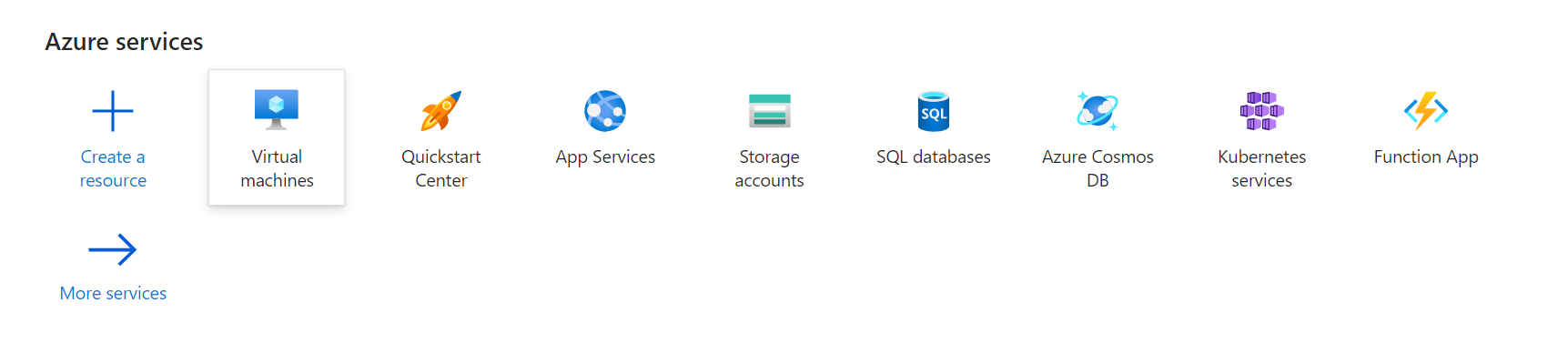
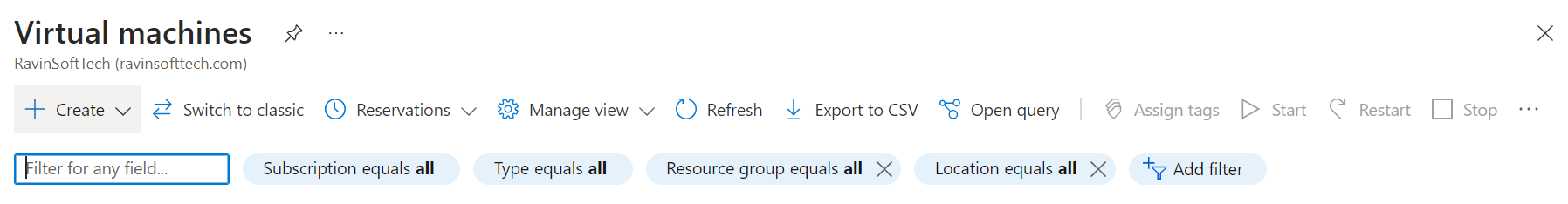
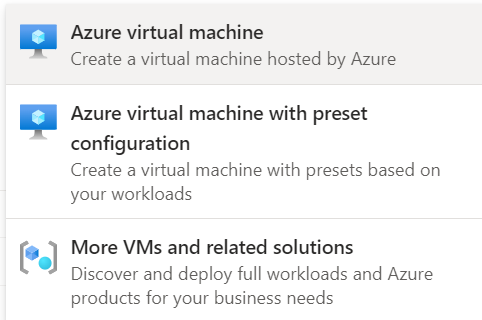
Case Study Document

