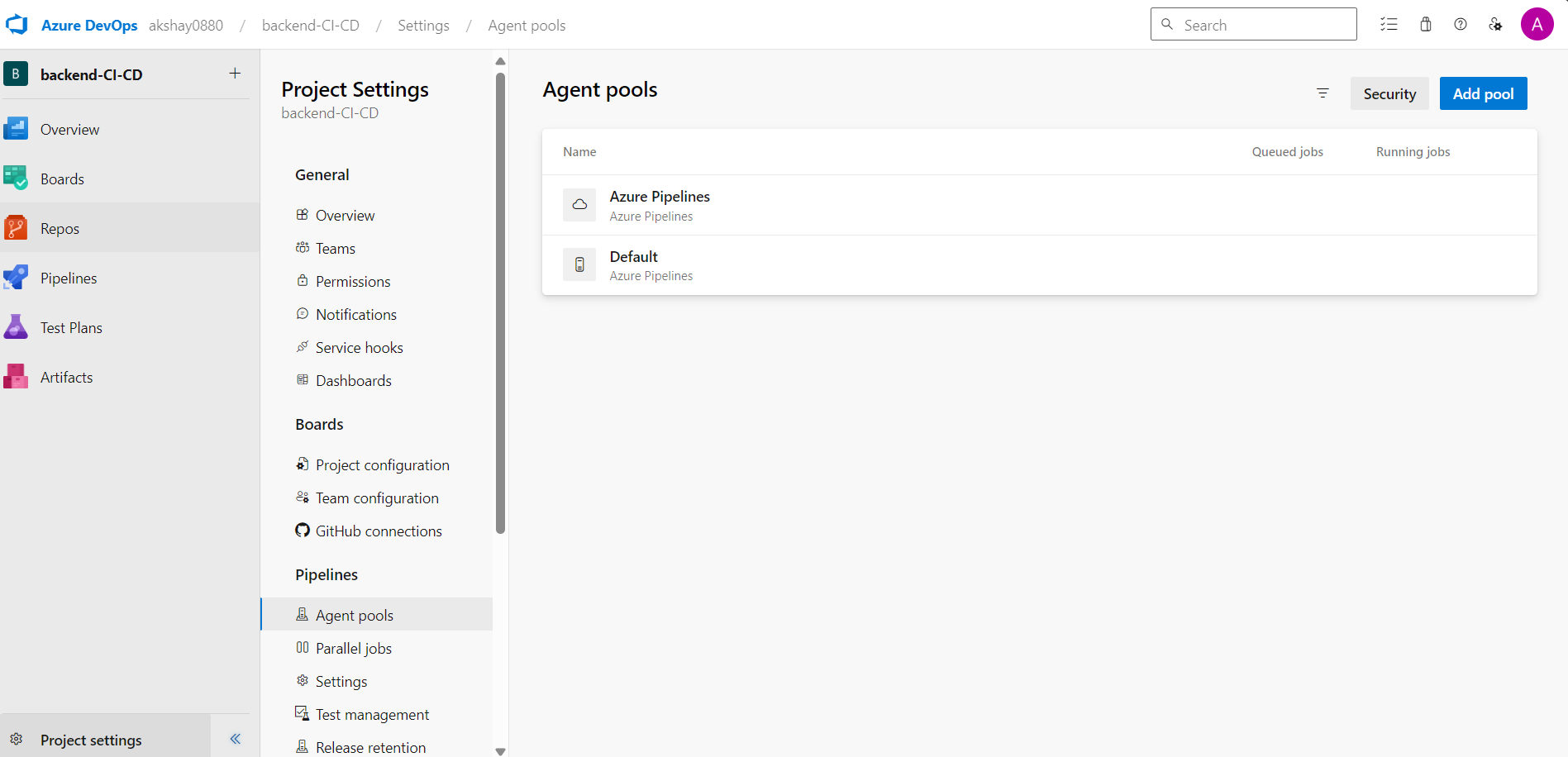
Task 1 : Set up DevOps Project

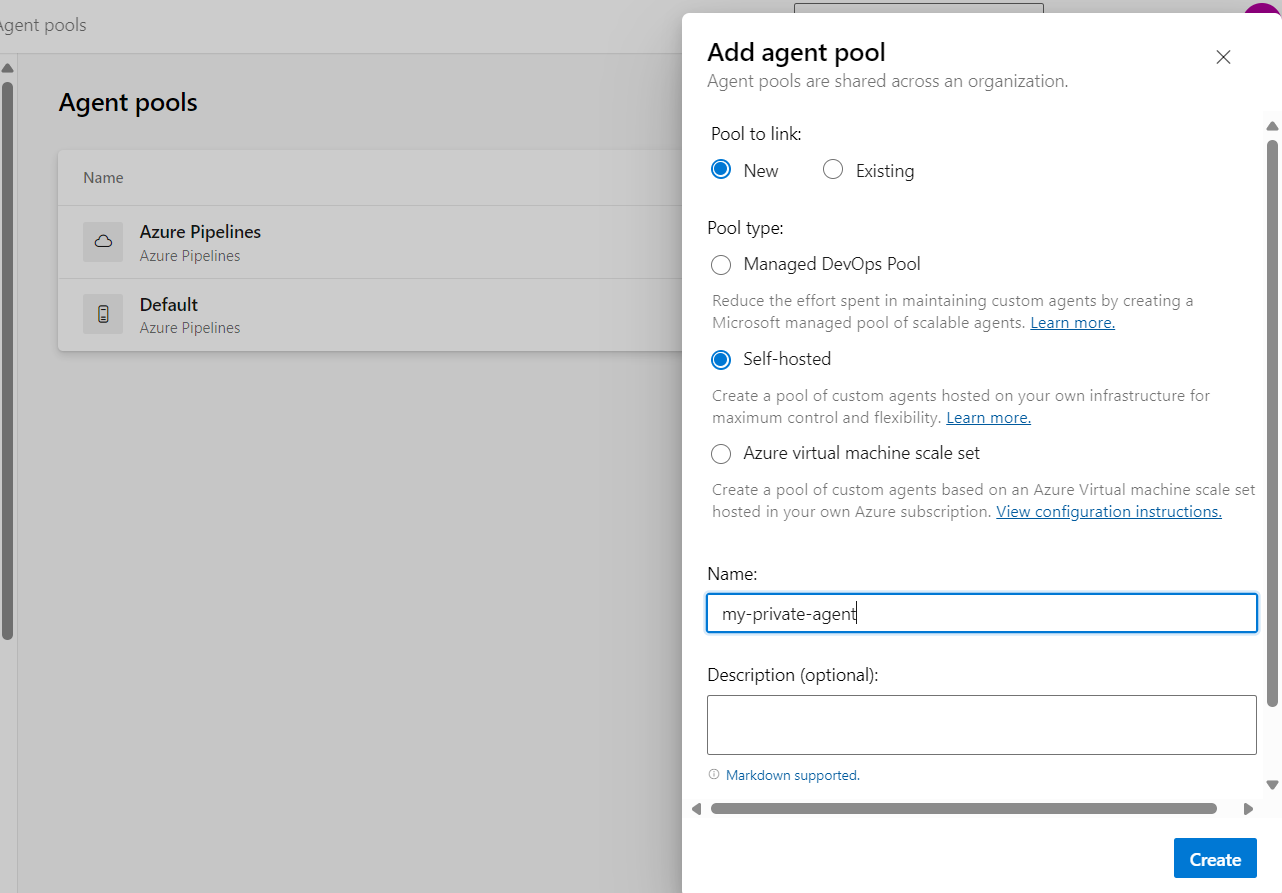
* Go to Azure Portal and search for 'Azure DevOps Organization'
* then click on 'My Azure DevOps Organizations'
* now select your organization or create new one if you dont have.
* Now click on ' + New Project '



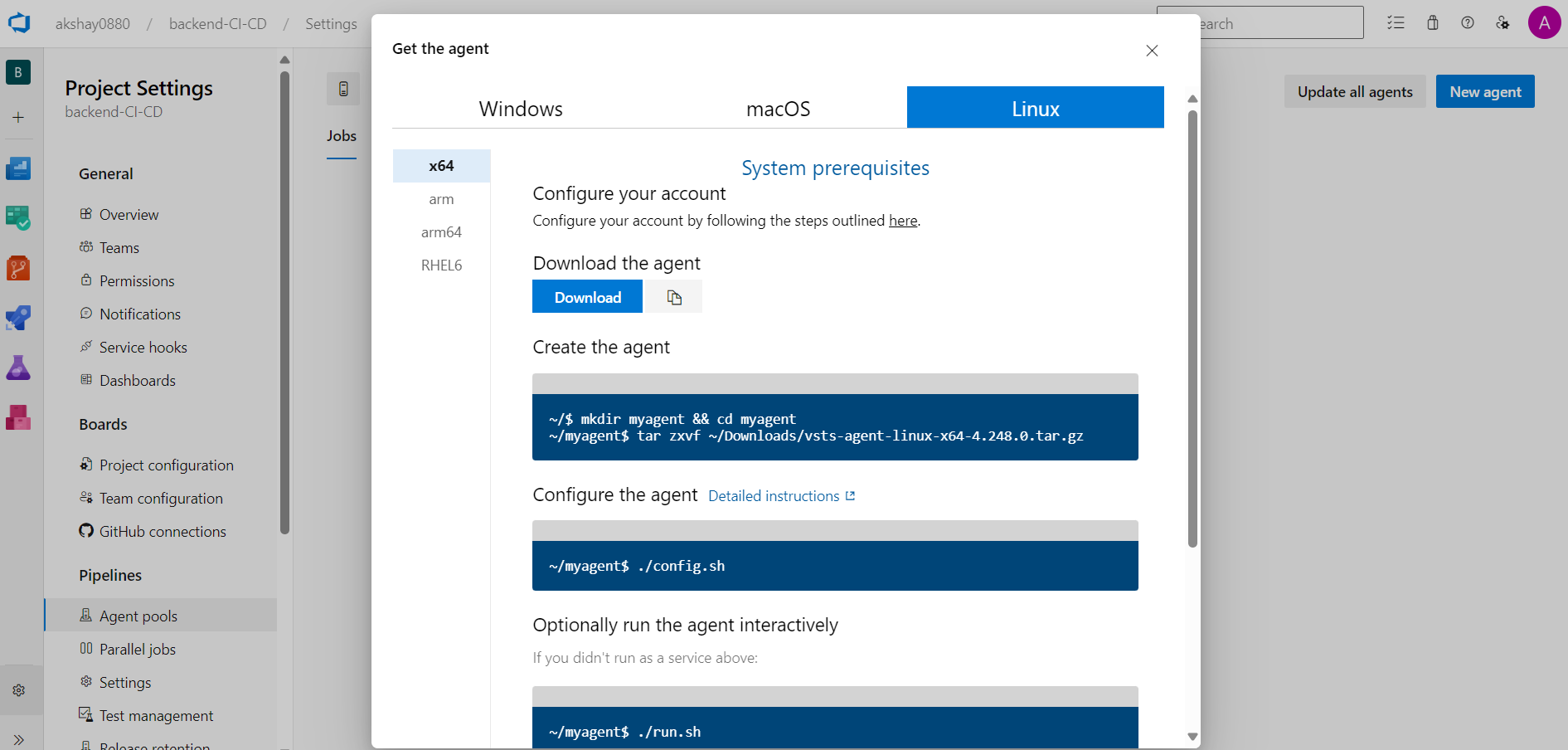
* Then click on 'Create'.

