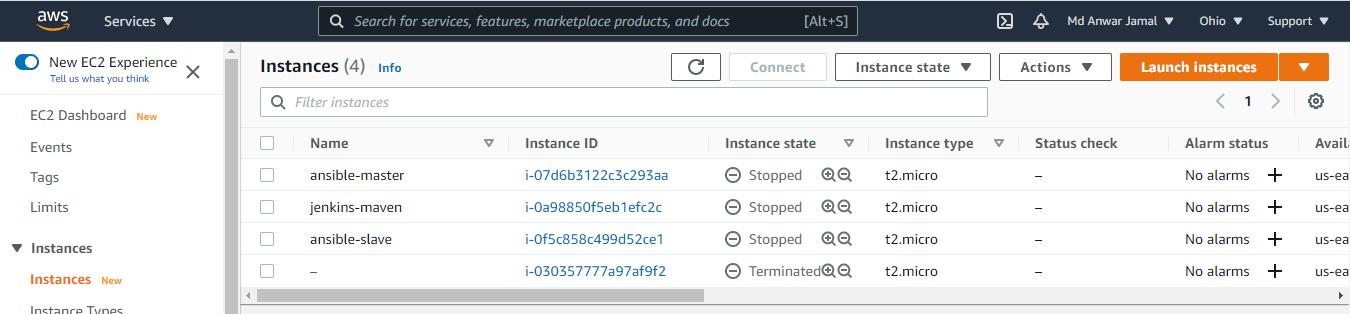
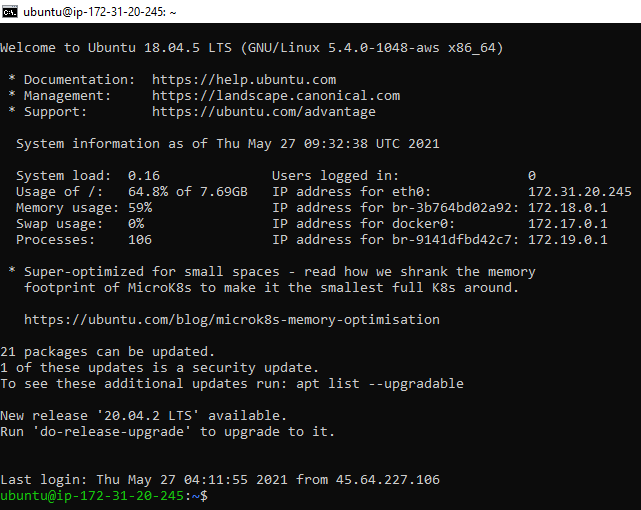
Bootcamp Day 4: Assignment on Ansible

Step 1: Created two EC2 instance (Master node and Slave Node for Ansible)