# Creating a Virtual Machine on Azure

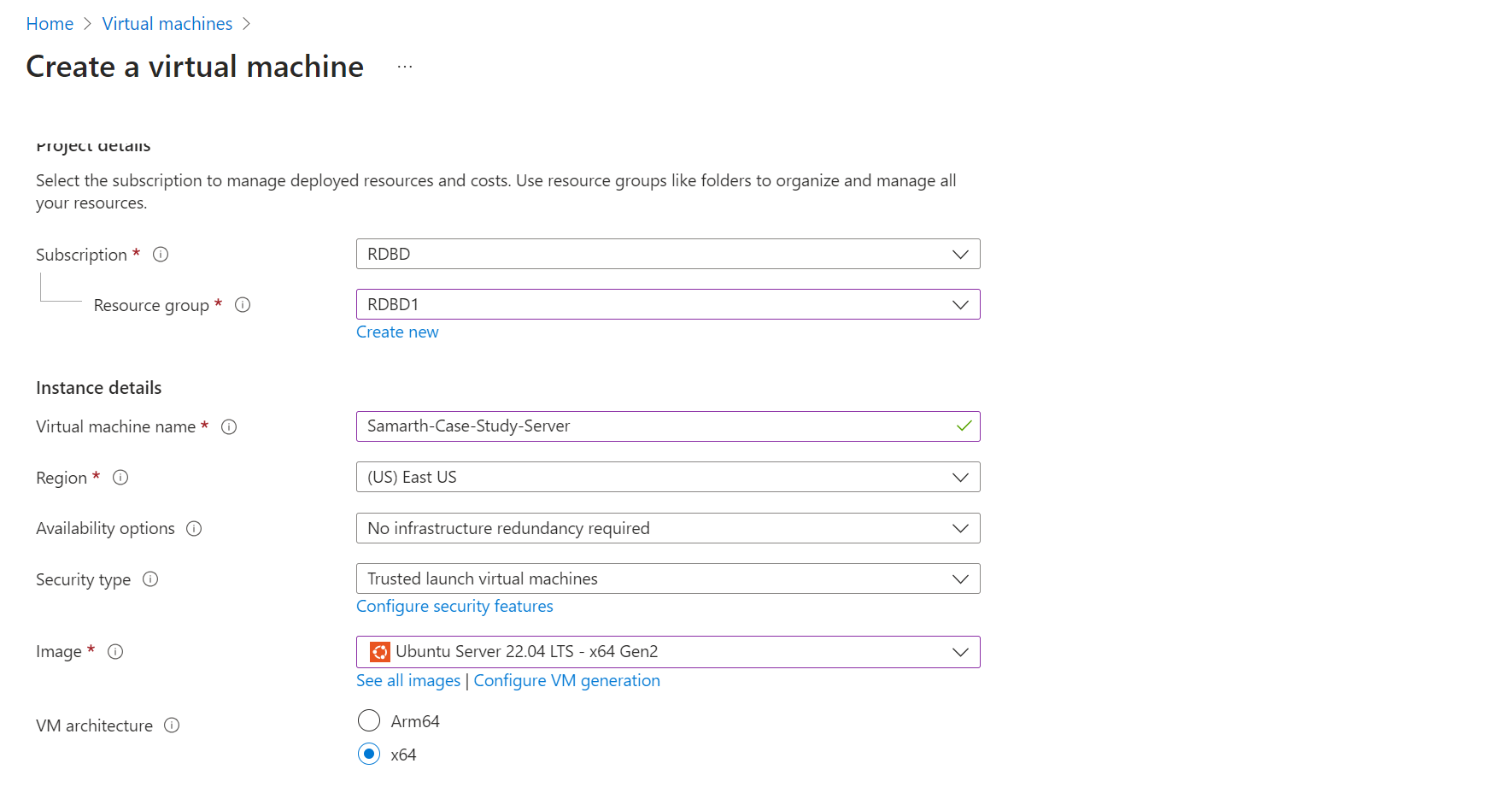


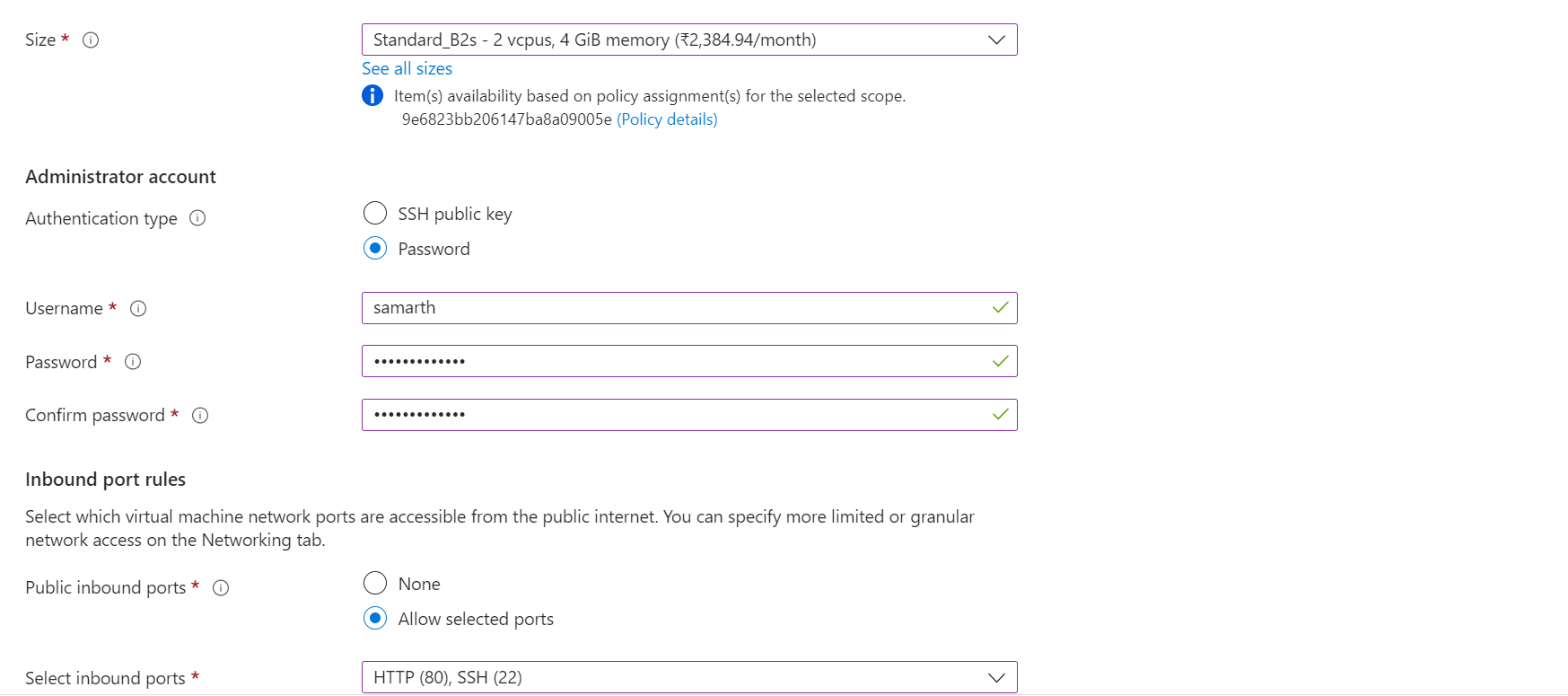
Click on the Virtual Machine button, to navigate to create a virtual machine.

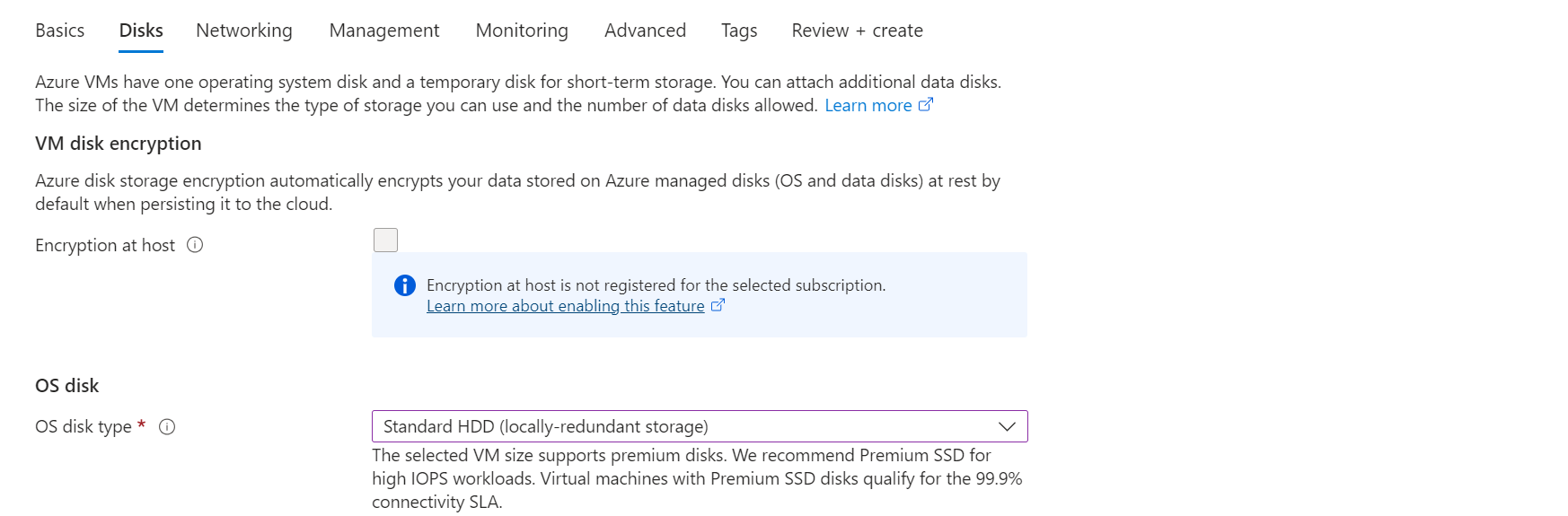




After which we have to click on the ‘Create’ Button, and select the ‘Azure virtual Machine’ option.