Task 2 : Create a VM and connect

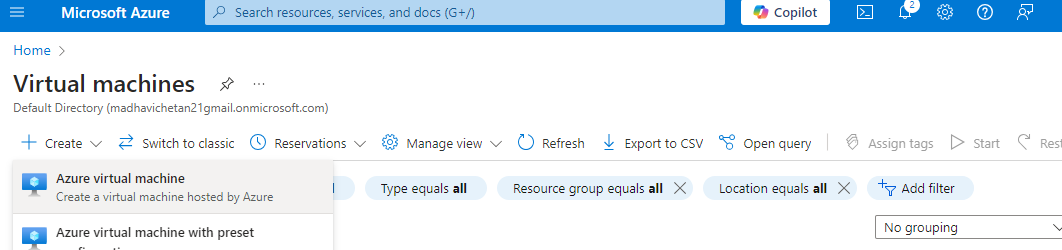
* Now go to Project settings -> Agent pools and click on Add pool
* now add below details



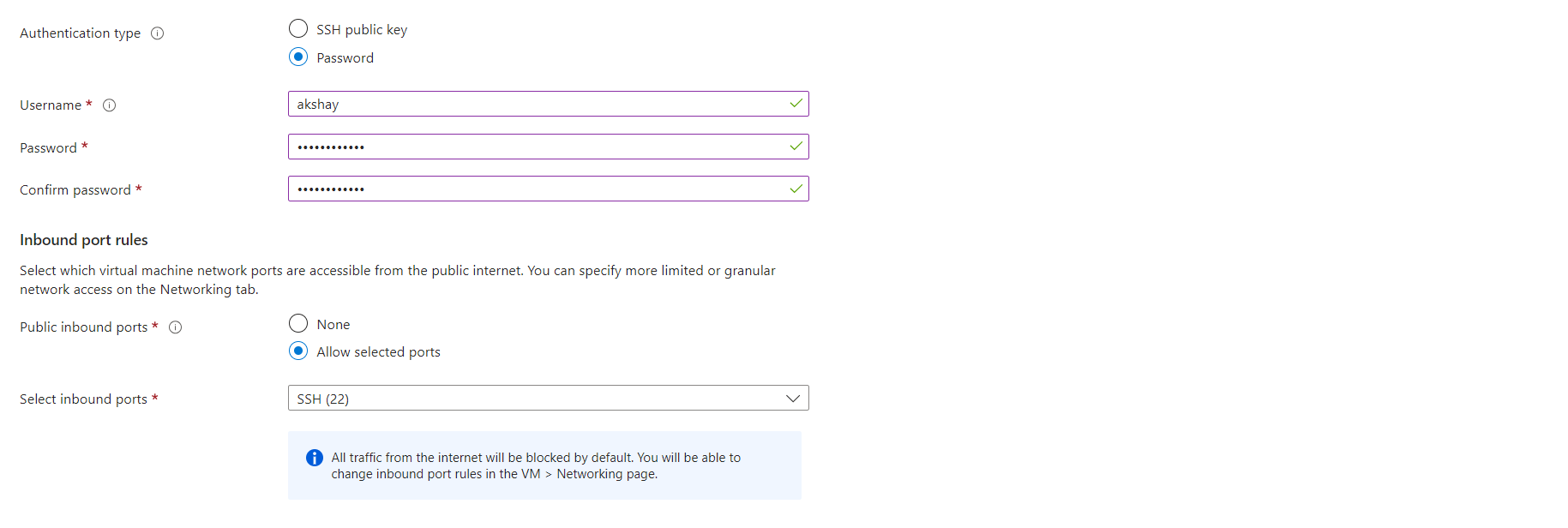
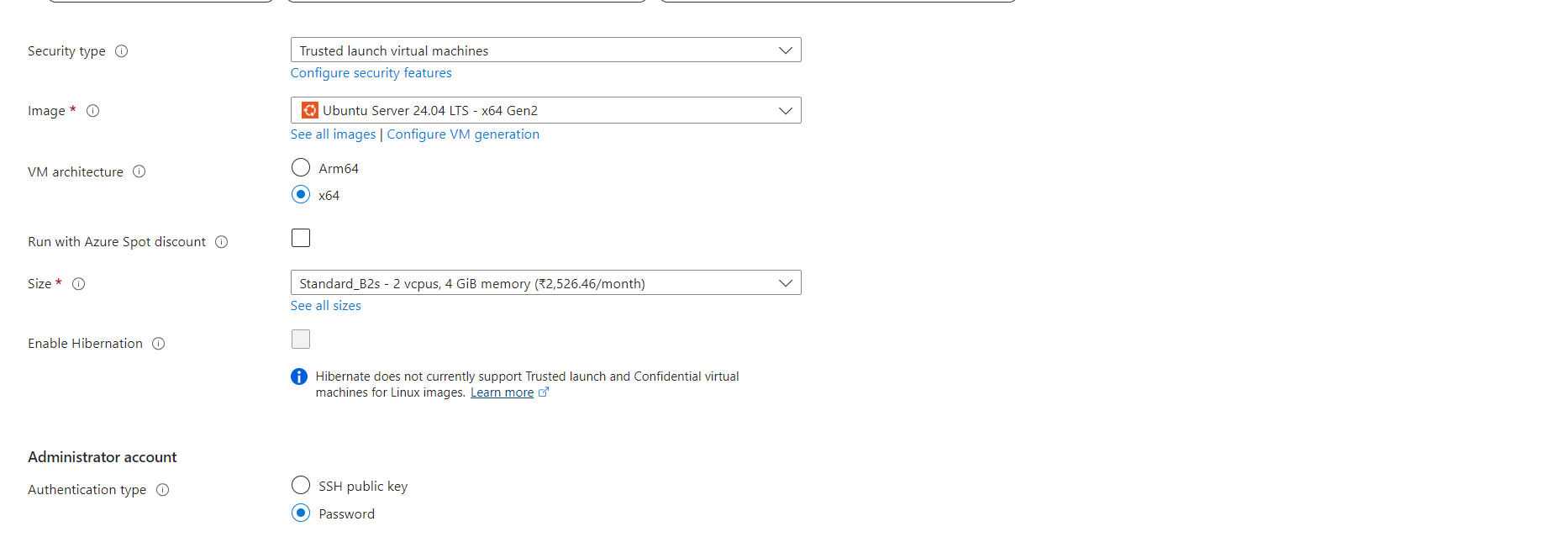
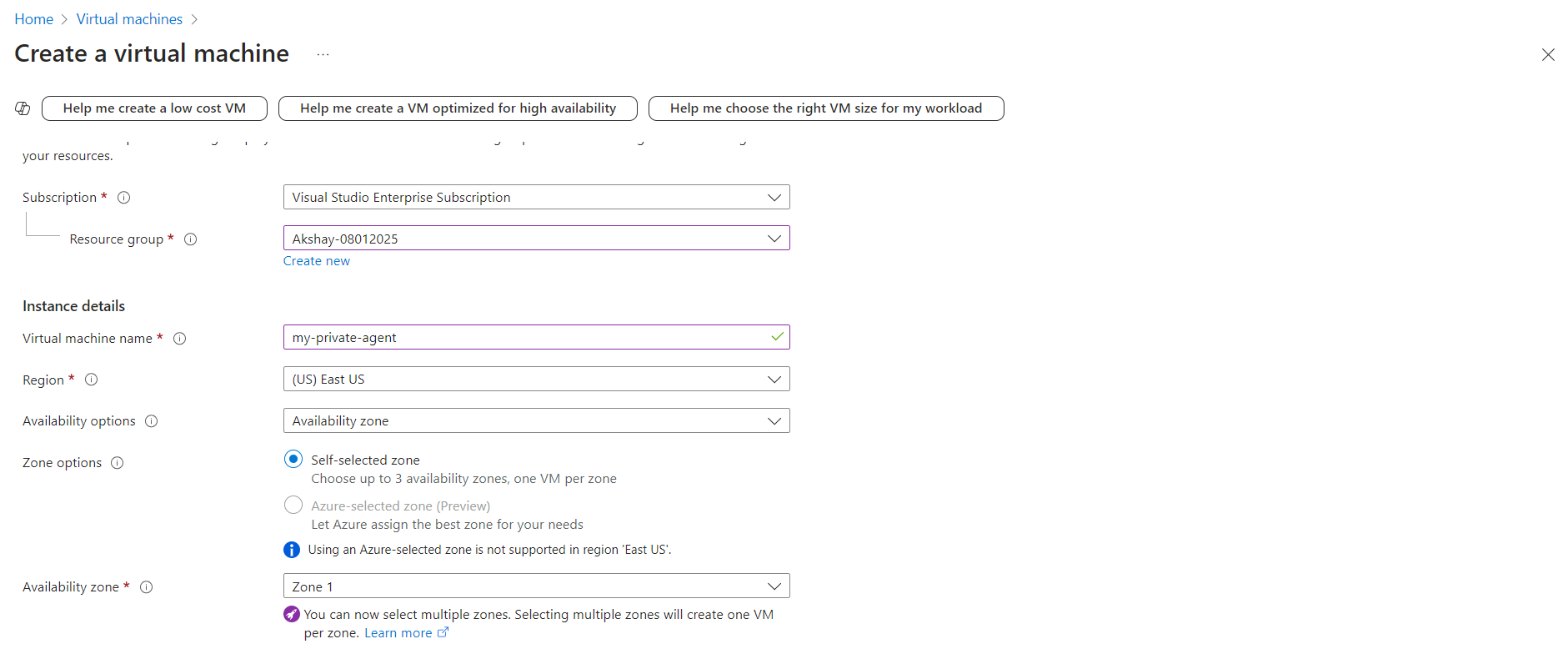
* now click on your 'my-private-agent' and click on 'New Agent'



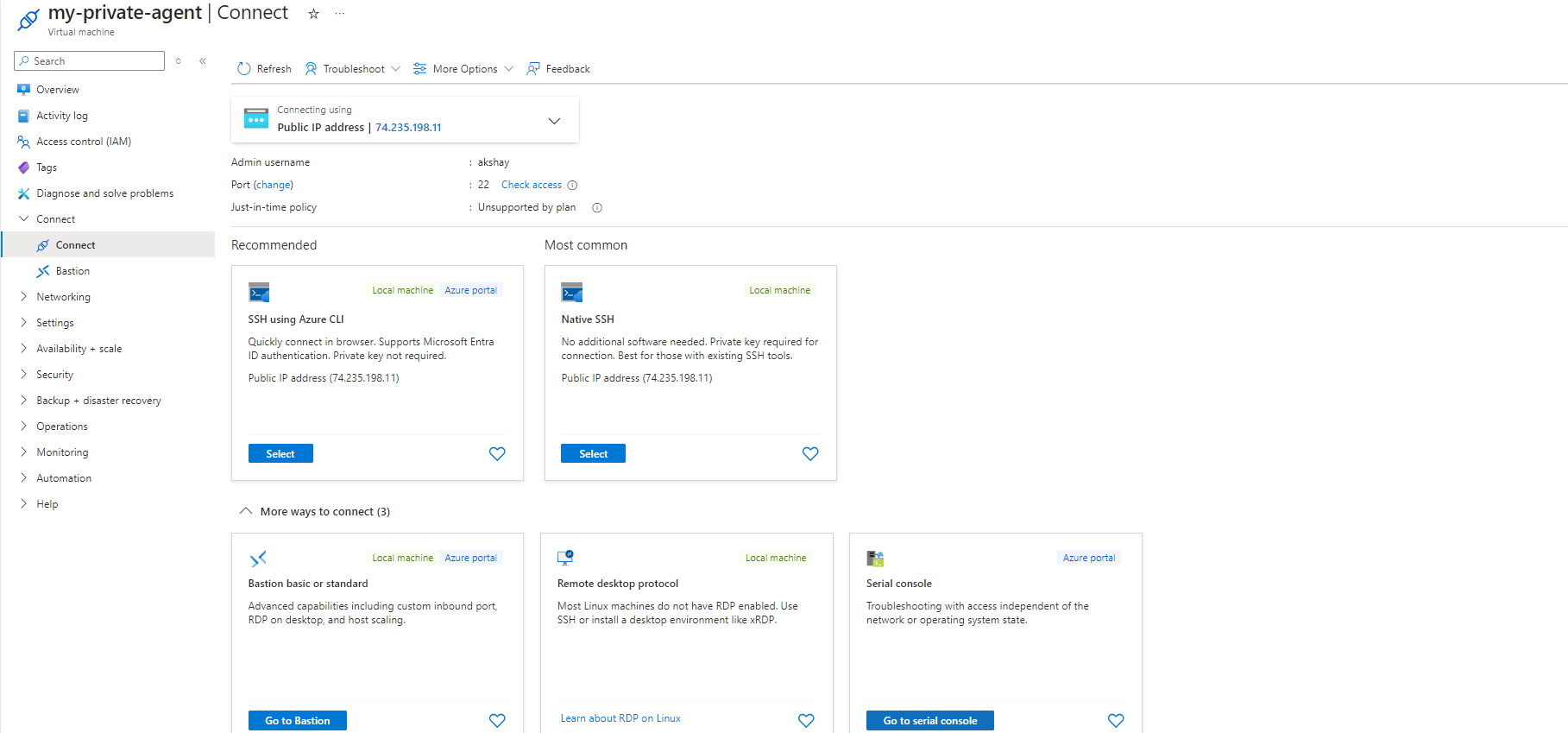
* now keep this screen tab open and open a new one and go to azure portal
* now search for 'Azure Virtual Machine' and click on '+ Create'



* now provide below configuration for the VM.



* Keep the password as **Admin@123456**
* then click on 'Review + Create' then 'Create'
* wait for this machine to get running
* now click on 'connect' button
* now from more option choose 'serial console option'



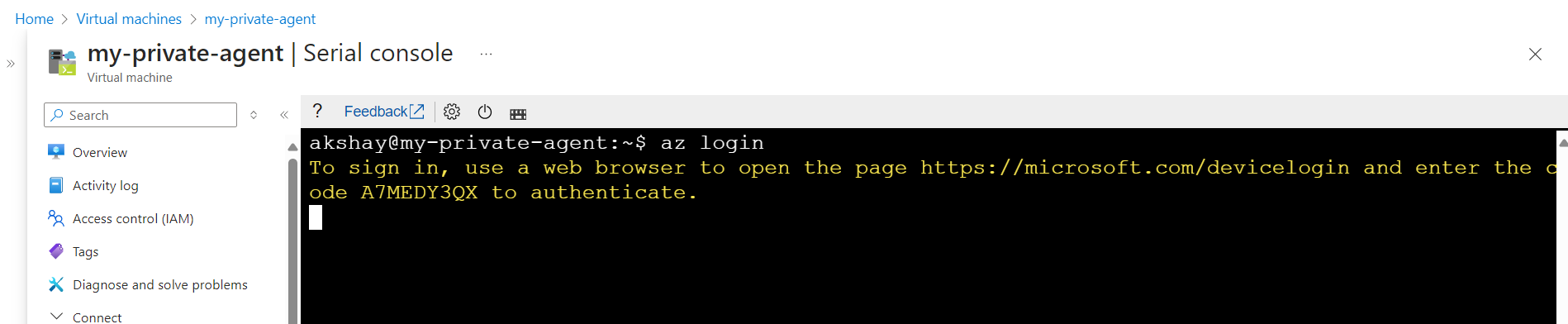
* now login with your credentials.

