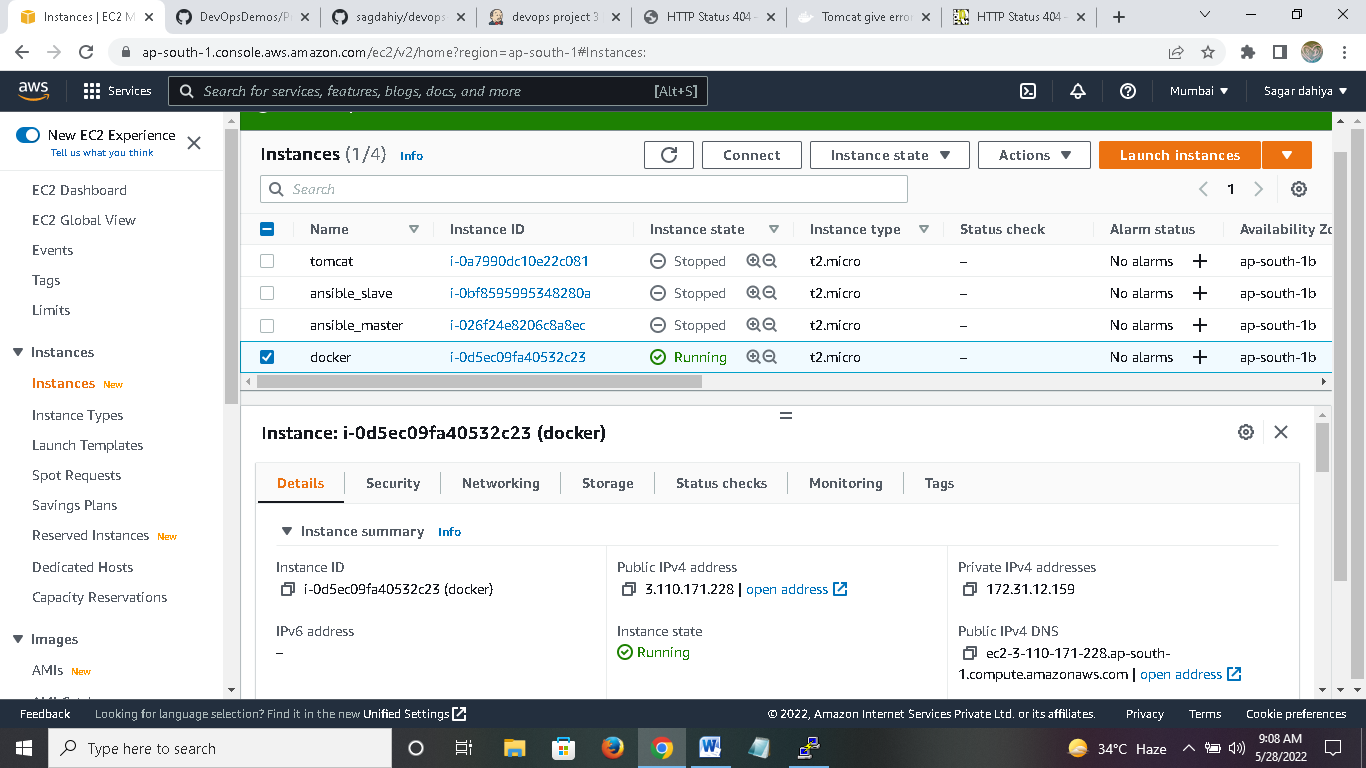
**DevOps Project -3**



