**Shell- Final Case Study Assessment**

**Activities:**

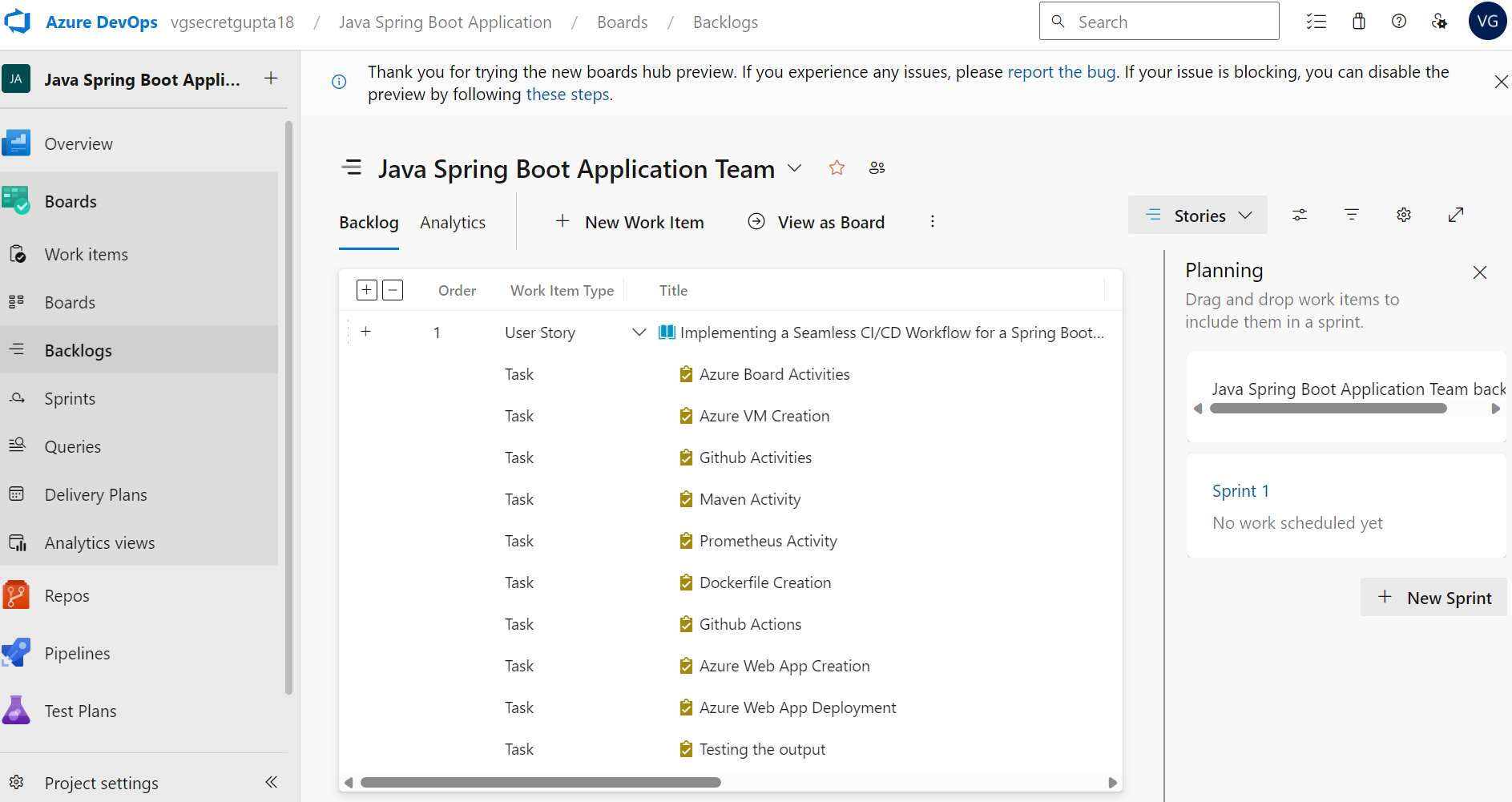
1. Project Management Setup:

**Step-1:** Go to azure devops and create a new project.

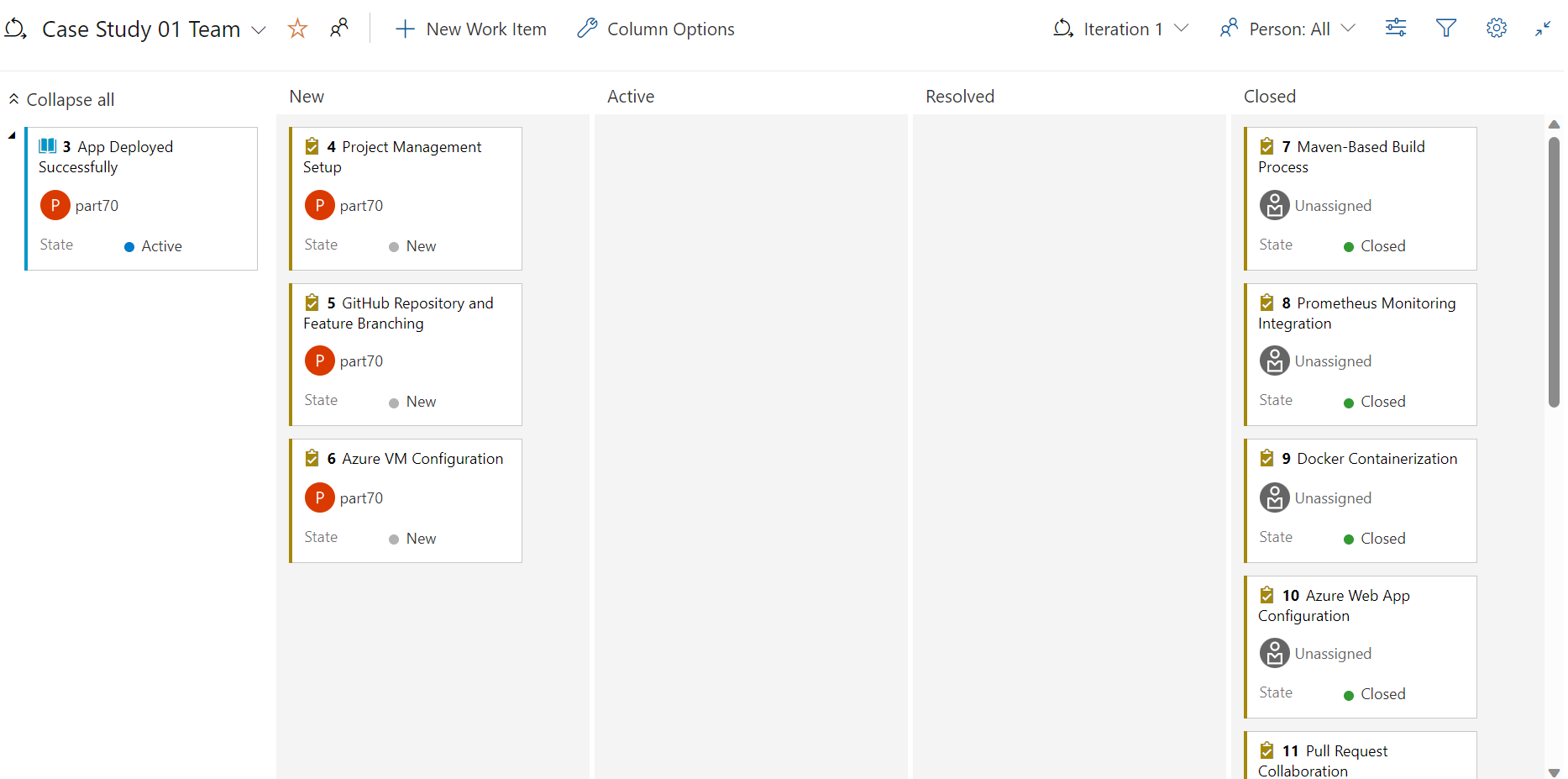
**Step-2:** Add epic, features and tasks according to the problem statement.

