**Deploy Code to GitHub via Git​​​​​​**

This section will guide us to:

* Install Git and set up your GitHub account
* Execute the most popular commands in Git
* Push all the files from the local repository to GitHub

**Install Git**

Git is already installed in our lab. we can check the version of git by executing the below command in the terminal. : **git -version**

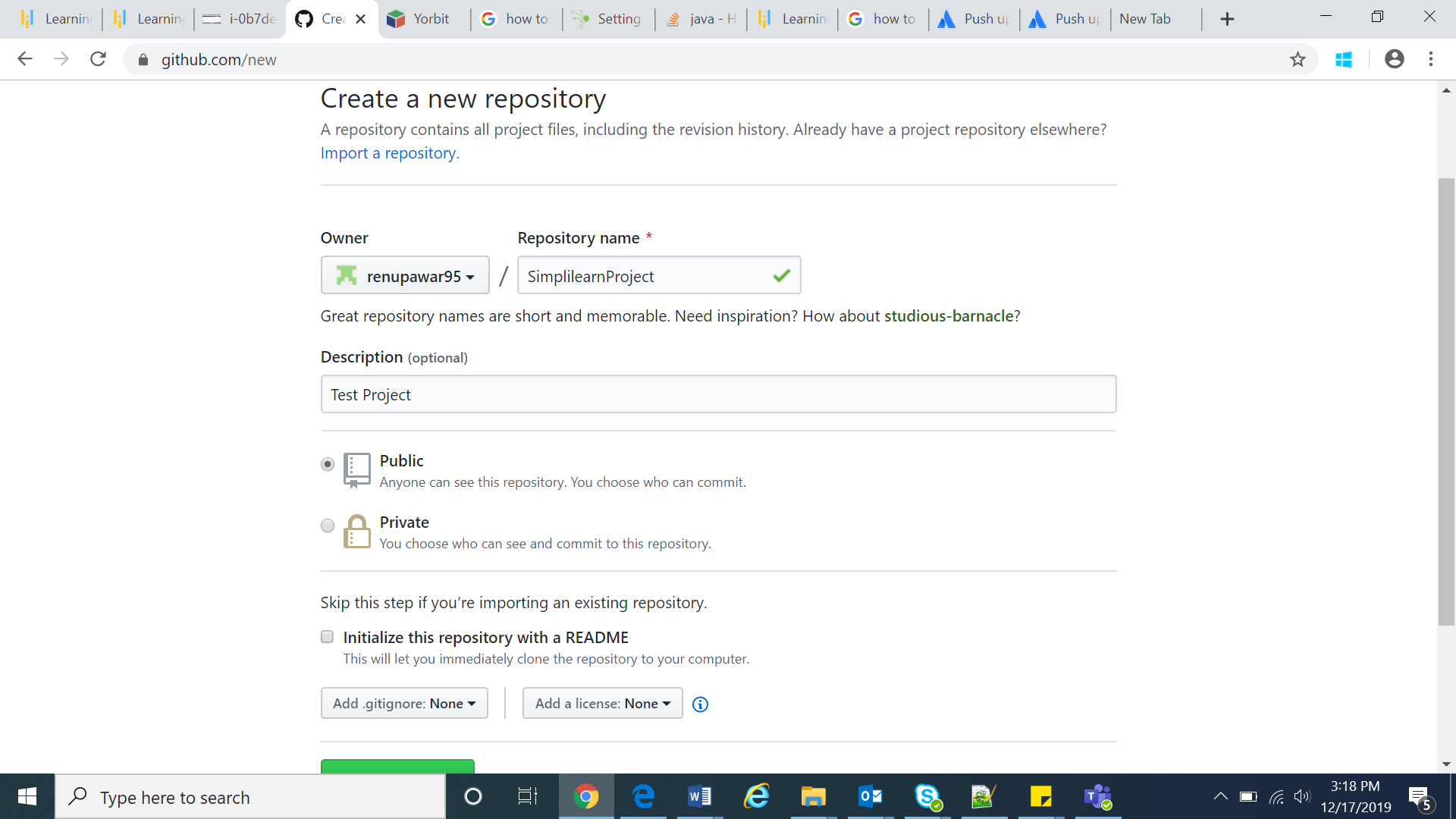
**Set up your GitHub account**

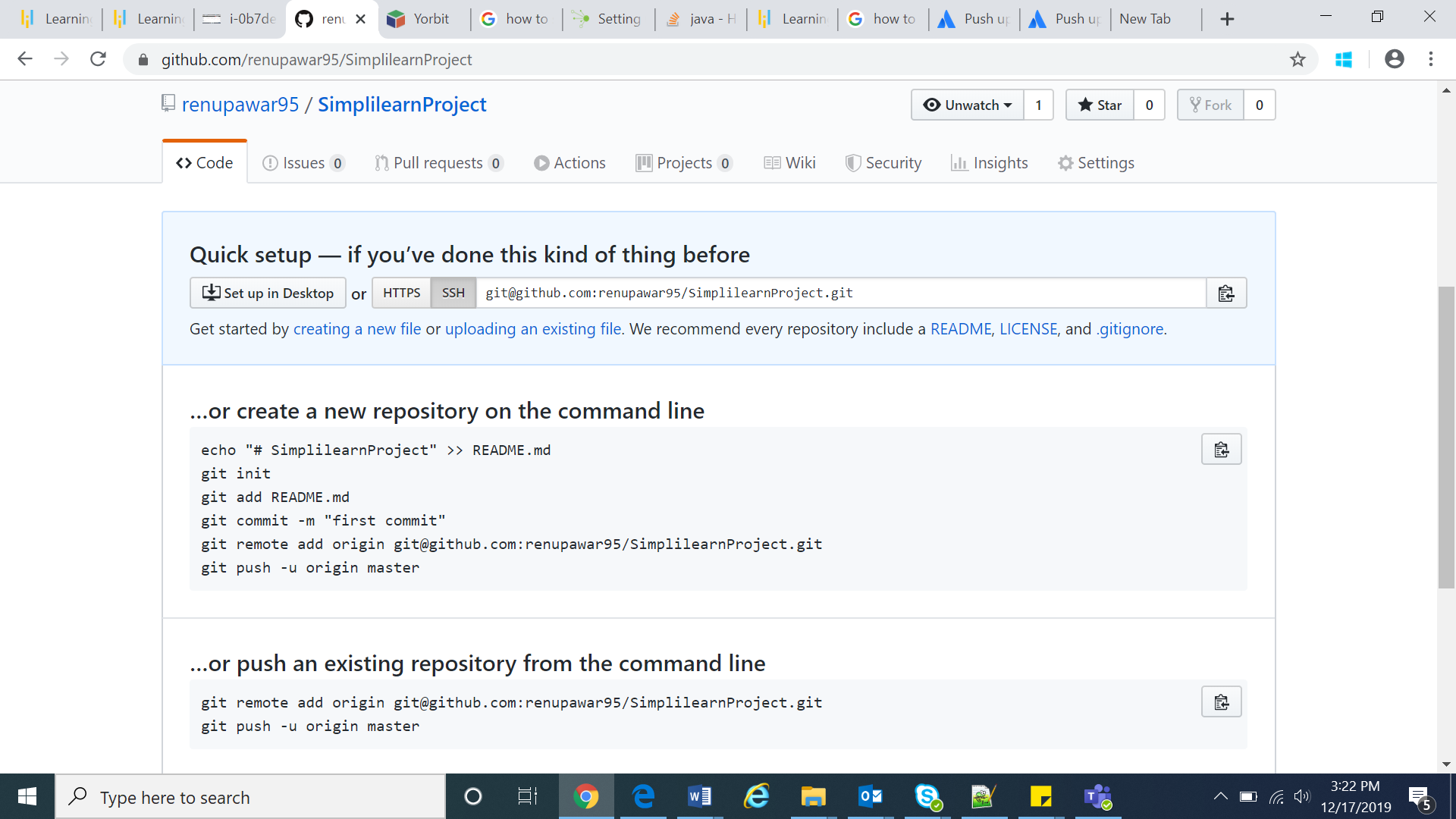
**The github account has already been setup.**

**So the next step was to create a Repository and Push our code to that repo.**

**Steps:**

1. Create a Directory in VM – **mkdir SimplilearnProject**
2. Once the directory is created add the file which you need to push: **touch fileName**
