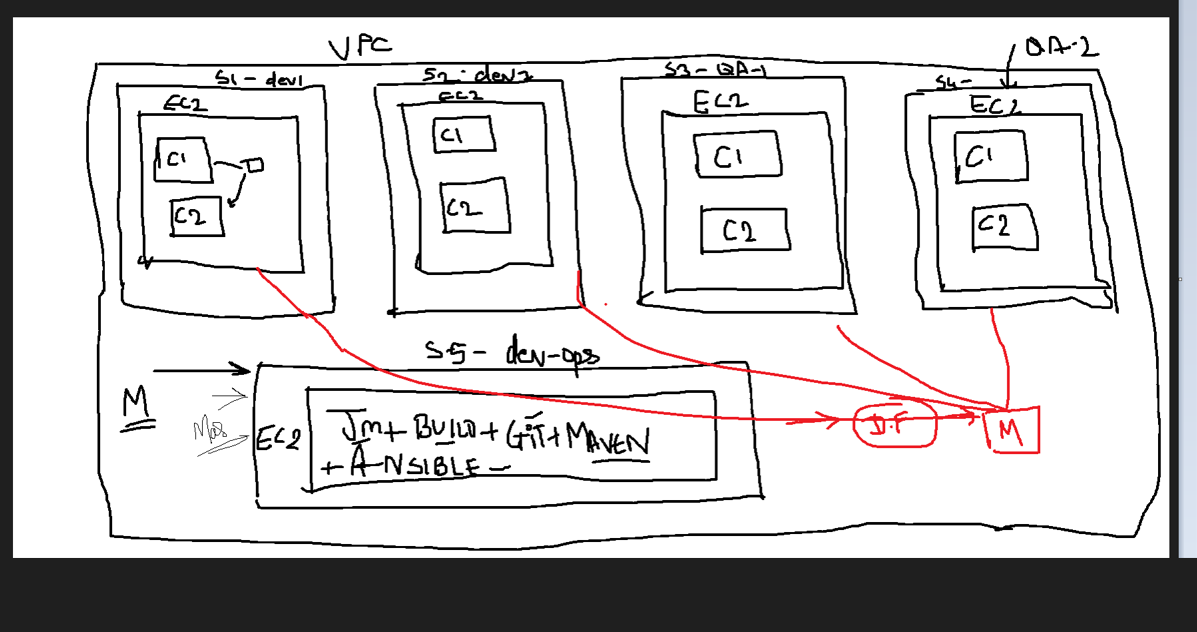
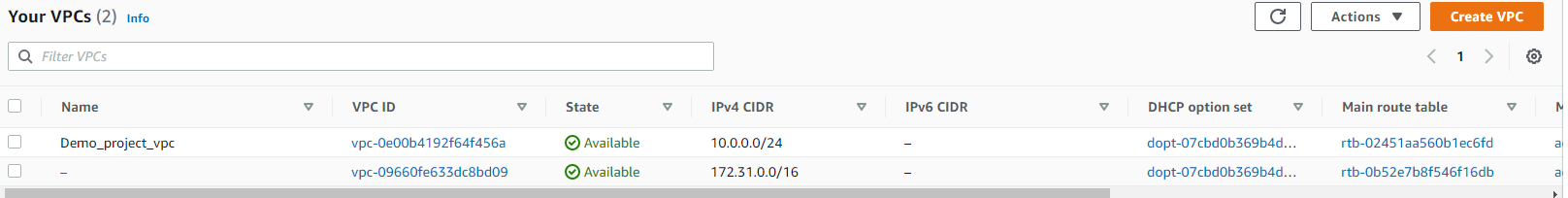
**MAIN\_PROJECT\_USING\_ANSIBLE\_JENKINS\_DOCKER**

**PROJECT: Need to build Game of life on Master and deploy it on 2container on various env such as Dev1,Dev2,QA1,QA2.** 

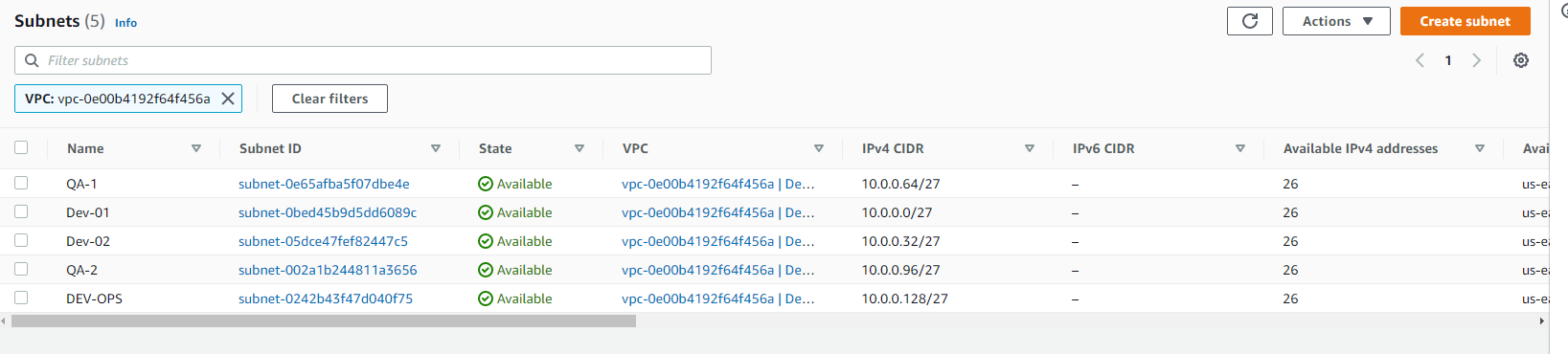
**STEP1) CREATE ONE VPC**

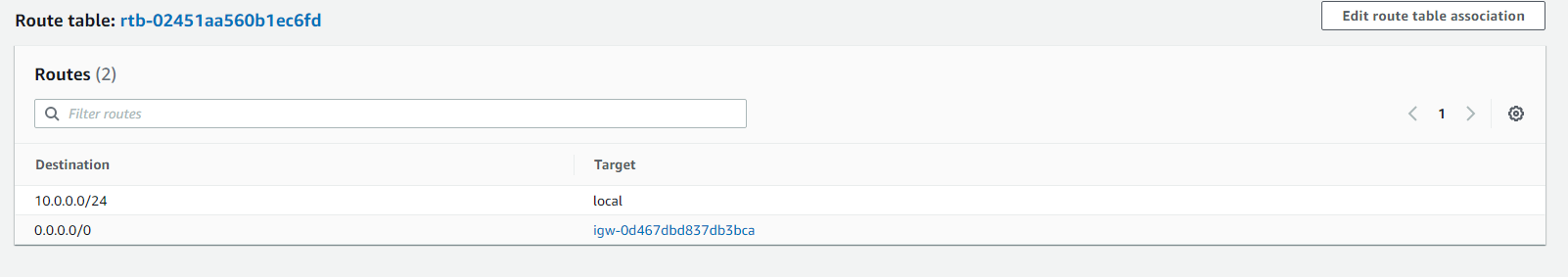
**NAME IT AS DEMO VPC**

****

**STEP2) CREATE 5 SUBNET ( DEV1, DEV2, QA1, QA2 AND DEV\_OPS)**

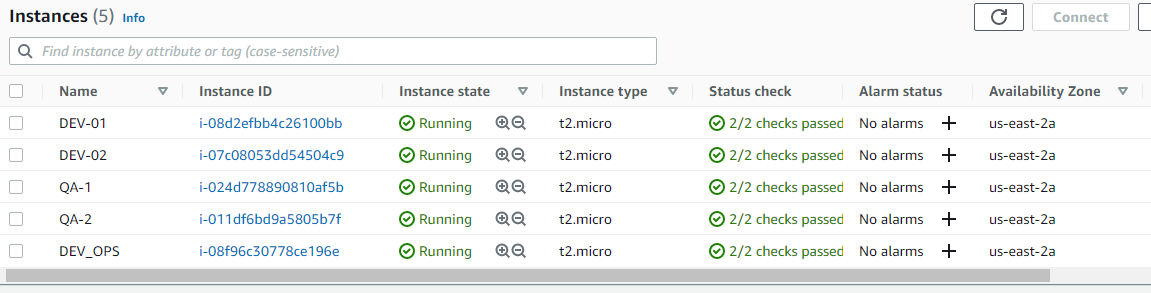
**ATTACH IGW TO THE SUBNET**

****

****

**STEP3) LAUNCH ONE MACHINE IN EACH SUBNET:**

**(NAME THE MACHINE AS PER THE SUBNET NAME)**

****

**STEP4) DEV\_OPS MACHINE WILL BE YOUR MASTER MACHINE IN WHICH WE WILL NEED JENKINS, TOMCAT, ANSIBLE\_PLAYBOOK, MAVEN. AS WE ARE GOING TO BUILD THE JOB ON MASTER.**

**STEP4A)--- CREATE A SERVICE USER NAME AS VELOCITY ( IN ALL THE MACHINE DEV1, DEV2, QA1, QA2 AND DEV\_OPS)**

**COMMAND: useradd velocity**

**ASSIGN PASSWD TO USER**

**COMMAND: passwd velocity**

**(ENTER THE PASSWORD)**

**NOW GIVE THE SUDO PERMISSON TO VELOCITY USER ON EACH MACHINE (DEV1, DEV2, QA1, QA2 AND DEV\_OPS)**

**COMMAND: visudo**

****

**NOW AS WE ARE GOING TO INSTALL THE REQ DEPENDENCY USING ANSIBLE WE WILL NEED TO MAKE SOME CHANGES ON EACH MACHINE (DEV1, DEV2, QA1, QA2 AND DEV\_OPS)**

**ON MASTER MACHINE (DEV\_OPS) INSTALL ANSIBLE**

**COMMAND: sudo yum install ansible -y**

****

**FOR THE MASTER MACHINE ( FOR ANSIBLE) WE NEED TO EDIT THE INVENTORY FILE CALLED AS ANSIBLE-CONF IN THE LOCATION /etc/ansible/ansible.cfg**

**COMMAND: sudo vi /etc/ansible/ansible.cfg**

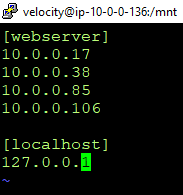
**WE NEED TO ENABLE THE INVENTORY AND SUDO USER AS SHOWN BELOW**

****

**NOW WE NEED TO EDIT THE HOST FILE FOR ANSIBLE ON MASTER ( DEV\_OPS)**

**COMMAND: sudo vi /etc/ansible/hosts**

**WE NEED TO PROVIDE THE SERVERNAME AND PRIVATE IP ADD OF OTHER HOSTS (DEV1, DEV2, QA1, QA2)**

****

**NOW WE HAVE TO MAKE CHANGES IN THE SSHD FILE SO THAT OUR SERVICE USER (VELOCITY) CAN MAKE SSH CONNECTION WITH OTHER HOSTS WITHOUT USING THE PASSWORD**

**FOR THAT WE NEED TO MAKE CHANGES IN THE SSHD FILE**

**COMMAND: sudo vi /etc/ssh/sshd\_config**

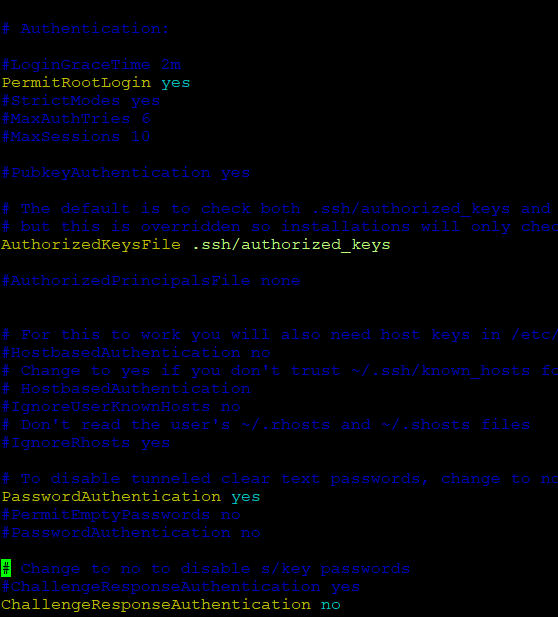
**MAKE SURE YOU UNCOMMENT THE FOLLOWING**