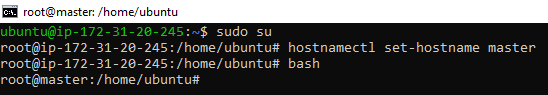


Step 2: Configuring Master node for password-less authentication with slave node

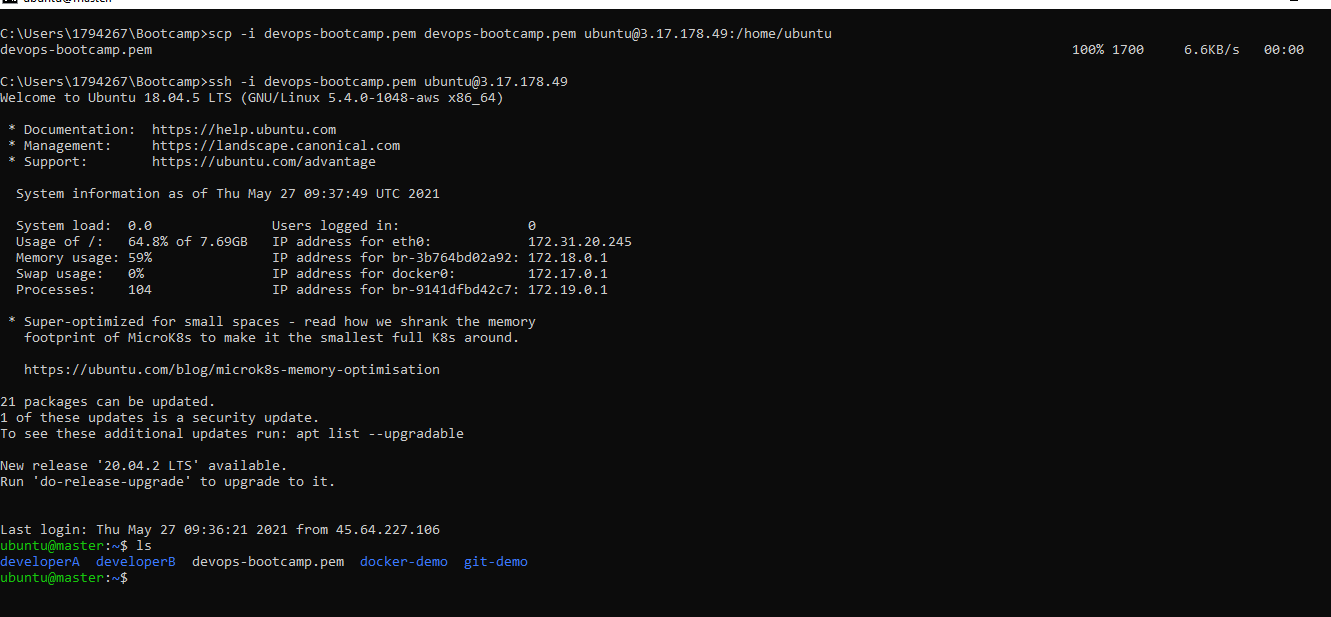
* SSH into Master Node



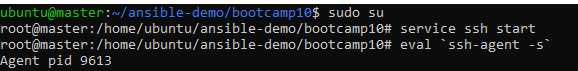
* Becoming root user: **$sudo su**
* Changing hostname of master from its IP to master: **$hostnamectl set-hostname master**
* Running **$bash** in order to see hostname change



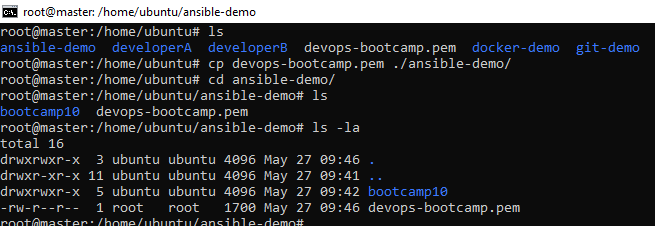
* Copying the **devops-bootcamp.pem** from Local System to Master Node as this .pem file only is used for both Master and Slave node
* **$scp -I devops-bootcamp.pem devops-bootcamp.pem ubuntu@[IP]:/home/ubuntu**
* Verifying file is transferred to Master node **$ls**



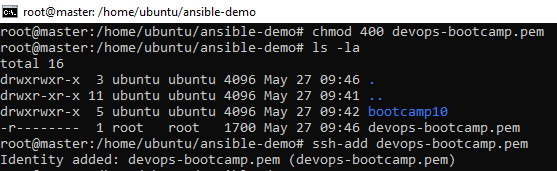
* Login as root **$sudo su**
* Starting ssh service **$service ssh start**
* Starting ssh-agent service in background **$eval `ssh-agent –s`**



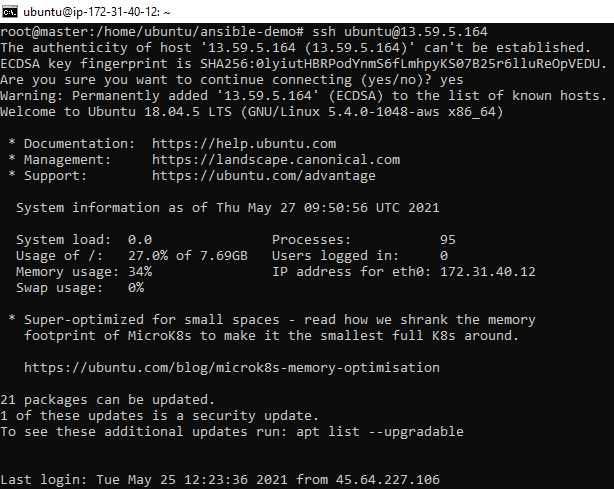
* **Checking file permission before executing chmod**



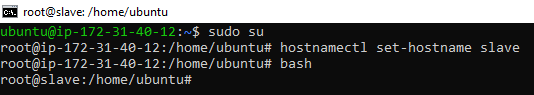
* Changing file permission for devops-bootcamp.pem to 400 **$chmod 400 devops-bootcamp.pem**
* Checking change in file permission: **$ls -la**



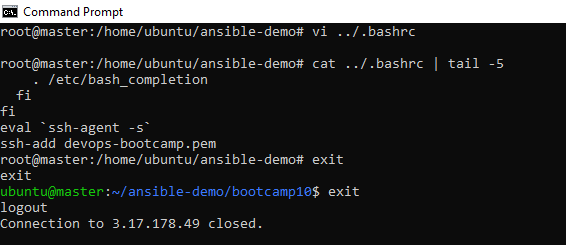
* Using ssh command to login into Slave node directly from Master node
* **$ssh ubuntu@13.59.164**