After selecting the Respective Location, Resource Group, B2s as size, Ubuntu as a OS, Username, Password, Standard HDD for disk space as instructed we then have to click upon the Review+Create button to review our selection and inputs.

Once reviewed click on Create button to create the VM.

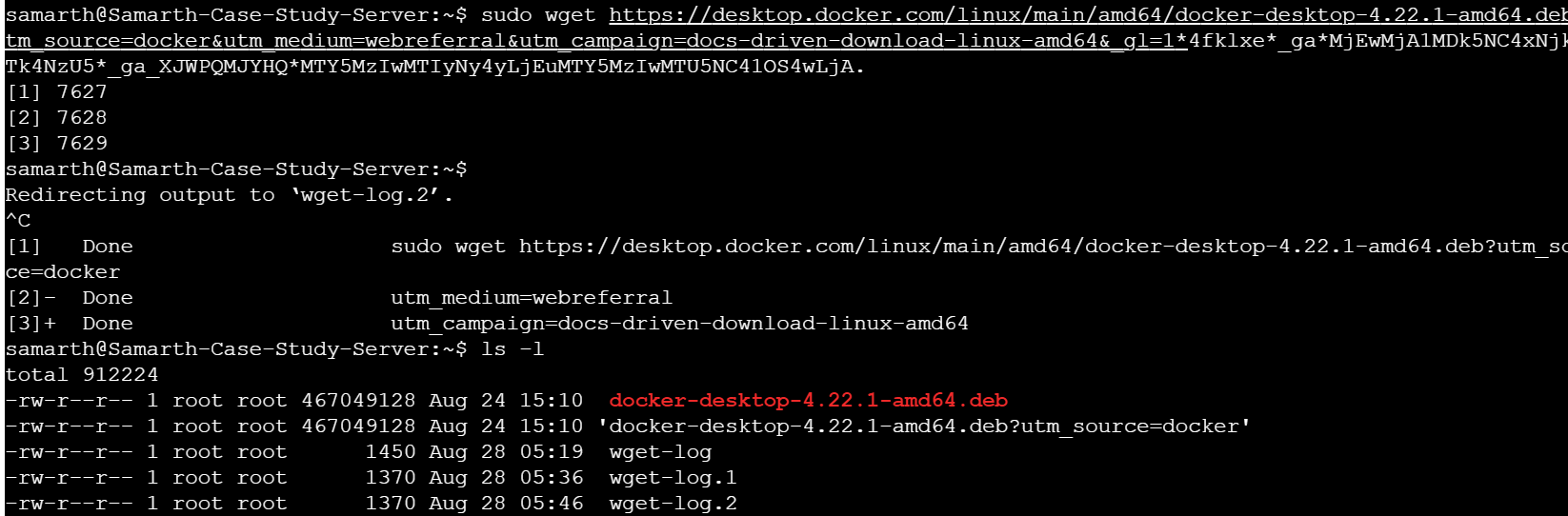
# Creation of Docker Image

Go to <https://ssheasy.com> to run for the CLI

Enter the Public IP Address, Username and Password to connect to the VM

Now enter the command ***sudo apt update*** to update the existing packages.

To get the docker package use the following command: ***sudo wget <link>.***

**

After enter the command to start installing it will redirect to another command which we have to kill using: ***Ctrl + C***.

After the command is killed now move the installed file to .deb file using the command: ***sudo mv <file-name-to-be-moved> <destination-file>****.*



To install the package now enter the command: ***sudo apt-get install ./ <file-name>***

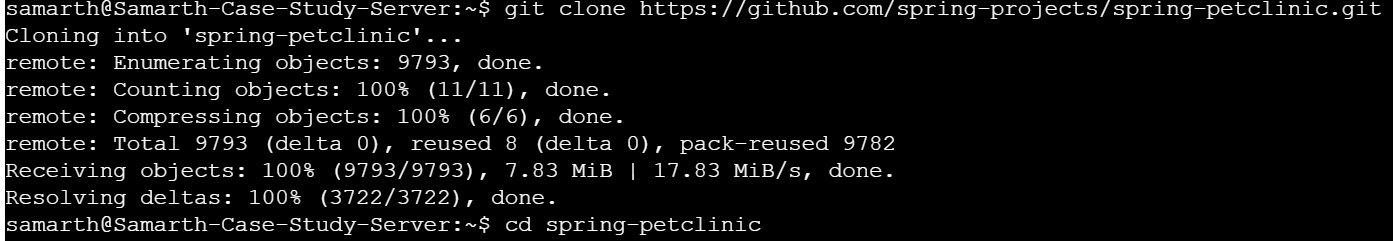
**

It will start installing the packages and once it is installed we have to enable it using the following command: ***systemctl --user enable docker-desktop***

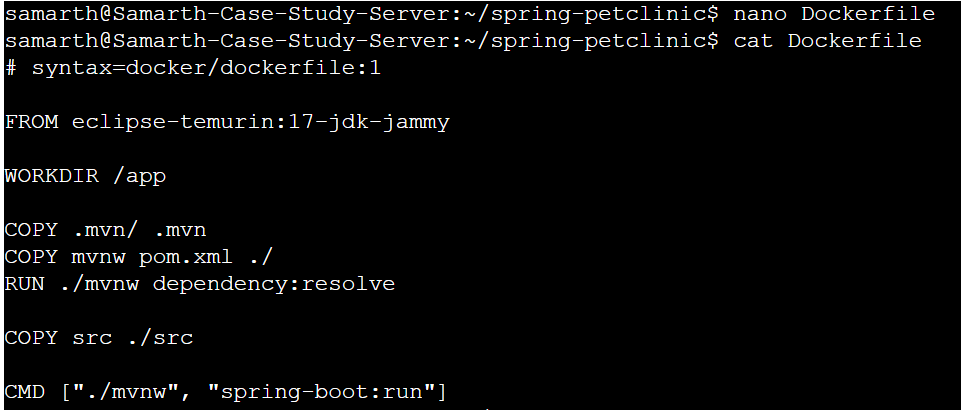
Now you have installed and enabled the docker-desktop

The next step is to install a sample application so we used the following commands: ***git clone*** [***https://github.com/spring-projects/spring-petclinic.git***](https://github.com/spring-projects/spring-petclinic.git)

***cd spring-petclinic***



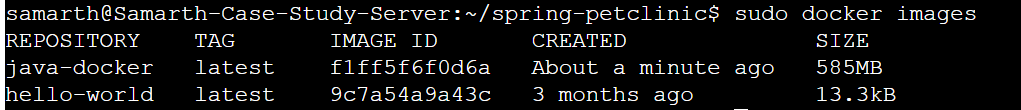
After the repository has been cloned move to the cloned directory and create a file named ‘Dockerfile’ using the command ***touch Dockerfile***and enter the content given.

