

**Day 15: In this task, I have used Github, Docker, Jenkins CI/CD and ansible to deploy a docker container on the target machine.**

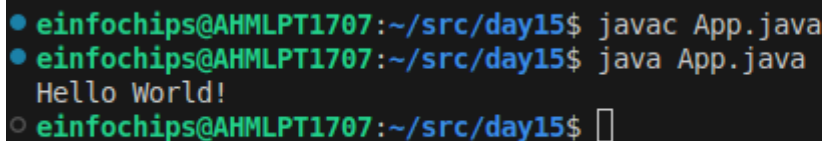
**First of all, I created a simple application in Java as App.java which is given below:**

```
package com.example;

public class App {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

**I tried running this app in local first using below commands, which was running successfully.**

```
javac App.java
java App.java
```

A terminal window with a dark background and light-colored text. It shows the execution of a Java application. The prompt is 'einfochips@AHMLPT1707:~/src/day15\$'. The first command is 'javac App.java'. The second command is 'java App.java', which outputs 'Hello World!'. The prompt then returns to 'einfochips@AHMLPT1707:~/src/day15\$' with a cursor.

```
einfochips@AHMLPT1707:~/src/day15$ javac App.java
einfochips@AHMLPT1707:~/src/day15$ java App.java
Hello World!
einfochips@AHMLPT1707:~/src/day15$
```

**Secondly, I created a Dockerfile which will create an image for this app.**

```
FROM openjdk:11-slim
WORKDIR /app
COPY . /app
RUN javac App.java
CMD ["java", "App"]
```

**Next, I created a Jenkinsfile which uses my GitHub repo to build the image from the Dockerfile, will push that docker image to my docker repository and make a container out of that image.**

```
pipeline {
    agent any

    environment {
        DOCKERHUB_CREDENTIALS = credentials('docker-cred')
        registry = 'docker.io'
        registryCredential = 'personal-docker'
    }

    stages {
        stage('Checkout SCM') {
```

```

    steps {
        git url: 'https://github.com/yashmahi88/day15_2.git', branch: 'master'
    }
}

stage('Build Docker Image') {
    steps {
        script {
            sh 'docker build -t yashmahi04/test15:latest .'
        }
    }
}

stage('Push Docker Image') {
    steps {
        script {
            withDockerRegistry([credentialsId: 'docker-cred', url: 'https://index.docker.io/v1/']) {
                sh 'docker push yashmahi04/test15:latest'
            }
        }
    }
}

stage('Deploy Container') {
    steps {
        script {
            sh 'docker run -d -p 8099:80 yashmahi04/test15:latest'
        }
    }
}

post {
    always {
        cleanWs()
    }
}
}

```



created an inventory.ini file on local host

```
[local]
localhost ansible_connection=local
```

writing deploy.yaml playbook

```
---
- name: Deploy Java Application
  hosts: local
  become: yes
  tasks:
    - name: Install Docker
      apt:
        name: docker.io
        state: present
        update_cache: yes

    - name: Ensure Docker service is started
      service:
        name: docker
        state: started
        enabled: yes

    - name: Pull Docker image from registry
      docker_image:
        name: yashmahi04/test15
        tag: latest
        source: pull

    - name: Run Docker container
      docker_container:
        name: java-app
        image: yashmahi04/test15:latest
        state: started
        restart_policy: always
        ports:
          - "8099:80"
```

Used the following command to execute the ansible file

```
ansible-playbook -i inventory.ini deploy.yaml --ask-become-pass
```

```
PLAY [Deploy Java Application] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host localhost is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more information.
ok: [localhost]
TASK [Install Docker] *****
ok: [localhost]
TASK [Ensure Docker service is started] *****
ok: [localhost]
TASK [Pull Docker image from registry] *****
ok: [localhost]
TASK [Run Docker container] *****
changed: [localhost]
PLAY RECAP *****
localhost : ok=5    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

einfochips@AHMLPT1707:~/src/day15$
```