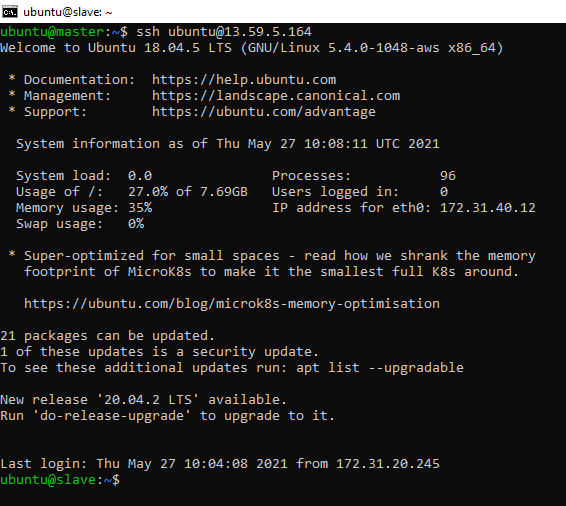
* Login as Root in Slave **$sudo su**
* Changing hostname of Slave from IP to slave **$ hostnamectl set-hostname slave**



* Adding **eval `ssh-agent –s`** and **ssh-add devops-bootcamp.pem** in the last two lines of .bashrc

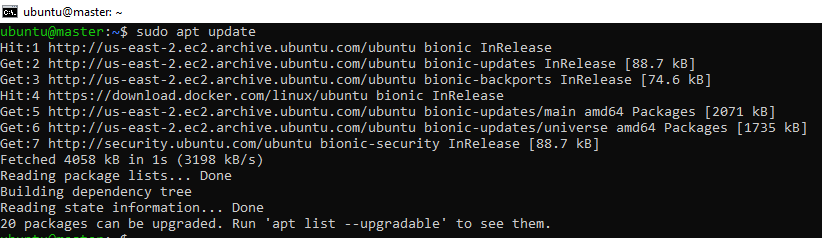


* SSH into slave from master and hostname is reflecting for slave

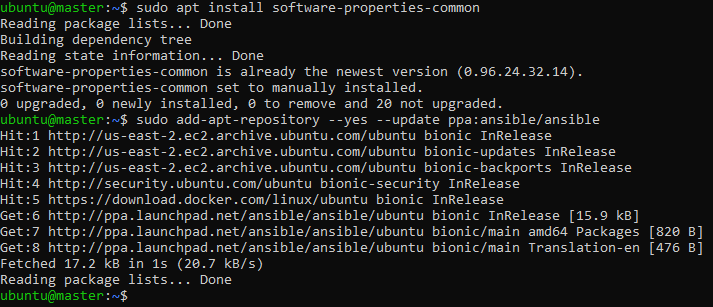


Step 3: Install Ansible on Master Node

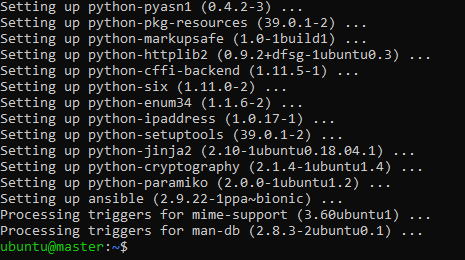
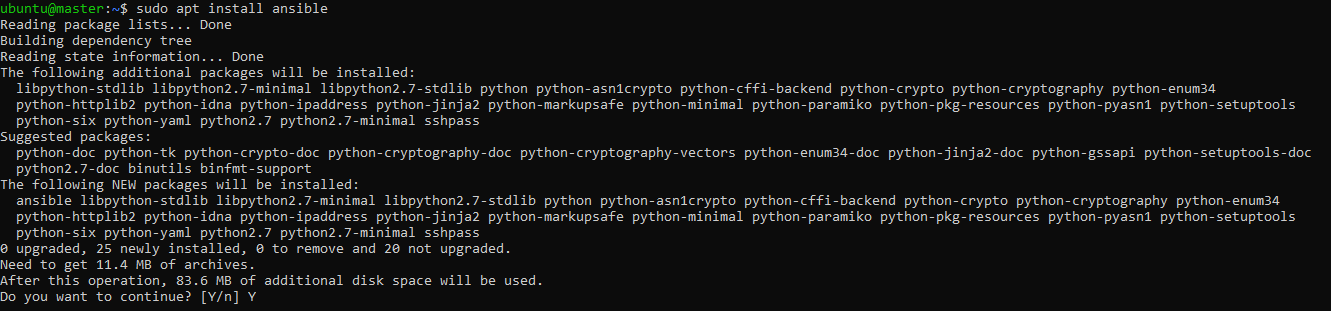
* Updating apt libraries **$sudo apt update**