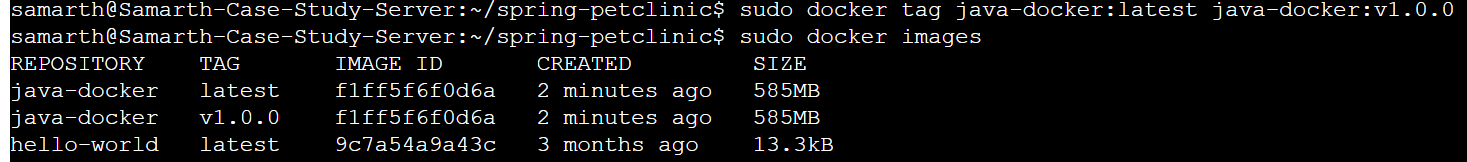


Create a file named .dockerignore file and enter the contents in the similar fashion.

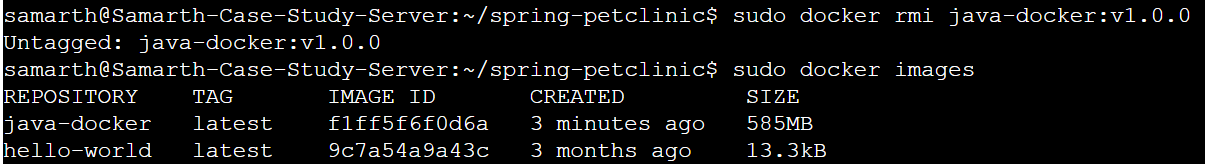
Now to create the docker image we will be first running the following command: ***sudo docker build --tag java-docker.***

And run the command: ***sudo docker images*** to view the created docker images.



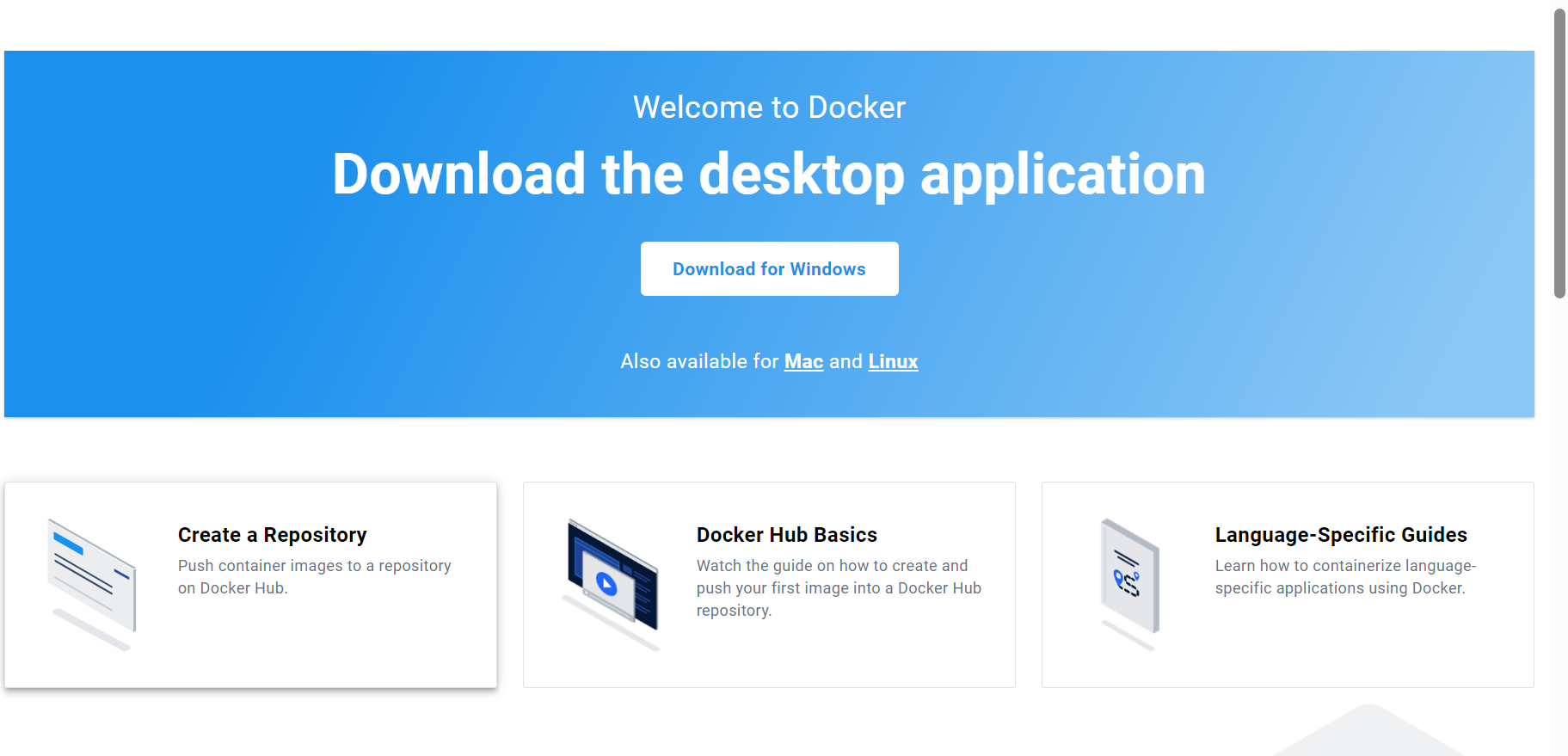
Now to create an image use the following command: 

