Task 1

Configured three EC2 instance using cloudformation and a yaml file

```yaml
Resources:
  Ec2OneSG:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: String
      GroupName: "Ec2OneSG"
      SecurityGroupIngress:
        - IpProtocol: TCP
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
        - IpProtocol: TCP
          FromPort: 8080
          ToPort: 8080
          CidrIp: 0.0.0.0/0
  Ec2TwoSG:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: "Security group for anywhere and port 3000"
      GroupName: "Ec2TwoSG"
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
        - IpProtocol: tcp
          FromPort: 3000
          ToPort: 3000
          CidrIp: 0.0.0.0/0
  Ec2ThreeSG:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupDescription: "Security group allowing ssh from anywhere"
      GroupName: "Ec2ThreeSG"
      SecurityGroupIngress:
        - IpProtocol: tcp
```

```
        FromPort: 22
        ToPort: 22
        CidrIp: 0.0.0.0/0
Ec2One:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-0629230e074c580f2
    InstanceType: t2.micro
    KeyName: deployment-08
    SecurityGroupIds:
      - !Ref Ec2OneSG
    Tags:
      - Key: "Name"
        Value: "Ec2One"
Ec2Two:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-0629230e074c580f2
    InstanceType: t2.micro
    KeyName: deployment-08
    SecurityGroupIds:
      - !Ref Ec2TwoSG
    Tags:
      - Key: "Name"
        Value: "Ec2Two"
Ec2Three:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-0629230e074c580f2
    InstanceType: t2.micro
    KeyName: deployment-08
    SecurityGroupIds:
      - !Ref Ec2ThreeSG
    Tags:
      - Key: "Name"
        Value: "Ec2Three"
```

Then ssh-keygen to make a key called deployment08 and deploument08 in the .ssh directory

- sudo scp -i deployment-08.pem deployment08.pub ubuntu@18.217.51.88:/home/ubuntu/.ssh

- sudo scp -i deployment-08.pem deployment08.pub ubuntu@13.59.10.97:/home/ubuntu/.ssh
- sudo scp -i deployment-08.pem deployment08.pub ubuntu@18.224.199.253:/home/ubuntu/.ssh

Also I copied my deployment08.pub into each of the three instance's authorized_key files.

After that I Used the command to make sure the hosts were correct
ansible all --list-hosts

Go inside authorized_key file of each of the three ec2, and copy paste the contents deployment08.pub of the file.

Lastly I configured my EC2s using ansible

Sudo nano Master-config.yaml

```
- name: ssh into Production and  installs java/docker
  hosts: Master
  become: yes
  gather_facts: True
  tasks:
  - name: update ec2
    shell: sudo apt-get update && sudo apt-get upgrade -y

  - name: installing java
    shell: sudo apt install openjdk-11-jre-headless -y

  - name: install openjdk11
    shell: sudo apt install openjdk-11-jdk -y

  - name: curl for jenkins
    shell: curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key
| sudo tee \ /usr/share/keyrings/jenkins-keyring.asc > /dev/null

  - name: echo variable
    shell: echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

  - name: upgrade && update
    shell: sudo apt-get update && sudo apt-get upgrade -y
```

```
  - name: installing jenkins
    shell: sudo apt-get install jenkins -y

  - name: installing git tool
    shell: sudo apt install git -y

  - name: checking if jenkins is active
    shell: sudo systemctl status jenkins | head -n 3
    register: command_output
  - debug:
        var: command_output.stdout_lines
```

Ran this command: ansible-playbook Master-config.yaml

Sudo nano agent-config.yml

```
- name: ssh into agent and install java, nodejs, npm
  hosts: Agent
  become: yes
  gather_facts: True
  tasks:
  - name: update ec2
    shell: sudo apt-get update && sudo apt-get upgrade -y

  - name: installing java
    shell: sudo apt install openjdk-11-jre-headless -y

  - name: download openjdk11
    shell: sudo apt install openjdk-11-jdk -y

  - name: installing nodejs
    shell: sudo apt install nodejs -y

  - name: installing npm
    shell: sudo apt install npm -y
```

Ran the yml using ansible-playbook agent-config.yaml

The last ec2 I configured using the command:
Sudo nano production-config.yml

```
- name: ssh into agent and install java and docker
  hosts: Production
  become: yes
  gather_facts: True
  tasks:
  - name: update ec2
    shell: sudo apt-get update && sudo apt-get upgrade -y

  - name: installing java
    shell: sudo apt install openjdk-11-jre-headless -y

  - name: install modules
    shell: sudo apt-get install \ ca-certificates \ curl \ gnupg \
lsb-release -y

  - name: curl command for long term release
    shell: curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

  - name: echo command for long term release
    shell: echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null

  - name: upgrade repo
    shell: sudo apt-get update && sudo apt-get upgrade -y

  - name: install jenkins
    shell: sudo apt-get install docker-ce docker-ce-cli containerd.io -y

  - name: starting docker
    shell: sudo systemctl start docker

  - name: docker status
    shell: sudo systemctl status docker
    register: command_output
  - debug:
```

```
    var: command_output.stdout_lines
```

Ran the yml using ansible-playbook production-config.yaml

Errors I had was my EC2 were not connecting as hosts. So I had to recreate the cloudformation file using a new key. ANd then it worked.