**Step-3:** Link them with the project iteration.

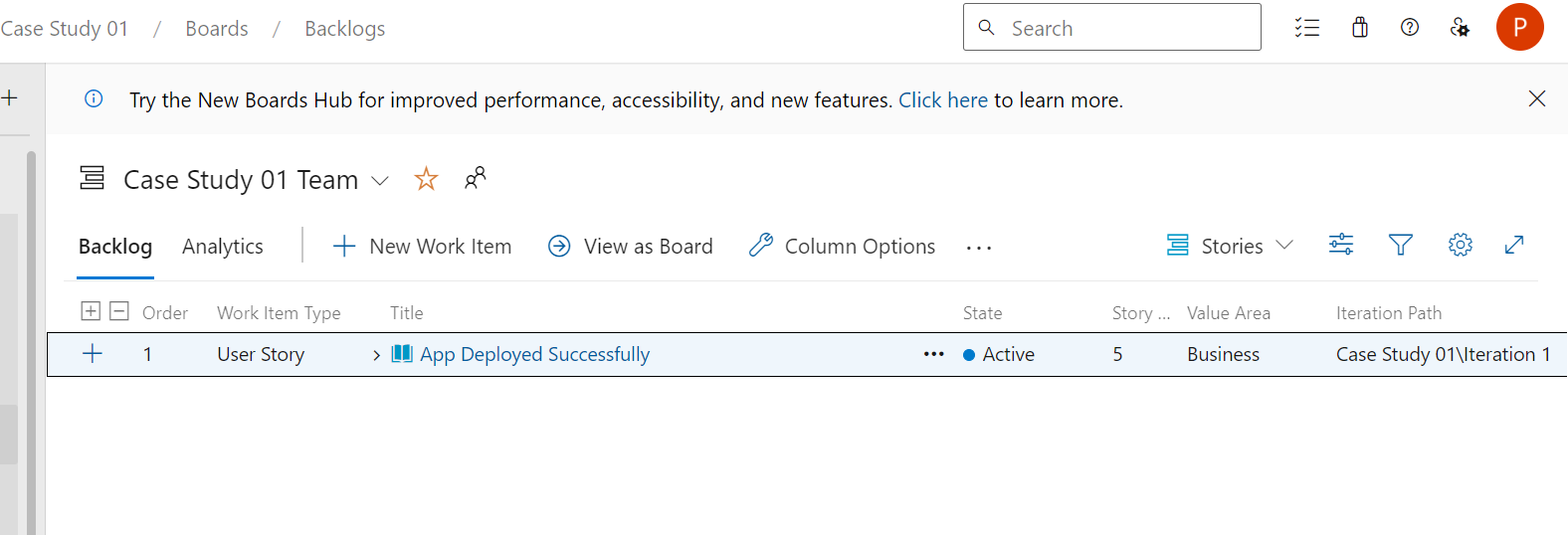
****

****

**Task board:**

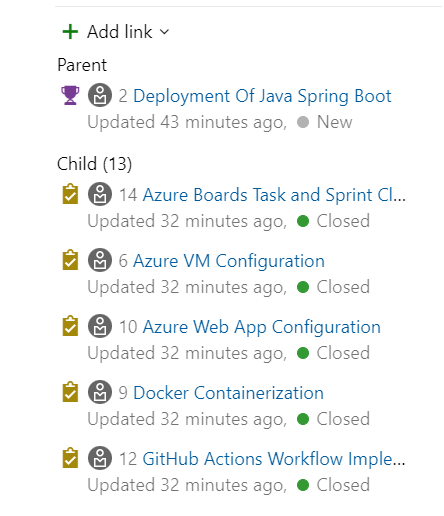


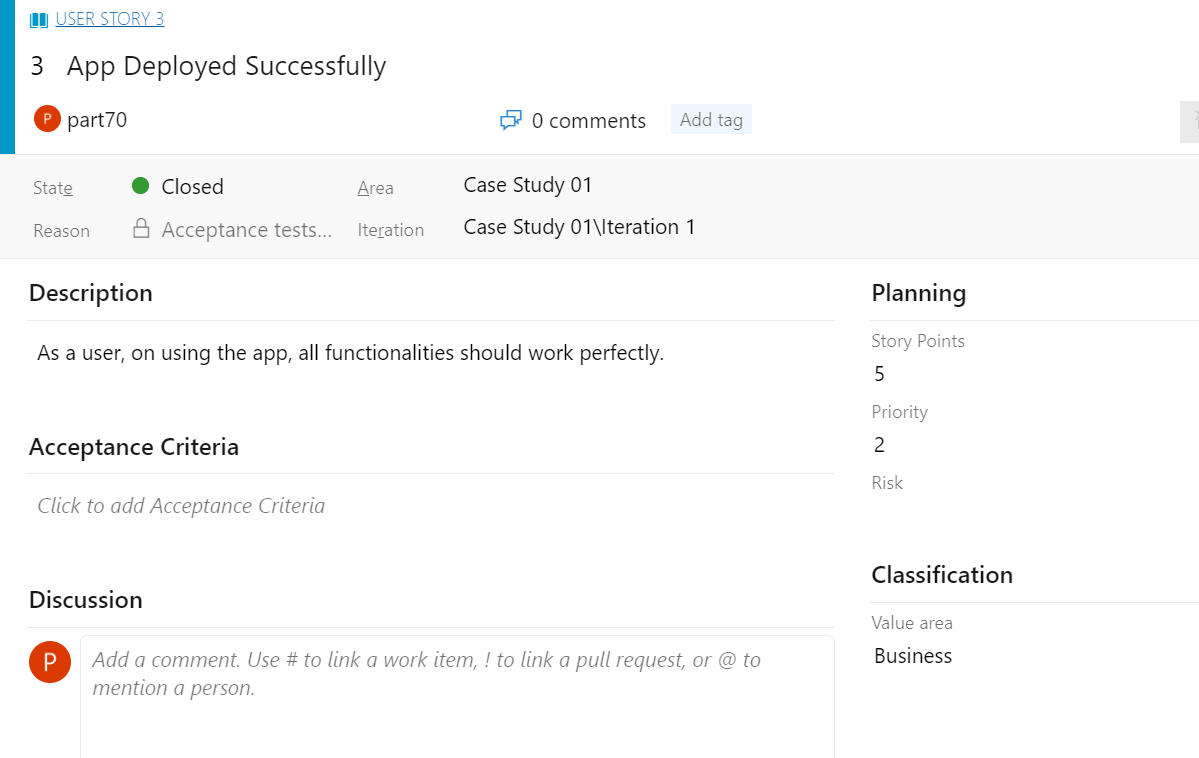
**Project backlog:**



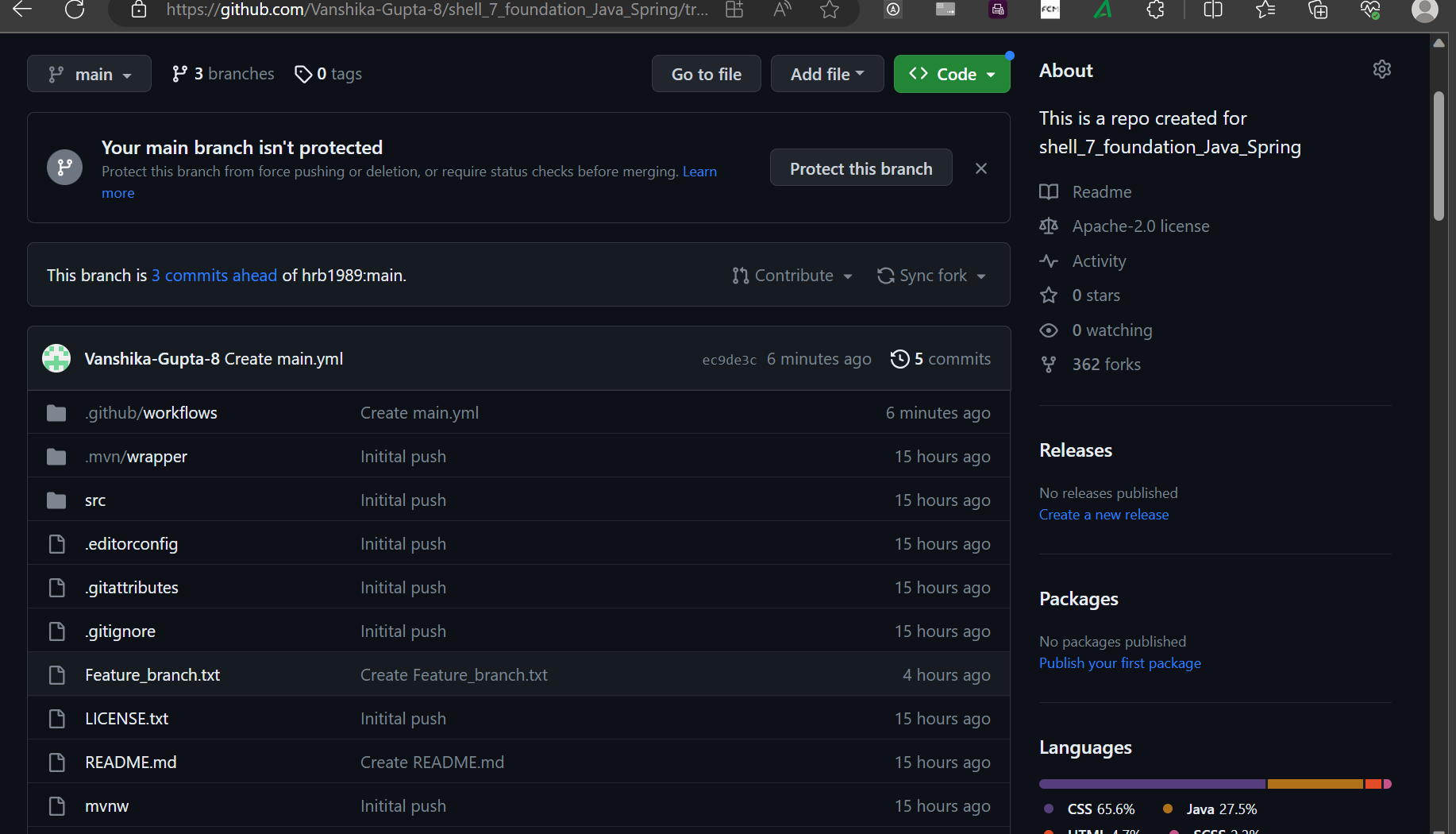
**Project board:**







1. GitHub repository and feature branching:



1. Azure VM Configuration:





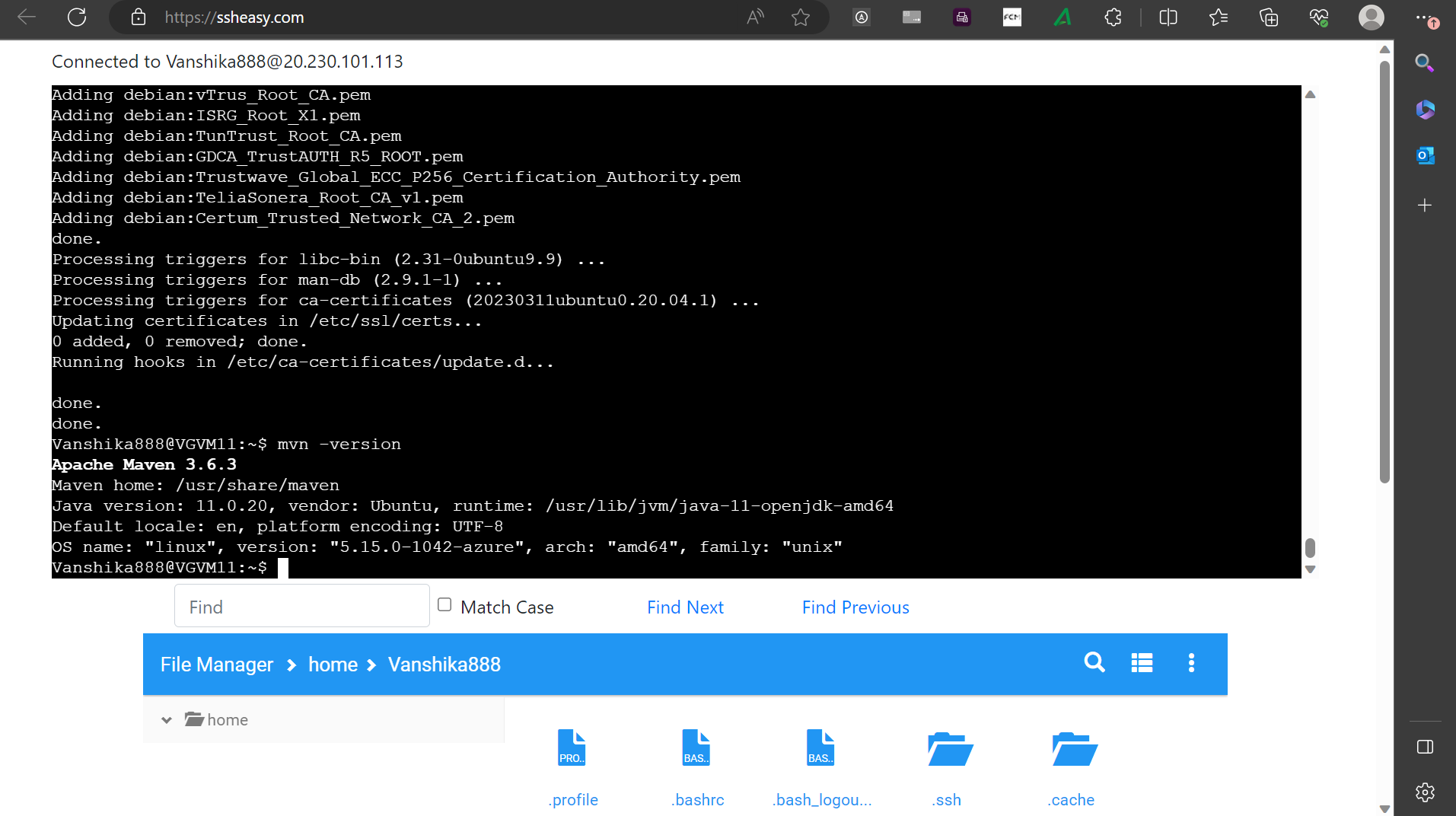
1. Maven-Based Build Process:

Installing Maven-

**Step-1:** sudo apt update

**Step-2:** sudo apt install maven

**Step-3:** mvn -version



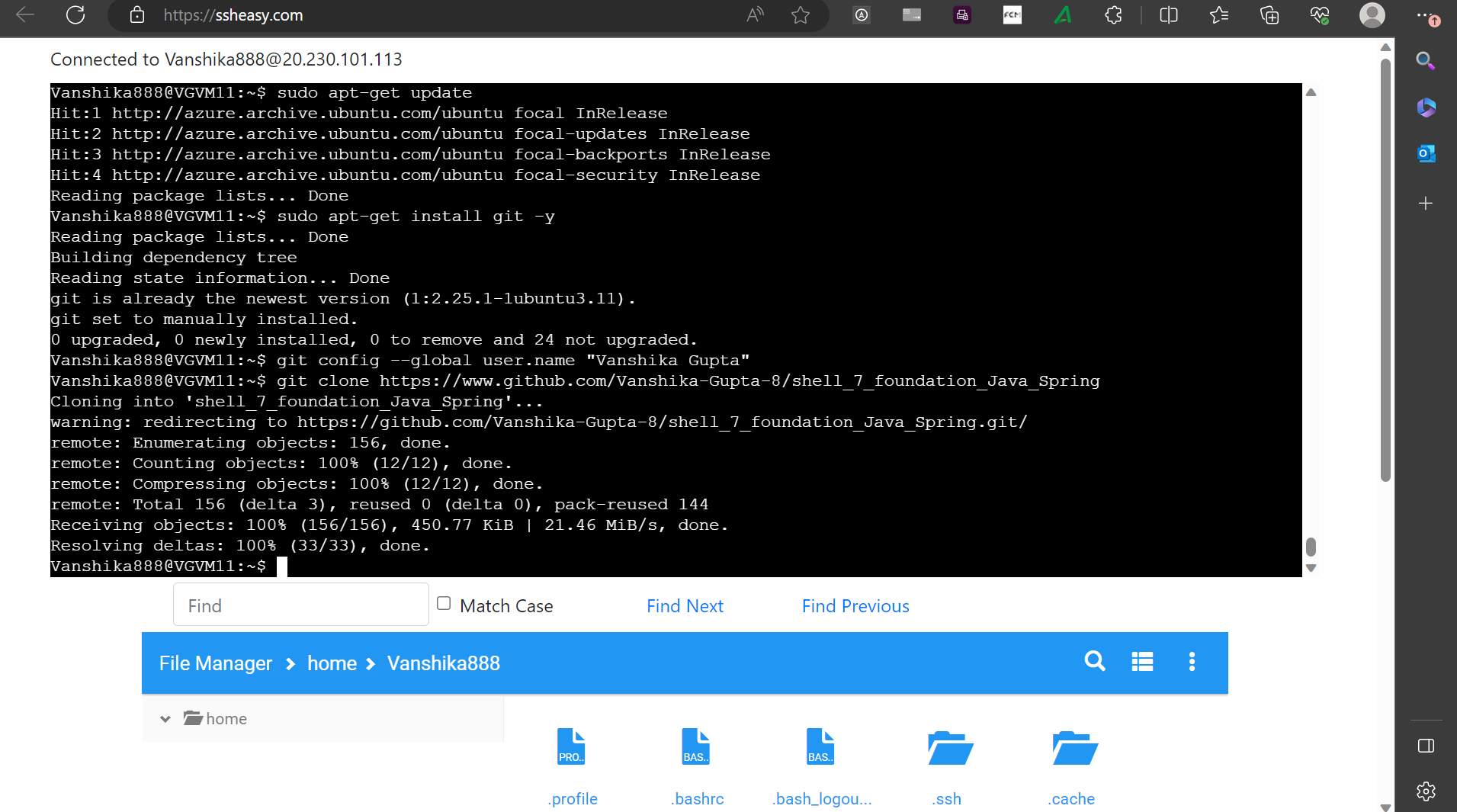
Installing git and cloning the GitHub repository-