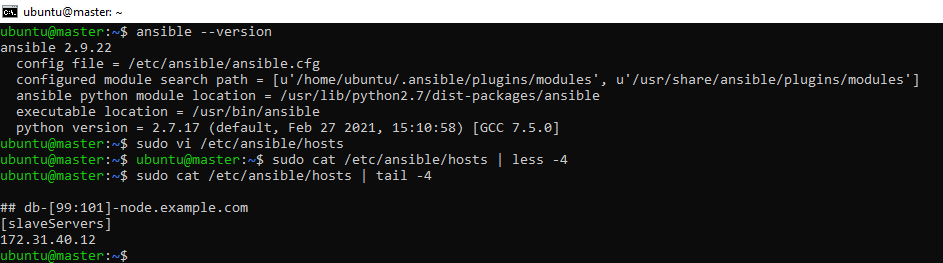
* Installing **software-properties-common $sudo apt install software-properties-common**
* Adding apt-repository for ansible installation **$sudo add-apt-respository –yes –update ppa:ansible/ansible**



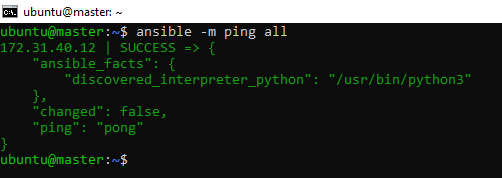
* Installing ansible **$sudo apt install ansible**



* Checking Ansible version **$ansible –version**
* Adding Slave Private IP into **/etc/ansible/hosts**

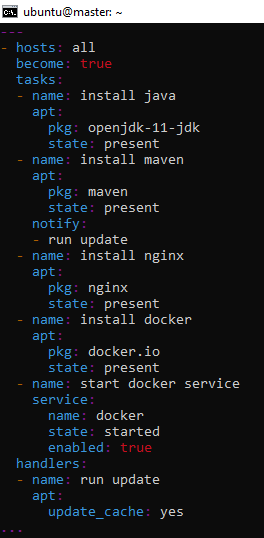


* Checking connection between Master and Slave Node **$ansible –m ping all**

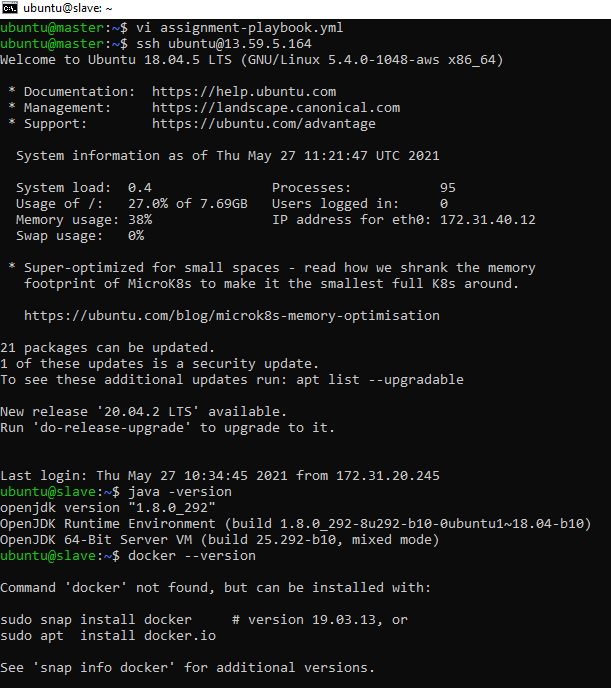


Step 4: Install software on slave

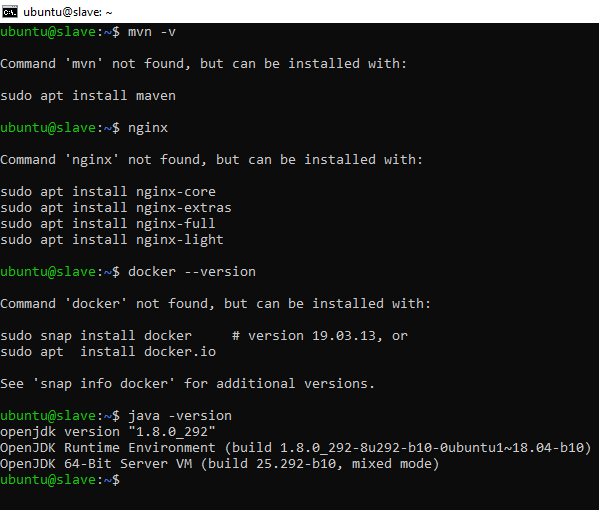
* Assignment-playbook.yml to install **java, maven, nginx and docker** on slave node
* Starting docker service



