QA FINAL PROJECT DOCUMENTATION

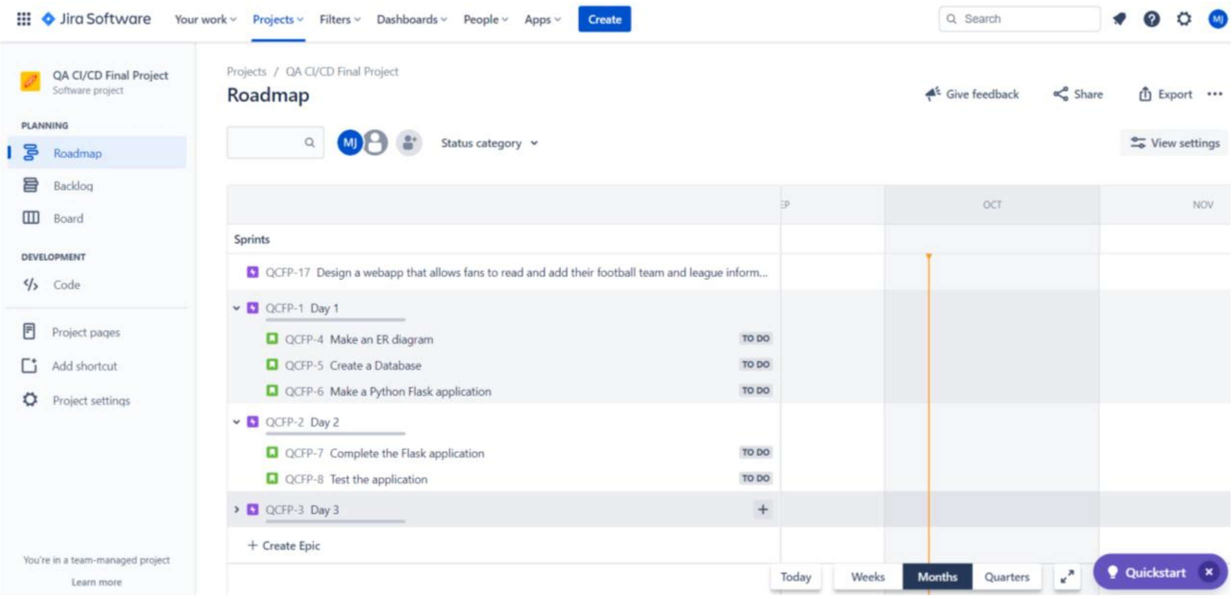
Overview:

I have created a simple webapp that the user can access, read, update the teams in the European League. I used the Flask framework with SQLite. I then uploaded the code onto Github and used Jenkins to automatically build and deploy a container that runs my webapp. I had many complications to get this working and its still not fully complete. I will document the process till now.

Tools used: Python using Flask with SQLite, HTML, Azure VM, Jira, Github, Docker, Jenkins

Jira and ER Diagram:

First the ER diagram and making an outline of the project on Jira:



Jira Software · Your work · Projects · Filters · Dashboards · People · Apps · Create · Search