Task 3: install all the packages and dependencies

* install docker, azure cli

sudo apt update  
sudo apt install sshpass -y  
sudo apt install docker.io -y  
sudo usermod -aG docker $USER  
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

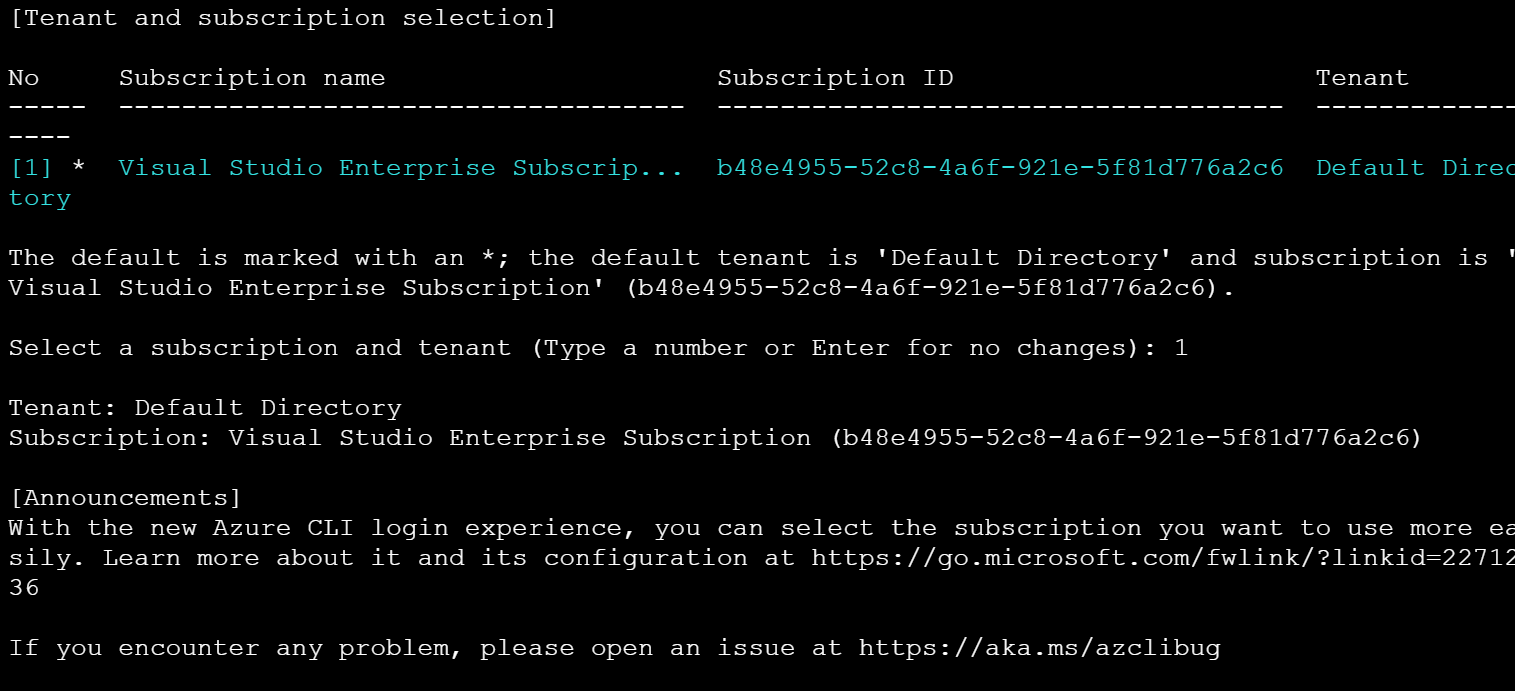
* Now type 'az login' command you will see something like this



* now open a new tab with below website and enter the code which is displayed on CLI

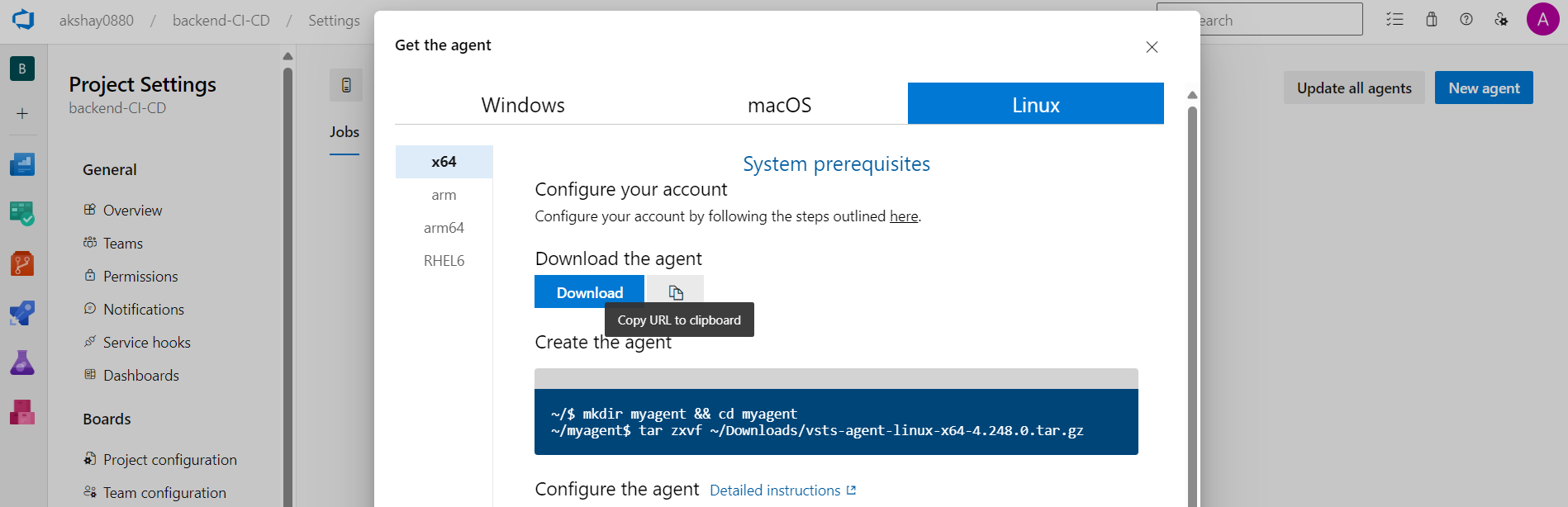
<https://microsoft.com/devicelogin>

* now once you enter the code it will ask you for the permissions do provide that
* now go back to your VM CLI choose your subscription



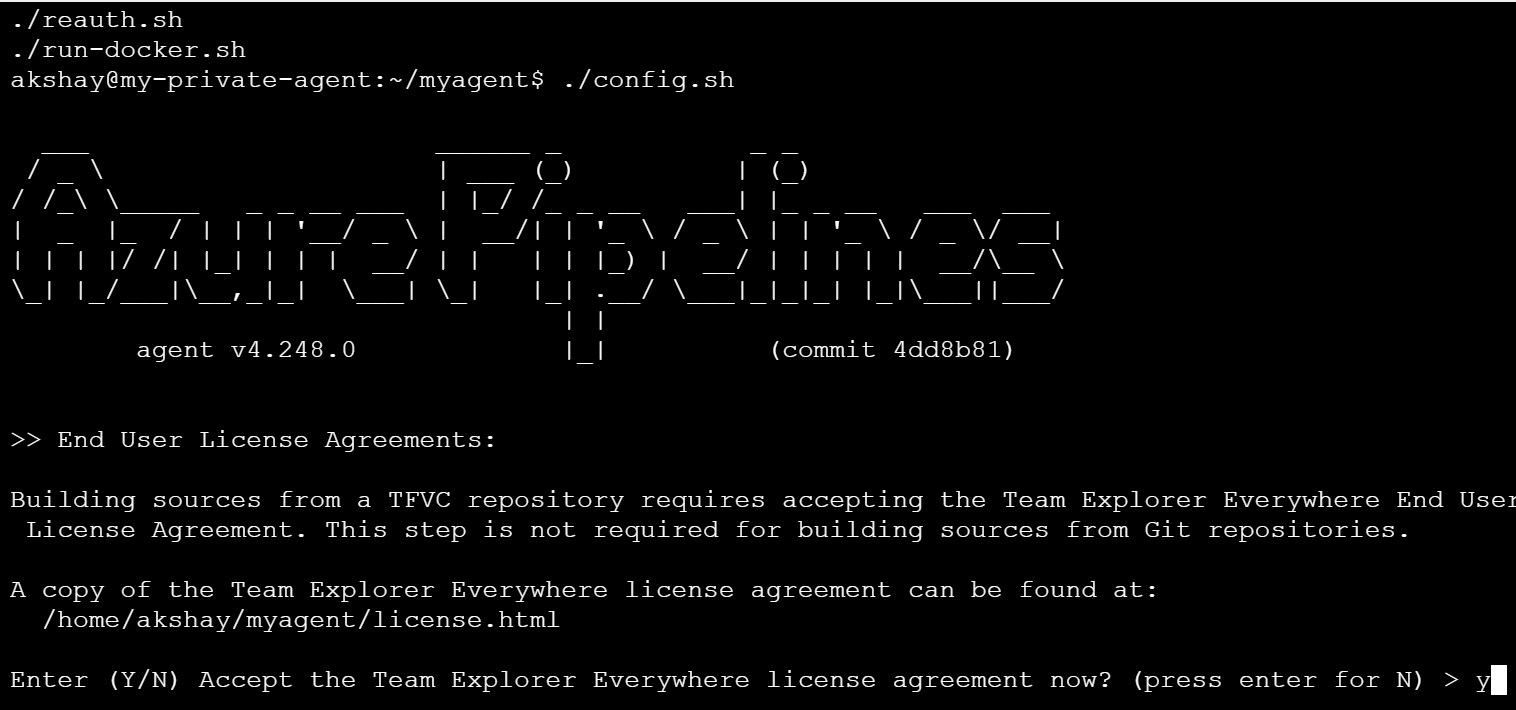
Task 4: Configuring self hosted VM for pipeline

* now go to 'Azure DevOps' tab -> Linux -> and click on copy

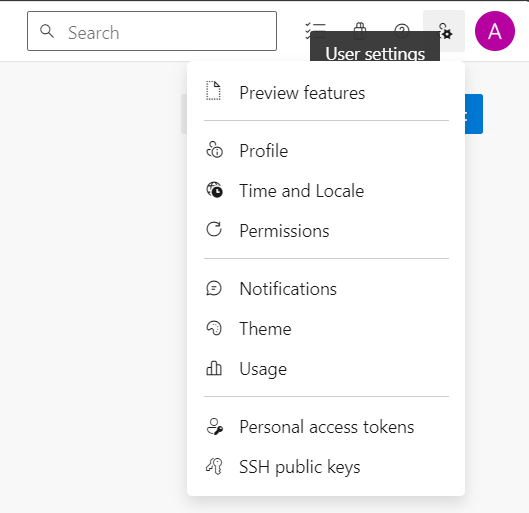


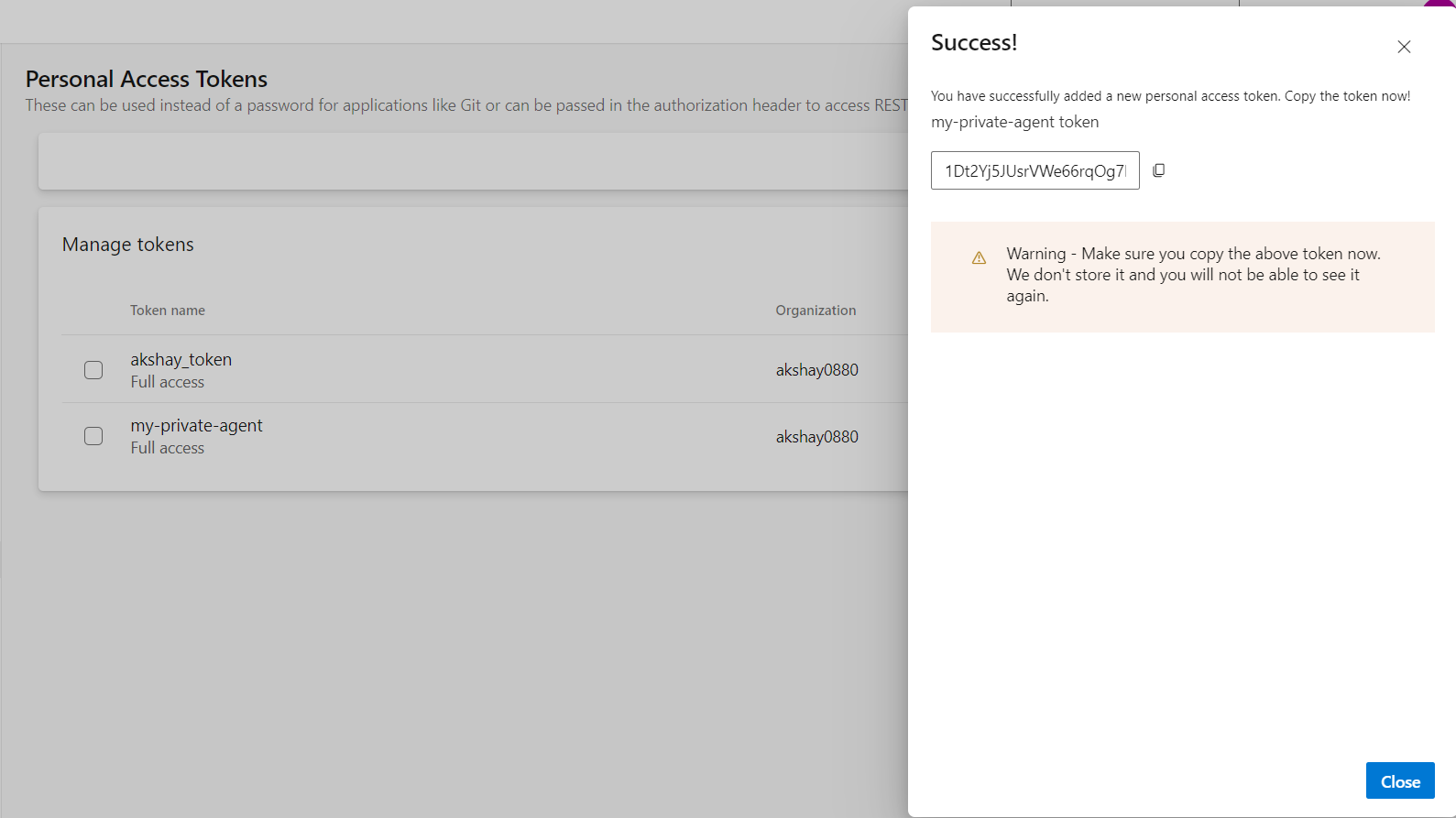
wget https://vstsagentpackage.azureedge.net/agent/4.248.0/vsts-agent-linux-x64-4.248.0.tar.gz  
mkdir myagent && cd myagent  
tar zxvf ~/vsts-agent-linux-x64-4.248.0.tar.gz  
./config.sh