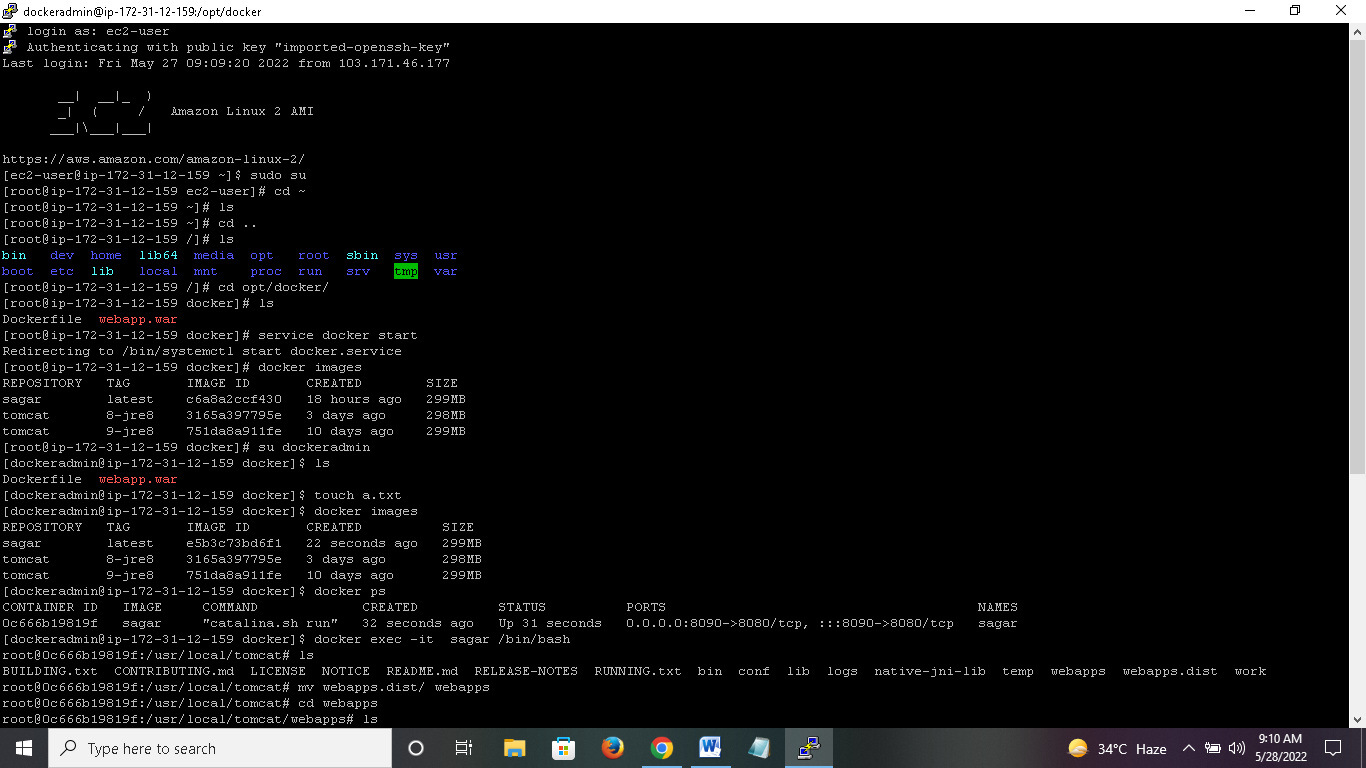
1. Launch an EC2 instance for Docker host