QA CI/CD Final Project
Software project

PLANNING
- Roadmap
- Backlog
- Board

DEVELOPMENT
- Code

- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
Learn more

Projects / QA CI/CD Final Project

Roadmap

Give feedback · Share · Export · View settings · Status category

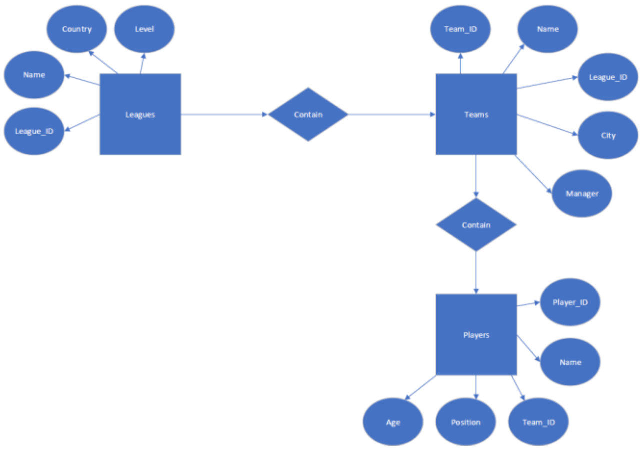
EP · OCT · NOV

Sprints

- QCFP-17 Design a webapp that allows fans to read and add their football team and league inform...
- QCFP-1 Day 1
  - QCFP-4 Make an ER diagram — TO DO
  - QCFP-5 Create a Database — TO DO
  - QCFP-6 Make a Python Flask application — TO DO
- QCFP-2 Day 2
  - QCFP-7 Complete the Flask application — TO DO
  - QCFP-8 Test the application — TO DO
- QCFP-3 Day 3