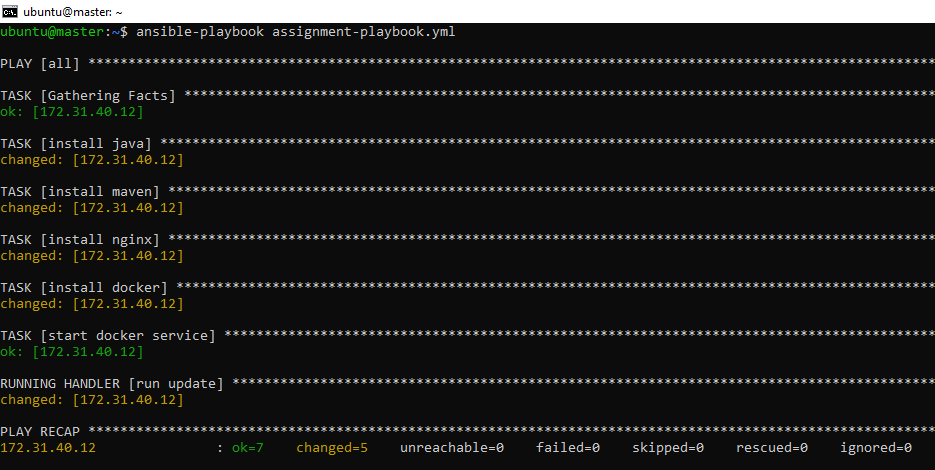
* Before running ansible playbook, SSH into slave node from master



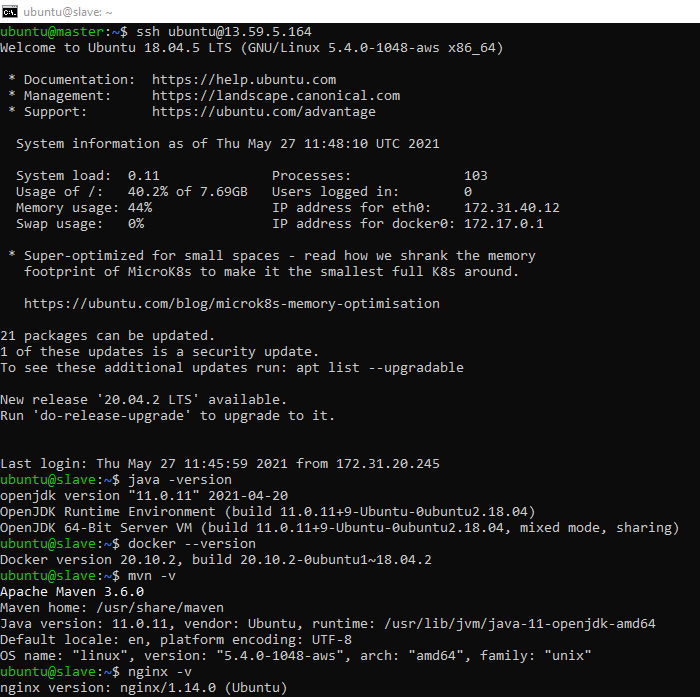
* Checking installations of required software before running ansible playbook
* Java is **1.8**



* Running **$ansible-playbook assignment-playbook.yml** for running tasks on Slave node

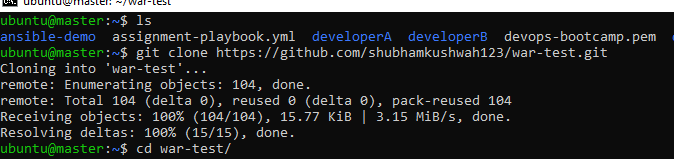


* Checking version of software in slave node
* Java is update to 1.11 to 1.8
* Maven, nginx and docker is installed