1. Install docker on EC2 instance and start services

yum install docker

service docker start



1. create a new user for Docker management and add him to Docker (default) group

useradd dockeradmin

passwd dockeradmin

usermod -aG docker dockeradmin

1. Write a Docker file under /opt/docker

mkdir /opt/docker

### vi Dockerfile

# Pull base image

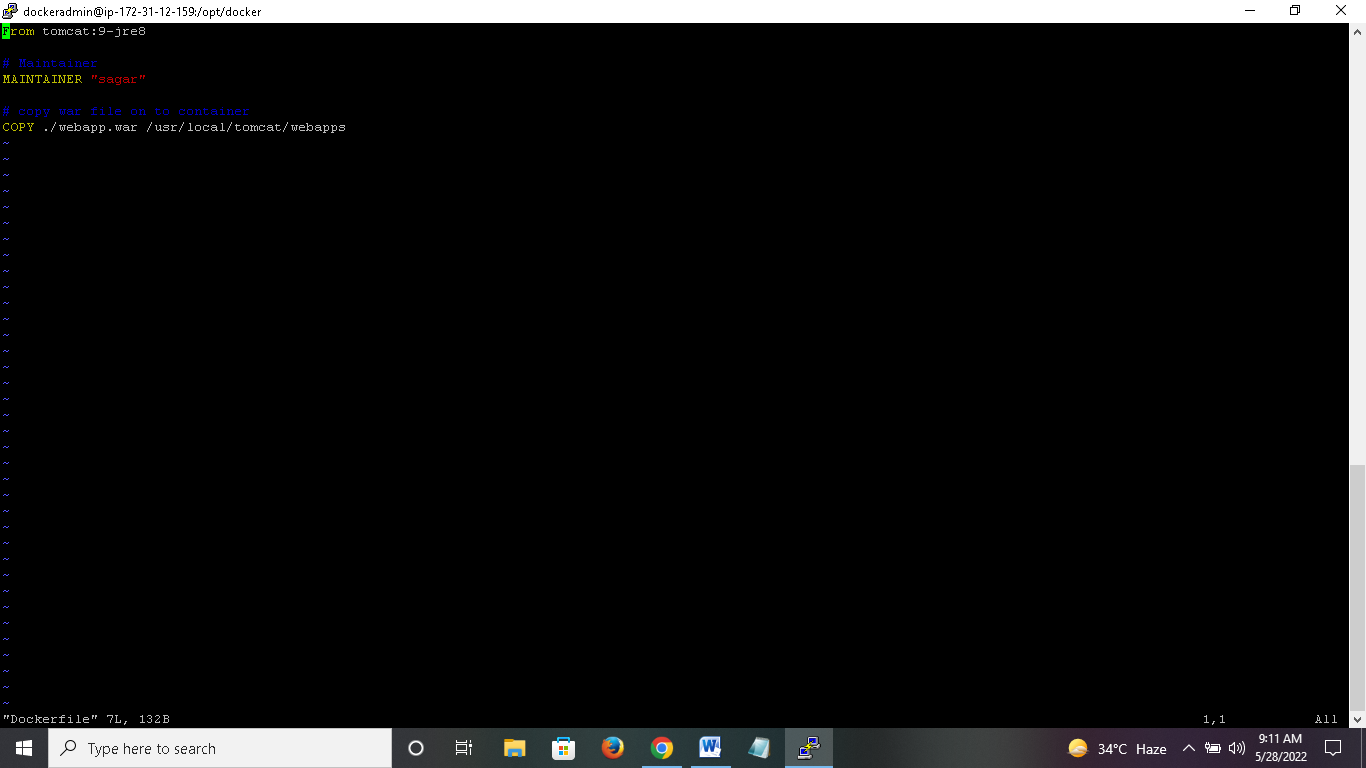
From tomcat:8-jre8

# Maintainer

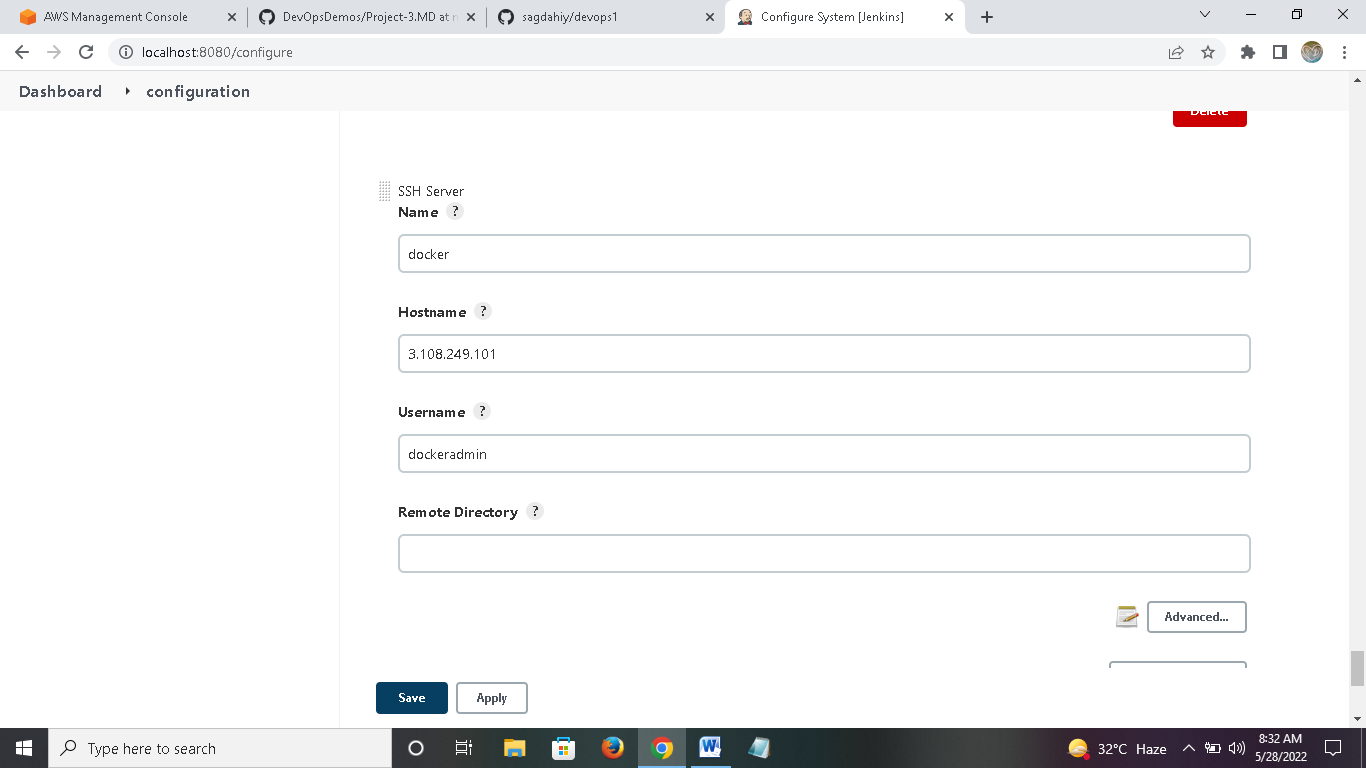
MAINTAINER "sagar"

# copy war file on to container

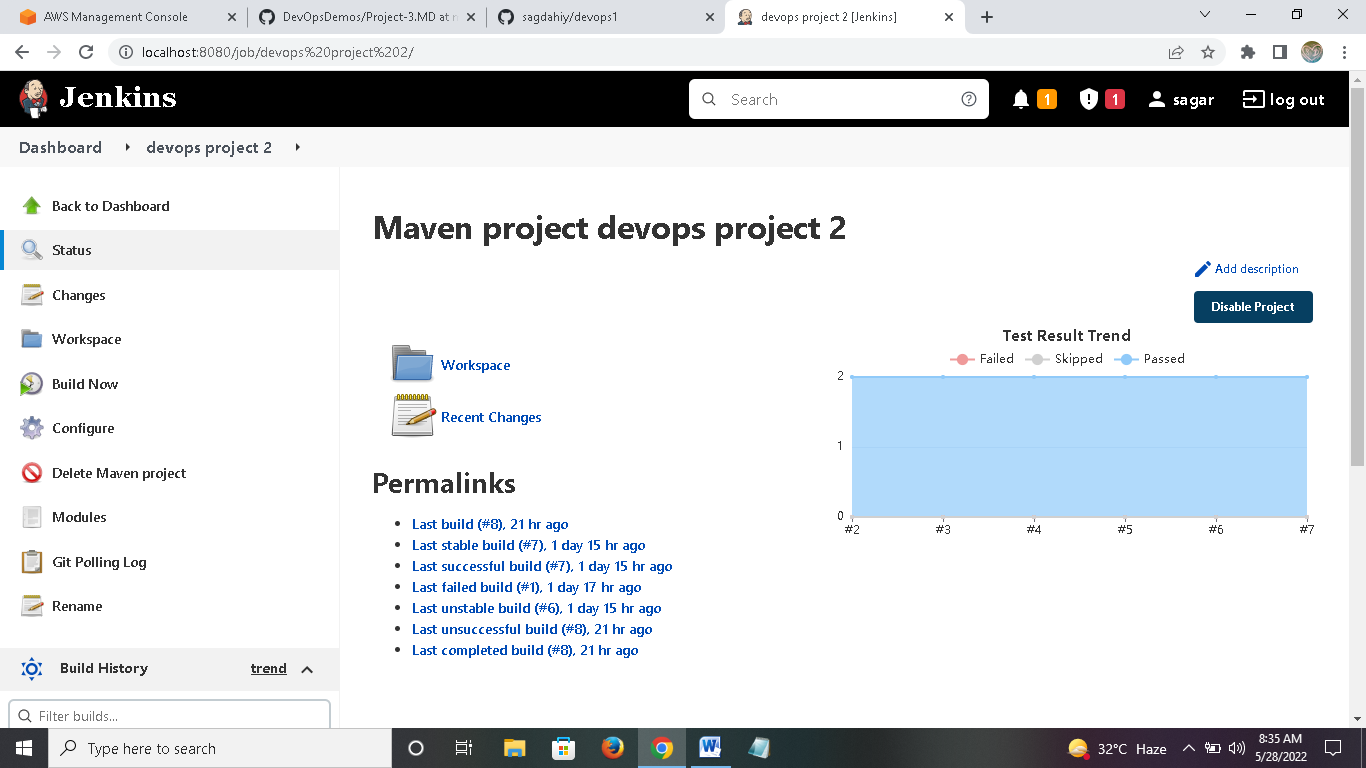
COPY ./webapp.war /usr/local/tomcat/webapps