**PERMITROTTLOGIN YES**

**PASSWORDAUTHENTICATION YES**

**ADD # TO COMMENT**

**PASSWORDAUTHENTICATION NO**

****

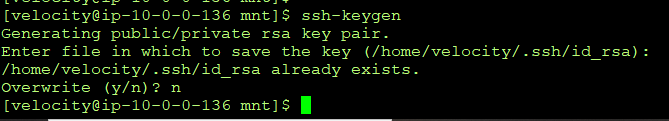
**THE ABOVE STEP HAS TO BE EXECUTED AT ALL THE MACHINE (DEV1, DEV2, QA1, QA2 AND DEV\_OPS)**

**MAKE SURE YOU RESTART SSHD SERVICES**

**COMMAND: service sshd restart**

**NOW WE HAVE TO GENERATE SSH KEY ON THE MASTER MACHINE SO THAT WE CAN CONNECT WITH HOST MACHINE USING THE SERVICE\_USER ( VELOCITY)**

**COMMAND: ssh-keygen**

****

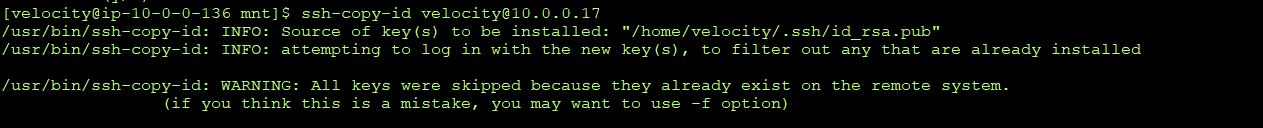
**NOW AS YOUR KEY IS GENERATED WE NEED TO COPY THE KEY TO ALL THE MACHINE USING THE FOLLOWING COMMAND:**

**COMMAND: ssh-copy-id** [**velocity@10.0.0.17**](mailto:velocity@10.0.0.17)

**ssh-copy-id velocity@10.0.0.38**

**ssh-copy-id velocity@10.0.0.85**

**ssh-copy-id velocity@10.0.0.106**

****

**AS WE HAVE ALREADY COPIED SO IT WON’T PROMT FOR THE PASSWORD. THE VERY FIRST TIME YOU COPY THE KEY IT SHOULD ASK YOU FOR PASSWORD OF VELOCITY USER.**

**ONCE YOU ENTER THE PASSWORD THE NEXT TIME YOU DO SSH IT WON’T ASK YOU FOR KEY.**

**STEP5) ON DEV\_OPS MACHINE (MASTER MACHINE) WE WILL INSTALL JAVA.**

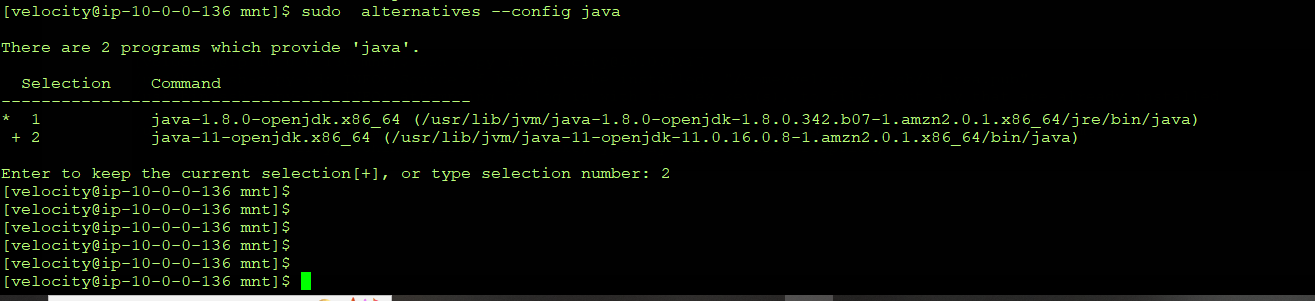
**COMMAND: sudo yum install java-1.8.0\_openjdk-devel.x86\_64 -y**

**sudo amazon-linux-extras install java-openjdk11=latest –y**

**MAKE SURE YOU SELECT THE JAVA 11 AS THE TOP PRORIOTY AS WE ARE GOING TO BUILD JENKINS ON JAVA 11. TO DO SO WE NEED TO USER ALTERNATIVES COMMAND**

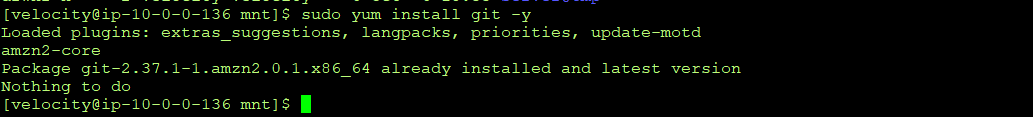
**COMMAND: sudo alternatives –config java**

**MAKE SURE YOU SELECT THE JAVA 11 OPTION.**

****

**NOW WE NEED TO INSTALL GIT:**

**COMMAND: yum install git -y**

****

**STEP6) NOW WE NEED TO CREATE VARIOUS MOUNT POINTS FOR VARIOUS PACKAGES**

**BUILD-TOOLS-----IT WILL BE OUR MAVEN HOME DIR**

**SERVER---- HOME DIR FOR OUR TOMCAT SERVER**

**PROJECT—WE WILL BUILD OUR PROJECT HERE ( ALSO WE WILL CLONE IT HERE USING GIT)**

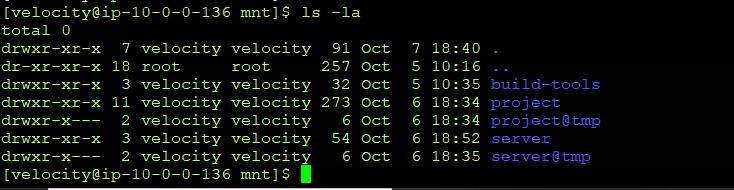
**COMMAND: cd**

**cd /mnt**

**mkdir build-tools**

**mkdir projects**

**mkdir server**

****

**NOW TO INSTALL MAVEN FIRST CD TO BUILD-TOOLS**

**COMMAND: cd /mnt/build-tools**

**GO ON THE MAVEN HOME PAGE AND COPY THE .ZIP LINK TO DOWNLOAD MAVEN**

**COMMAND:** [**https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.zip**](https://dlcdn.apache.org/maven/maven-3/3.8.6/binaries/apache-maven-3.8.6-bin.zip)

****

**ONCE YOU HAVE DOWNLOADED IT INTO YOUR REQ DIR WE NEED TO UNZIP IT USING UNZIP COMMAND : unzip apache-maven-3.8.6**

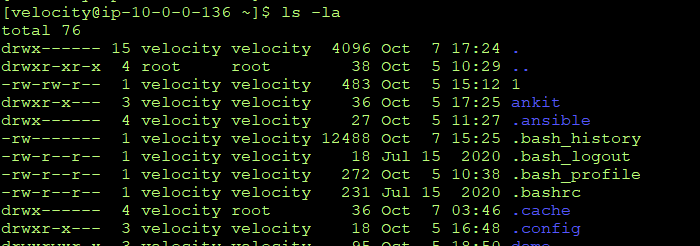
**NOW WE NEED TO SET THE ENV VARIABLE FOR MAVEN TO WORK**

**TO DO SO EDIT IT .BASH\_PROFILE FILE.**

**COMMAND: cd**

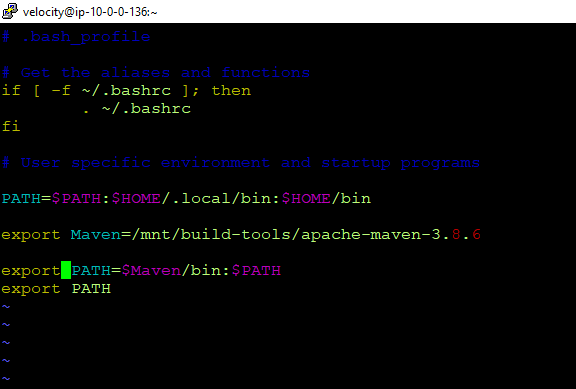
**ls -la**

**sudo vi .bash\_profile**

****

**ONCE YOU GOT INTO THE FILE WE NEED TO SET MAVEN ENV VAR PATH**

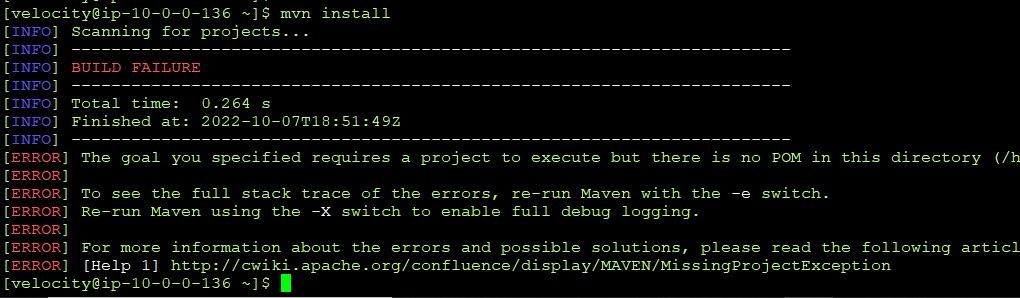
**REFER TO SCREENSHOT BELOW AND MAKE THE CHANGES AS SHOWN**

****

**LOUTOUT AND LOGIN BACK FOR CHANGES TO WORK**

**YOU CAN CHECKIF YOUR ENV VAR IS WORKING BY RUNING THE MVN COMMAND ANYWHERE.**

**YOU SHOULD SEE SOME ERRORS AS WE DO NOT HAVE THE POM FILE AS OF NOW**

****

**NOW TO INSTALL TOMCAT FIRST CD TO SERVER**

**COMMAND: cd /mnt/server**

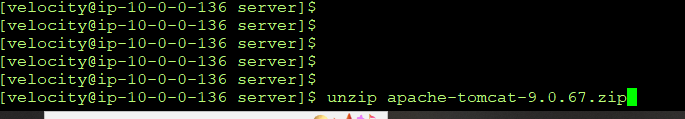
**GO ON THE APACHE TOMCAT HOME PAGE AND COPY THE APACHE TOMCAT 9.0 .ZIP LINK TO DOWNLOAD TOMCAT SERVER**

**COMMAND:** [**https://downloads.apache.org/tomcat/tomcat-9/v9.0.68/bin/apache-tomcat-9.0.68.zip.asc**](https://downloads.apache.org/tomcat/tomcat-9/v9.0.68/bin/apache-tomcat-9.0.68.zip.asc)