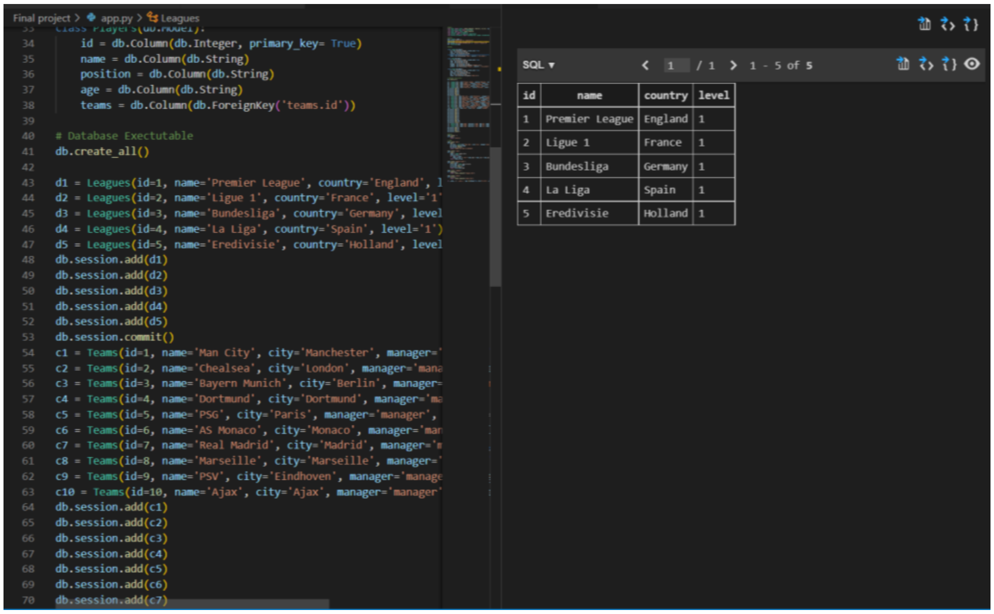
+ Create Epic

Today · Weeks · Months · Quarters · Quickstart



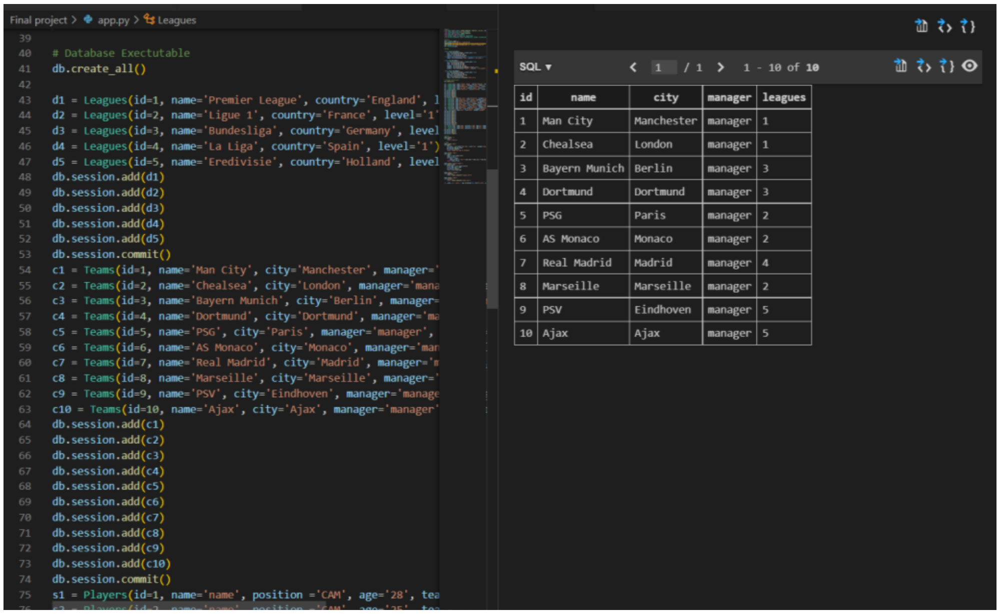
Database:

I created the database on VS code using Flask and SQLite. This database follows the same relationships as the ER diagram. One league many teams. One team many players. I also used a simple add and update to add and update teams in the database.



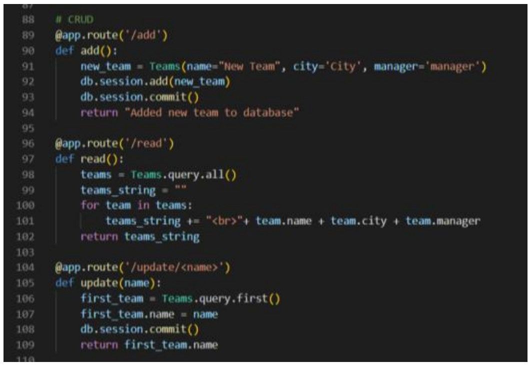
```
class Players(db.Model):
    id = db.Column(db.Integer, primary_key= True)
    name = db.Column(db.String)
    position = db.Column(db.String)
    age = db.Column(db.String)
    teams = db.Column(db.ForeignKey('teams.id'))

# Database Exectutable
db.create_all()

d1 = Leagues(id=1, name='Premier League', country='England', l
d2 = Leagues(id=2, name='Ligue 1', country='France', level='1'
d3 = Leagues(id=3, name='Bundesliga', country='Germany', level
d4 = Leagues(id=4, name='La Liga', country='Spain', level='1')
d5 = Leagues(id=5, name='Eredivisie', country='Holland', level
db.session.add(d1)
db.session.add(d2)
db.session.add(d3)
db.session.add(d4)
db.session.add(d5)
db.session.commit()
c1 = Teams(id=1, name='Man City', city='Manchester', manager='
c2 = Teams(id=2, name='Chealsea', city='London', manager='mana
c3 = Teams(id=3, name='Bayern Munich', city='Berlin', manager=
c4 = Teams(id=4, name='Dortmund', city='Dortmund', manager='ma
c5 = Teams(id=5, name='PSG', city='Paris', manager='manager',
c6 = Teams(id=6, name='AS Monaco', city='Monaco', manager='man
c7 = Teams(id=7, name='Real Madrid', city='Madrid', manager='m
c8 = Teams(id=8, name='Marseille', city='Marseille', manager='
c9 = Teams(id=9, name='PSV', city='Eindhoven', manager='manage
c10 = Teams(id=10, name='Ajax', city='Ajax', manager='manager'
db.session.add(c1)
db.session.add(c2)
db.session.add(c3)
db.session.add(c4)
db.session.add(c5)
db.session.add(c6)
db.session.add(c7)
```

id	name	country	level
1	Premier League	England	1
2	Ligue 1	France	1
3	Bundesliga	Germany	1
4	La Liga	Spain	1
5	Eredivisie	Holland	1