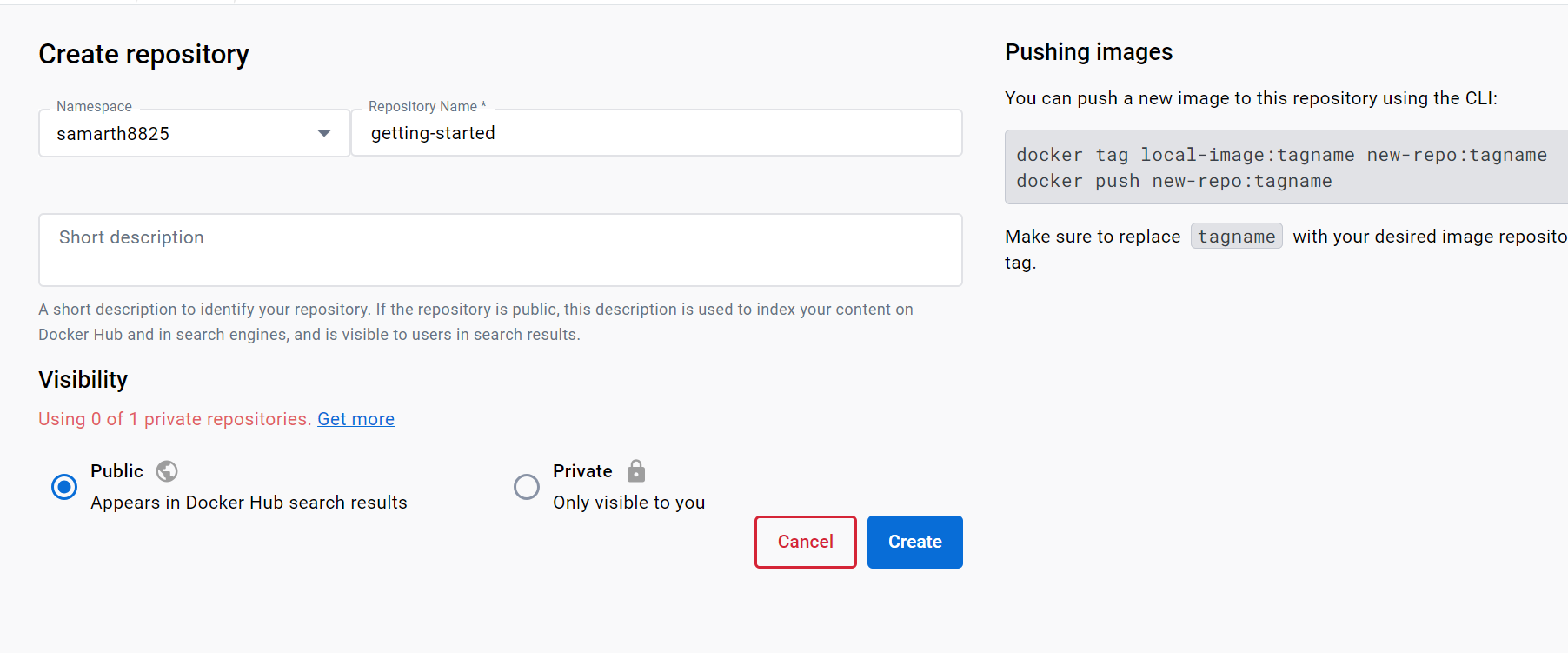
Since we have created and redundant image we will be removing it using *rmi* command



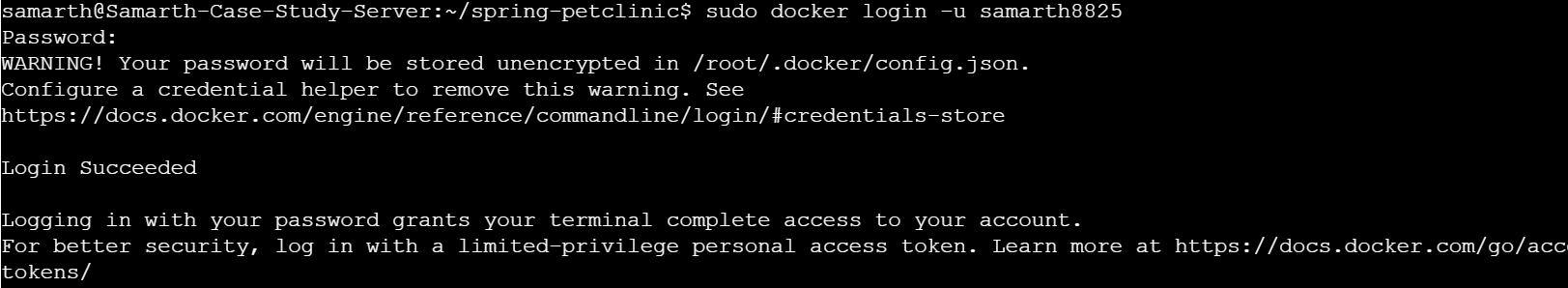
# Docker Containerization

To publish a docker image on docker hub we have to signup and click on the *Create Repository* button



Enter the details and make its visibility public 

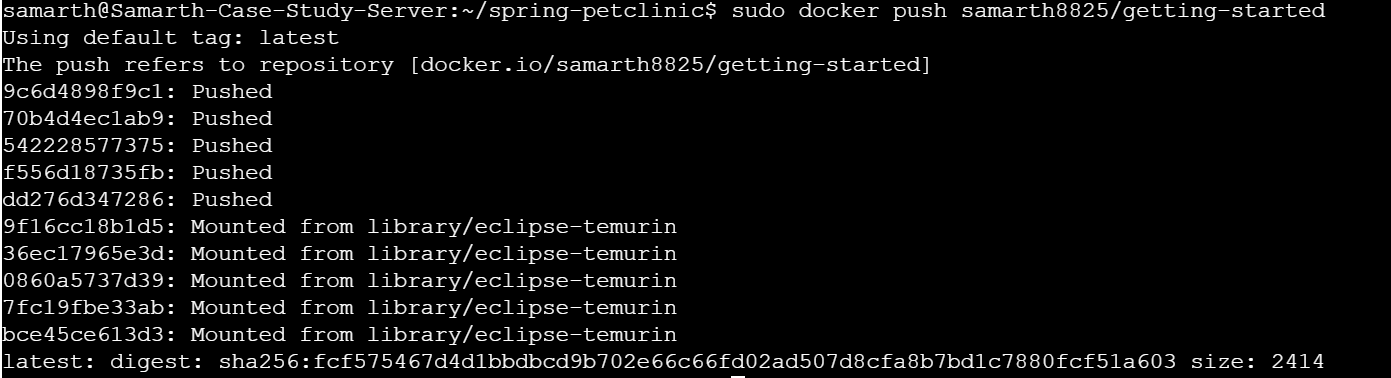
Now enter your username and password of docker hub in the command line using the command: ***sudo docker login -u USERNAME***

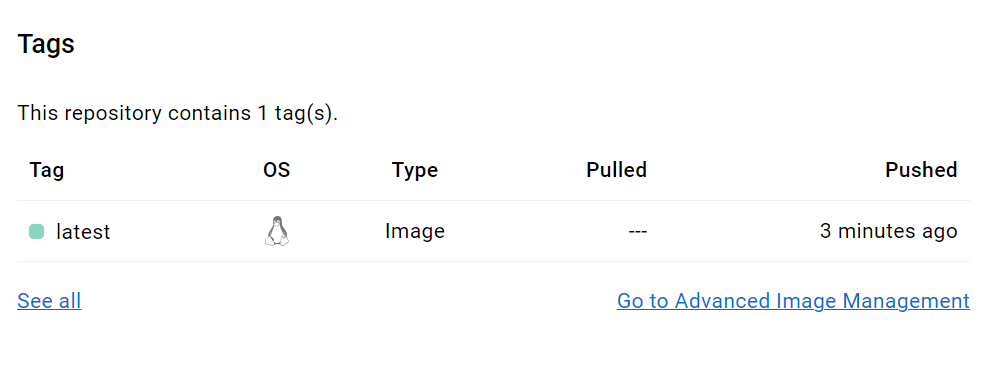


Once logged into the docker hub through the terminal enter the following command to queue the docker image: ***sudo docker tag java-docker:latest samarth8825/getting-started***