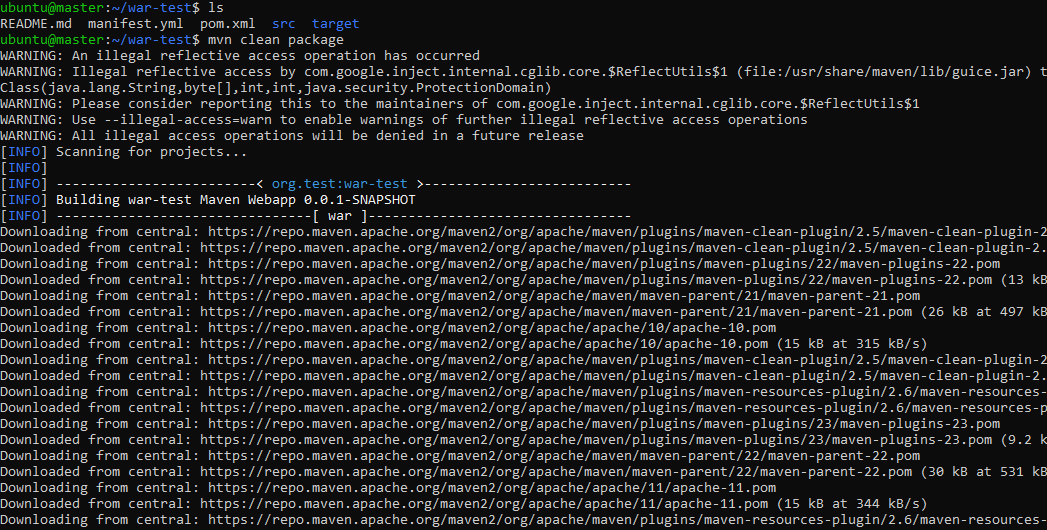


Step 5

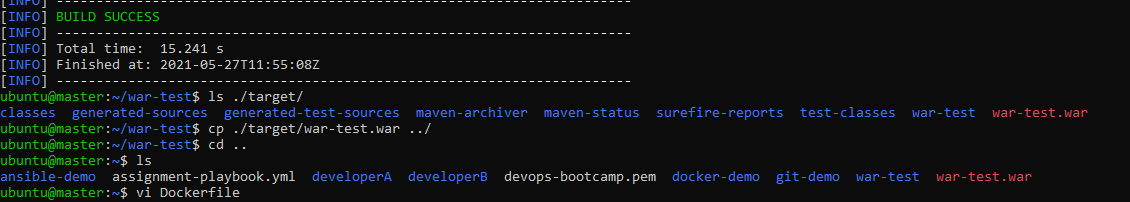
* Cloning war-test repo into master node



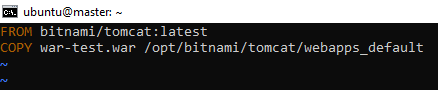
* Creating a war file by running **$mvn clean package**



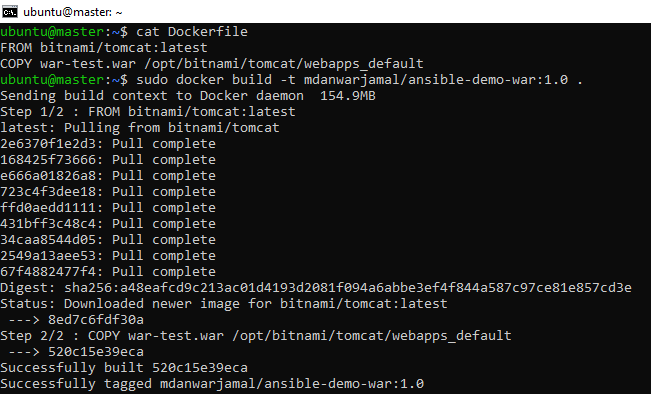
* Copying war file to home directory **$cp ./target/war-test.war ../**



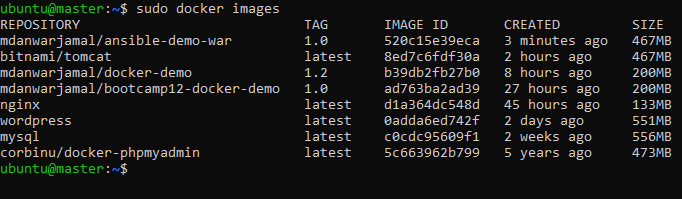
* Dockerfile for creating deployable docker image with our war file



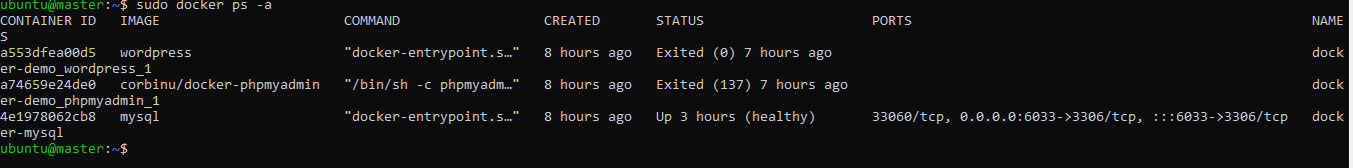
* Building up Dockerfile to create Docker Image
* **$sudo docker build –t mdanwarjamal/ansible-demo-war:1.0**



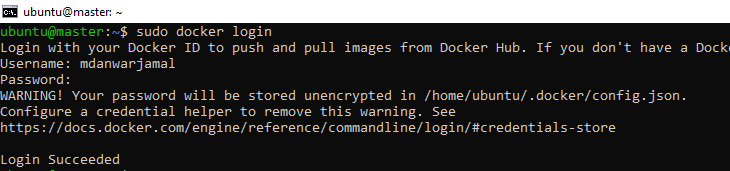
* Our Docker image is built successfully **$sudo docker images**