1. Login to Jenkins console and add Docker server to execute commands from Jenkins  
   Manage Jenkins --> Configure system --> Publish over SSH --> add Docker server and credentials



1. Create maven project named devops project 2

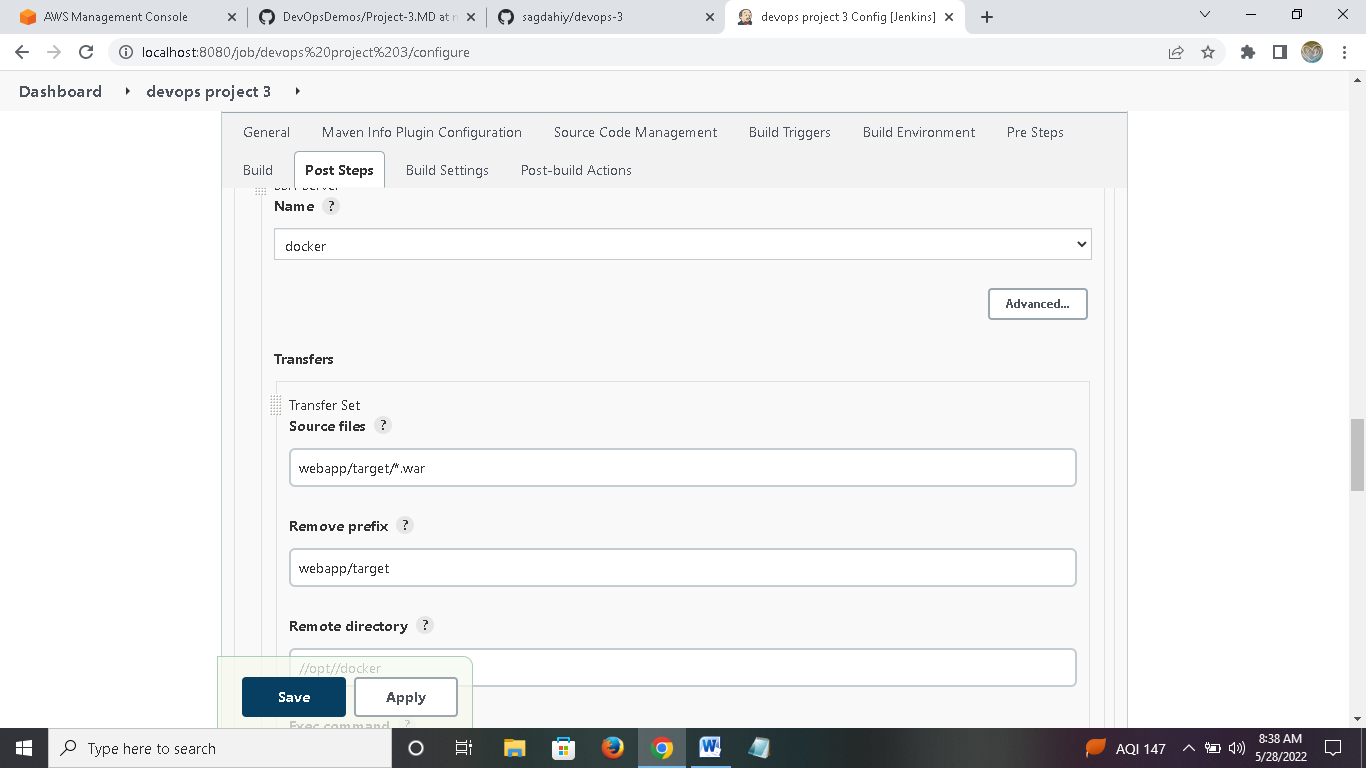


7. Go to configure ,do below changes:

A) Source Code Management  
Repository : https://github.com/sagdahiy/devops-3.git   