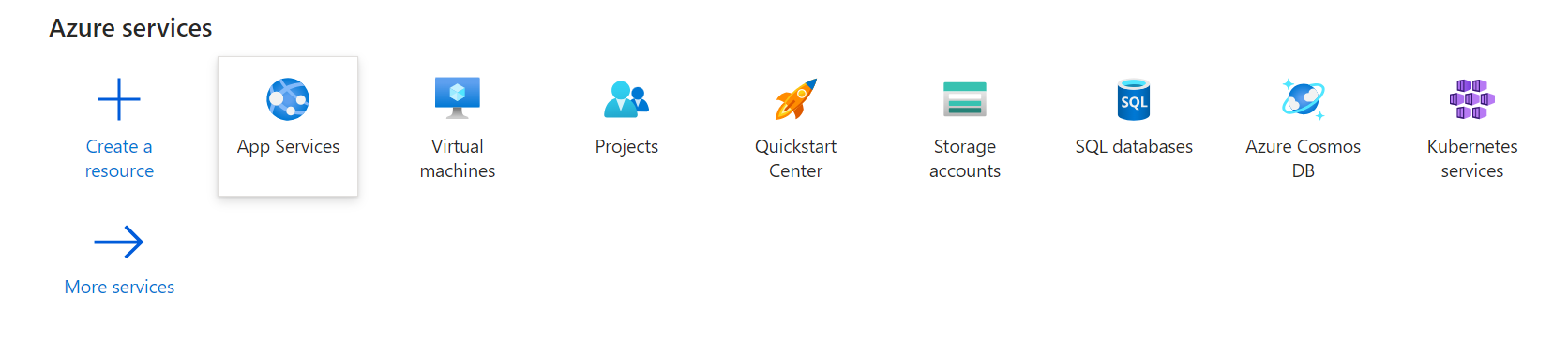
After queueing is done now we have to push it into the repository and to do that use the following command:***sudo docker push samarth8825/getting-started***

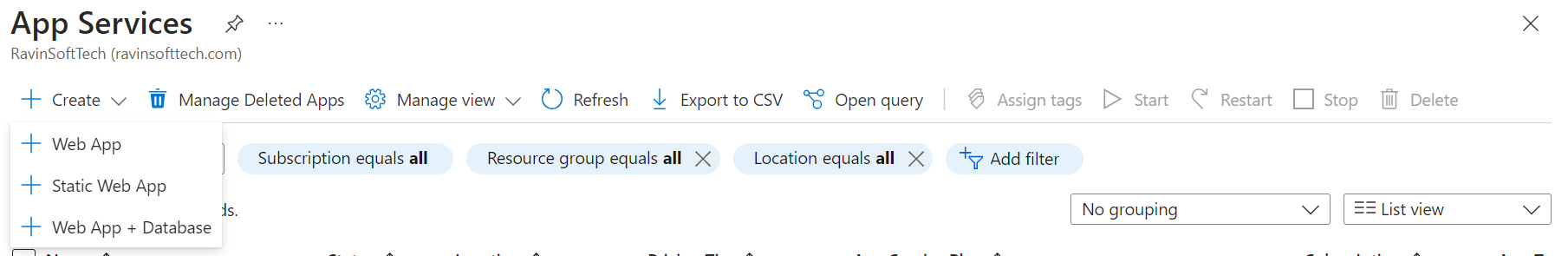




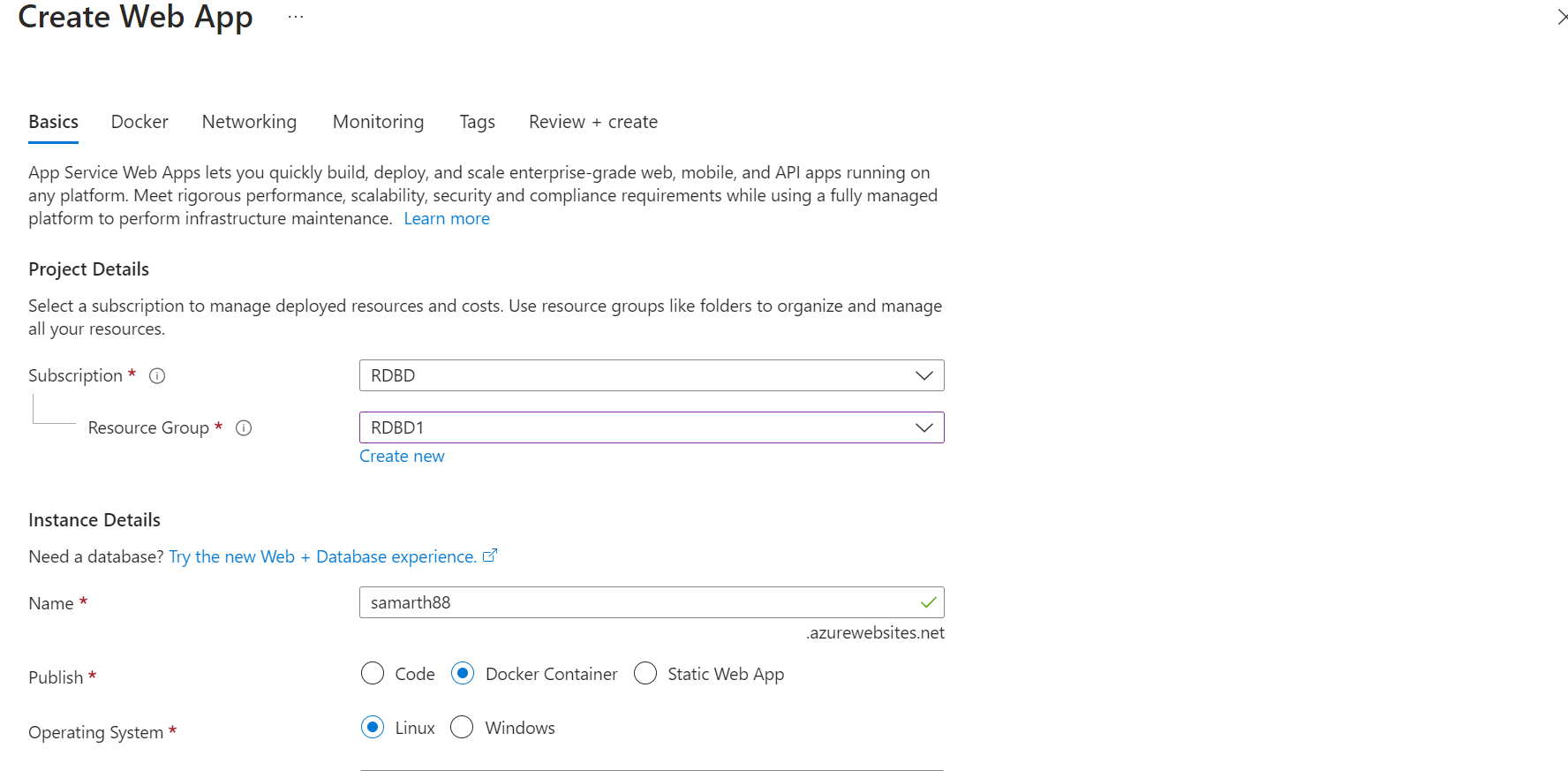
Once the image has been published onto the docker hub, Github repository has to be set. To do that a template was provided with the help of which we get a file named DockerFile.