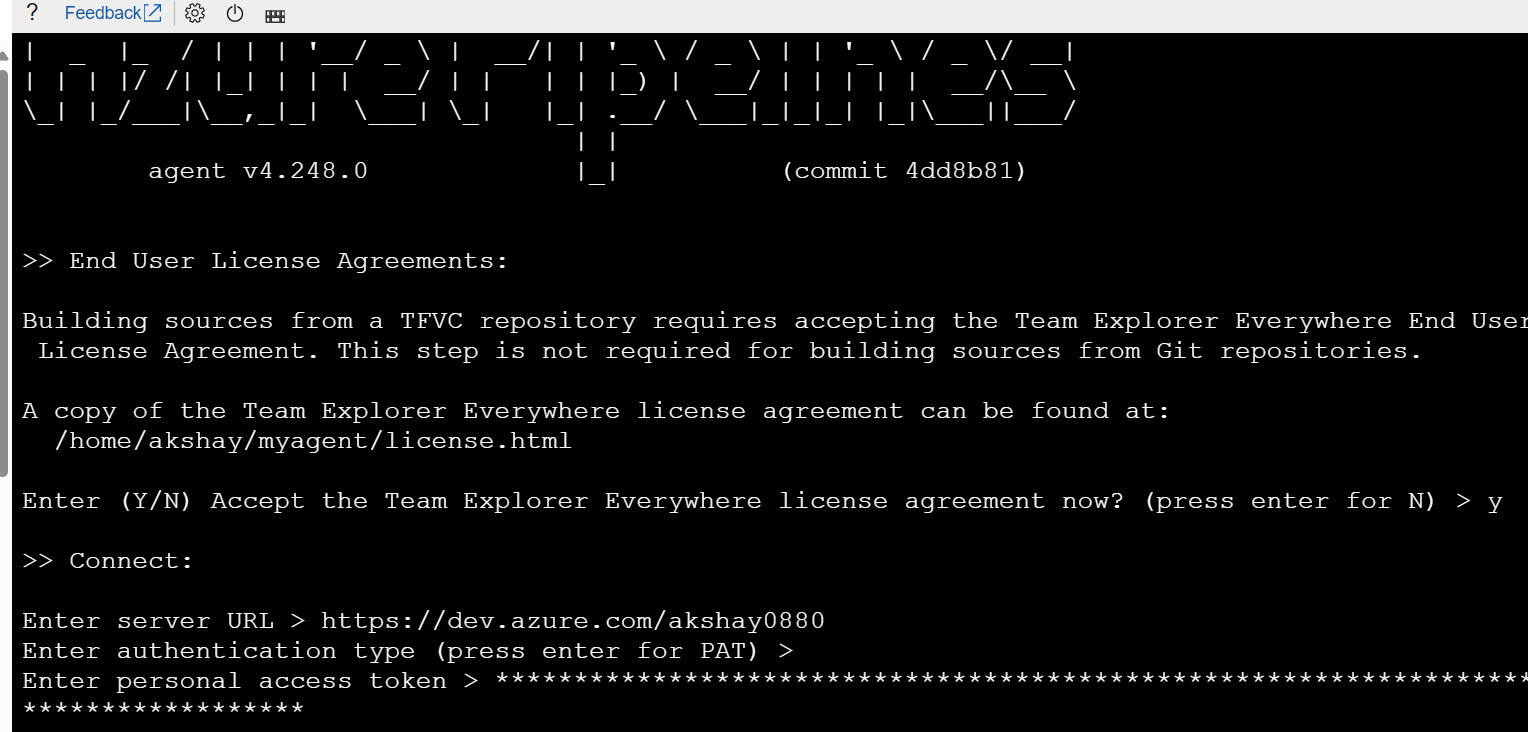
* now once script is executed succesfully you will see something like this



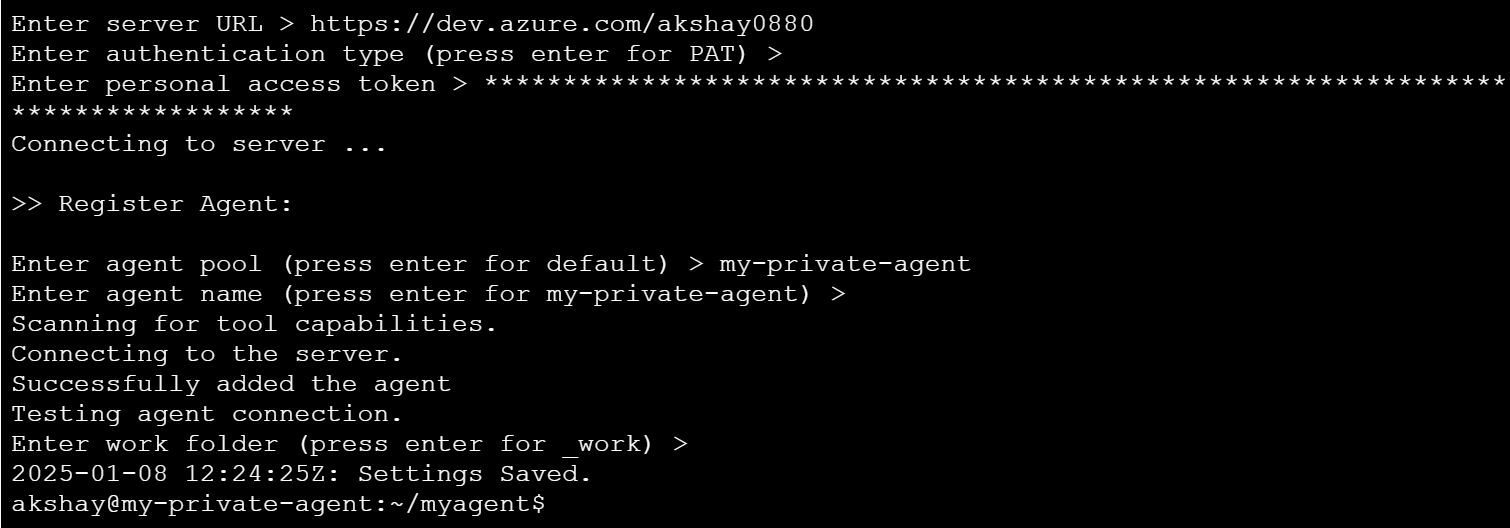
* enter 'Y' and for server url is address of your azure devops server
* like this <https://dev.azure.com/{your-organization}>
* now for PAT press enter
* now go to azure devops tab and on click on before your profile picture



* Now click on Personal Access Tokens and click on '+ New Token'
* Enter the name 'my-private-agent' , check on 'full access' and then clck on create
* now quickly copy this code and paste it on the terminal



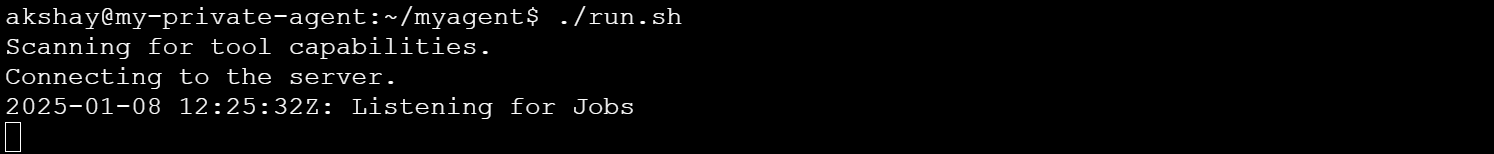
* for agent pool enter 'my-private-agent'



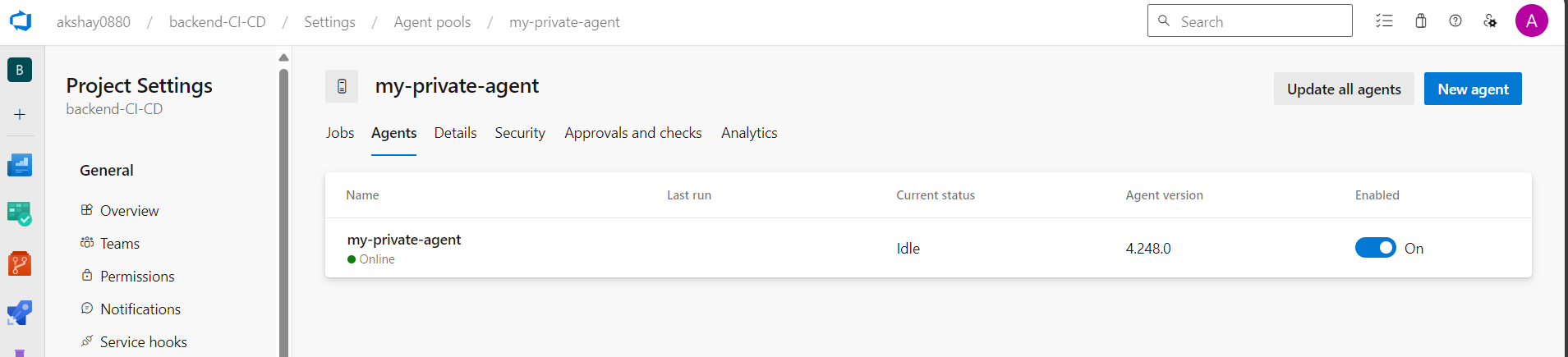
* now last at the end you can run below command to test if installation and connection is established