**Step-4:** sudo apt-get update

**Step-5:** sudo apt-get install git -y

**Step-6:** config --global user.name “Vanshika Gupta”

**Step-7:** git clone **https://www.github.com/Vanshika-Gupta-8/shell\_7\_foundation\_Java\_Spring**



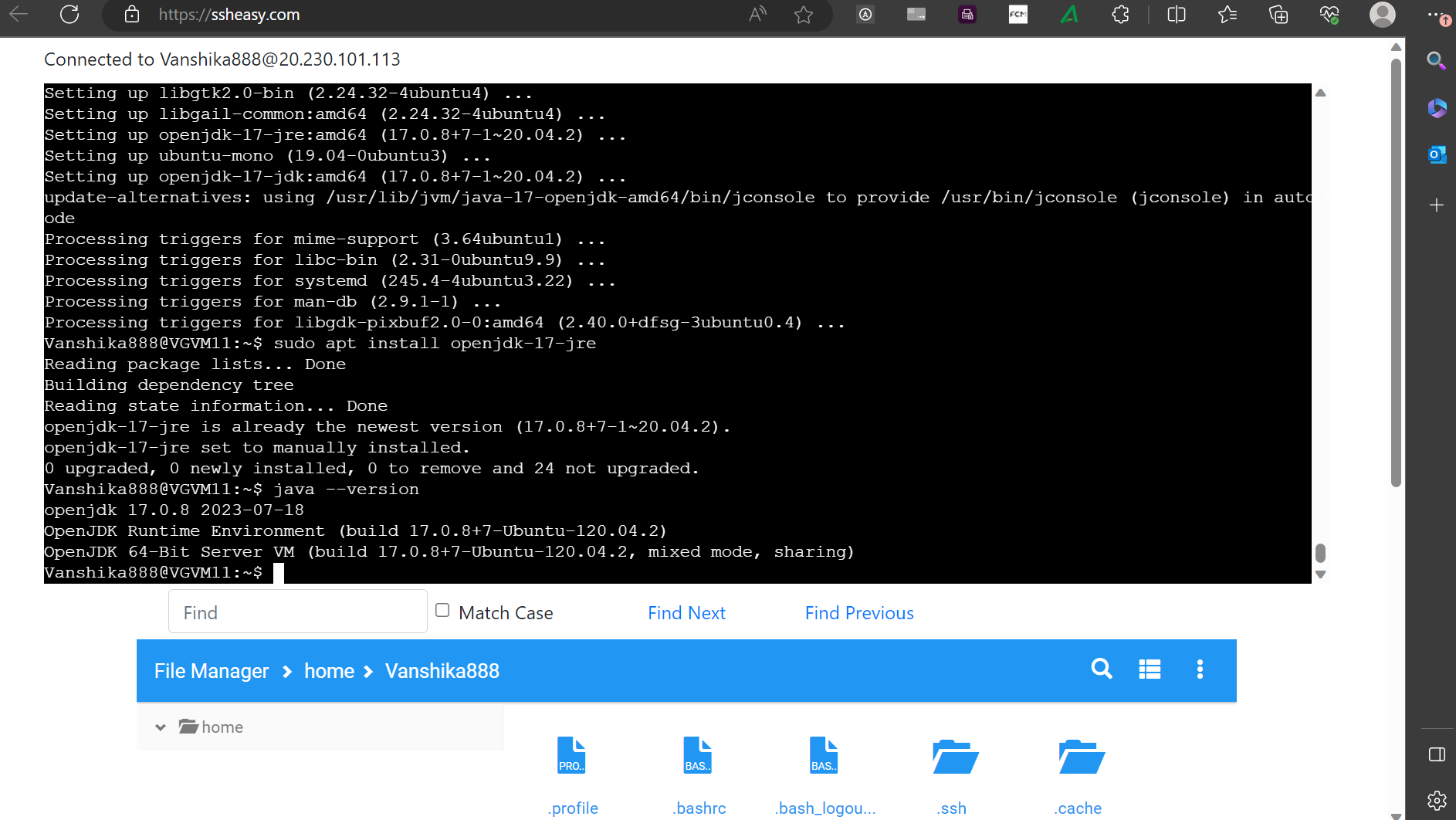
Install Java17-