Branches to build : \*/master

B) Build Root POM: pom.xml  
Goals and options : clean install package

C) send files or execute commands over SSH Name: docker  
Source files : webapp/target/\*.war Remove prefix : webapp/target Remote directory : //opt//docker



Build image:  
Exec command[s] :

docker stop sagar;

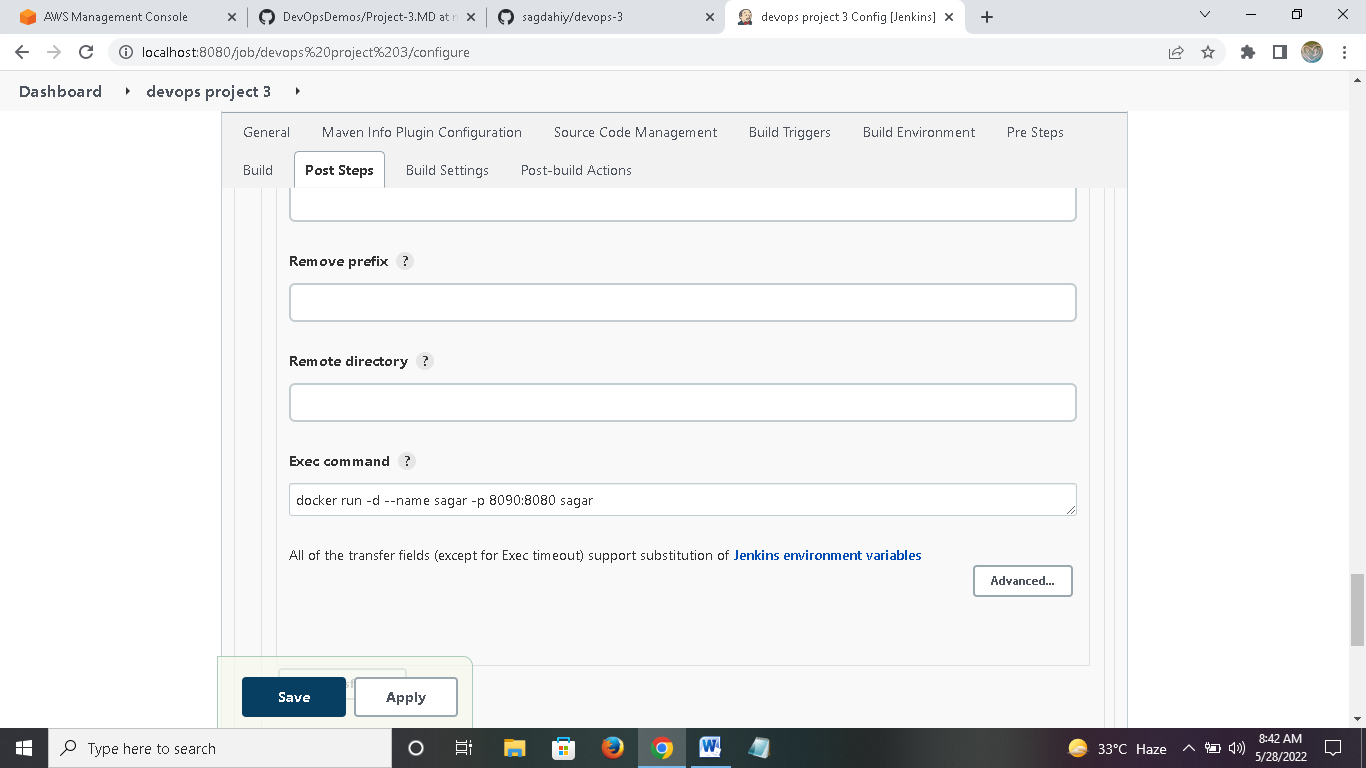
docker rm -f sagar;