****

**UNZIP IT USING THE UNZIP COMMAND**

**COMMAND: unzip**

****

**NOW AFYER UNZIP WE NEED TO GIVE (777 PERMISSION TO THE ENTIRE TOMCAT DIR)**

**COMMAND: sudo chmod –R 777 apache-tomcat-9.0.67.zip**

**NOW WE NEED TO START THE TOMCAT SERVICES**

**FOR DOING SO WE NEED TO GO INTO /apache-tomcat-9.0.67/bin**

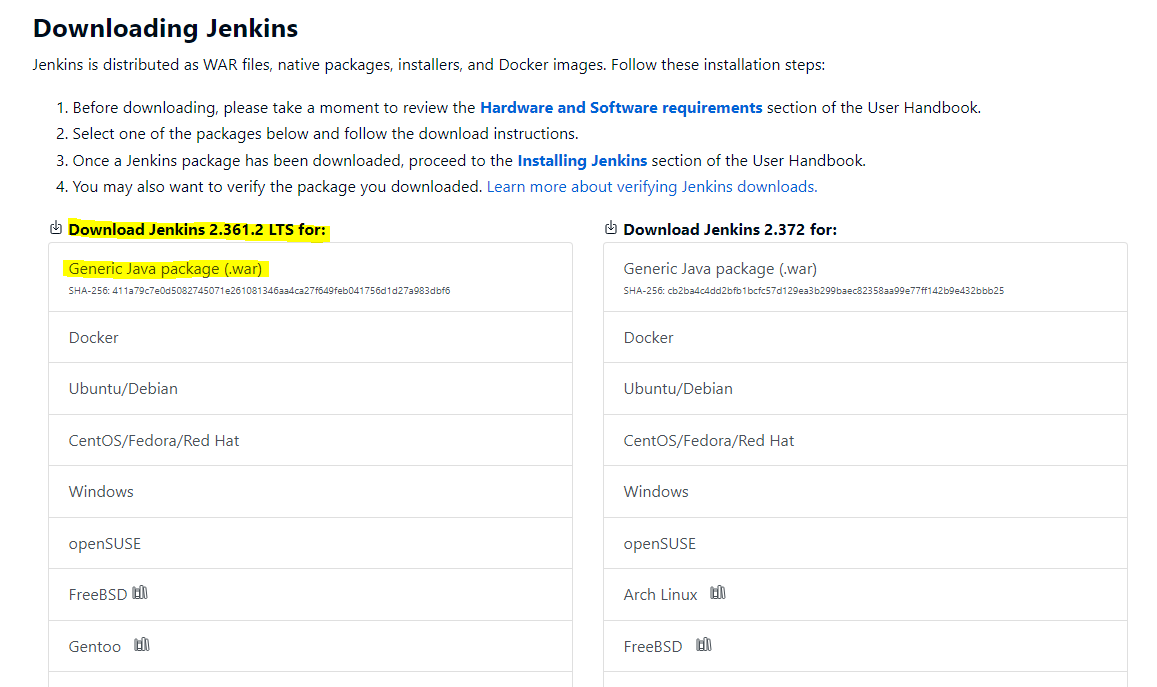
**HERE WE NEED TO START THE TOMCAT SERVICES BY RUNNING FOLLWING COMMAND**

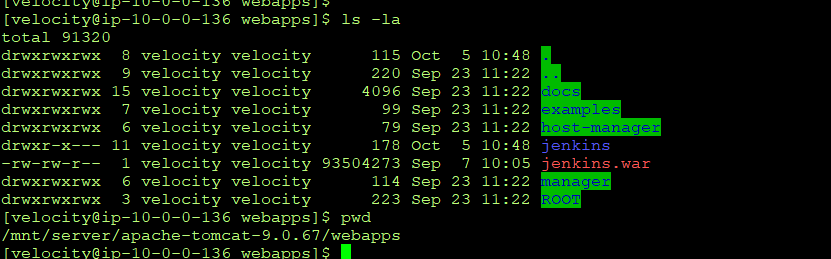
**COMMAND: ./startup.sh**

****

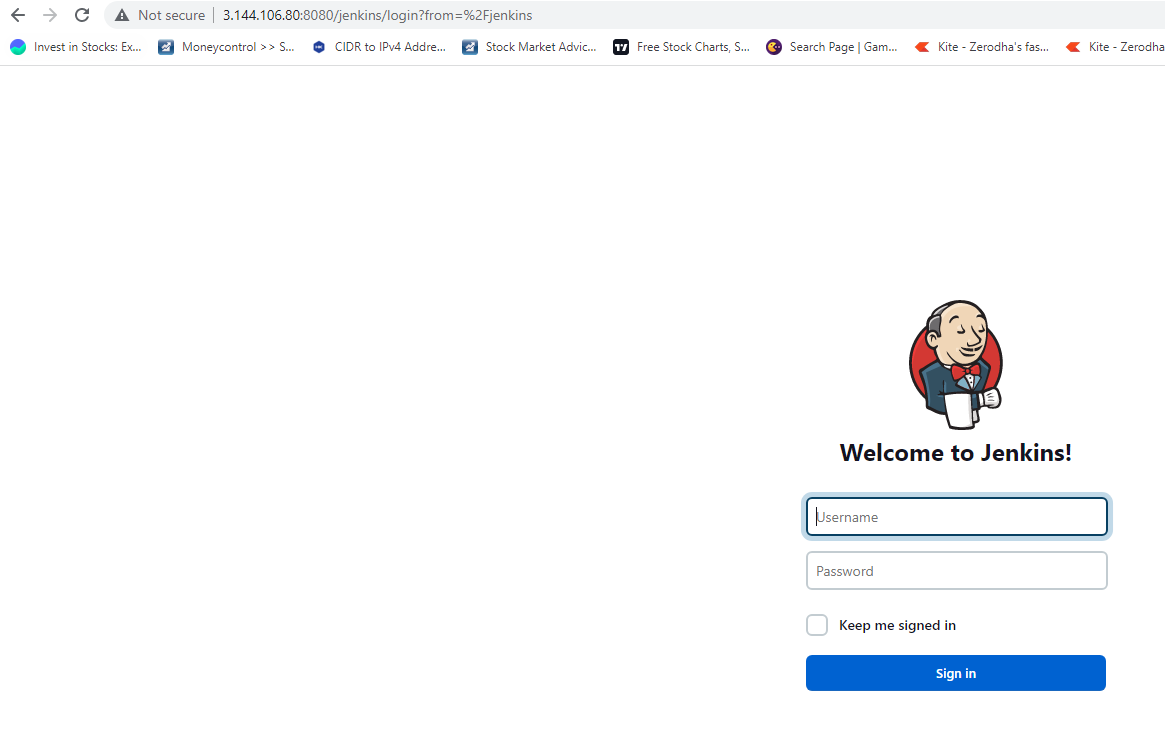
**NOW THAT YOUR TOMCAT IS UP AND RUNNING WE NEED TO INSTALL JENKINS ONTO IT**

**FOR THAT WE NEED TO VISIT JENKINS OFFICIAL SITE AND COPY THE .WAR JENKINS TO /apache-tomcat-9.0.67/webapps/**

**  
COPY THE URL AND WGET INTO THE WEBAPPS PATH**

****

**NOW AS TOMCAT IS ALREADY UP AND RUNNING, WE NEED TO JUST GO THE THE INSTANCE ( MASTER-DEV\_OPS) PUBLIC IP AND ACCESS THE FOLLOWING URL TO INSTALL THE JENKINS**

****

**MALE SURE TO OPEN THE PORTS ON SECURITY GROUP.**

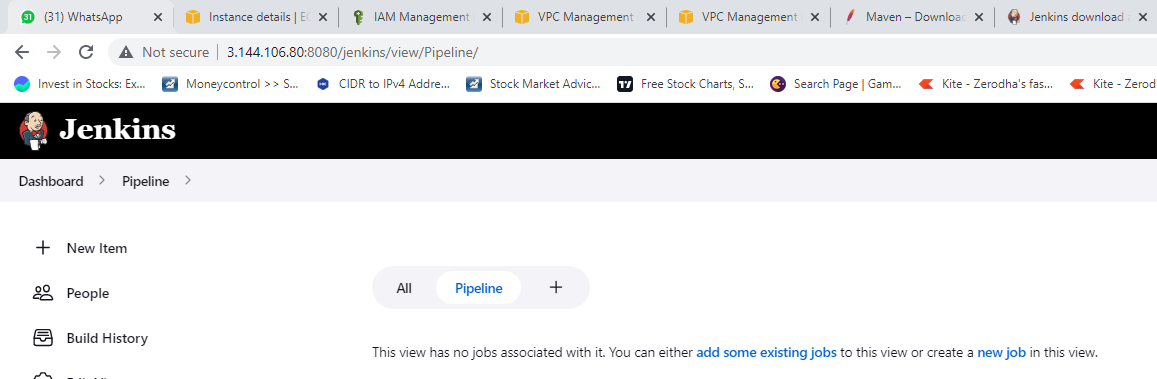
**NOW WE HAVE ALREADY INSTALLED AND CONFIGURE JENKINS.**

**FOR THE FIRST TIME IT WILL ASK YOU TO PROVIDE A PASSWORD FROM YOUR MACHINE AND THEN WILL ASK YOU TO ASIGN USER NAME AND ORTHER DETAILS.**

**THEN IT WILL ASK YOU FOR THE PLUGIN.**

**ONCE YOUR ARE DONE WITH IT YOU WILL BE ON YOUR**

**JENINKS HOMEPAGE**

****

**STEP7) NOW WE NEED TO ANSIBLE-PLAYBOOK ON MASTER TO INSTALL DEPENDENCY ON ALL THE NODE MACHINE ( DEV1, DEV2, QA1, QA2)**

**WE ARE CREATING PLAYBOOK ONTO THE SERVER DIR ( WE CAN USE IT FROM GIT HUB AS WELL BUT AS OF NOW WE ARE CREATING IT OVER THIS DIR AND WILL USE IT FROM HERE TO INSTALL THE REQ SOFT ON NODES)**

**COMMAND: cd /mnt/server**

**Sudo vi project.yaml**

**---**

**- hosts: webserver**

**user: velocity**

**become: yes**

**connection: ssh**

**gather\_facts: yes**

**tasks:**

**# - name: insatlling java**

**# action: yum pkg=java\* state=absent**

**# - name: insatlling java11**

**#iaction: yum pkg=java-openjdk11 state=present**

**- name: install java 11**

**command: sudo amazon-linux-extras install java-openjdk11=latest -y**

**- name: install docker**

**action: yum pkg=docker state=present**

**- name: start docker**

**action: service name=docker state=started**

**# - name: docker-compose install-1**

**# command: curl -L https://github.com/docker/compose/releases/download/1.21.0/docker-compose-`uname -s`-`uname -m` | sudo tee /usr/local/bin/docker-compose > /dev/null**

**- name: Install docker-compose from official github repo**

**remote\_user: velocity**

**get\_url:**

**url : https://github.com/docker/compose/releases/download/1.29.2/docker-compose-Linux-x86\_64**

**dest: /usr/local/bin/docker-compose**

**mode: 'u+x,g+x'**

**remote\_src: yes**

**- name: docker-compose install-2**

**command: sudo chmod +x /usr/local/bin/docker-compose**

**# - name: docker-compose install-3**

**# command: ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose**

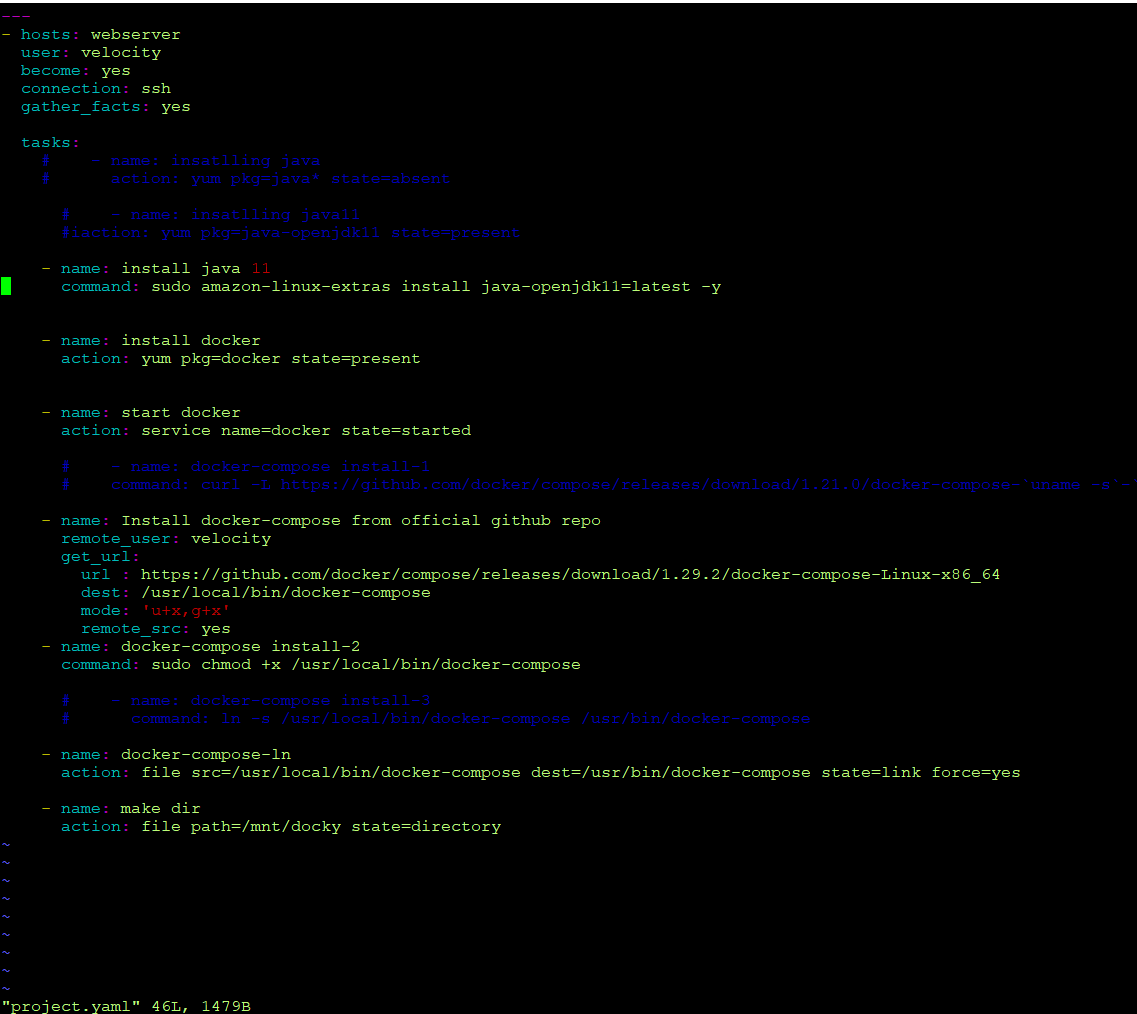
**- name: docker-compose-ln**

**action: file src=/usr/local/bin/docker-compose dest=/usr/bin/docker-compose state=link force=yes**

**- name: make dir**

**action: file path=/mnt/docky state=directory**

**(NOTE: YOU HAVE TO RUN THE PLAYBOOK HERE ONLY AS WE WILL NEED JAVA 11 ON EACH OF NODE MACHINE TO CONNECT WITH OUR JENKINS SERVER)**