./run.sh

* you should see output like this

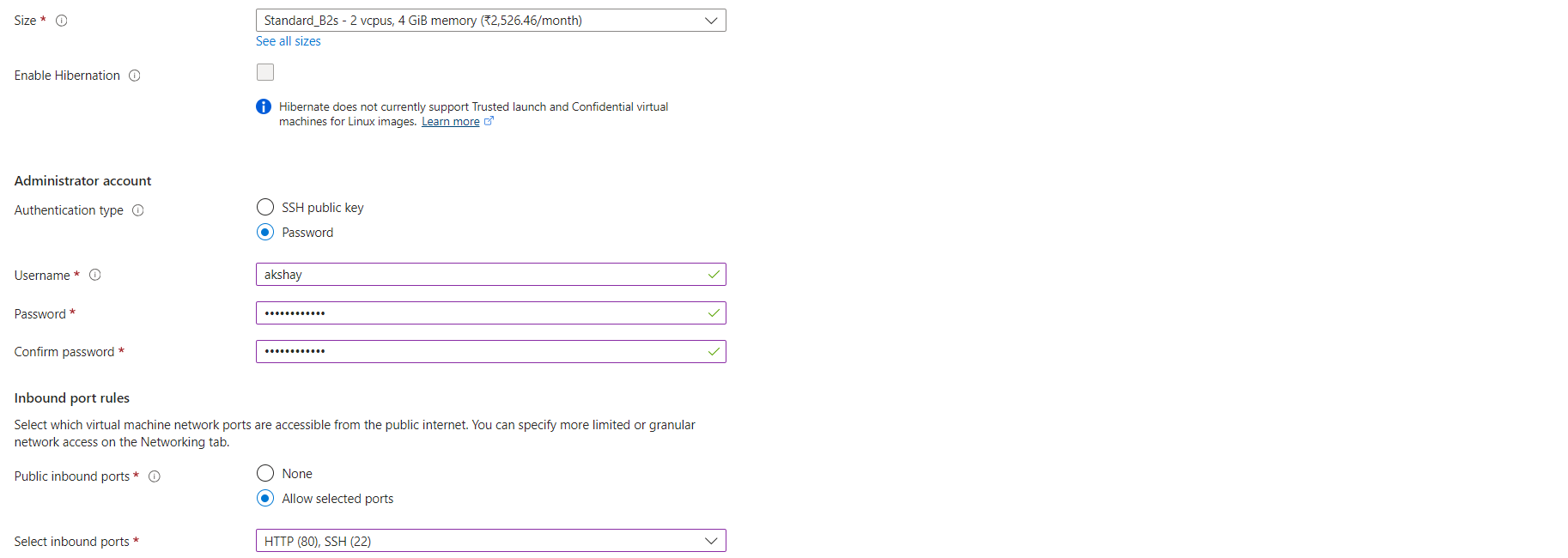
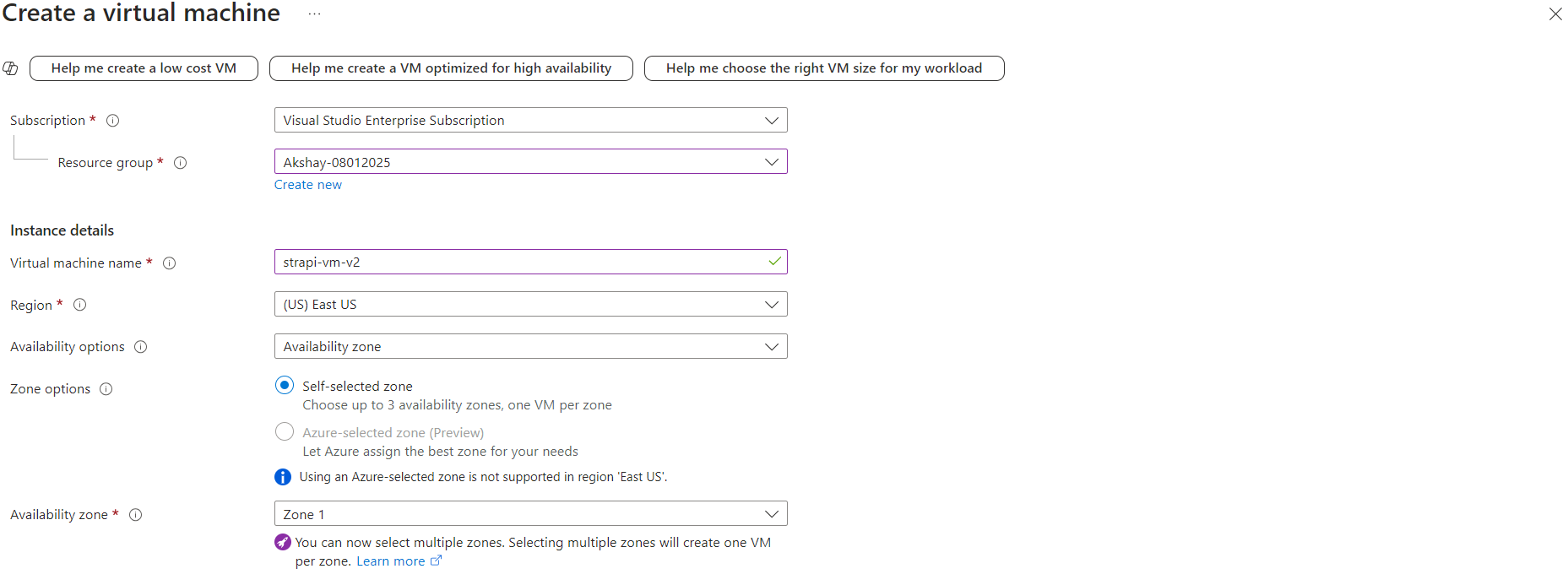


* you can go to azure devops portal to verify there as well



Step 4: Now create Strapi-VM and install all the dependencies to run

Task 1: creat Azure VM

* Go to Azure Portal and search for 'Azure VM'
* now choose below configurations for VM

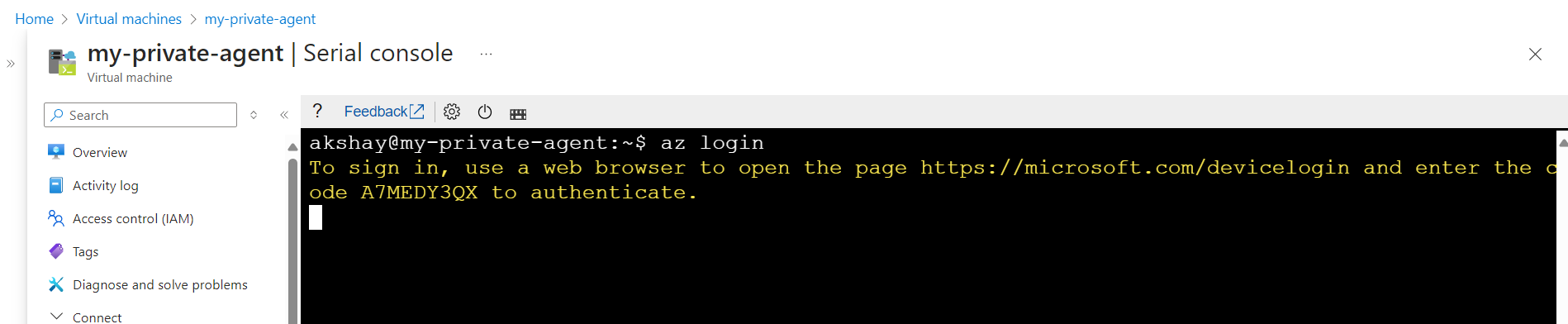
-Then click on 'review + create ' then 'create'

Task 2: install all the packages and dependencies

* install docker, azure cli

sudo apt update  
sudo apt install docker.io -y  
sudo usermod -aG docker $USER  
curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash

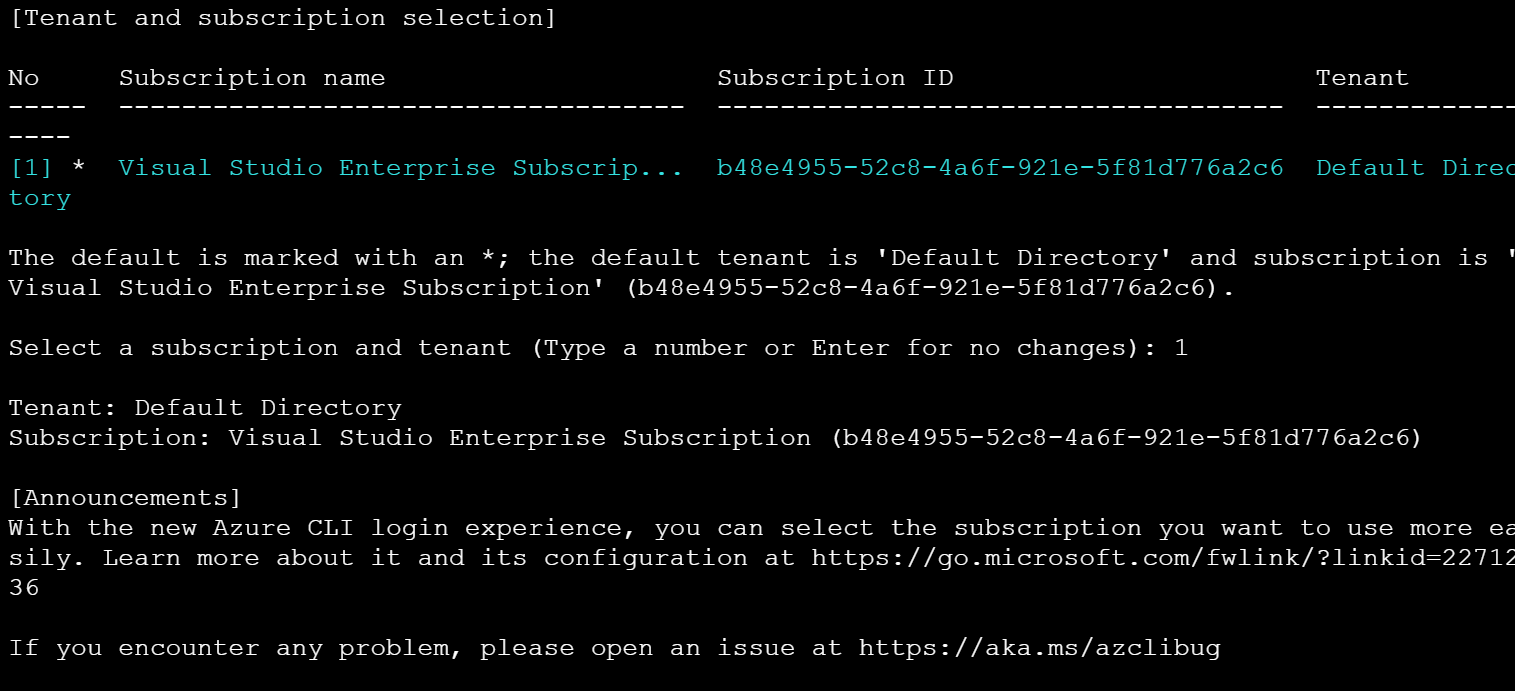
* Now type 'az login' command you will see something like this



* now open a new tab with below website and enter the code which is displayed on CLI

<https://microsoft.com/devicelogin>

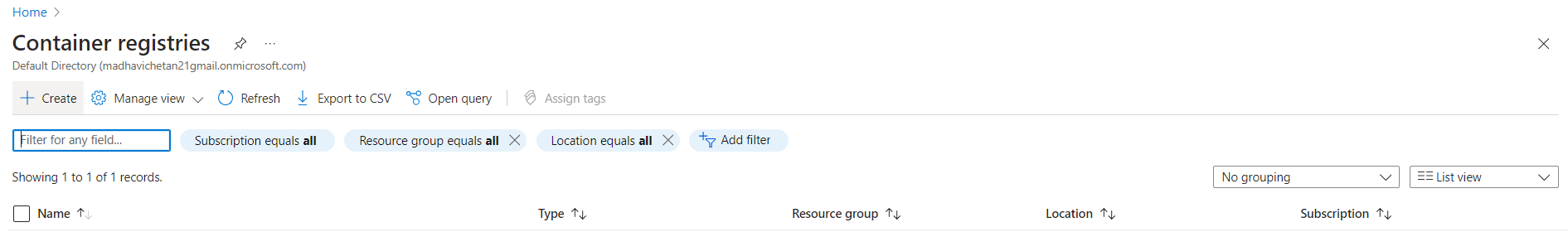
* now once you enter the code it will ask you for the permissions do provide that
* now go back to your VM CLI choose your subscription



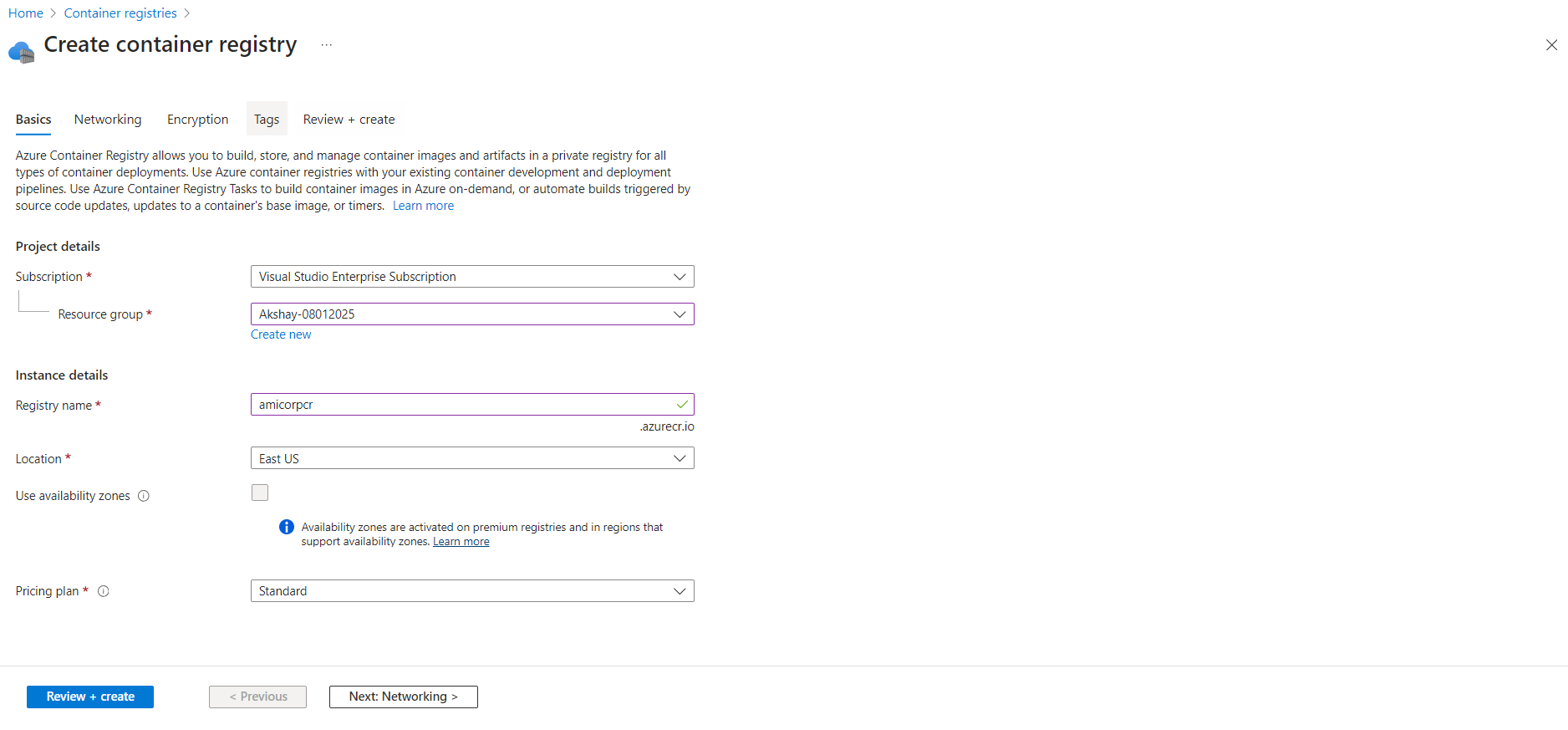
Step 5: Create Azure DevOps Pipeline

Task 1: set up Azure Container Registry to store images

* Go to azure portal and search for 'Azure container registery'