docker image rm -f sagar;

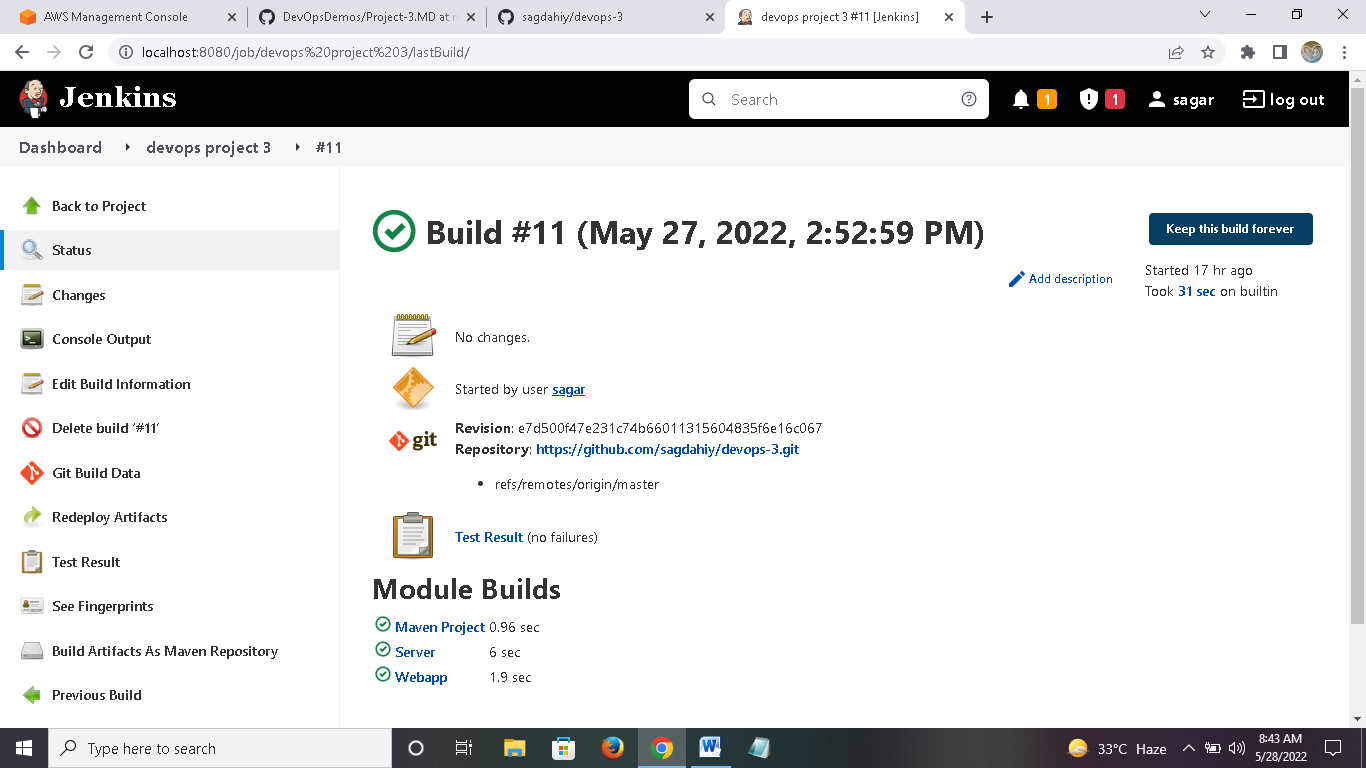
cd /opt/docker;

docker build -t sagar .