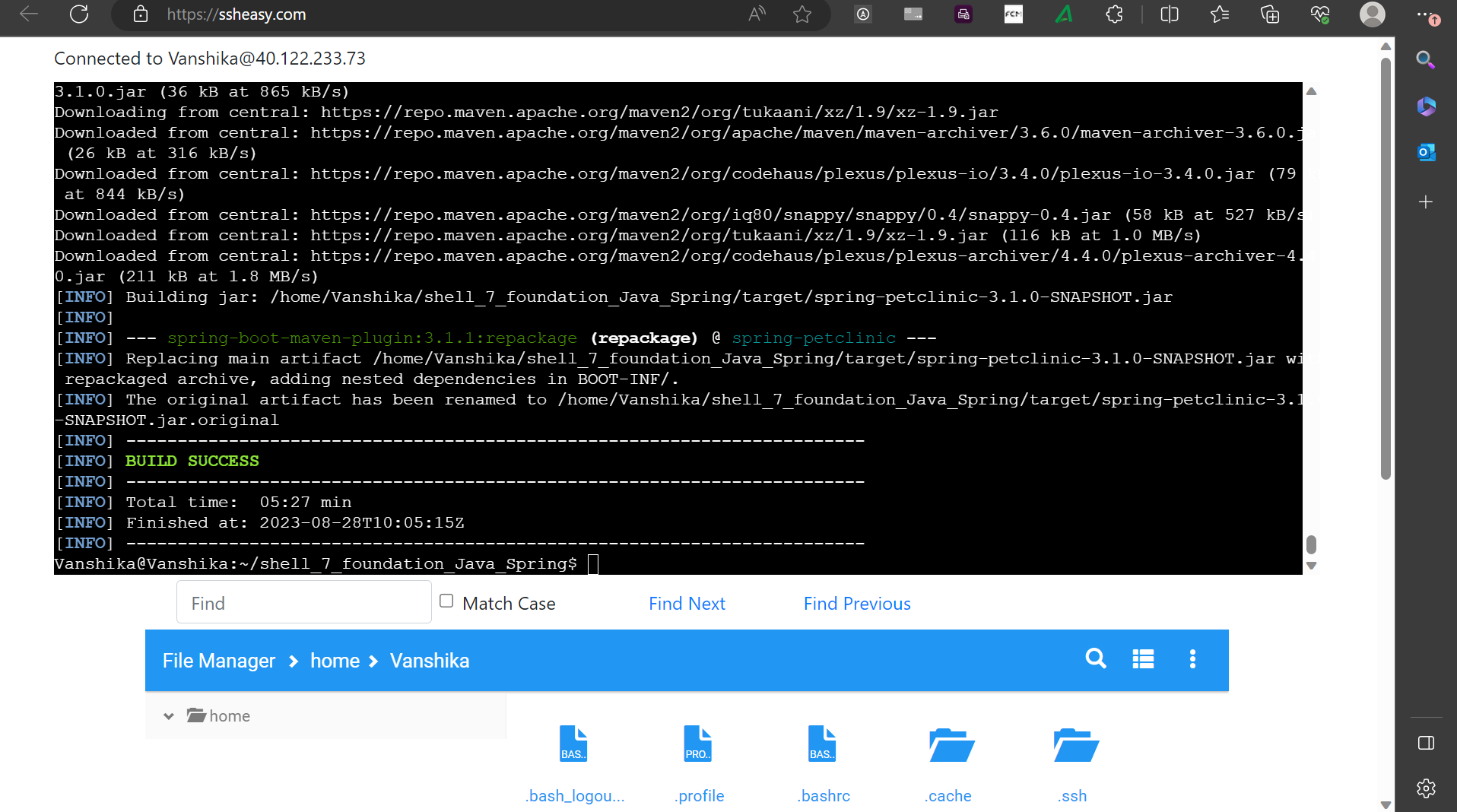
**Step-8:** sudo apt update

**Step-9:** sudo apt install openjdk-17-jdk

**Step-10: s**udo apt install openjdk-17-jre

**Step-11:** java --version



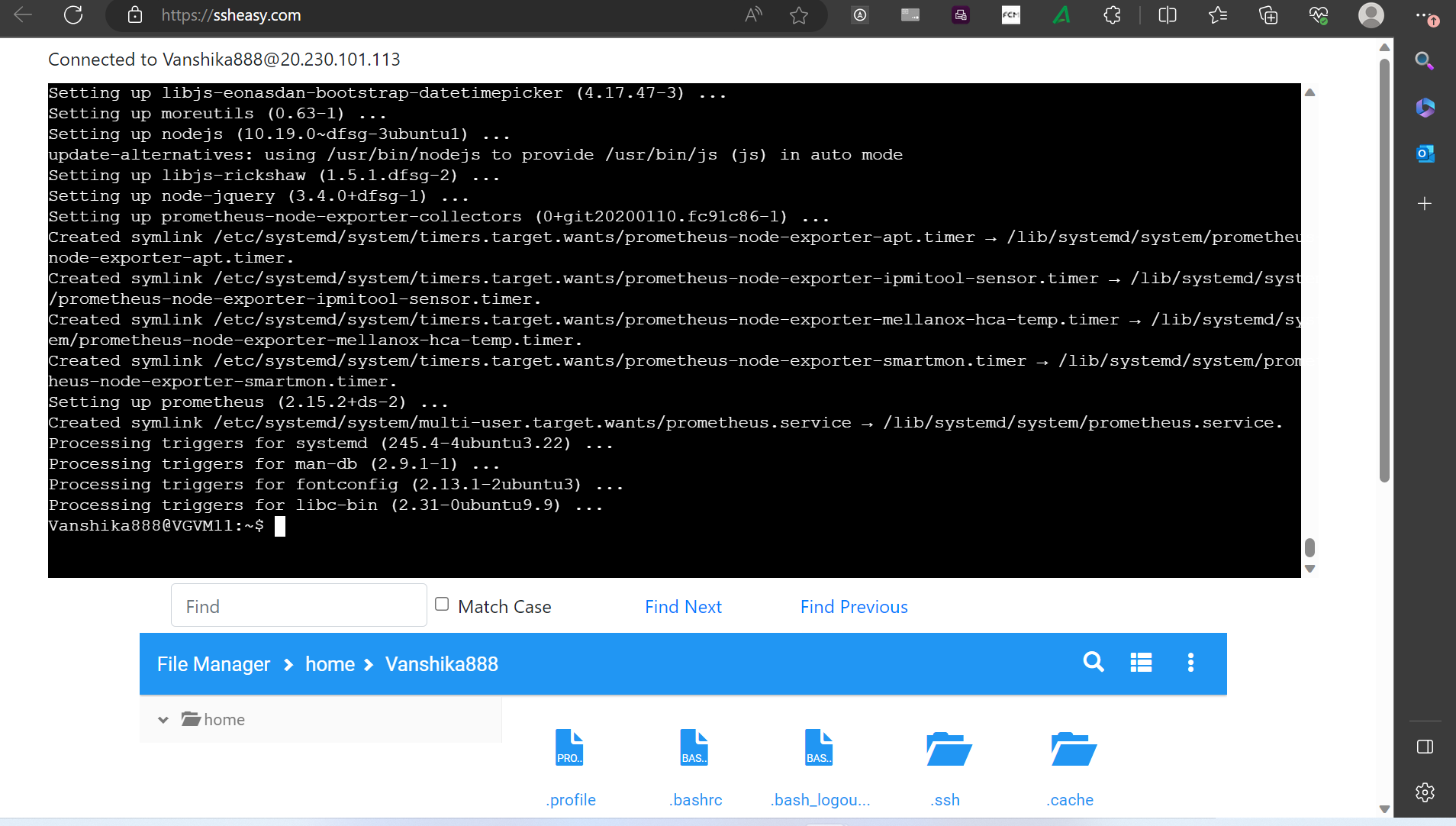


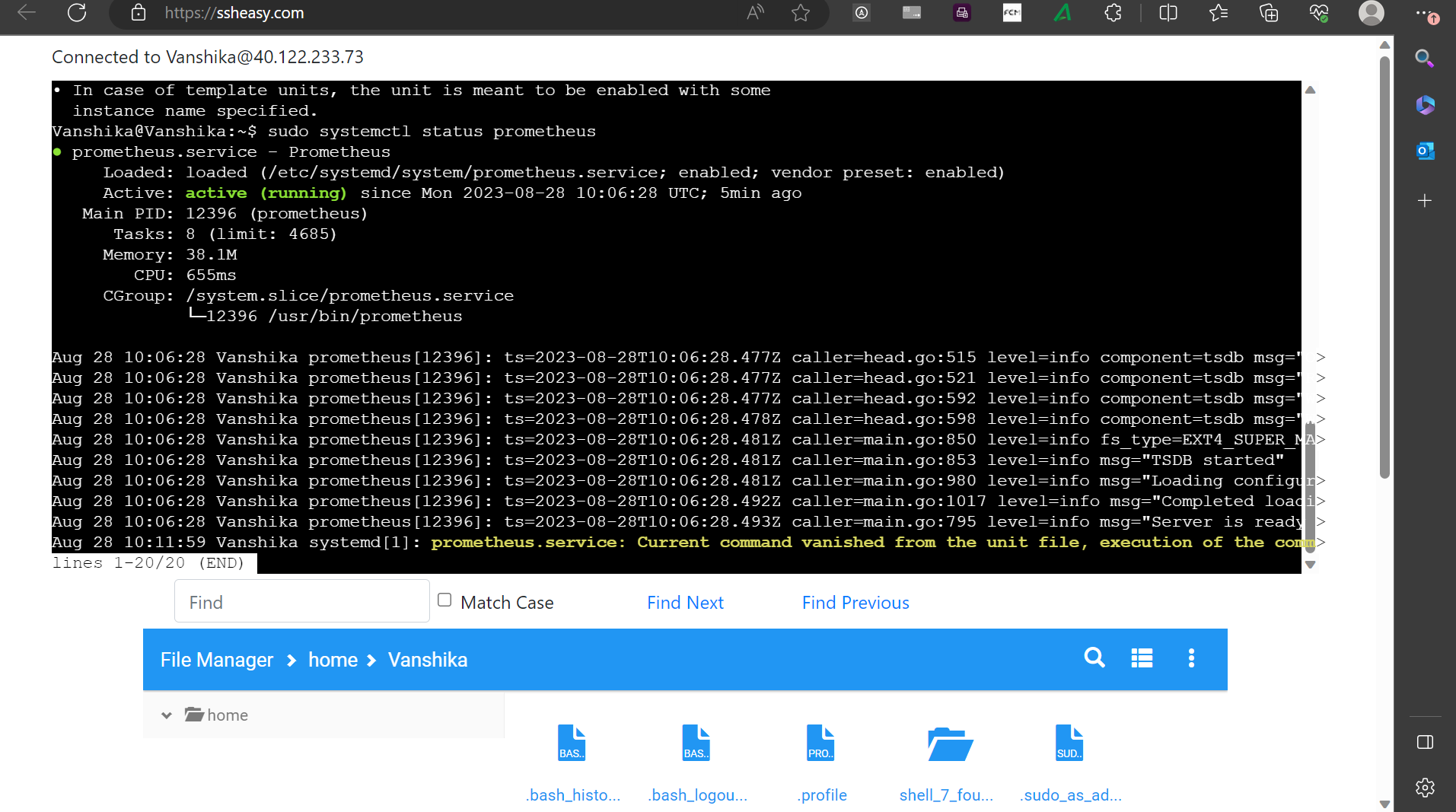
1. Prometheus Monitoring Integration:

Installing Prometheus-

**Step-1:** sudo apt update

**Step-2:** sudo apt install prometheus



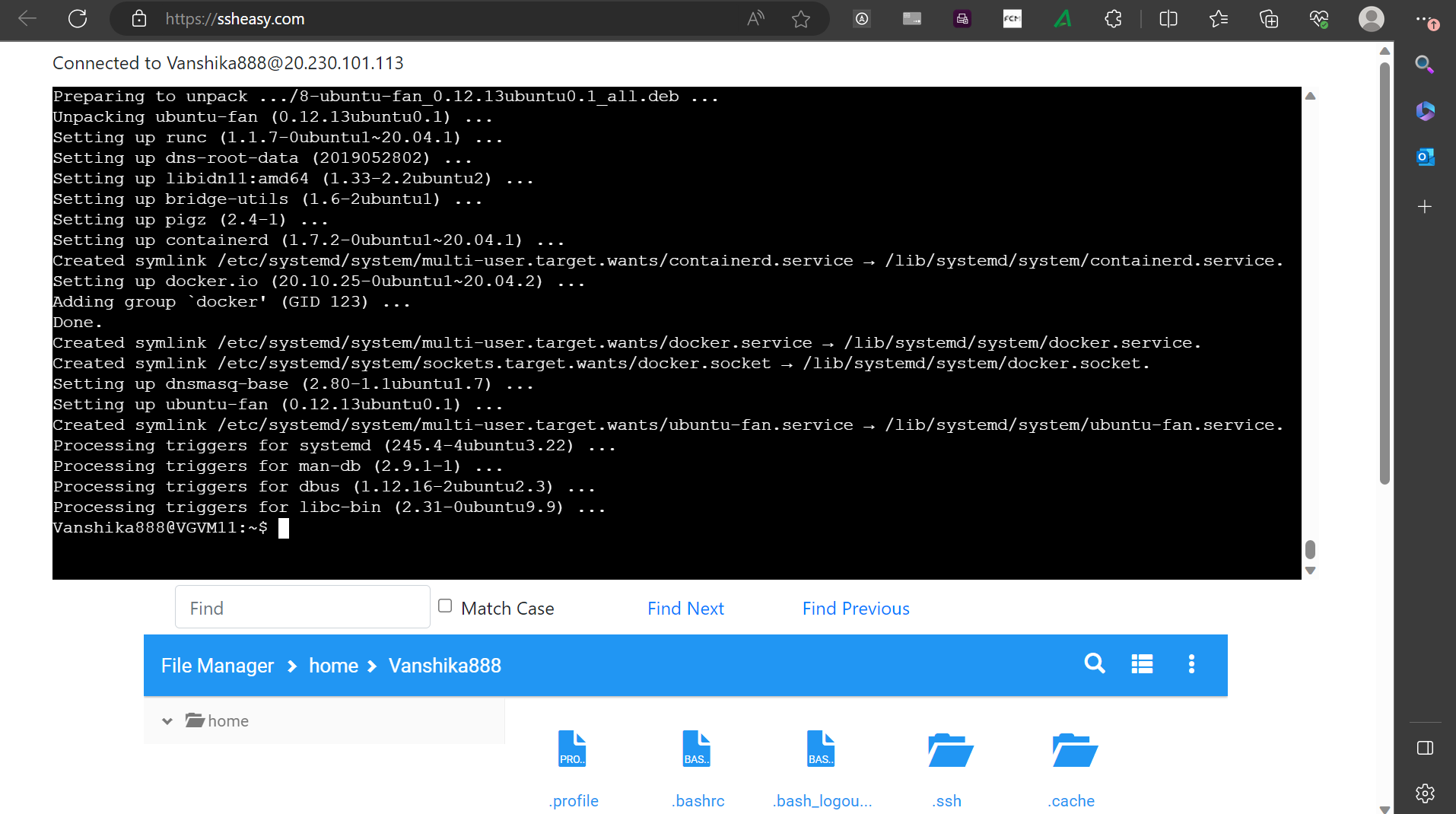


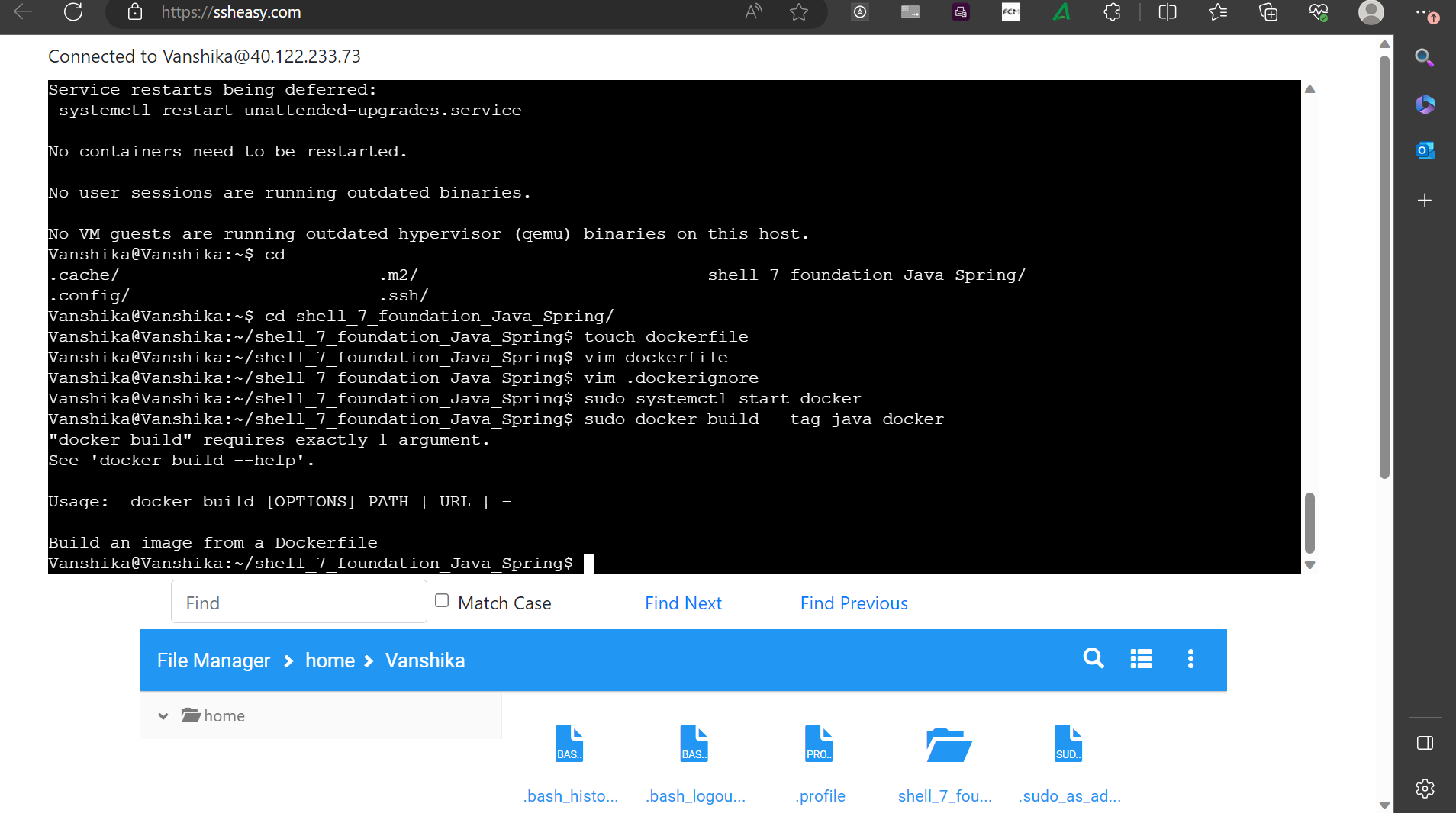
1. Docker containerization:

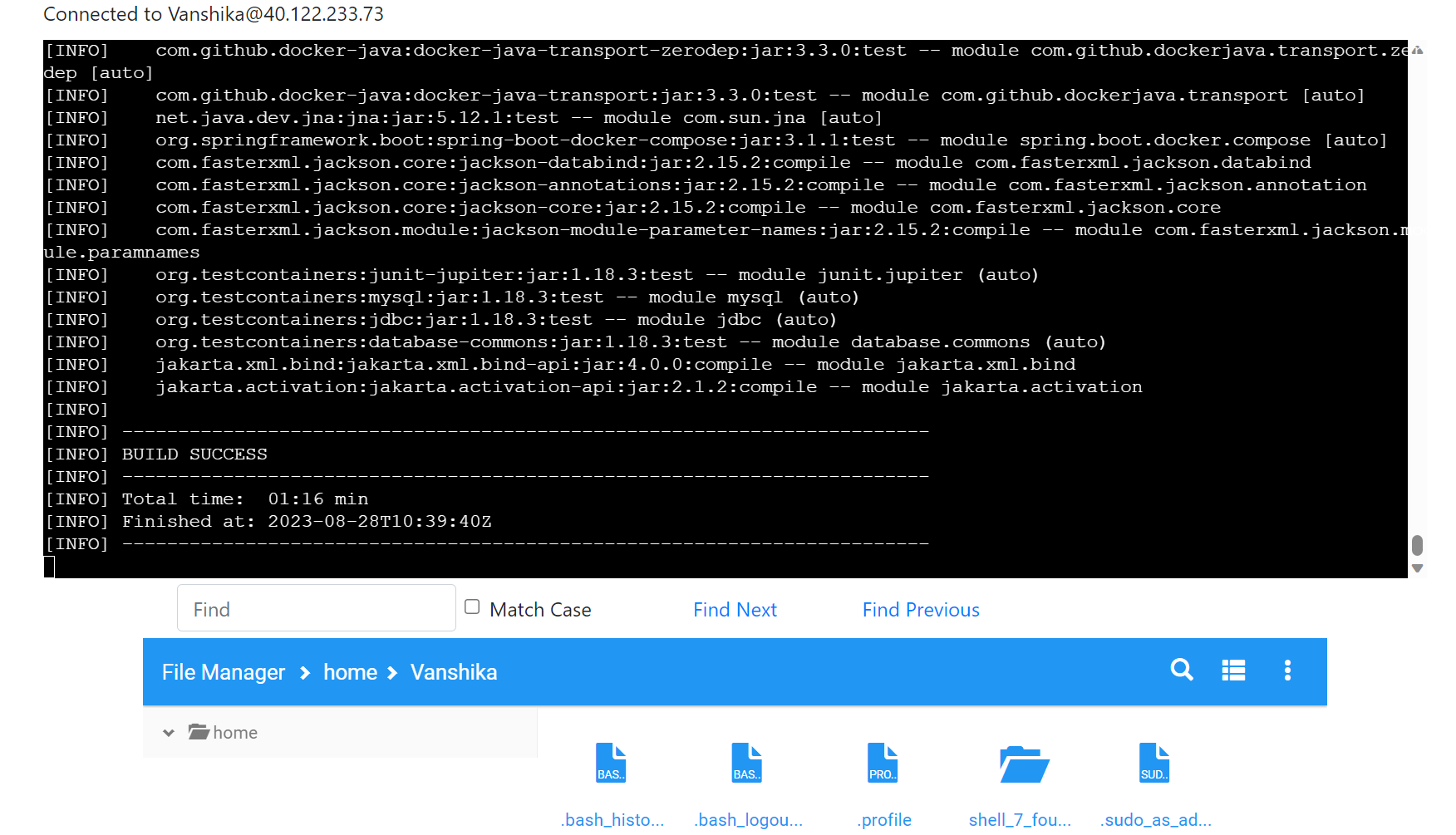
Installing docker-

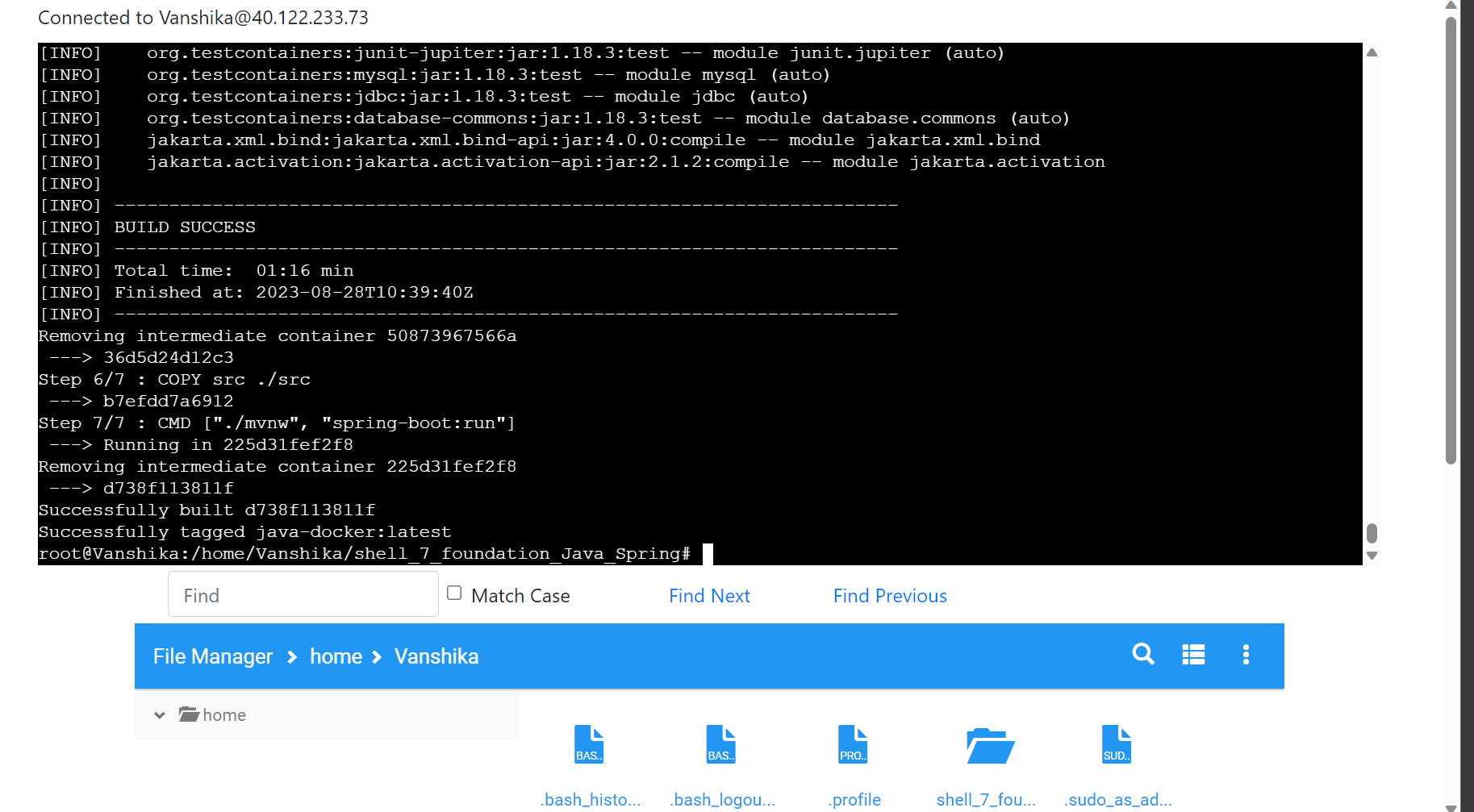
**Step-1:** sudo apt update

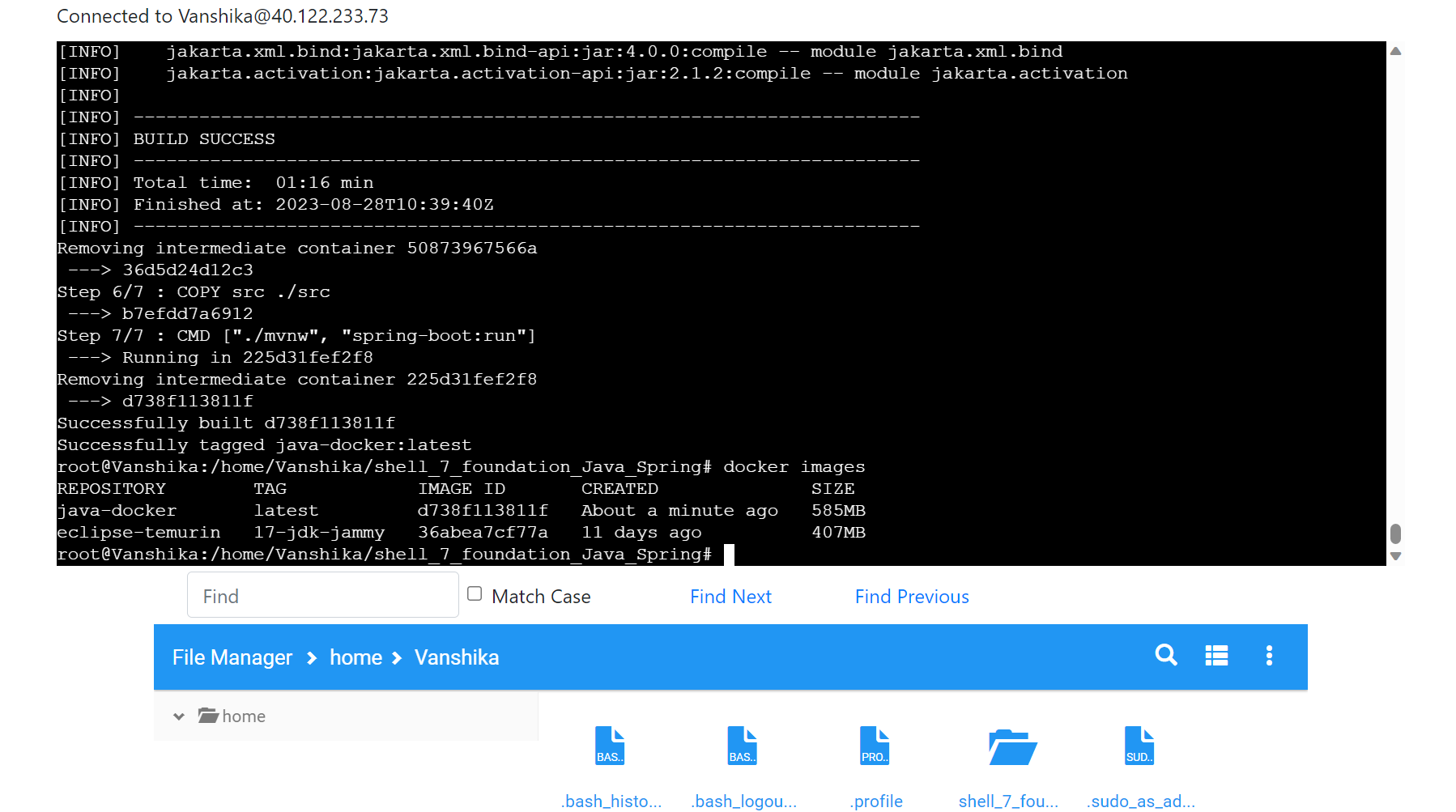
**Step-2:** sudo apt install docker.io -y