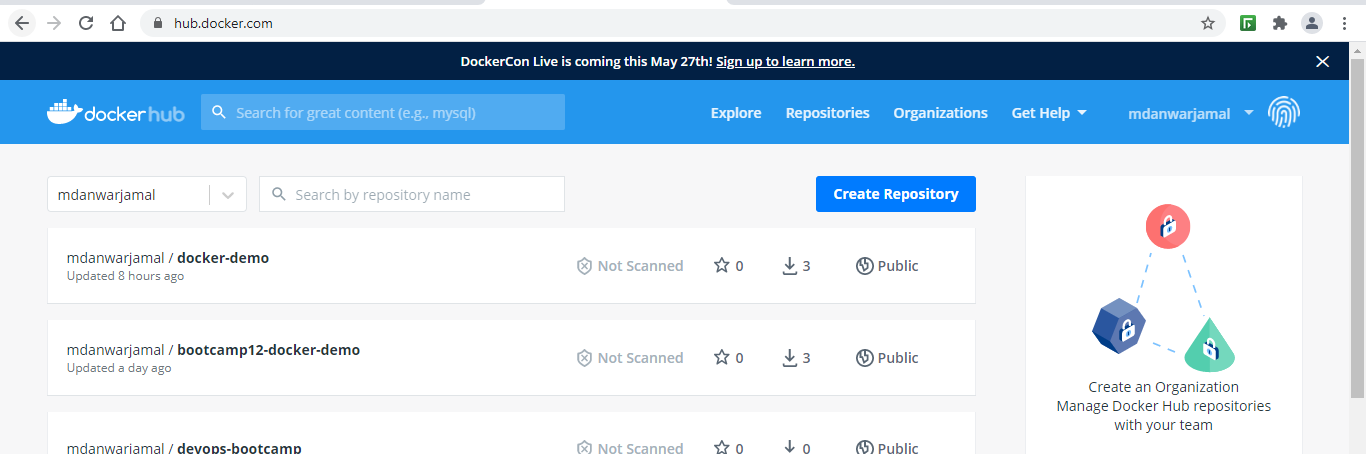
* Checking existing containers before running ansible playbook **$sudo docker ps -a**



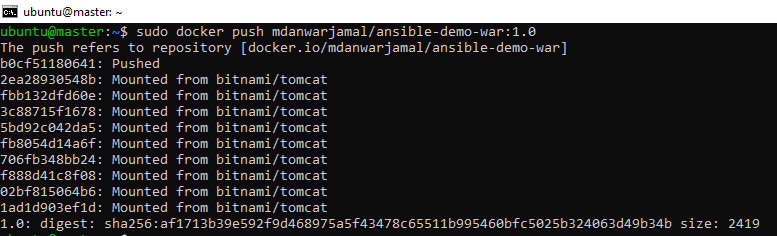
* Login into Docker Hub Account before we push **$sudo docker login**