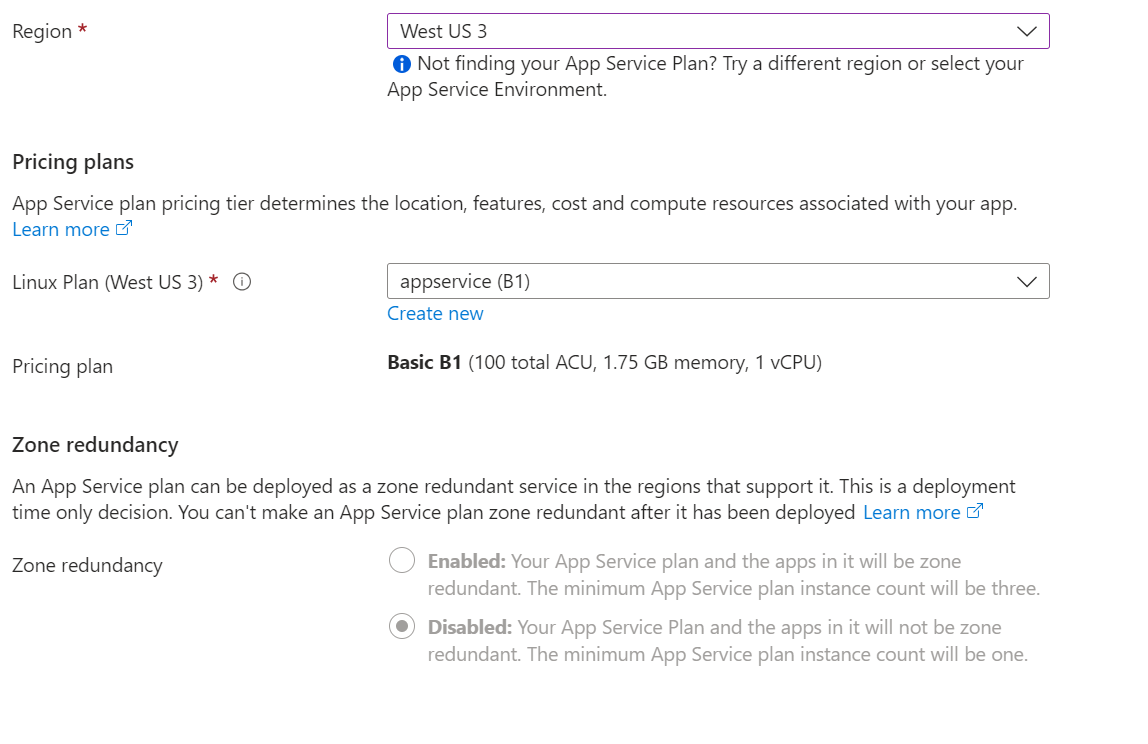
# Azure Web App Configuration

Click the Web App option under the Create option

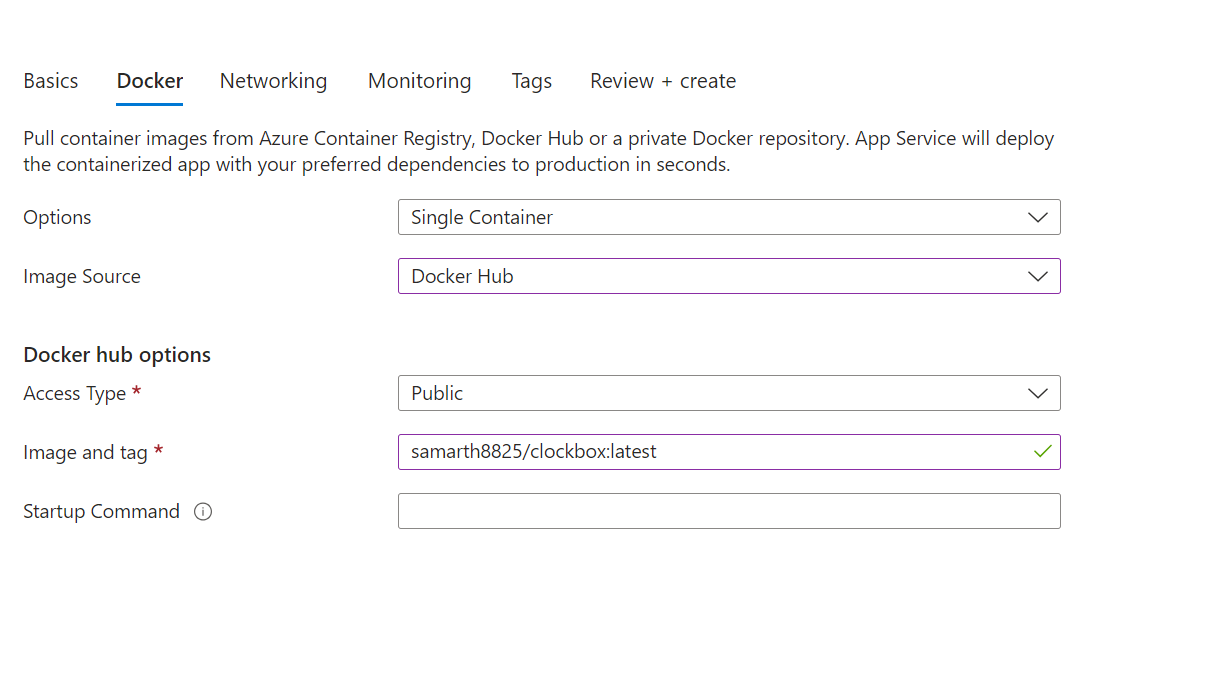


Fill the details as shown in the below images:





Then go to the Docker section and fill in the details

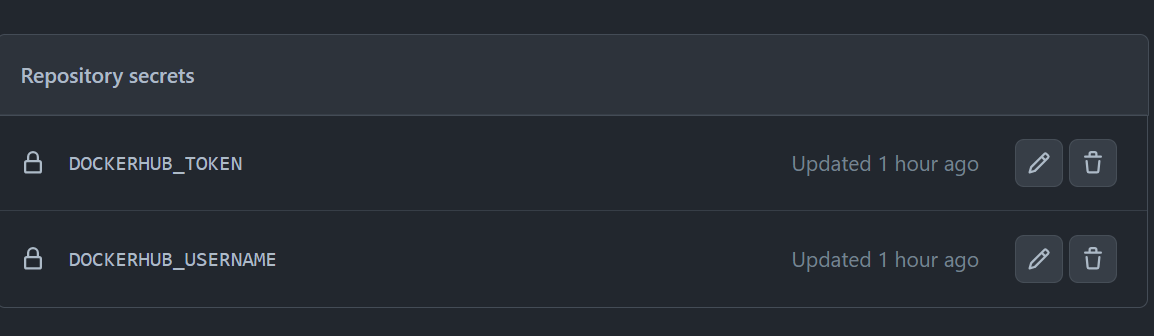


After filling the details in, click on the Create button to deploy the web app.

# GitHub Actions Workflow Implementation

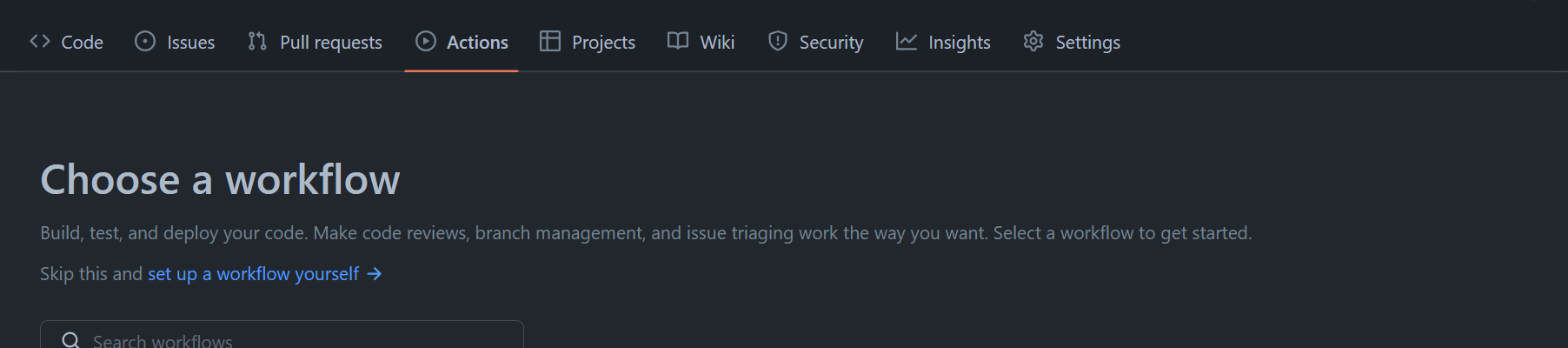
Go to the settings > Secrets and variables > Actions to set up the USERNAME and Personal Access Token.

To set up the personal access token or PAT go to Account Settings > Security to create PAT. Copy the PAT and paste it int the secret variable.

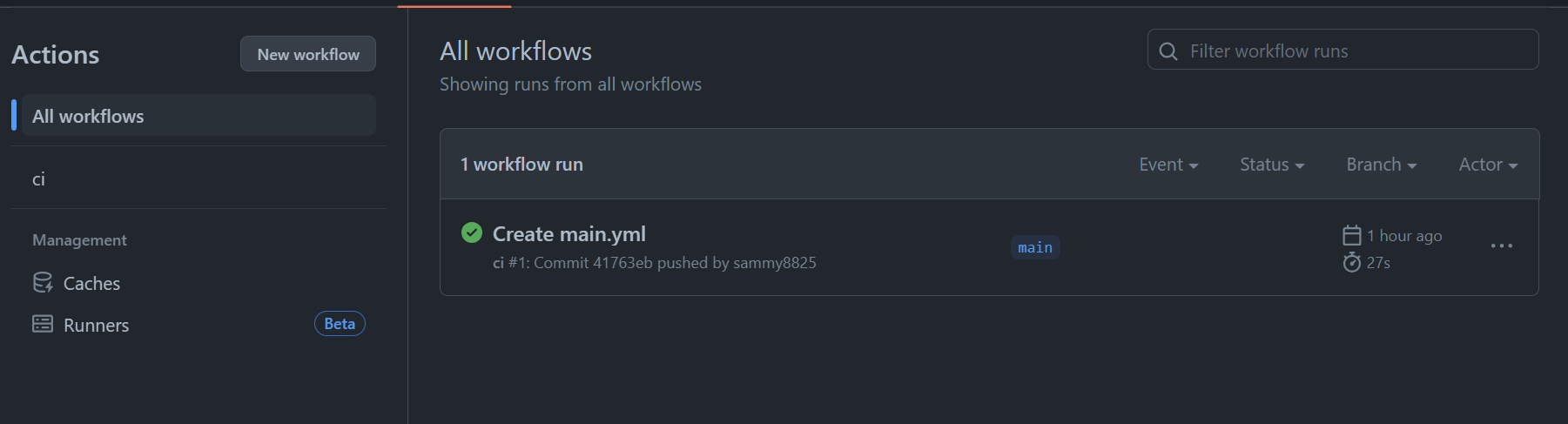


This is how it will look after the tokens have been updated.

After the setting up of variables go to Actions tab to set up the workflow.



Click on *set up workflow yourself* option and enter the contents into main.yml file and commit the changes.



After successfully setting up the workflow it will look like this.

To create a web app go to the app services option

# Maven-Based Build Process:

While installing Maven we encountered that our machine had Java 11 and it required Java 17.

Run the following commands to complete the process:

***sudo apt install maven***

***sudo apt update***

***git clone*** [***https://github.com/hrb1989/shell\_7\_foundation\_Java\_Spring.git***](https://github.com/hrb1989/shell_7_foundation_Java_Spring.git)

***cd shell\_7\_foundation\_Java\_Spring***

***mvn install -DskipTests***

***sudo apt install -y openjdk-17-jdk***

***docker build -t <name> -f /home/azureuser/shell\_7\_foundation\_Java\_Spring/Dockerfile***

***vi Dockerfile***

***docker build --tag samarth8825:latest .***

***sudo docker build --tag samarth8825:latest***

***sudo docker login***

***sudo docker images***

***sudo docker tag samarth8825:latest samarth8825/case-study-docker***

***sudo docker push samarth8825/case-study-docker***

# Outcome

As a result, you will get the following result.