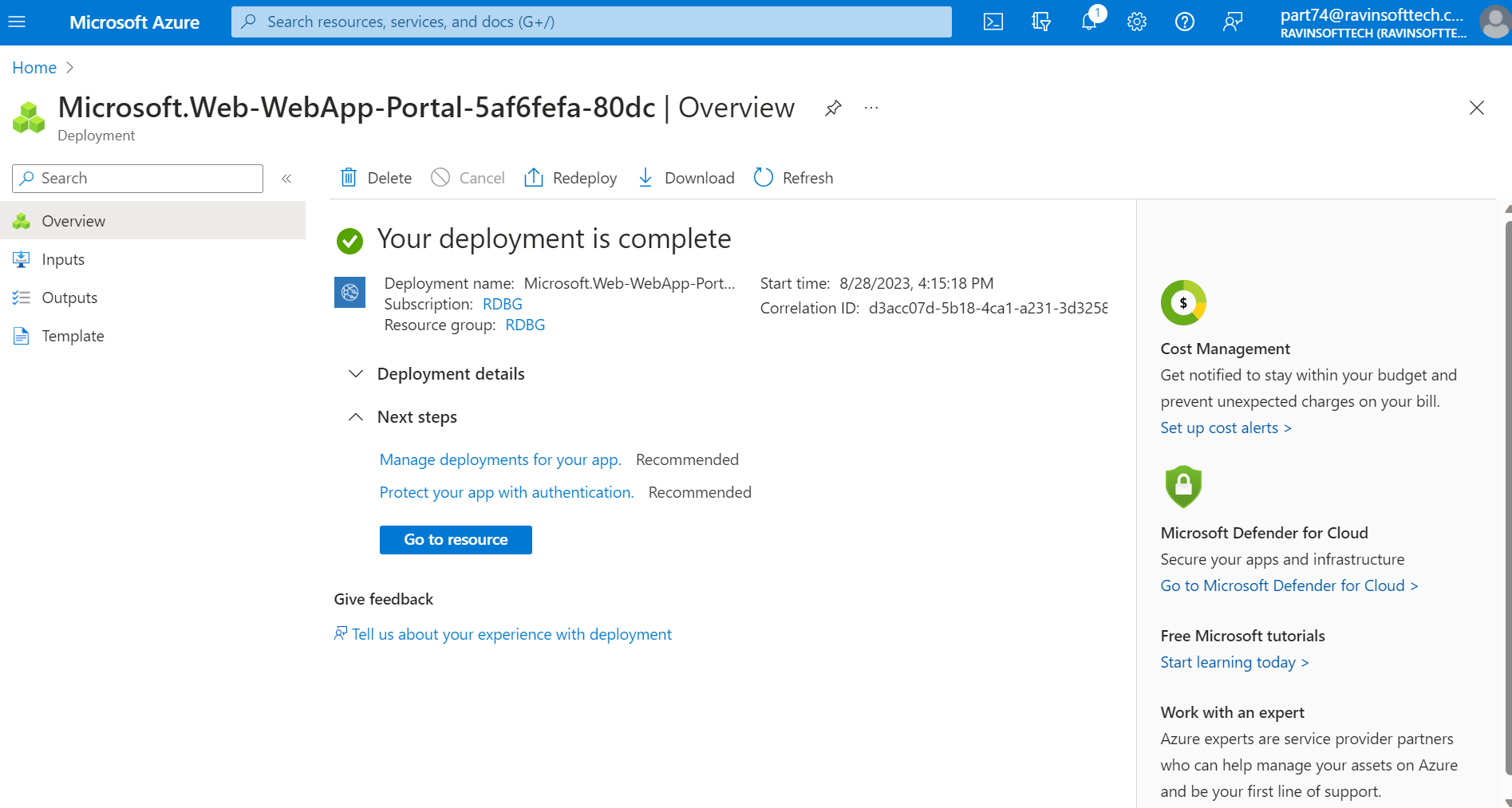


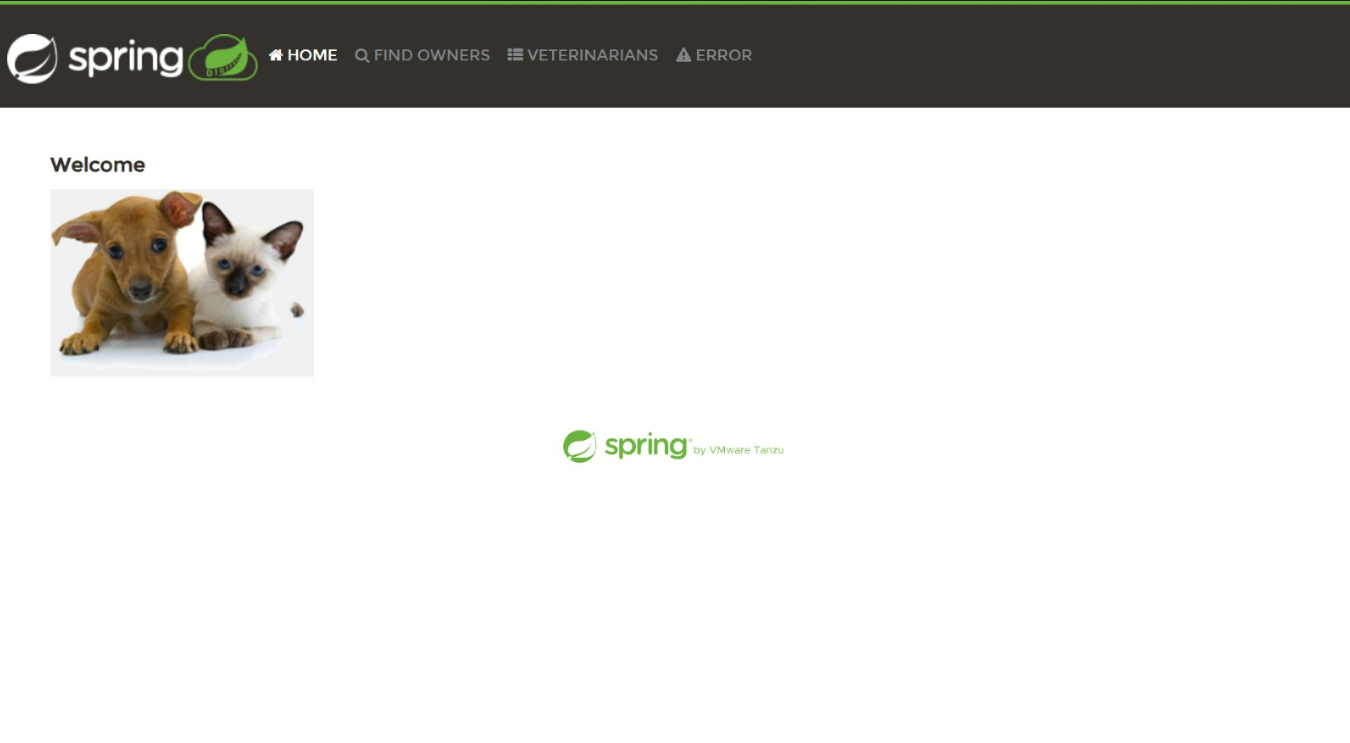






1. Azure Web App Configuration:





1. Github Actions Workflow Implementation:

**Step 1:** uses: actions/checkout@v2

- name: Set up Docker Buildx

uses: docker/setup-buildx-action@v2

- name: Log in to container registry

uses: docker/login-action@v2

with:

registry: https://index.docker.io/v1/

username: ${{

secrets.AzureAppService\_ContainerUsername\_aaa07ab7057541748b79e07390bd9ed0 }}

password: ${{

secrets.AzureAppService\_ContainerPassword\_f4492f67520e4da6ba8258c0d107c282 }}

- name: Build and push container image to registry

uses: docker/build-push-action@v3

**Step 2:** with: context: .

push: true

tags: index.docker.io/${{

secrets.AzureAppService\_ContainerUsername\_aaa07ab7057541748b79e07390bd9ed0

}}/hrb1989/javasp1:${{ github.sha }}

file: ./Dockerfile

**Step 3:** deploy:

runs-on: ubuntu-latest

needs: build

environment:

name: 'production'

url: ${{ steps.deploy-to-webapp.outputs.webapp-url }}

steps:

- name: Deploy to Azure Web App

id: deploy-to-webapp

uses: azure/webapps-deploy@v2

with:

app-name: 'shell7SB1'

slot-name: 'production'

publish-profile: ${{

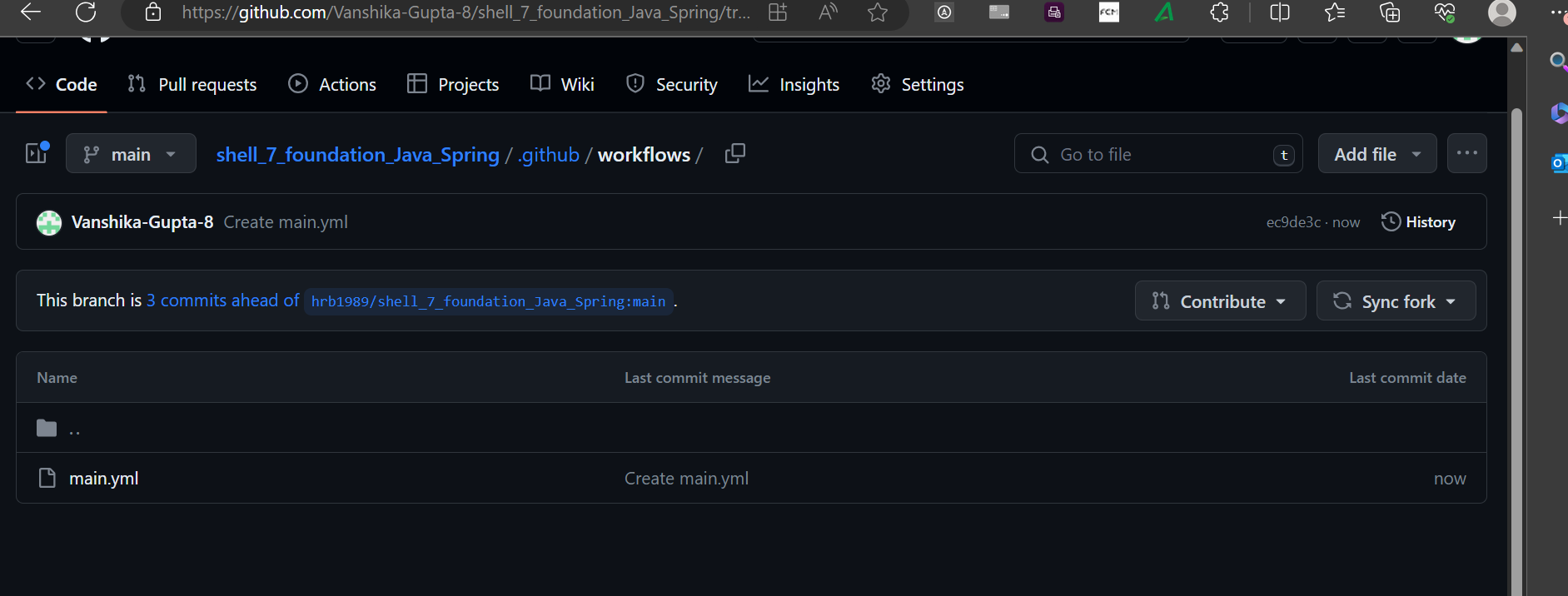
secrets.AzureAppService\_PublishProfile\_0425239420454e2d87725d19a92feb3d }}

images: 'index.docker.io/${{

secrets.AzureAppService\_ContainerUsername\_aaa07ab7057541748b79e07390bd9ed0

}}/hrb1989/javasp1:${{ github.sha }}





1. Outcome

I have learnt about how a sprint works and having implemented the app in a strict time constraint, I have learnt about time management as well as how to work under pressure. I learnt about deploying an app on azure, created a dockerfile and docker image, devops, agile, git, github workflow, etc. Having systematically implemented the CI/CD workflow, Streamline Solutions Inc. aims to achieve rapid and reliable software deployments. The company's Java Spring Boot application will reach customers seamlessly, ensuring a high-quality user experience and setting the stage for future innovation.