3. Initial the Git Repository : **git init**
4. Add the git code to stack by command : **git add .**
5. Then check the status of the code : **git status**
6. Once the status check is completed you need to commit the code : **git commit –m “code checkin”**
7. Now add the path in which Repository you need to add the code to **: git remote add origin git@github.com:Sandeepkr93/SimpleLearnProject.git**
8. Push the code to the Repository: **git push –u origin master**

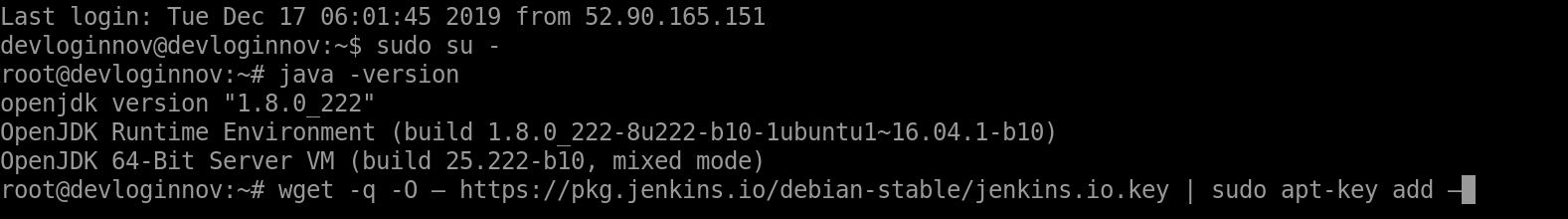




**Jenkin’s Installation**

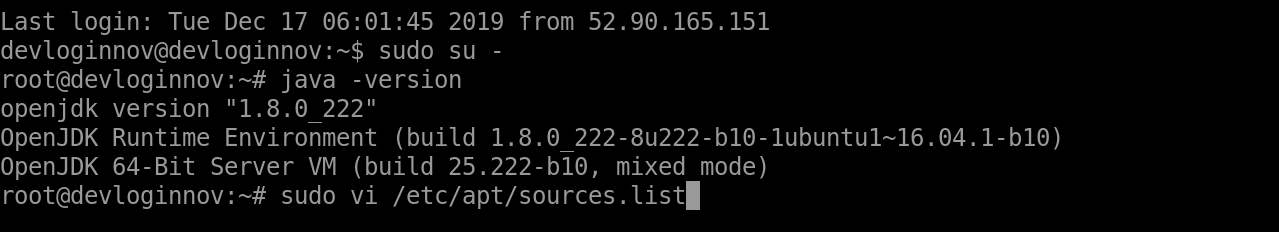
**To use the Debian repository of Jenkins to automate installation and upgrade, first add the key to your system using the following command:**

wget -q -O –<https://pkg.jenkins.io/debian-stable/jenkins.io.key> | sudo apt-key add –