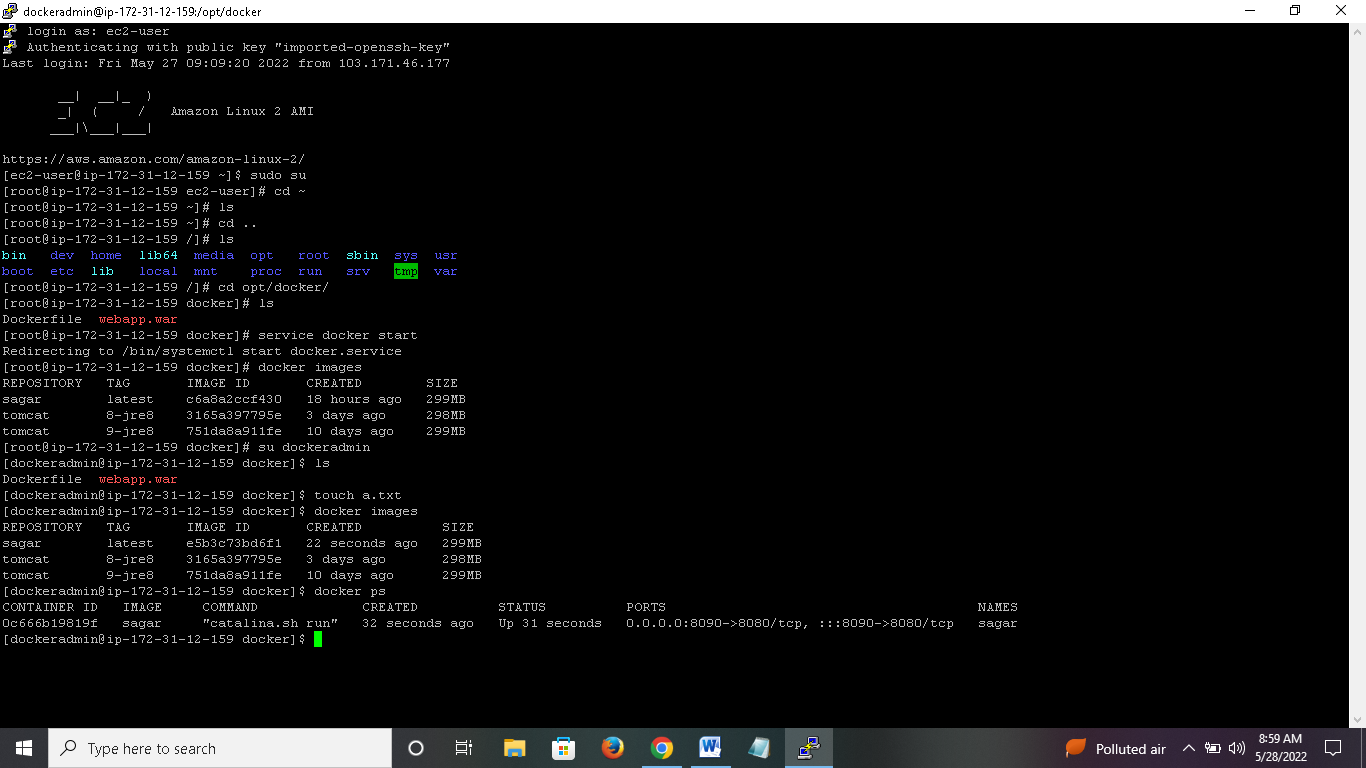
D) send files or execute commands over SSH:it run container  
Name: docker\_host  
Exec command : docker run -d --name sagar -p 8090:8080 sagar



1. Login to Docker host and check images and containers. (no images and containers)
2. Execute Jenkins job



1. check images and containers again on Docker host. This time an image and container get creates through Jenkins job



1. Access web application from browser which is running on container

<http://3.110.171.228:8090/webapp/>