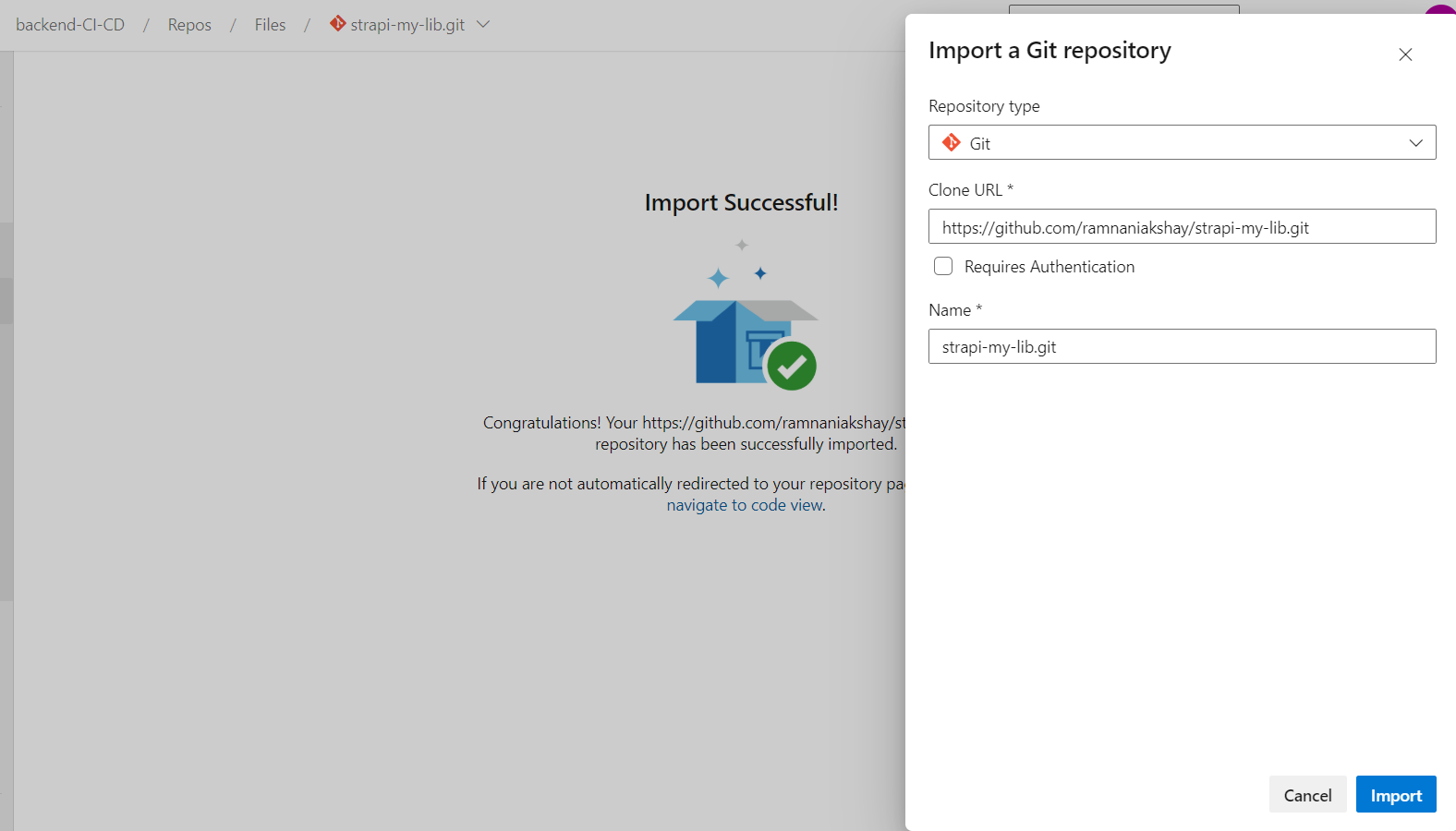


* click on create choose RG and enter registery name as **amicorpcr** then click on 'Review + Create' & create.

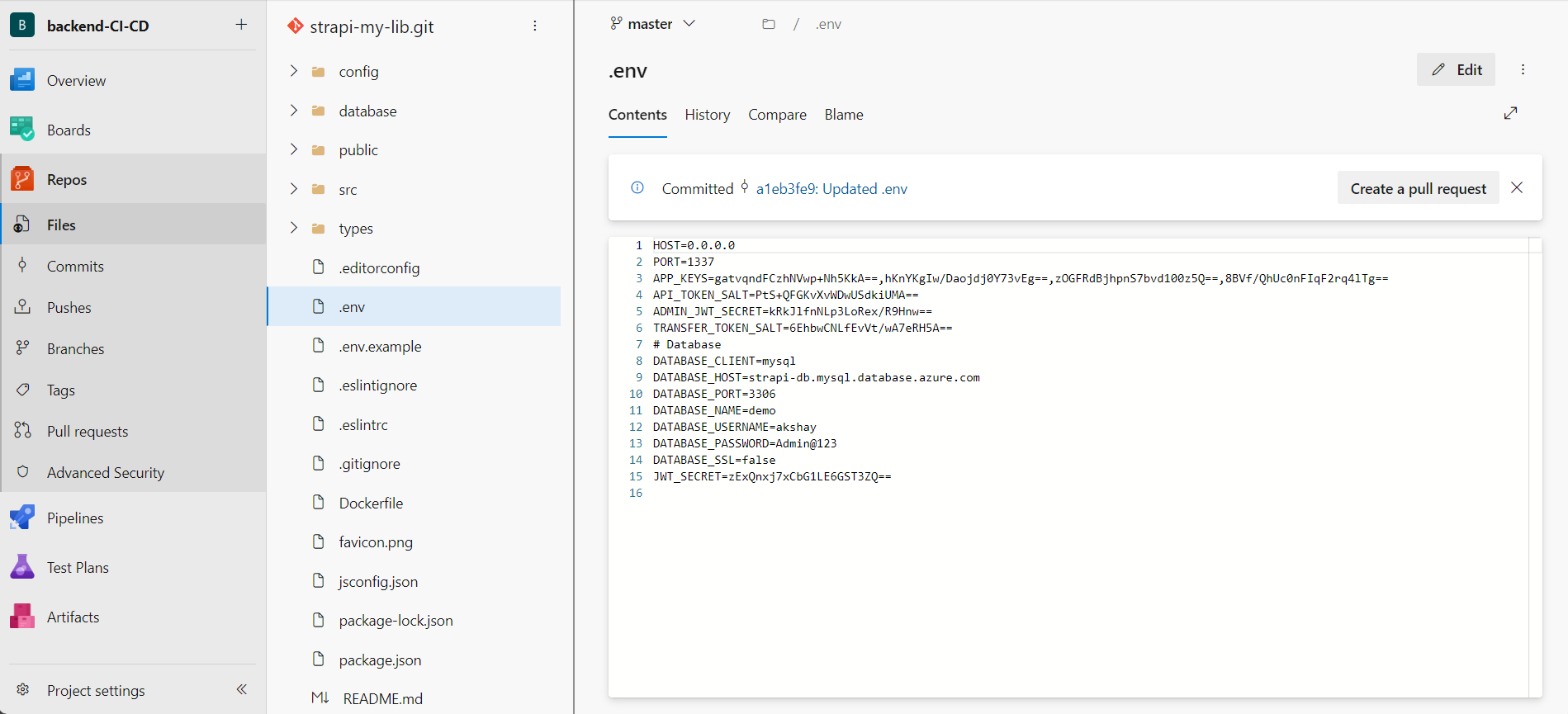


Task 2: set up basic configuration for pipeline

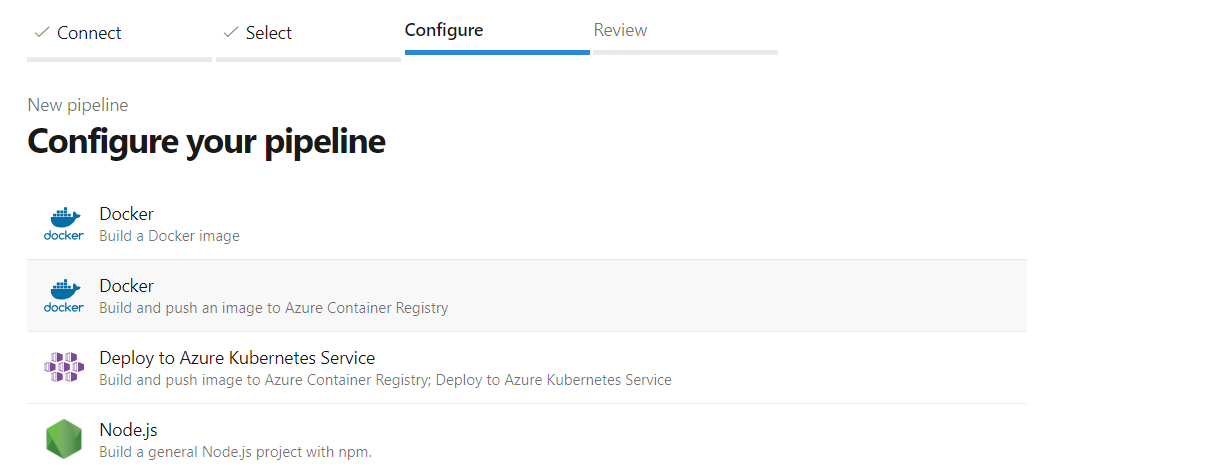
* Go to Azure Repos and click on import repsitory add this URL '<https://github.com/ramnaniakshay/strapi-my-lib.git>'



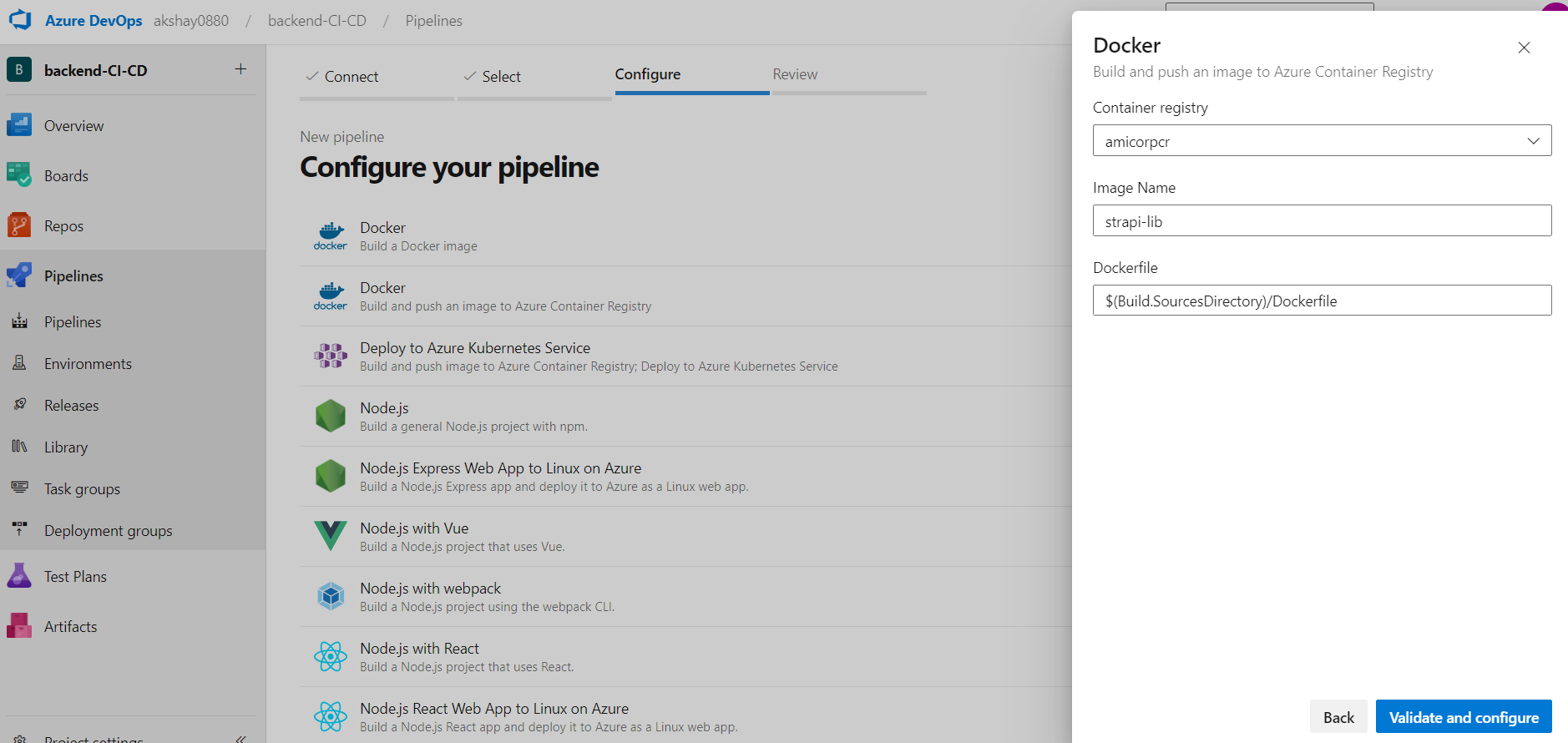
* now go to .env file and update your credentials of DB