**Then, add the following entry in your /etc/apt/sources.list:**

sudo vi /etc/apt/sources.list



**Now add the following command:**

deb[https://pkg.jenkins.io/debian-stable binary/](https://pkg.jenkins.io/debian-stable%20binary/)

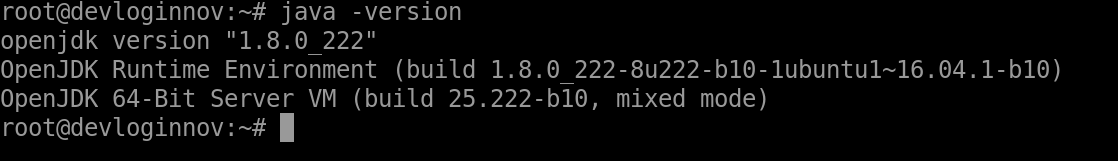
Save and exit using **:wq!**

**Update your local package index.**

sudo apt-get update

**Make sure you have jdk installed.**

**java –version**



**If not, install it using the following command:**

sudo apt-get install openjdk-8-jdk

**We need to set an Environment/Home Variable**

/usr/lib/jvm/java-1.8.0-openjdk-amd64



export JAVA\_HOME="path that you found" –(/usr/lib/jvm/java-1.8.0-openjdk-amd64 )  
export PATH=$JAVA\_HOME/bin:$PATH