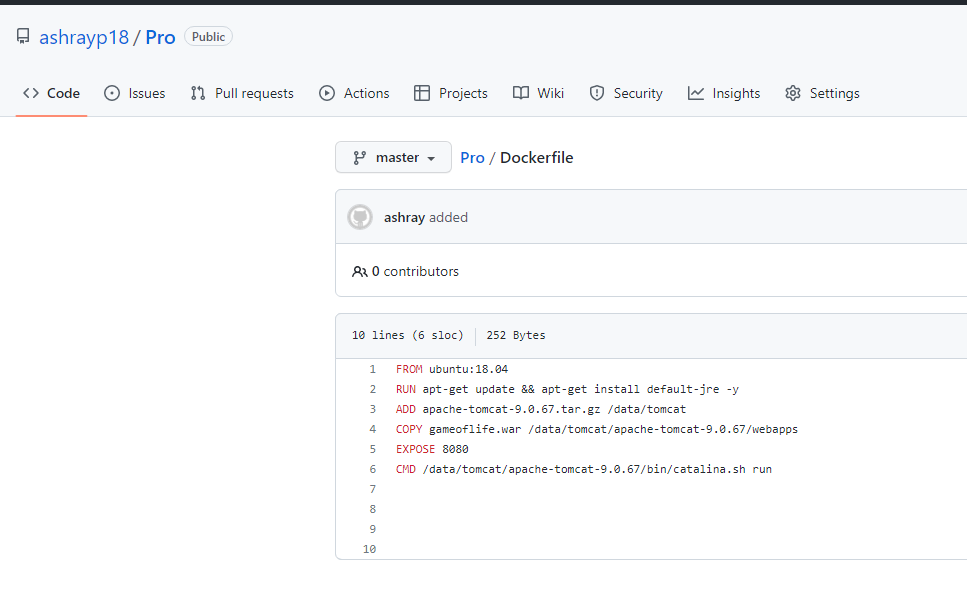
****

**( NOTE : WE WILL INSTALL JAVA 11, DOCKER, START DOCKER SERVICES, DOCKER-COMPOSE INSTALLATION, SETTING THE SL FOR DOCKER-COMPOSE, AND WE WILL ALSO CREATE ONE DIR CALLED AS DOCKY WHICH WE WILL USE LATER TO CLONE DOCKERFILE AND DOCKER-COMPOSE FILE FROM GIT HUB)**

**STEP8) NOW WE NEED TO CREATE DOCKERFILE AND DOCKER-COMPOSE ON MASTER MACHINE AND PUSH IT OVER THE GITHUB.**

**COMMAND: vi Dockerfile**

|  |
| --- |
| FROM ubuntu:18.04 |
|  | RUN apt-get update && apt-get install default-jre -y |
|  | ADD apache-tomcat-9.0.67.tar.gz /data/tomcat |
|  | COPY gameoflife.war /data/tomcat/apache-tomcat-9.0.67/webapps |
|  | EXPOSE 8080 |
|  | CMD /data/tomcat/apache-tomcat-9.0.67/bin/catalina.sh run |
|  |  |
|  |  |

****

**vi docker-compose.yaml**

|  |
| --- |
| version:'3' |
|  | services: |
|  | one: |
|  | build: |
|  | context: /mnt/docky/ |
|  | image: server-1 |
|  | ports: |
|  | - "8091-8092:8080" |

|  |  |
| --- | --- |
|  |  |

**AS YOU CAN SEE WE HAVE ALREADY UPDATED BOTH THE FILE TO OUR GIT REPO CALLED AS PRO**

**STEP9) NOW WE WILL GO ONTO JENKINS SERVER TO CONFIGURE GLOBAL TOOL SETTING**

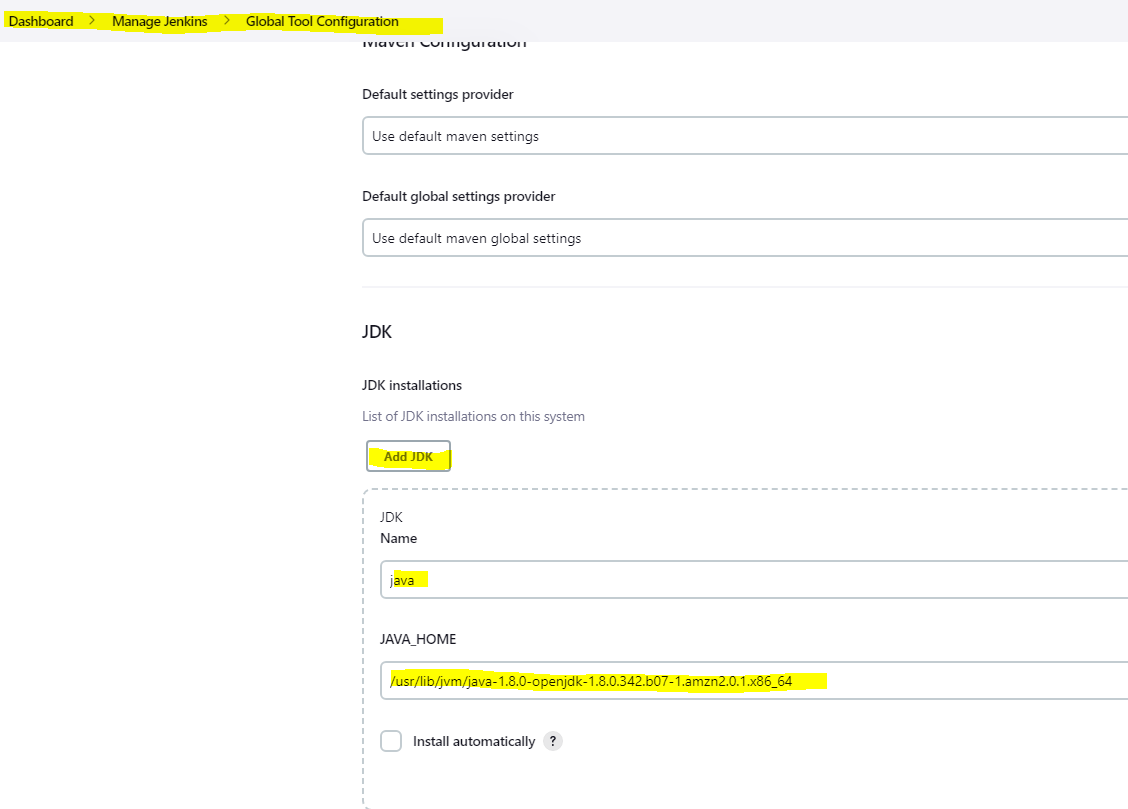
**HERE WE ARE RUNNING JENKINS MASTER ON JAVA 11 AND OUR PROJECT ( GAMEOFLIFE) NEEDS JAVA 1.8 TO BUILD. HENCE WE NEED TO DEFINE TOOLS.**

**TOOLS WILL HELP OUR JENKINS TO RUN ON JAVA 11 BUT AND USE JAVA 1.8 IN THE BACKGROUND TO RUN JOB/STAGE.**

**SEE THE SETTING AS BELOW.**

**MAKE SURE YOU REMEMBER THE NAME YOU HAVE PROVIDED UNDER JDK AS WE ARE GOING TO USE IT FURTHER IN OUT JOBS.**

**PATH: DASHBOARD >> MANAGE JENKINS >> GLOBAL TOOL CONFIGURATION**

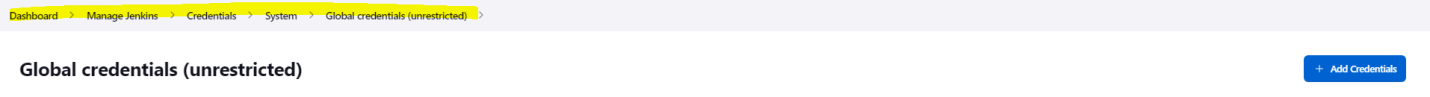
****

**STEP10) NOW WE NEED TO CONNECT JENKINS MASTER TO JENKINS HOST**

**FOR THAT WE FIRST NEED TO CREATE CREDENTIAL SO THAT OUR JENKINS MASTER CAN USE OUR SSH KEYS TO CONNECT TO HOST**

**FOR THIS WE NEED TO GO ON**

**MANAGE JENKINS >> CREDENTIALS >> SYSTEM >> GLOCBAL CREDENTIAL>>**

****

**CLICK ON CREATE CREDENTIAL ON THE RIGHT SIDE.**

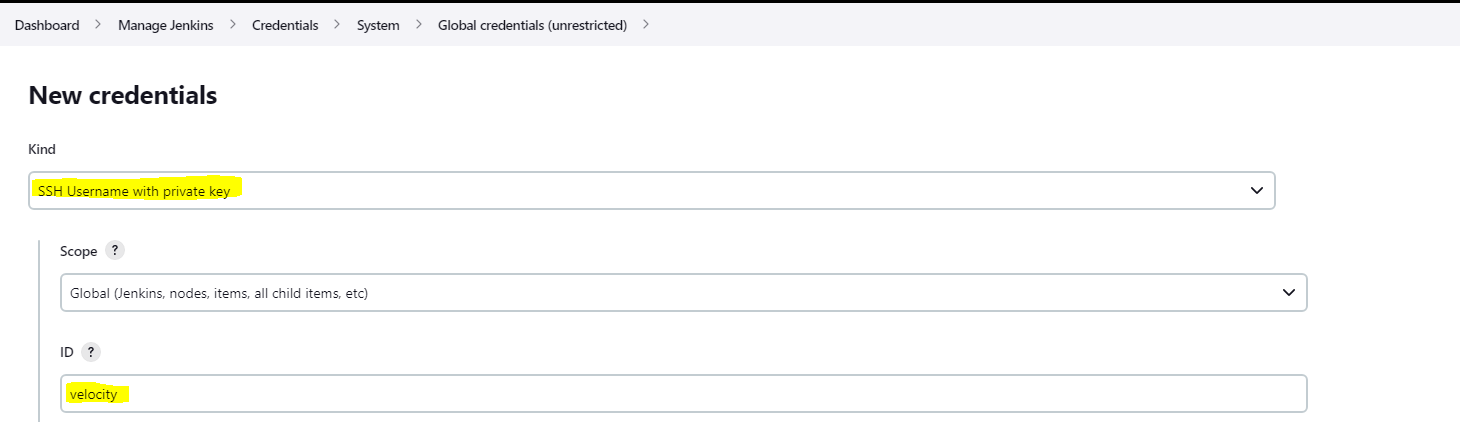
**FILL IN THE DETAILS**

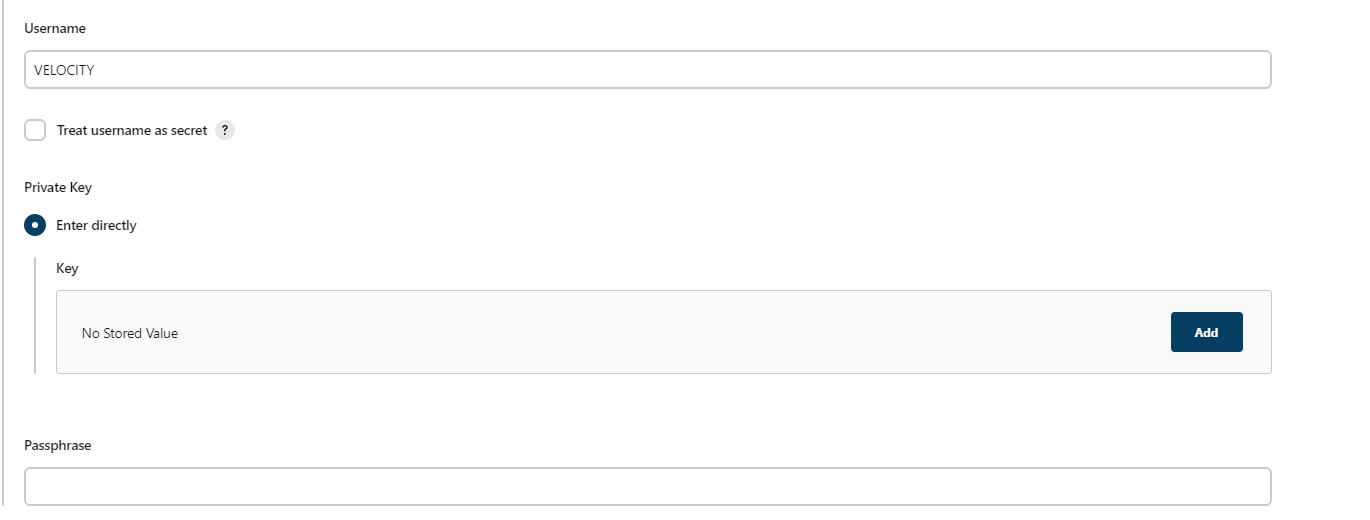
**SELECT “ SSH USERNAME WITH PRIVATE KEY”**