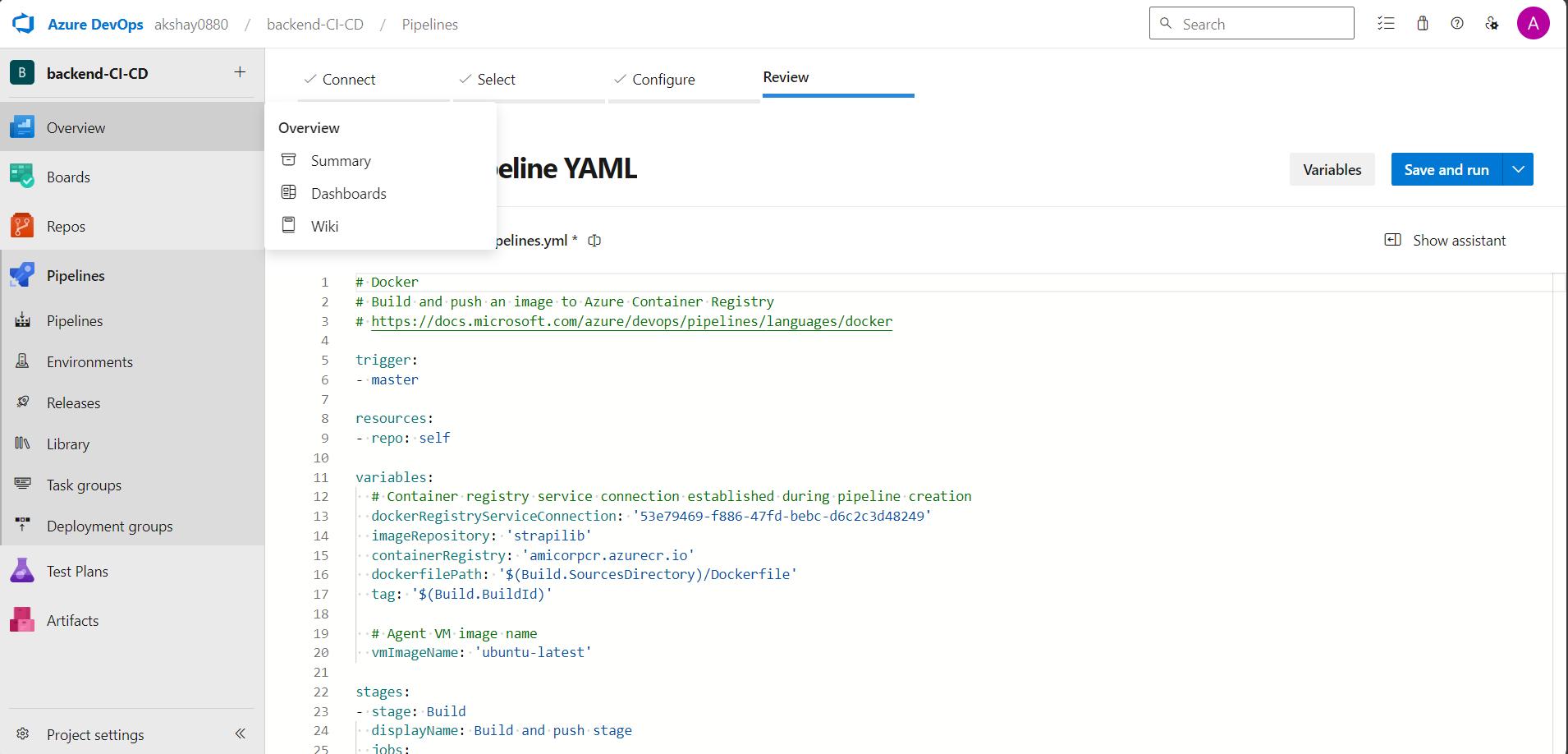
* Now go to Pipelines and click on 'Create Pipeline'
* choose azure Repos Git and pick your repository
* now in configure steps choose 'Docker: build and push to ACR'



* it will also ask for subscription so choose accordingly.
* now choose container registry which you created.



* now basic azure-pipelines.yml file will be generated

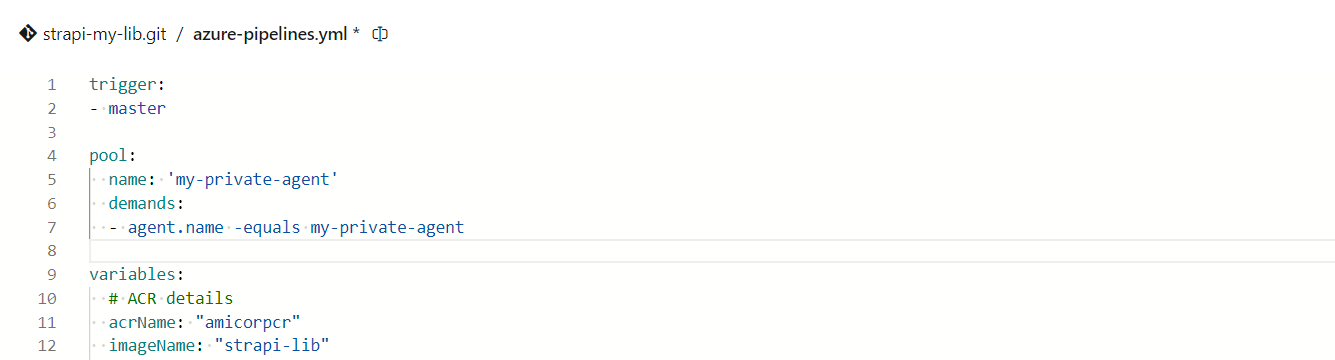


* now copy the below code and paste it on azure devops.

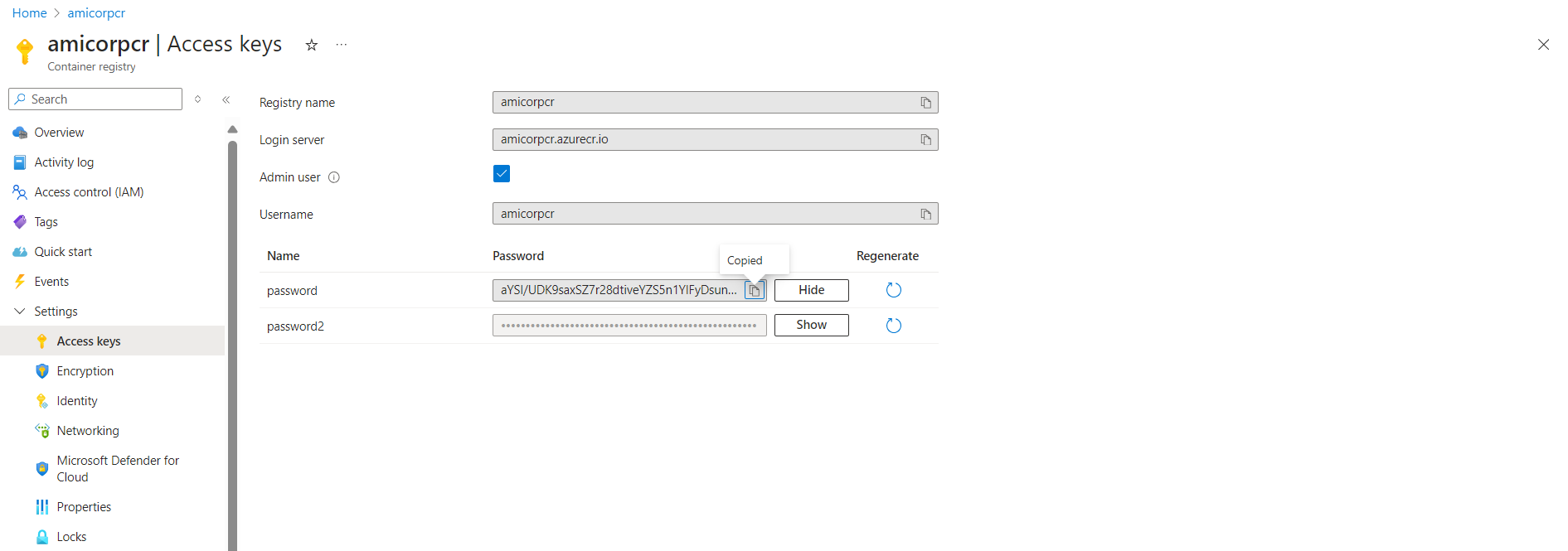
trigger:  
 - master  
  
pool:  
 name: "akshay-CI"  
 demands:  
 - agent.name -equals akshay-CI  
  
variables:  
 # ACR details  
 acrName: "amicorp"  
 imageName: "strapi-backend"  
 acrPassword: "2OoNKMrTr+2h7w9R9HsKuPs1rZkH6x34ZUUQ4tN/ue+ACRDtXCO5"  
 strapiVMIP: "20.106.194.17"  
 StrapiVMUsername: "Akshayr"  
 strapiVMPassword: "Akshayr\@cloudthat.com"  
  
steps:  
 # Step 1: Login to Azure  
 - task: AzureCLI\@2  
 displayName: "Login to Azure"  
 inputs:  
 azureSubscription: "Visual Studio Enterprise Subscription(b48e4955-52c8-4a6f-921e-5f81d776a2c6)"  
 scriptType: "bash"  
 scriptLocation: "inlineScript"  
 inlineScript: 'echo "Logging in to Azure"'  
  
 # Step 2: Docker build and tag  
 - task: Docker\@2  
 displayName: "Build Docker Image"  
 inputs:  
 containerRegistry: "$(acrName)"  
 repository: "$(imageName)"  
 command: "buildAndPush"  
 Dockerfile: "\*\*/Dockerfile"  
 tags: "latest"  
  
 # Step 3: Deploy it to VM  
 - task: Bash\@3  
 displayName: "SSH and Run Container"  
 inputs:  
 targetType: "inline"  
 script: |  
 sshpass -p "$(strapiVMPassword)" ssh $(StrapiVMUsername)\@$(StrapiVMIP) << EOF  
 echo "Pulling Docker image from ACR..."  
 docker login $(acrName).azurecr.io -u $(acrName) -p $(acrPassword)  
 docker pull $(acrName).azurecr.io/$(imageName):latest  
  
 echo "Stopping and removing existing container if exists..."  
 docker ps -a -q --filter "name=strapi-container" | grep -q . && docker stop strapi-container && docker rm strapi-container  
  
 echo "Running Docker container..."  
 docker run -d -p 1337:1337 --name strapi-container $(acrName).azurecr.io/$(imageName):latest  
 EOF