**Then install Jenkins.**

sudo apt-get install Jenkins

**Once Jenkins is installed, open a browser on your local machine and enter the**

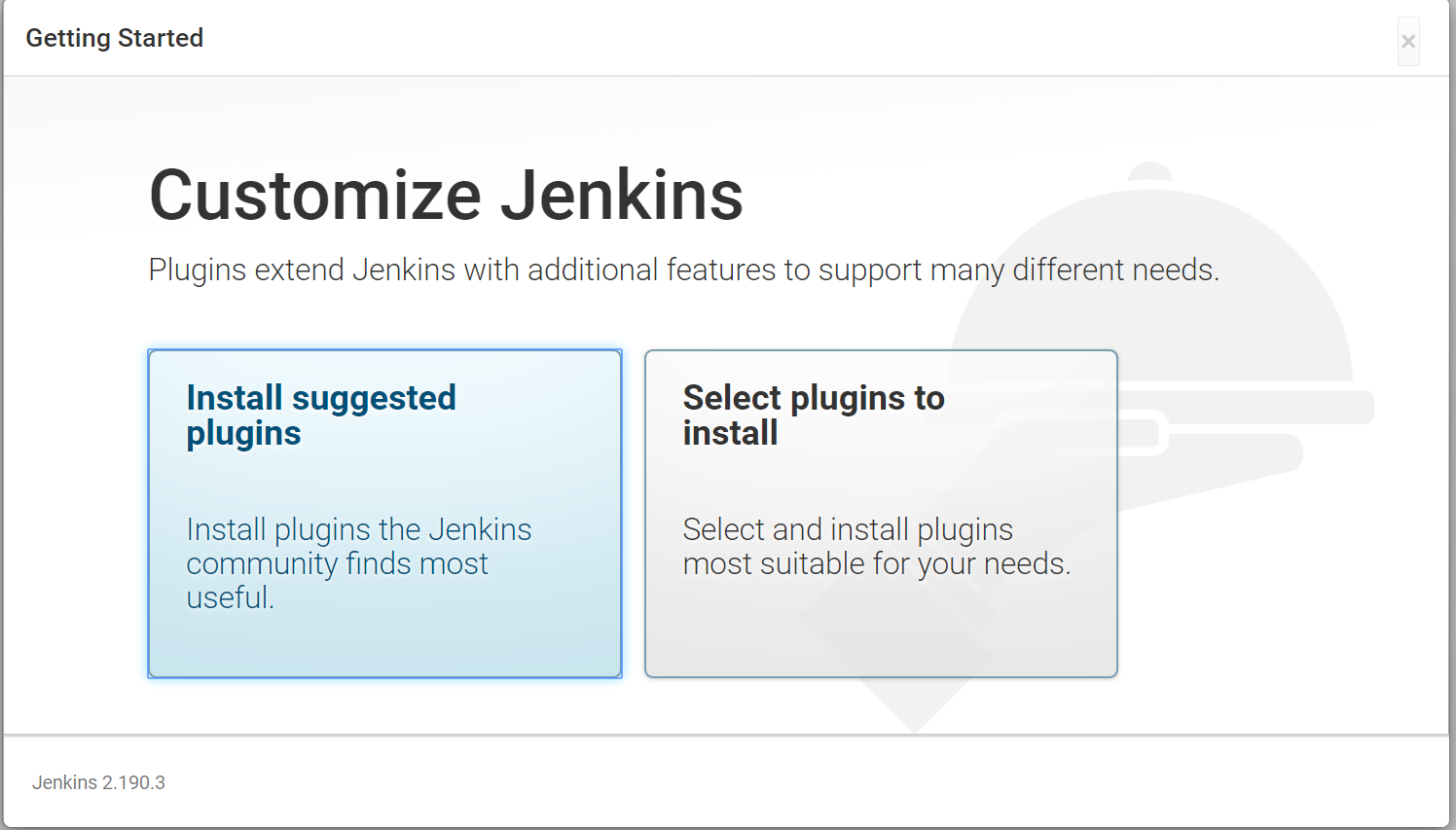
URL:    x.x.x.x:8080

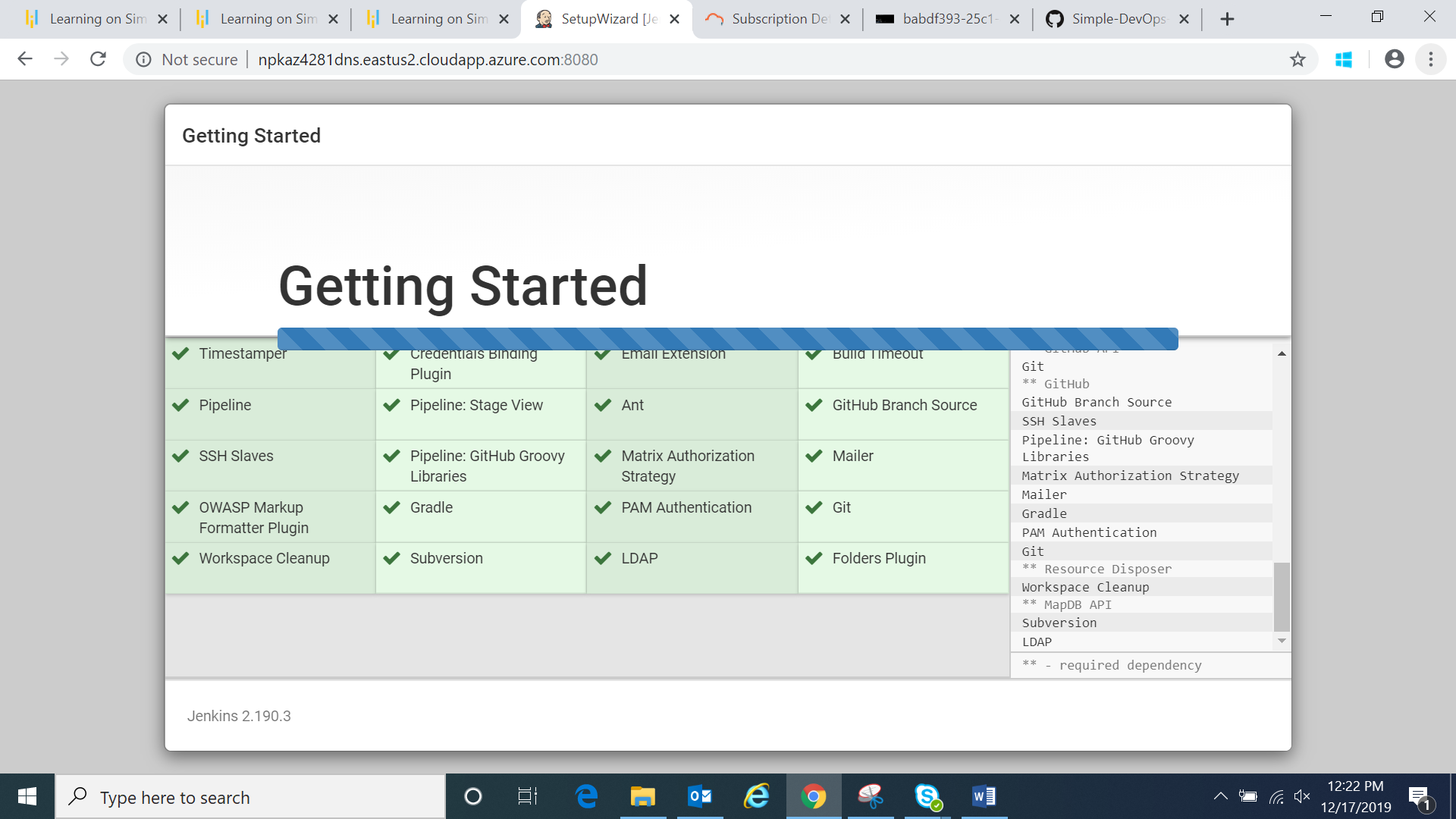
replacing x.x.x.x with the external IP address of your virtual machine.

You will need to enter the admin password.

**To find the password, type the following command:**

sudo cat /var/lib/jenkins/secrets/initialAdminPassword





Once that is done, start using Jenkins.

**Docker Installation**

This section will guide us to install Docker:

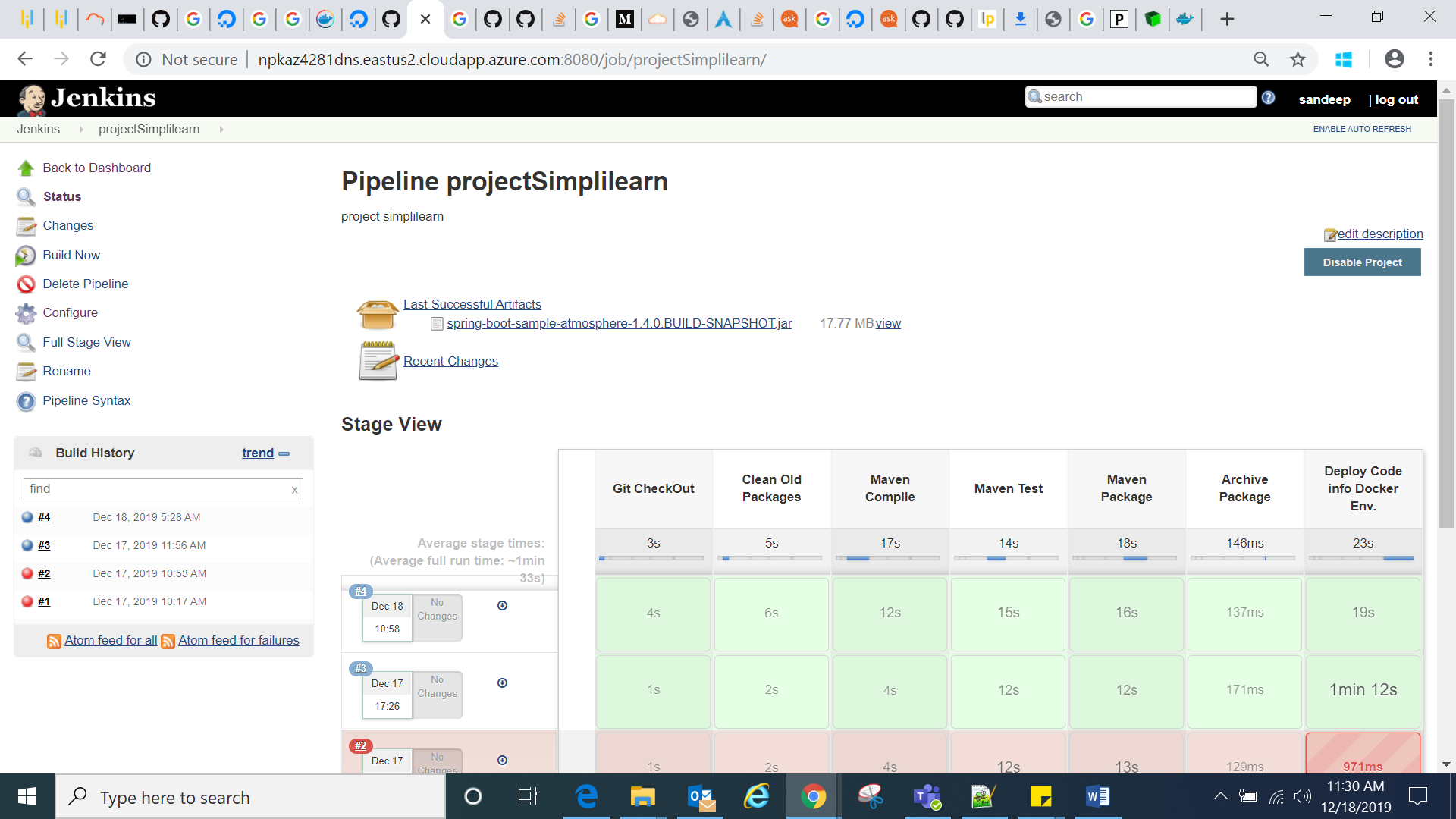
1. First, just update the apt package index: **sudo apt-get update**
2. Then Install packages to allow apt to use a repository over HTTPS:

**sudo apt-get install \apt-transport-https \ca-certificates \curl \gnupg-agent \software-properties-common**

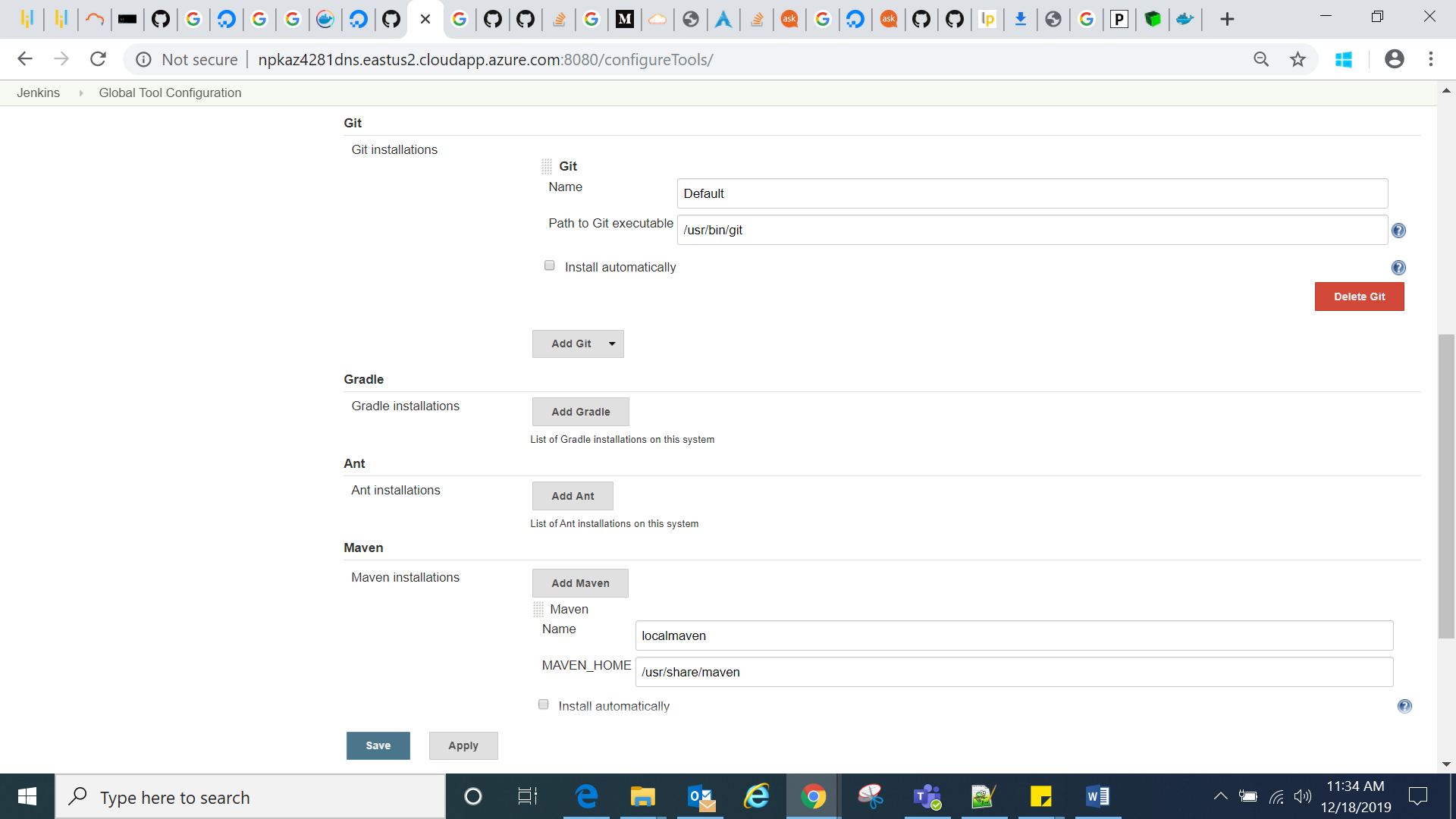
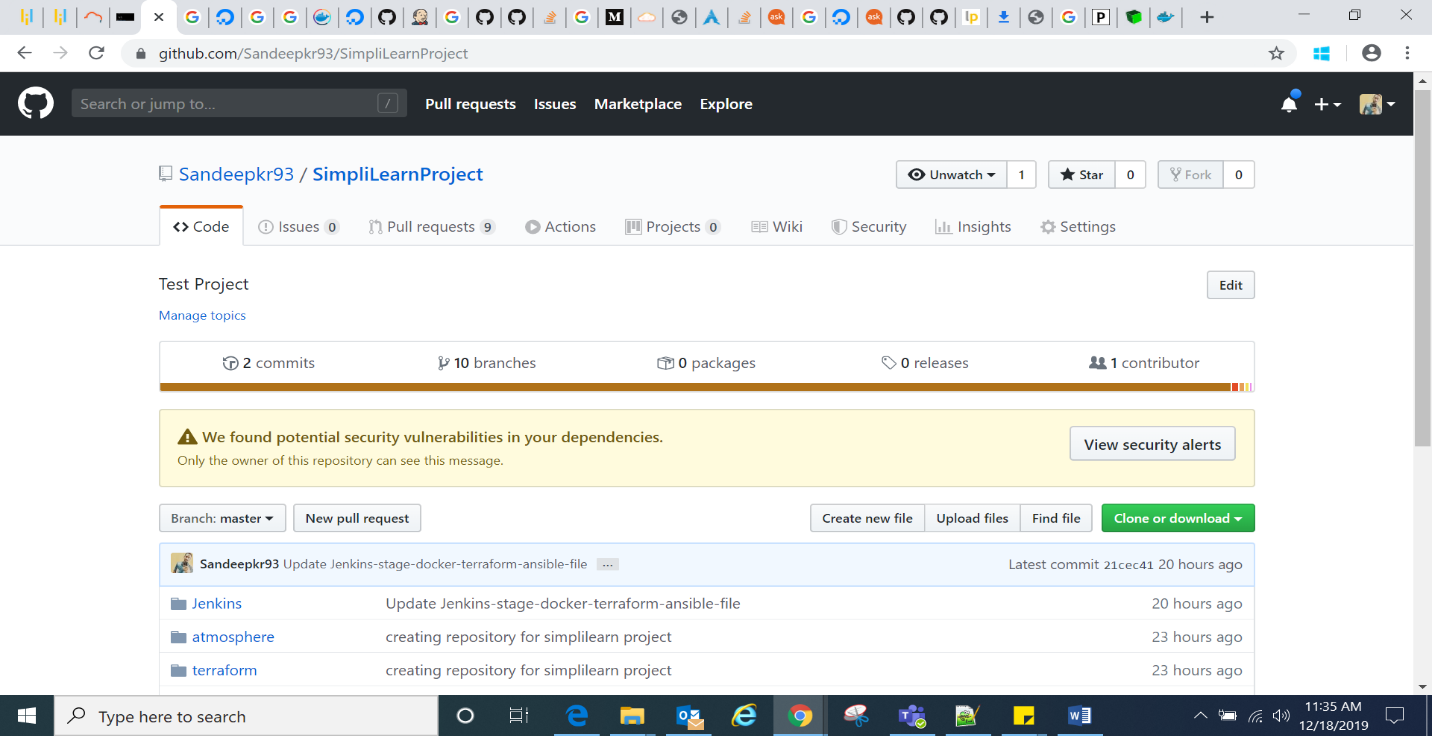
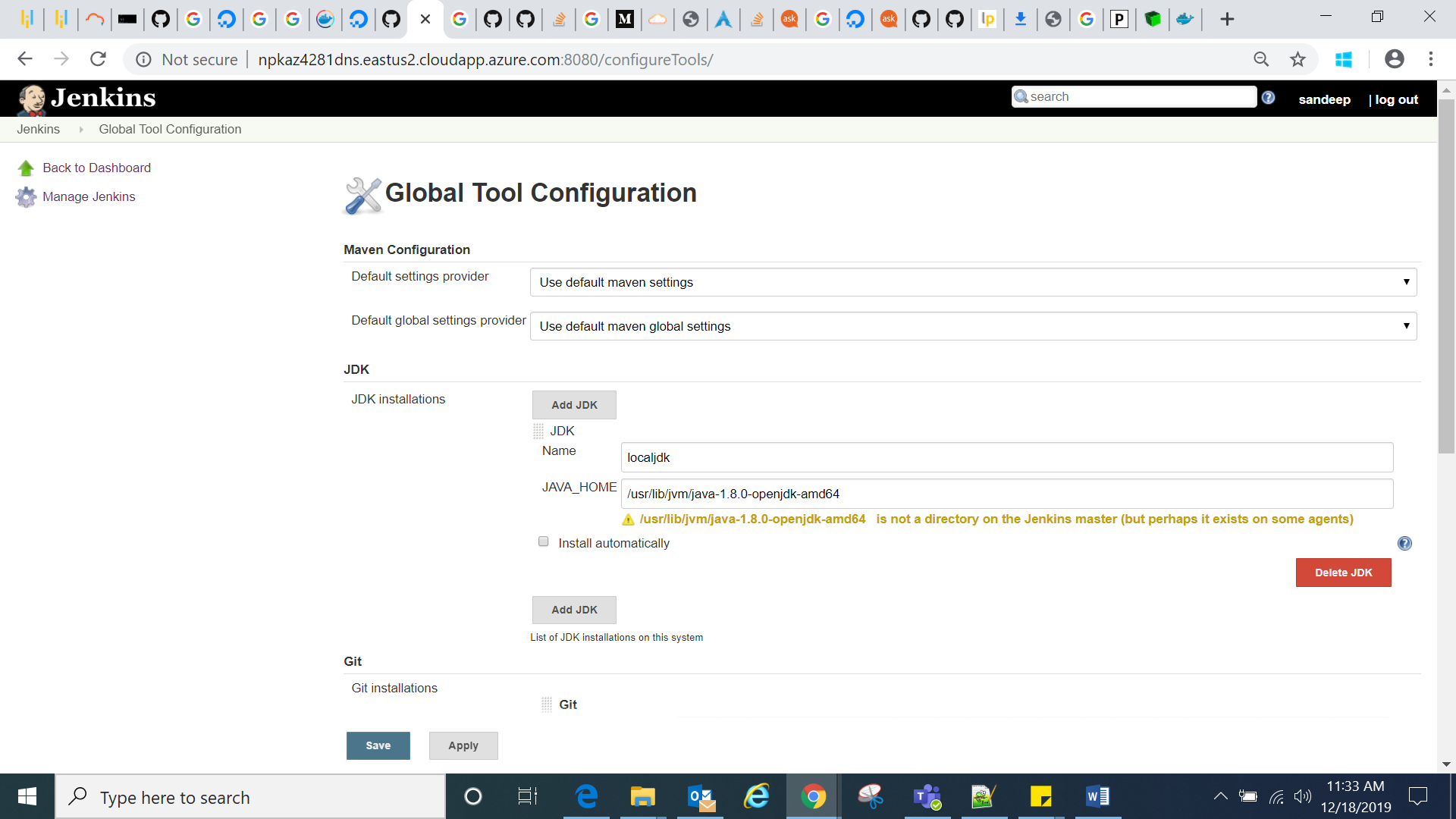
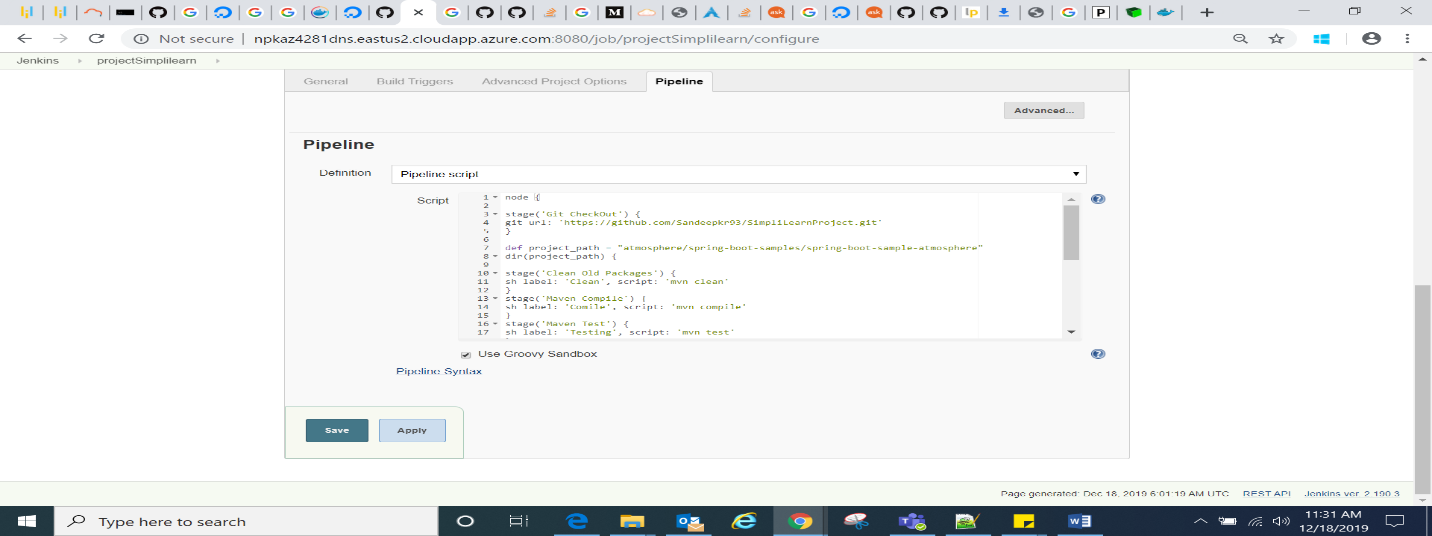
1. Add Docker’s official GPG key: **curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –**
2. Verify that you now have the key with the fingerprint 9DC8 5822 9FC7 : **sudo apt-key fingerprint 0EBFCD88**
3. Add the stable repository: **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"**
4. Update the apt package index again according to official site: **sudo apt-get update**
5. Install the latest version of Docker CE: **sudo apt-get install docker-ce**
6. We can check the version of the Docker: **sudo docker –v**
7. We can also run the hello world to see whether it is installed successfully: **sudo docker run hello-world**