**ID AS “VELOCITY”**

**USERNAME AS “VELOCITY”**

**PASSWORD AS “**

****

**CLICK ON ADD TO ADD THE PRIVATE KEY**

**SO WE NEED THE KEY NOW FOR USER VELOCITY.**

**FOR THAT USE THE FOLLOWING COMMAND:**

**COMMAND: cd**

**cd .ssh/**

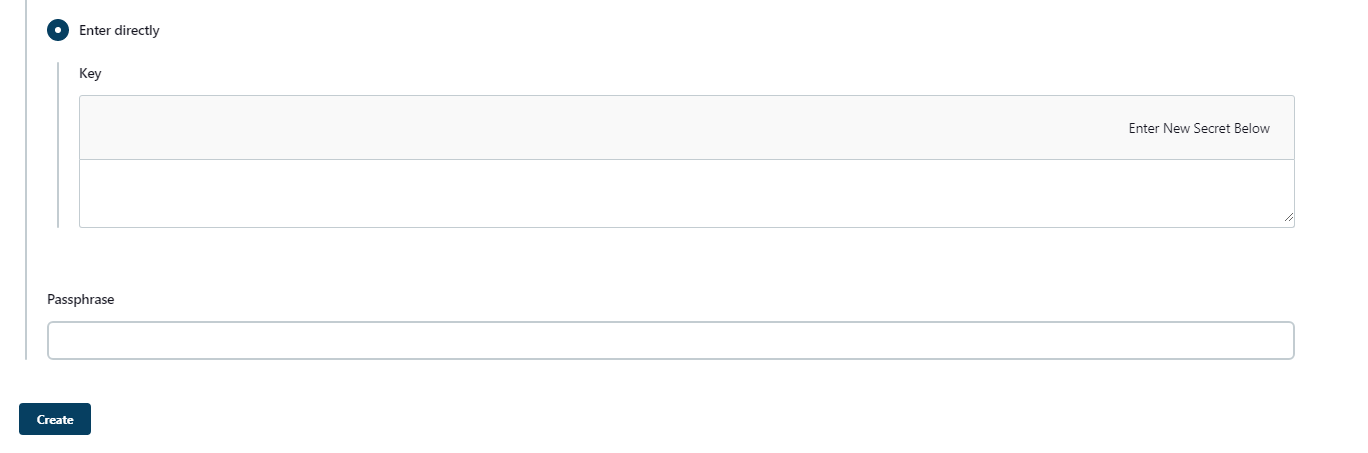
**ls -la**

**YOU WILL SEE THE PUB AND PRVATE KEY WHICH YOU HAVE CREATED EARLIER USING THE SSH-KEYGEN COMMAND.**

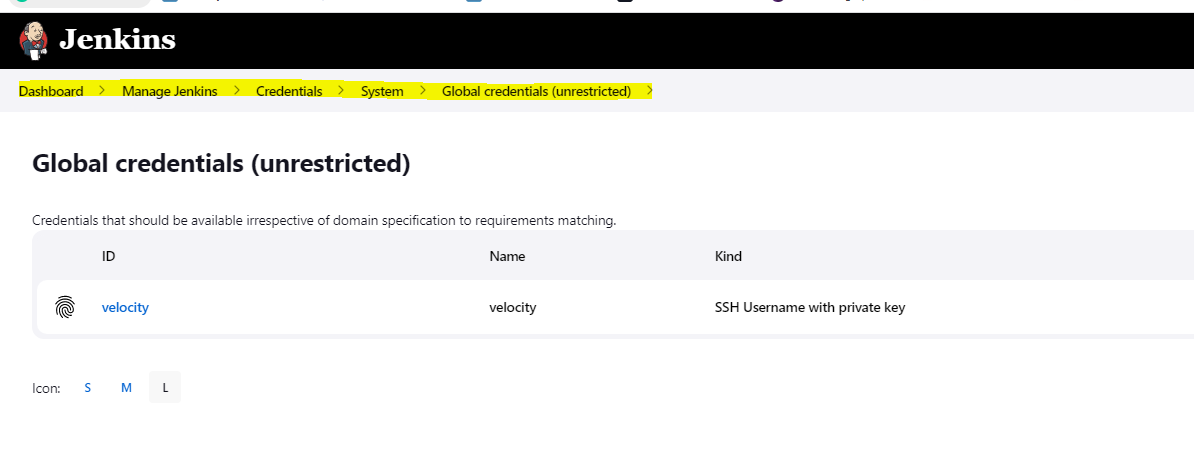
**NOW COPY THE CONTENT OF ID\_RSA FILE AS SHOWN BELOW**

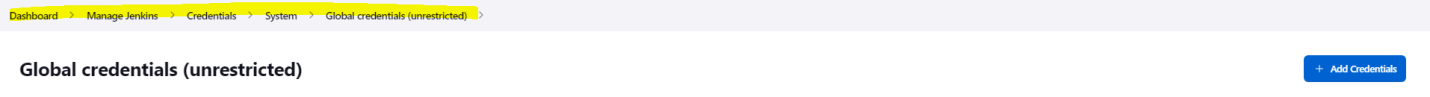
****

**NOW PASTE THE CONTENT OVER HERE UNDER THE KEY OPTION AND CLICK ON CREATE**

****

**YOU WILL SEE A CREDENTIAL HAS BEEN CREATED.**

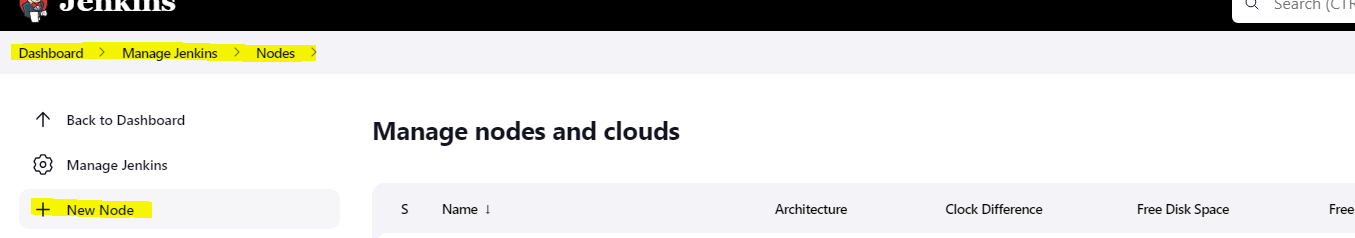
****

****

**STEP11) NOW WE NEED TO ADD THE SLAVE FOR OUR JENKINS MASTER**

**GO TO DASHBOARD >> MANAGE JENKINS >> NODES>>**

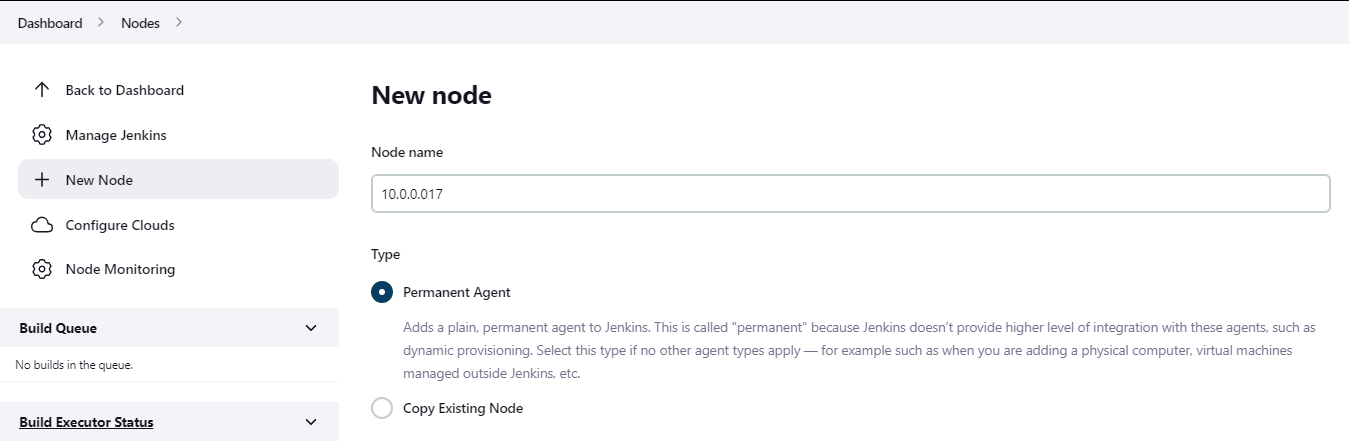
**CLICK ON +NEW NODE**

****

**ENTER THE NAME AS 10.0.0.17**

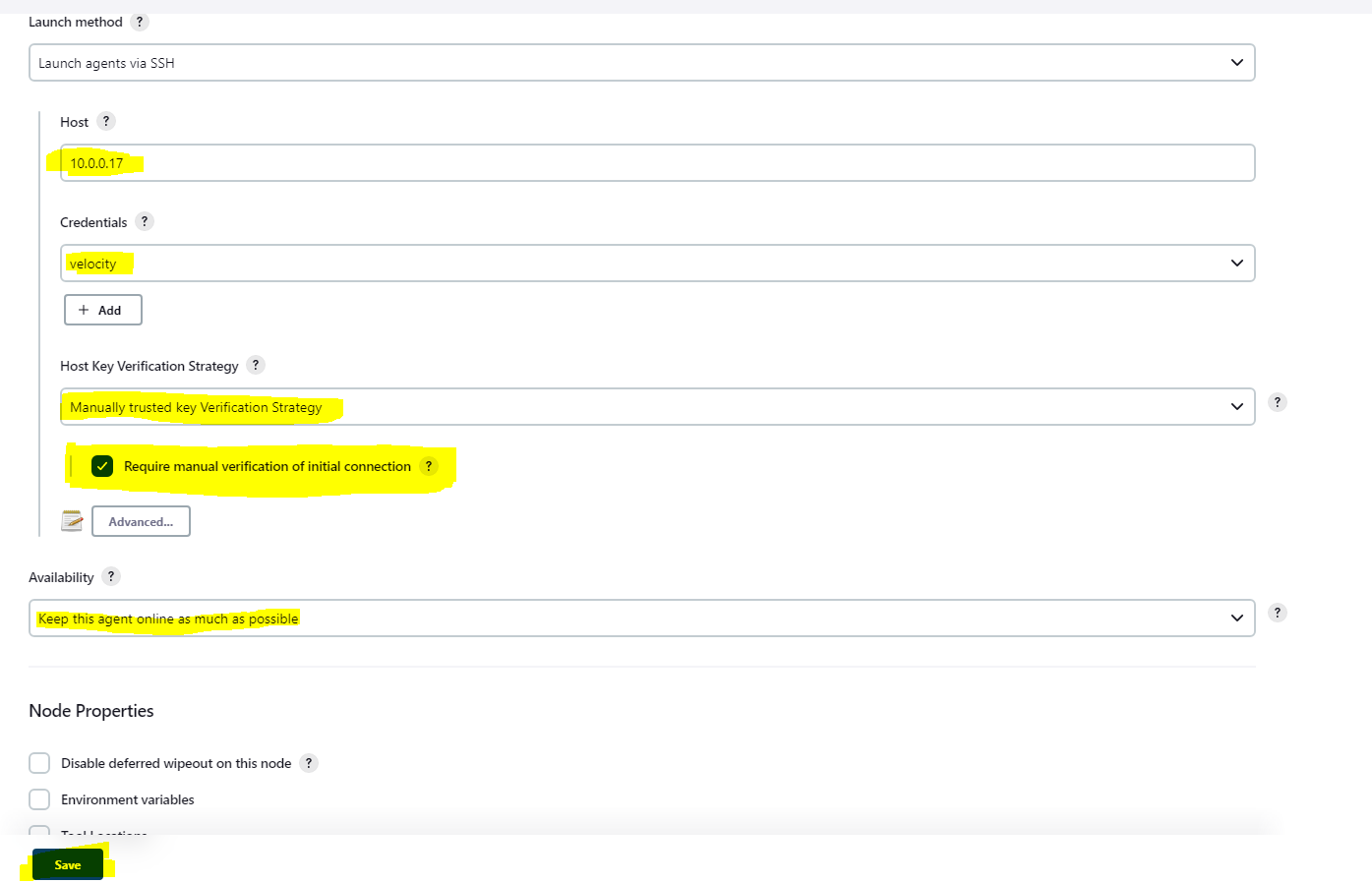
**NOTE ( WE ARE ADDING THE IP OF DEV1 MACHINE )**