Step 6: Lets modify the code

* now update this code for pool details, acrName and imageName



* now for acr password go to ACR from azure portal

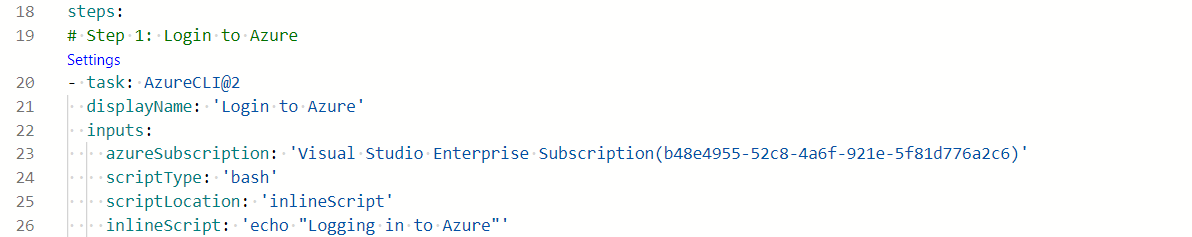


* now add your credentials here in variables



Task 1: modify step 1

* now click on settings



* select your subscription and click on add

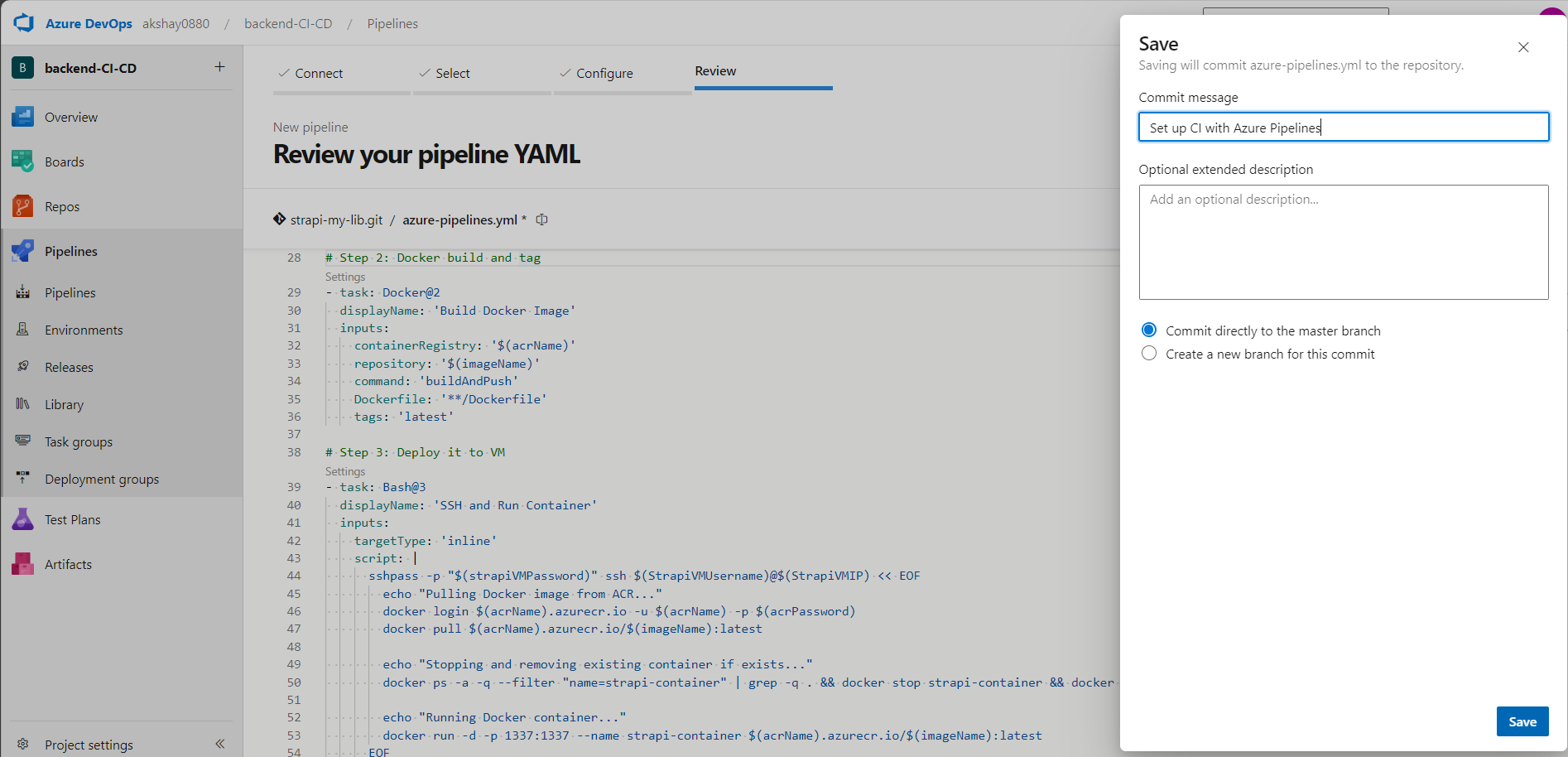


Task 2: modify step 2 & 3

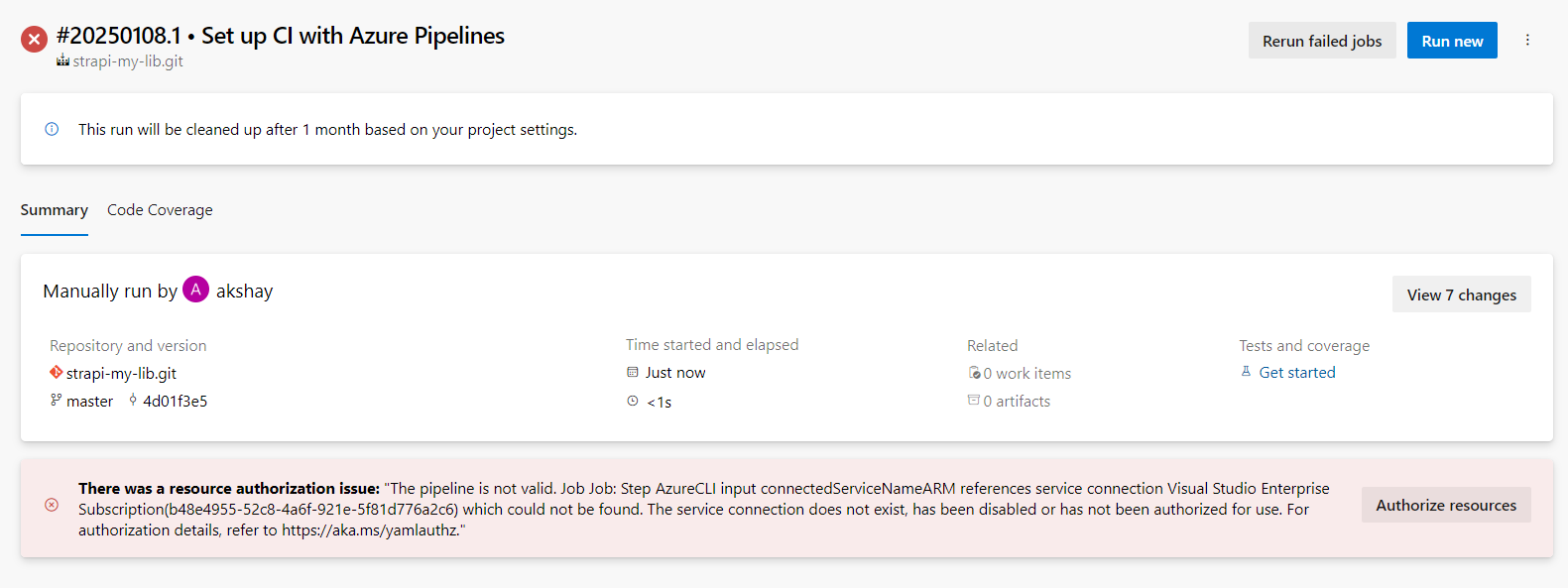
* keep all the fields as it as default



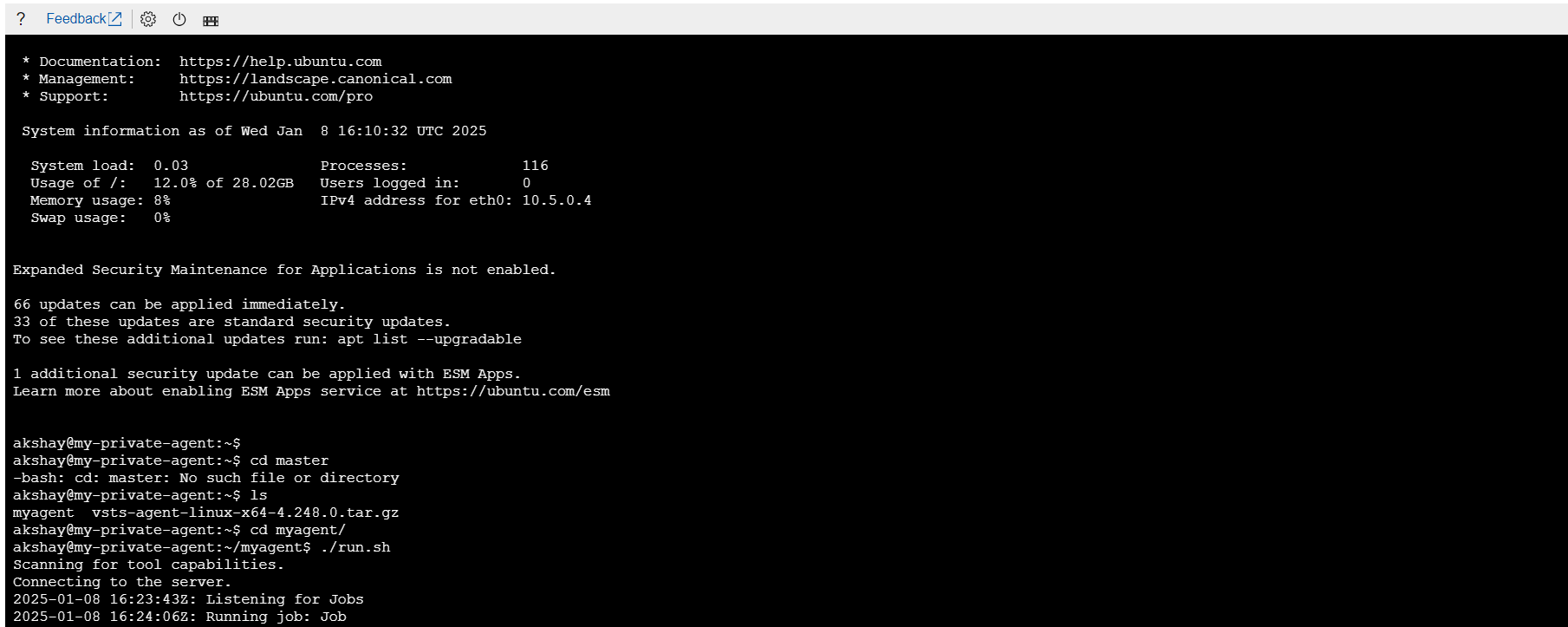
* now click on save and run pipeline



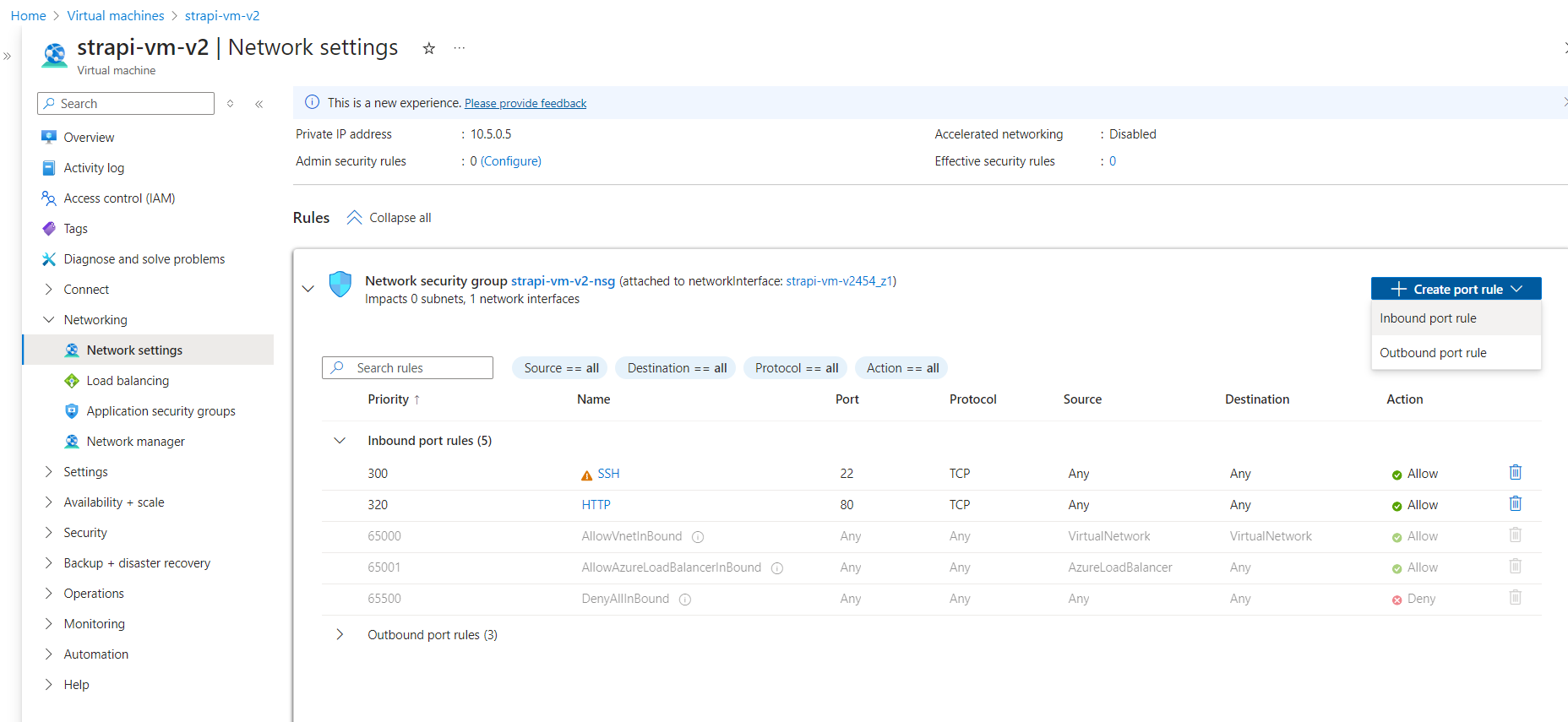
* now for the first time you have to authorize.



* provide another time permission
* Make sure your **my-private-agent** is running on another tab with /run.sh in myagent directory



* once its successful create inbound rule for our strapi-vm-v2 go VM -> network settings -> click on create port rule -> inbound port rule



* add 1337 port number and choose 'TCP'.
* now you can take your *Strapi-IP:1337* to check the deployment