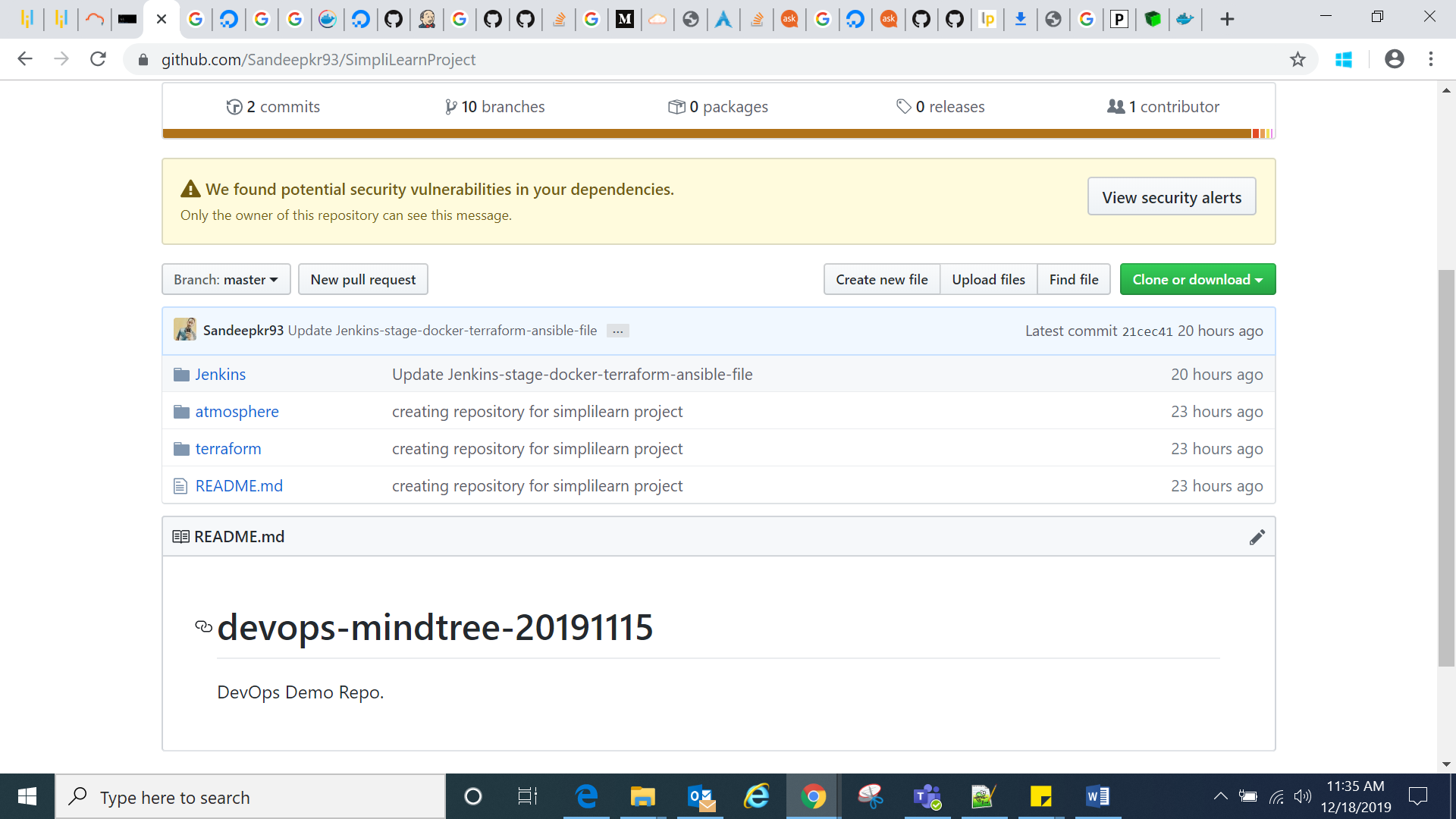
**We can see “Hello from Docker!”. Yeah! Docker is installed Successfully.**

After all these 3 steps are done we go and create the required pipeline.



Now that the pipeline is up and running our Atmosphere Chatbot is deployed in Docker environment.





The below screenshot depict the up and running application.