**CLICK ON CREATE AFTER ADDING THE NODE NAME**

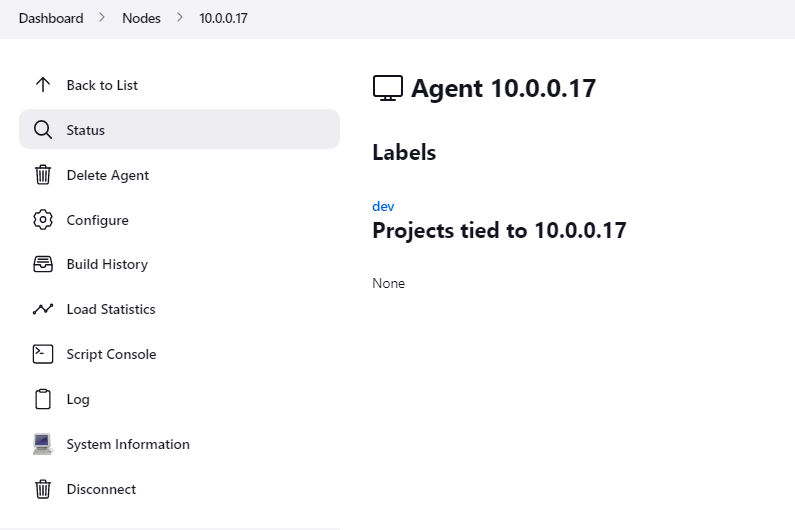
****

**CONFIGURE THE NODE AS SHOWN IN PICTURE BELOW**

****

****

**CLICK ON SAVE AND CLICK ON AGENT.**

****

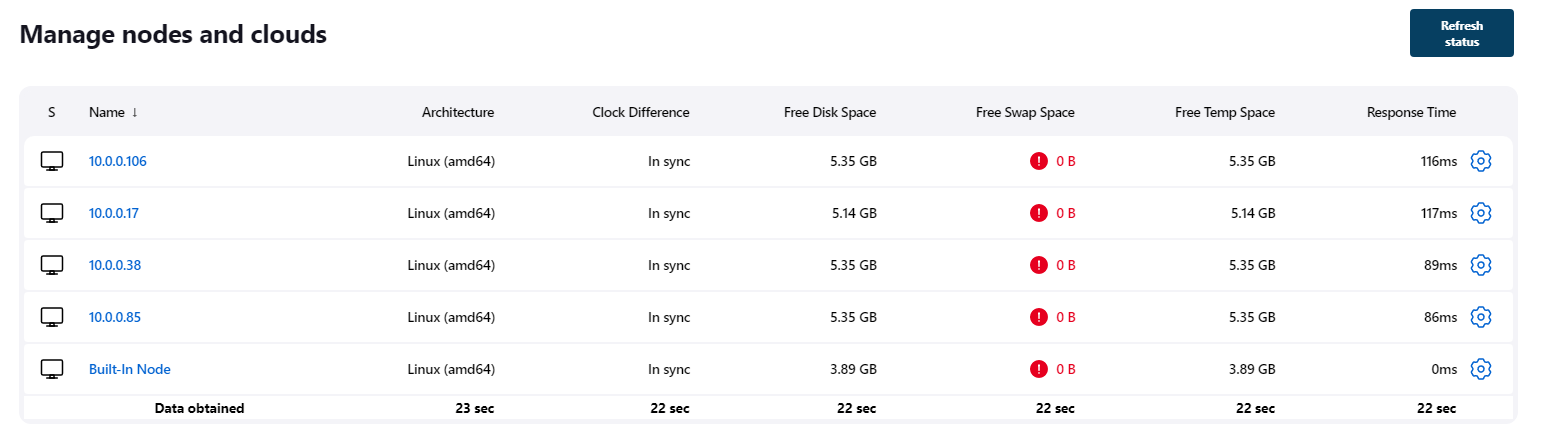
**NOTE: CURRENTLY WE HAVE ALREADY CONNECTED IT BUT YOU WON’T BE ABLE TO CONNECT IT UNTIL YOU GRANT THE MANNUAL SSH PERMISSION WHICH IT SHOWS BELOW LOG OPTION**

**ONCE YOU DID THIS THEN YOUR AGENT SHOULD BE ONLINE**

**( NOTE WE NEED TO HAVE JAVA 11 ON ALL THE NODES BEFORE FOR THE CONNECTION TO BE MADE**

**CONSIDER RUNNING THE PLAYBOOK FIRST FROM THE MASTER SO YOUR CONNECTION WILL BE SUCCESSFUL.)**

**SO NOW MAKE SUCH CONNECTION WITH EACH NODE ( DEV1, DEV2, QA1, Q2)**

****

**STEP12) NOW WE WILL CREATE JENKINS JOB:**

**WE WILL CREATE 3 JENKINS JOB**

**JOB1--- WE WILL CLONE THE PROJECT FROM GIT HUB AND BUILD IT ON MASTER**

**JOB2—WE WILL DEOPLY CONATINER USING DOCKER-COMPOSE AND DOCKERFILE ON EACH NODE (DEV1,DEV2, QA1, QA2), ALSO WE WILL COPY THE .WAR FROM OUR MASTER TO EACH SLAVE.**

**JOB3—HERE WE WILL BUILD JOB1-JOB2 SO THAT IT FIRST TRIGGERS JOB1 AND LATER JOB2.**

**JOB1:**

**WE WILL CLONE GAMEOFLIFE PROJECT UNDER /MNT/PROJECT DIR.**

**NOTE: WE WILL HAVE TO ASSIGN TOOLS WHICH WE HAVE CONFIGURED EARLIER**

**tools{**

**jdk 'java'**

**}**

****

**ALSO WE WARE RUNNING THE PLAYBOOK FROM JOB1. (BUT WE WILL HAVE TO RUN IN MANUALLY FIRST ON MASTER AFTER CREATING IT ELSE WE WILL NOT BE ABLE TO MAKE NODE CONNECTION DUE TO DEPENDENCY)**

**pipeline{**

**agent {**

**node {**

**label ' built-in'**

**}**

**}**

**stages {**

**stage ("clone project") {**

**tools{**

**jdk 'java'**

**}**

**steps{**

**sh " sudo chown -R velocity:velocity /mnt"**

**dir ("/mnt/project/"){**

**git 'https://github.com/ashrayp18/game-of-life.git'**

**sh 'mvn install -DskipTests'**

**}**

**sh " cp /mnt/project/gameoflife-web/target/gameoflife.war /mnt/docky/"**

**dir ("/mnt/docky/"){**

**sh " wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.67/bin/apache-tomcat-9.0.67.tar.gz"**

**}**

**dir ("/mnt/server"){**

**sh "ansible-playbook project.yaml"**

**}**

**}**

**}**

**}**

****

**JOB2: CREATE A NEW PIPELINE JOB CALLED AS JOB2.**

**HERE FIRST WE WILL COPY THE GAMEOFLIFE.WAR FILE WHICH WE HAVE BUILD AND COPIED ON ON JOB1. FOR DOING SO WE WILL USE THE SCP COMMAND**

**WE WILL NAME THE STAGE AS MASTER-SCP:**

**pipeline{**

**agent none**

**stages{**

**stage ('MASTER\_SCP') {**

**agent{**

**node {**

**label 'built-in'**

**}**

**}**

**steps{**

**sh "scp /mnt/docky/gameoflife.war velocity@10.0.0.17:/mnt/docky/"**

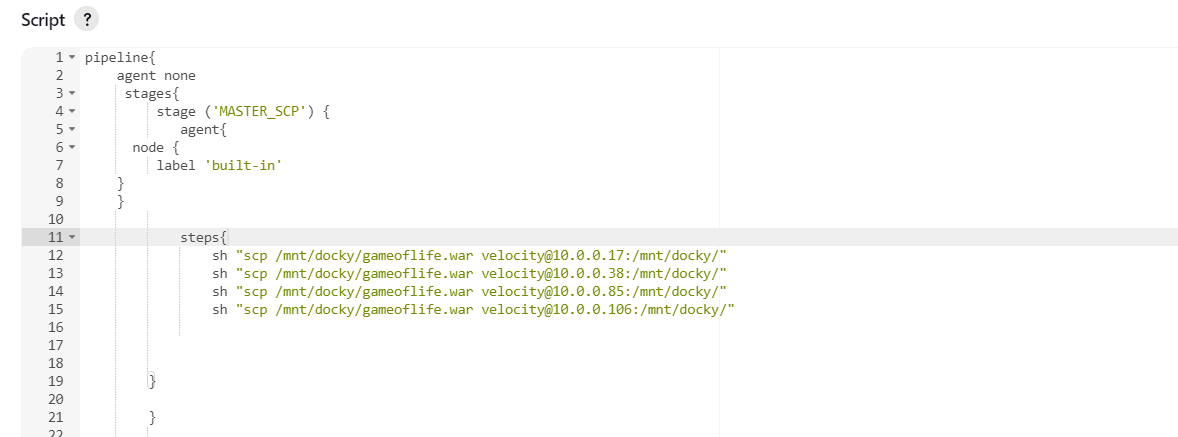
**sh "scp /mnt/docky/gameoflife.war velocity@10.0.0.38:/mnt/docky/"**

**sh "scp /mnt/docky/gameoflife.war velocity@10.0.0.85:/mnt/docky/"**

**sh "scp /mnt/docky/gameoflife.war velocity@10.0.0.106:/mnt/docky/"**

**}**

**}**

****

**NOW WE WILL WRITE ANOTHER STAGE FOR NODE1**

**HERE WE WILL CLONE THE DOCKERFILE AND DOCKERCOMEFILE ON A DIR CALLED AS DOCKY.**

**NOTE: WHILE WRITING DOCKERFILE WE HAVE USED THE COPY AND ADD COMMAND WHICH REQ GAMEOFLIFE.WAR FILE AND APACHE-TOMAT:9 TAR.GZ FILE INTO THE SAME DIR.**

**SO WE WILL WGET THE TOMCAT FILE AND ALSO INSTALL GIT INTO THE EACH SLAVE AND CLONE OUR GIT REPO (PRO) WHICH HAS DOCKERFILE AND DOCKER-COMPOSE FILE.**