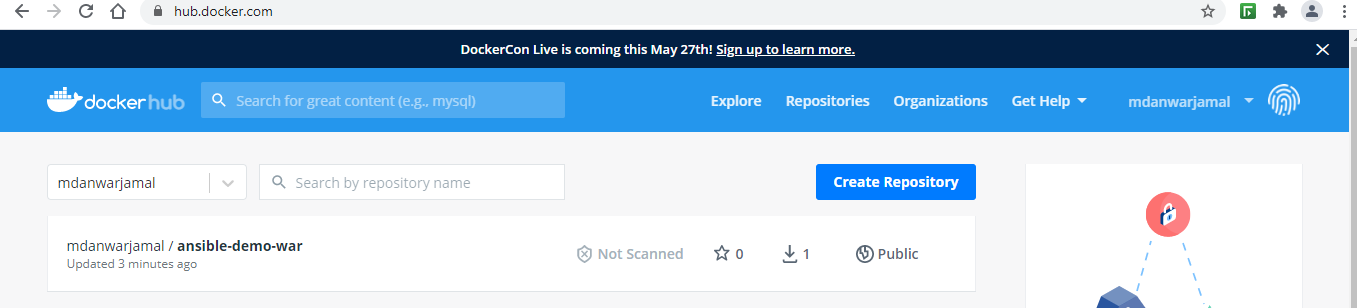
* Before Push State of Docker Hub



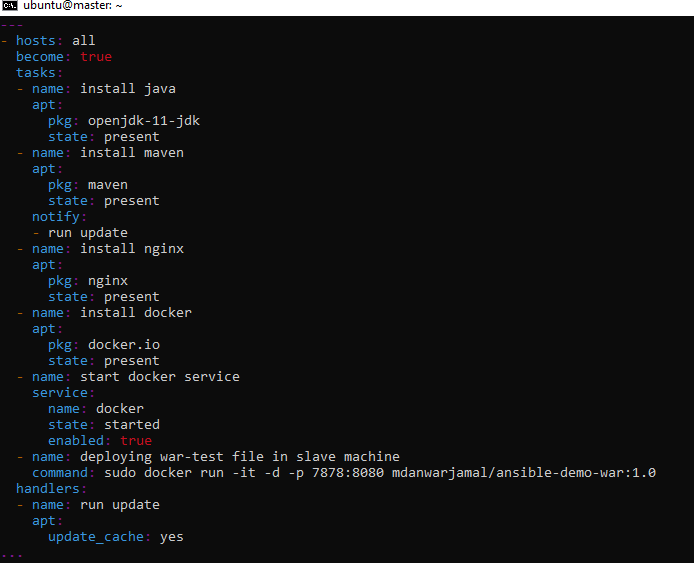
* Pushing Image to DockerHub **$ sudo docker push mdanwarjamal/ansible-demo-war:1.0**



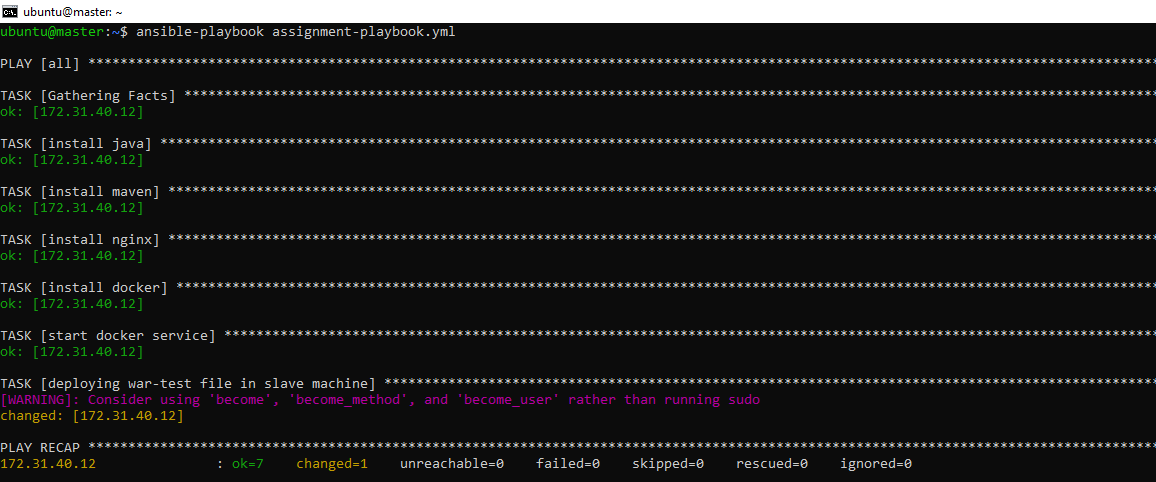
* State of Docker Hub after Push is done



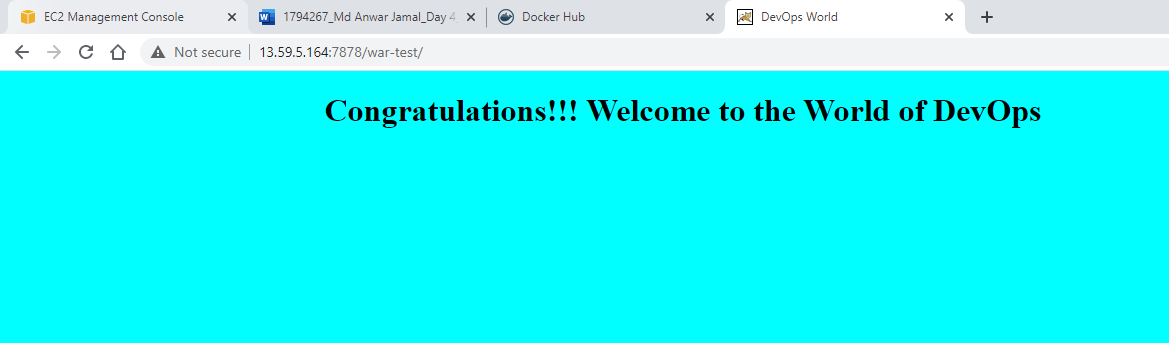
* Modifying **assignment-playbook.yml** to add new task for deploying our war application in docker container running on Slave node



* Running **$ansible-playbook assignment-playbook.yml** to deploy war application on Docker container running on Slave node



* Our war application is running on Slave Machine on port **7878**



* Checking all running containers and images available
* **$docker images**
* **$docker ps -a**