**stage ('node-1') {**

**agent{**

**node {**

**label '10.0.0.17'**

**}**

**}**

**steps{**

**sh "sudo yum install git -y"**

**sh " sudo chown -R velocity:velocity /mnt/"**

**dir ('/mnt/docky'){**

**git 'https://github.com/ashrayp18/Pro.git'**

**sh " wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.67/bin/apache-tomcat-9.0.67.tar.gz"**

**sh " sudo docker-compose up -d --scale one=2"**

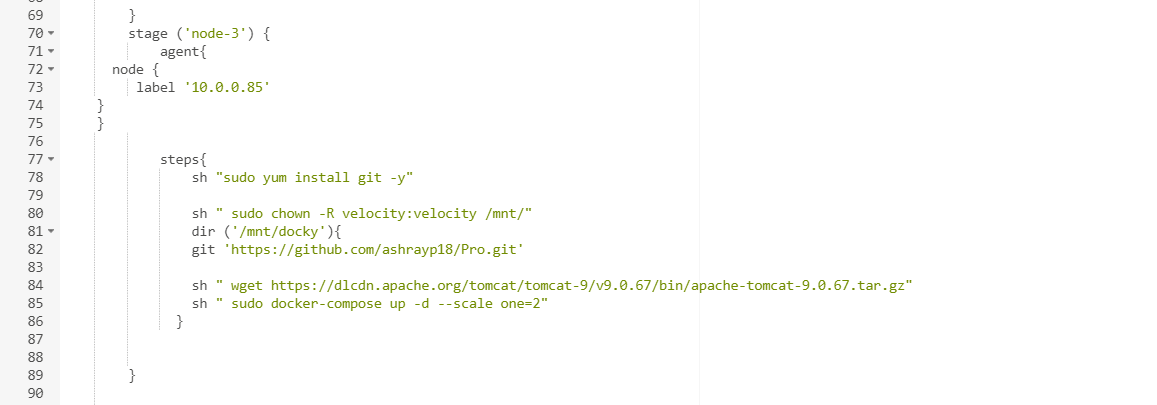
**}**

**}**

****

**SIMILARLY WRITE STAGES FOR EVERYNODE NODE2, NODE3 AND NODE4)**

****

****

****

**NOW THAT WE HAVE WRITTEN TWO JOB (JOB1 AND JOB2)**

**IT IS TIME TO CREATE A NEW JOB CALLED AS JOB3:**

**WE WILL CALL JOB1 AND JOB2 OVER HERE**

**pipeline { //indicate the job is written in Declarative Pipeline**

**agent any //agent specifies where the pipeline will execute.**

**stages {**

**stage ("JOB1-JOB2") { //an arbitrary stage name**

**steps {**

**build 'JOB1' //this is where we specify which job to invoke.**

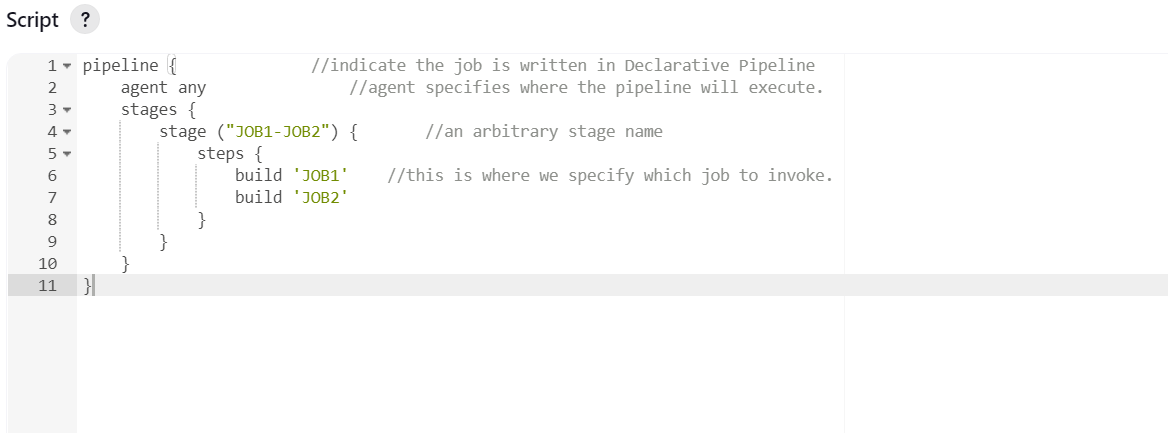
**build 'JOB2'**

**}**

**}**

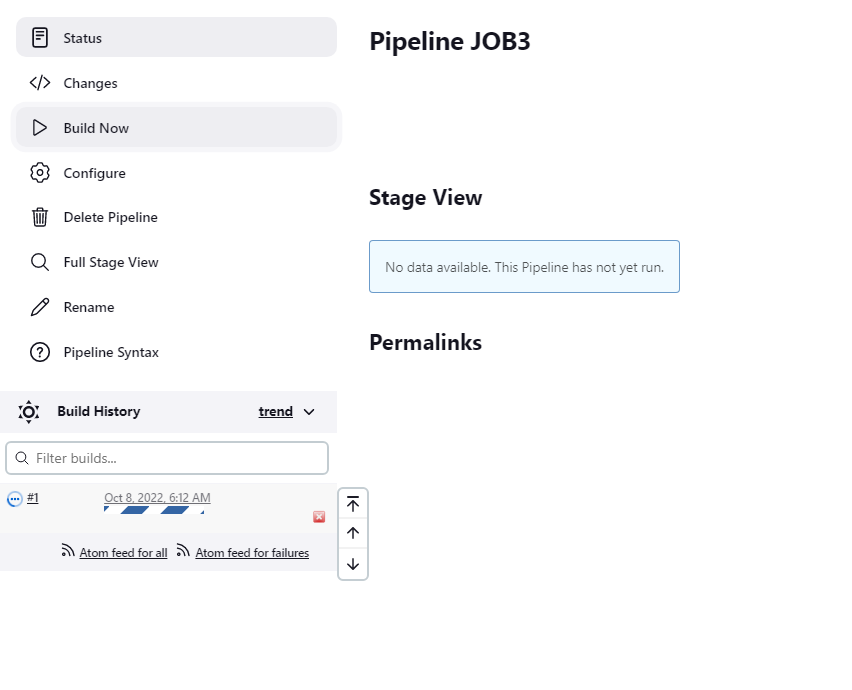
**}**

**}**

****

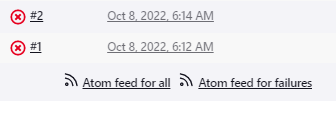
**VERY IMP: PLEASE READ THE ENTIRE CODE ATLEAST 2-3 TIMES TO MAKE SURE IF YOU HAVE NOT MADE ANY MISTAKE.**

**STEP13) RUN JOB3 WHICH WIL TRIGGER JOB1 AND JOB2**

****

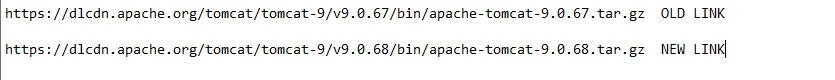
**WE WILL WAIT FOR PIPELINE TO FINISH.**

**WE HAVE ENCOUNTERED BUILD FAIL: TROUBLESHOOT ACCORDINGLY**

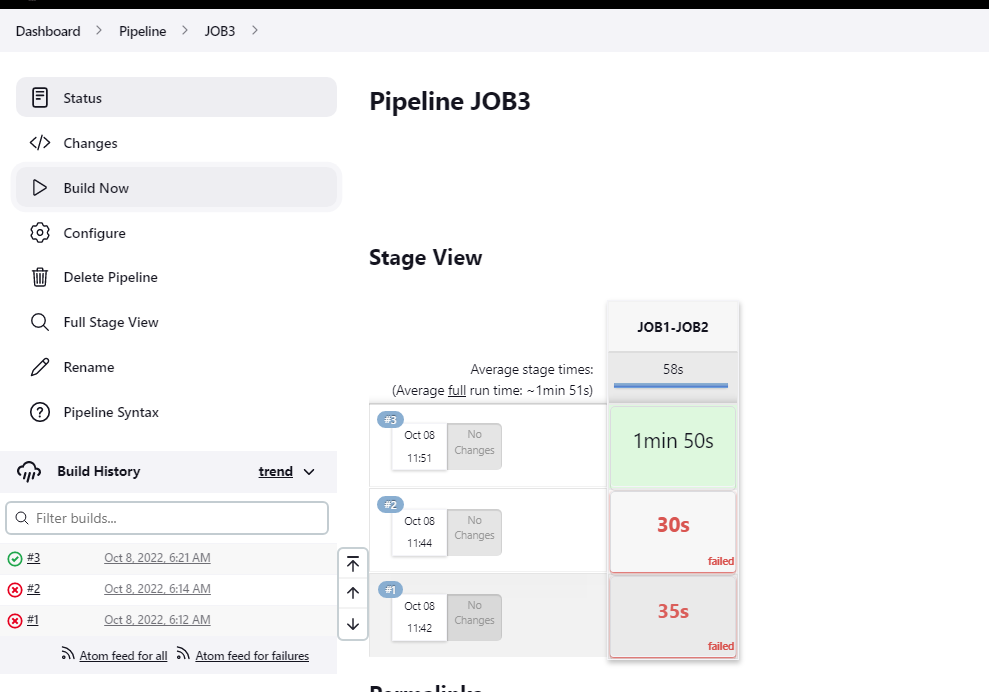
****

**WE GOT ERROR DUR TO APACHE TOMCAT: APACHE TOMCAT HAS NOW RELEASE 9.68 VERSION AND WE ARE REFERRING TO LINK WHICH WAS OF 9.67**

**SOLVED: UPDATED THE LINK ON BOTH THE JOB (JOB1 AND JOB2)**

****

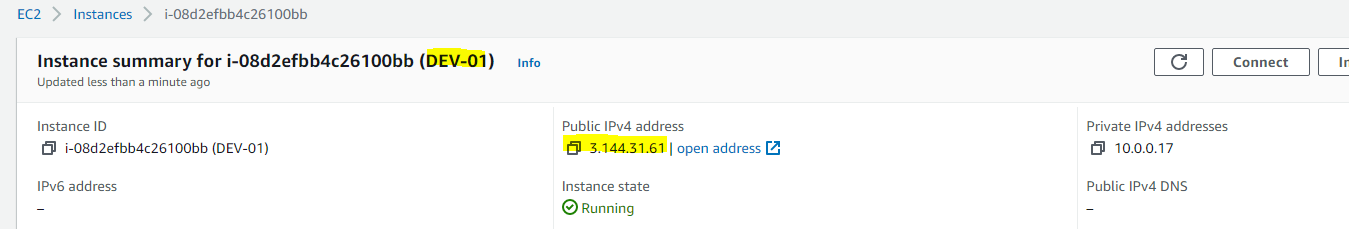
**LET’S CHECK THE STATUS OF OUR JOB3**

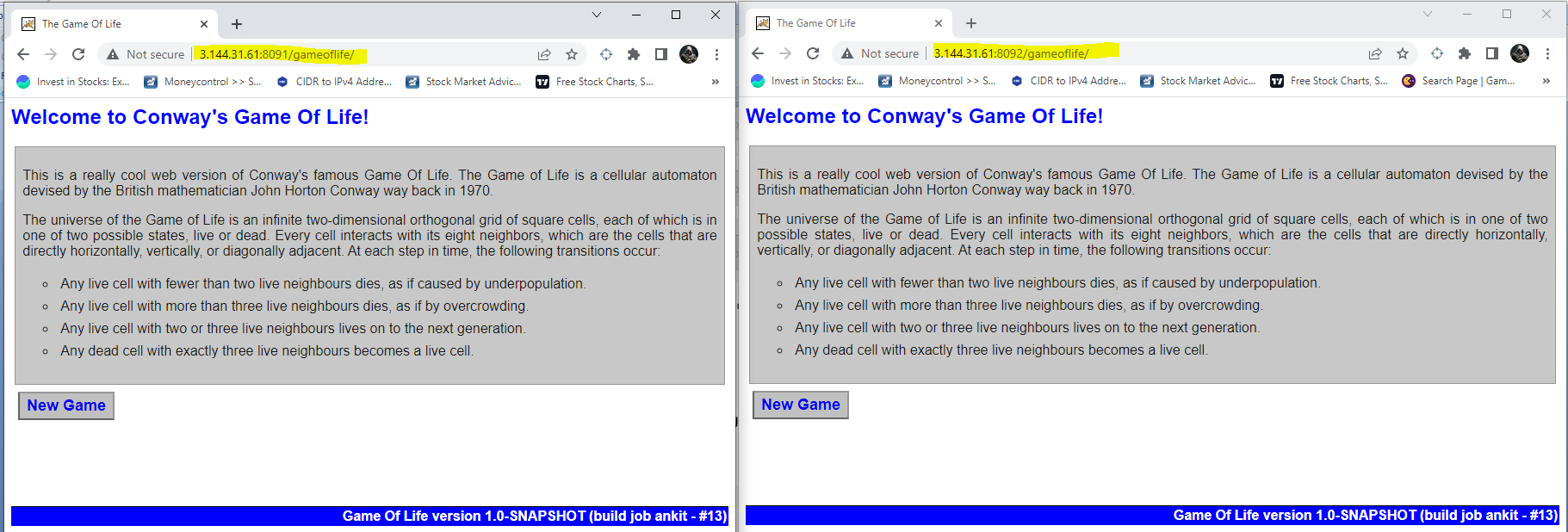
****

**BUILD SUCCESS !!!**

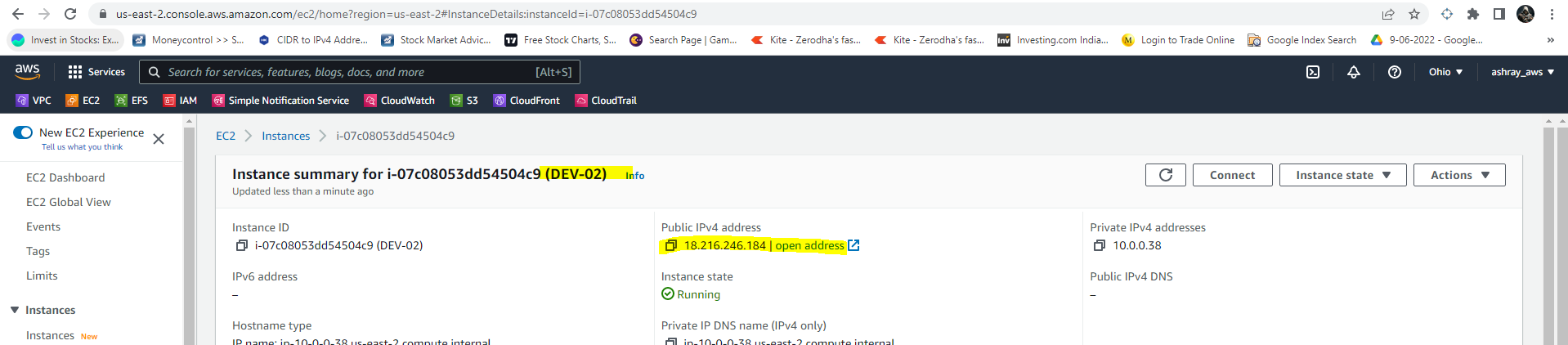
**NOW LETS CHECK DEV1 ENV AND LETS SEE IF OUR GAMEOFLIFE IS RUNNING**

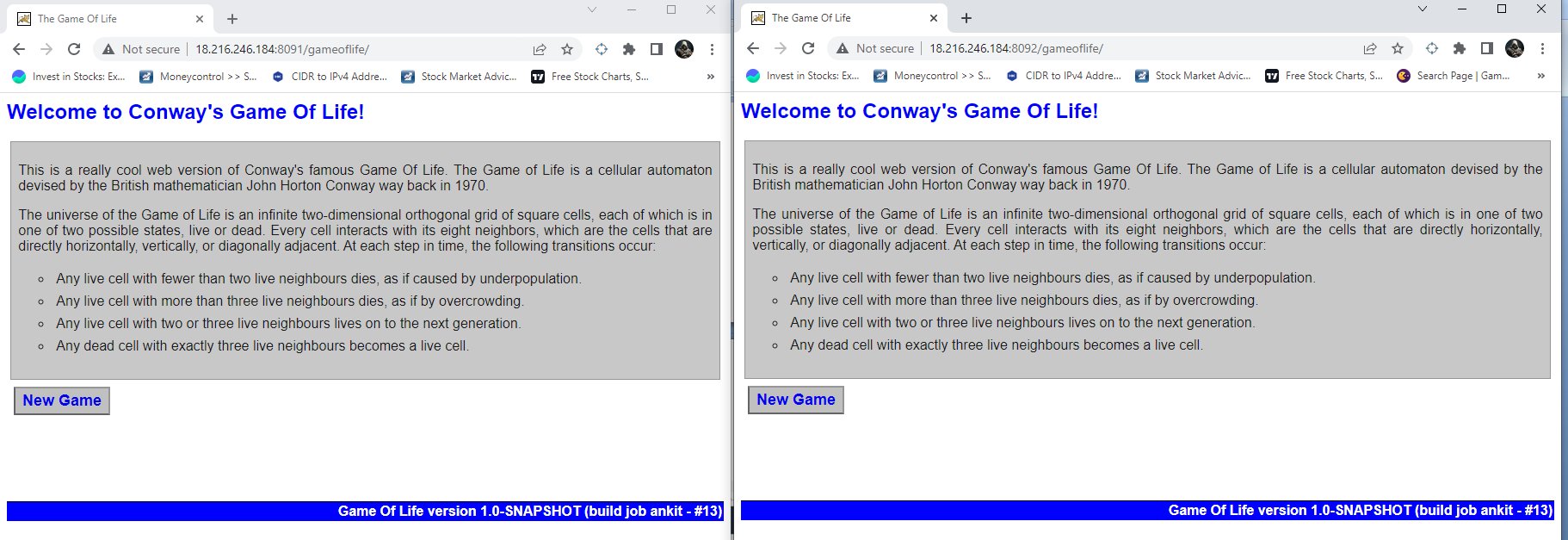
**DEV1 (OPEN THE PORTS YOU HAVE SPECIFIED ON DOCKERFILE AND CHECK )**

****

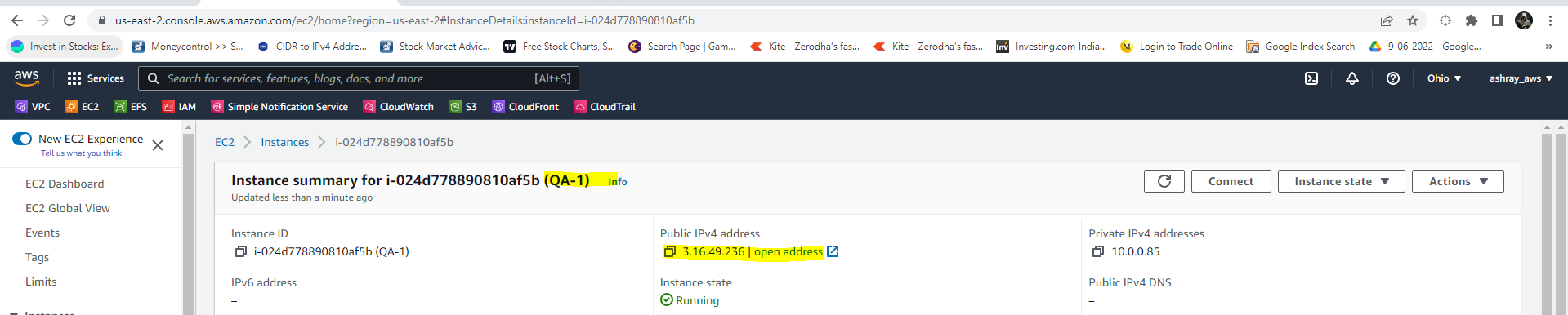
****

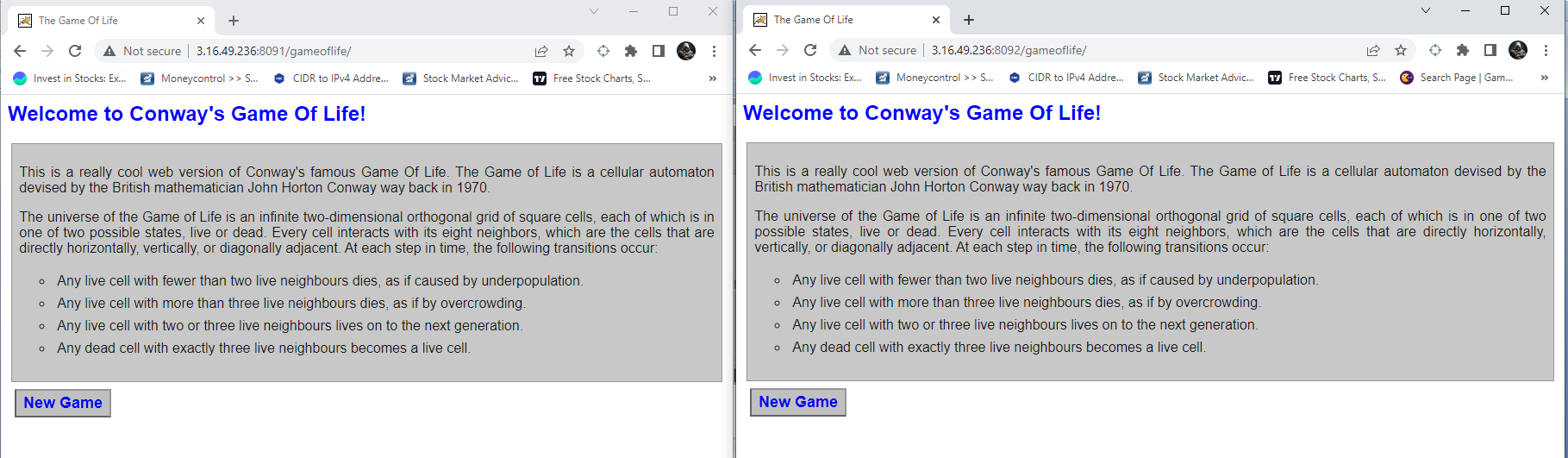
**DEV2:**

****

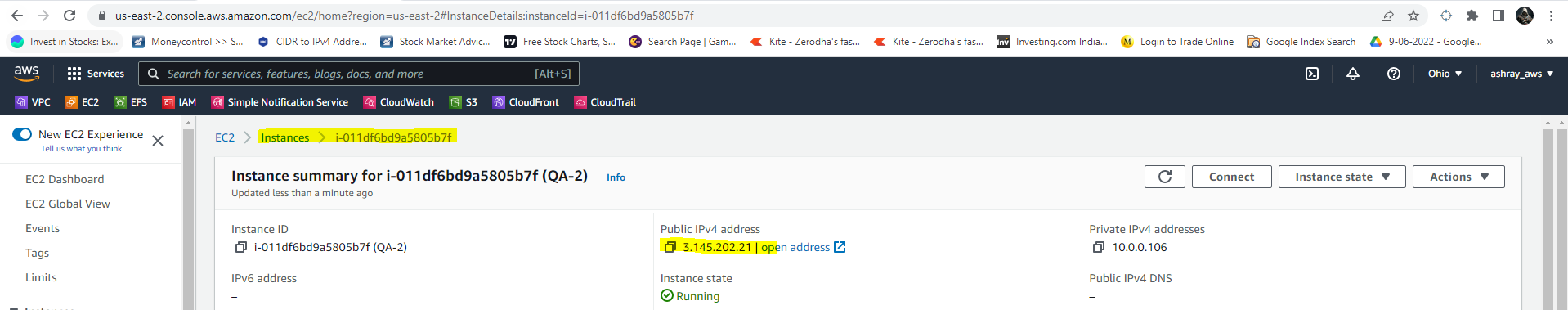
****

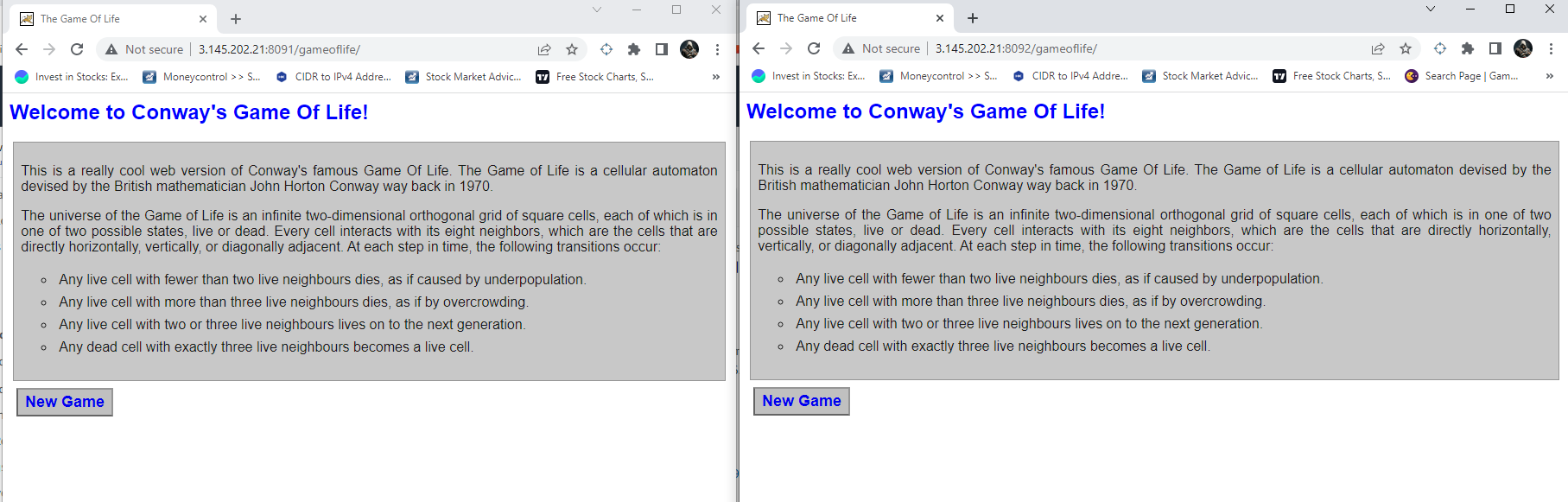
**QA1**

****

****

**QA2**

****

****