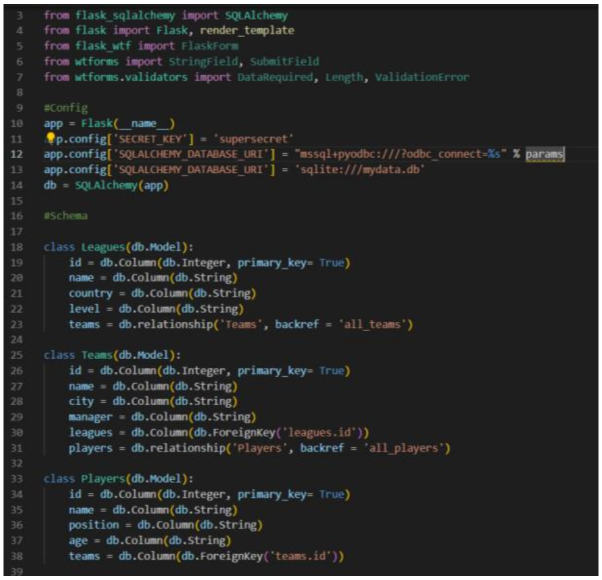
```
# Database Exectutable
db.create_all()

d1 = Leagues(id=1, name='Premier League', country='England', l
d2 = Leagues(id=2, name='Ligue 1', country='France', level='1'
d3 = Leagues(id=3, name='Bundesliga', country='Germany', level
d4 = Leagues(id=4, name='La Liga', country='Spain', level='1')
d5 = Leagues(id=5, name='Eredivisie', country='Holland', level
db.session.add(d1)
db.session.add(d2)
db.session.add(d3)
db.session.add(d4)
db.session.add(d5)
db.session.commit()
c1 = Teams(id=1, name='Man City', city='Manchester', manager='
c2 = Teams(id=2, name='Chealsea', city='London', manager='mana
c3 = Teams(id=3, name='Bayern Munich', city='Berlin', manager=
c4 = Teams(id=4, name='Dortmund', city='Dortmund', manager='ma
c5 = Teams(id=5, name='PSG', city='Paris', manager='manager',
c6 = Teams(id=6, name='AS Monaco', city='Monaco', manager='man
c7 = Teams(id=7, name='Real Madrid', city='Madrid', manager='m
c8 = Teams(id=8, name='Marseille', city='Marseille', manager='
c9 = Teams(id=9, name='PSV', city='Eindhoven', manager='manage
c10 = Teams(id=10, name='Ajax', city='Ajax', manager='manager'
db.session.add(c1)
db.session.add(c2)
db.session.add(c3)
db.session.add(c4)
db.session.add(c5)
db.session.add(c6)
db.session.add(c7)
db.session.add(c8)
db.session.add(c9)
db.session.add(c10)
db.session.commit()
s1 = Players(id=1, name='name', position ='CAM', age='28', tea
s2 = Players(id=2, name='name', position ='CAM', age='25', tea
```

id	name	city	manager	leagues
1	Man City	Manchester	manager	1
2	Chealsea	London	manager	1
3	Bayern Munich	Berlin	manager	3
4	Dortmund	Dortmund	manager	3
5	PSG	Paris	manager	2
6	AS Monaco	Monaco	manager	2
7	Real Madrid	Madrid	manager	4
8	Marseille	Marseille	manager	2
9	PSV	Eindhoven	manager	5
10	Ajax	Ajax	manager	5



```
# CRUD
@app.route('/add')
def add():
    new_team = Teams(name="New Team", city='City', manager='manager')
    db.session.add(new_team)
    db.session.commit()
    return "Added new team to database"

@app.route('/read')
def read():
    teams = Teams.query.all()
    teams_string = ""
    for team in teams:
        teams_string += "<br>"+ team.name + team.city + team.manager
    return teams_string

@app.route('/update/<name>')
def update(name):
    first_team = Teams.query.first()
    first_team.name = name
    db.session.commit()
    return first_team.name
```

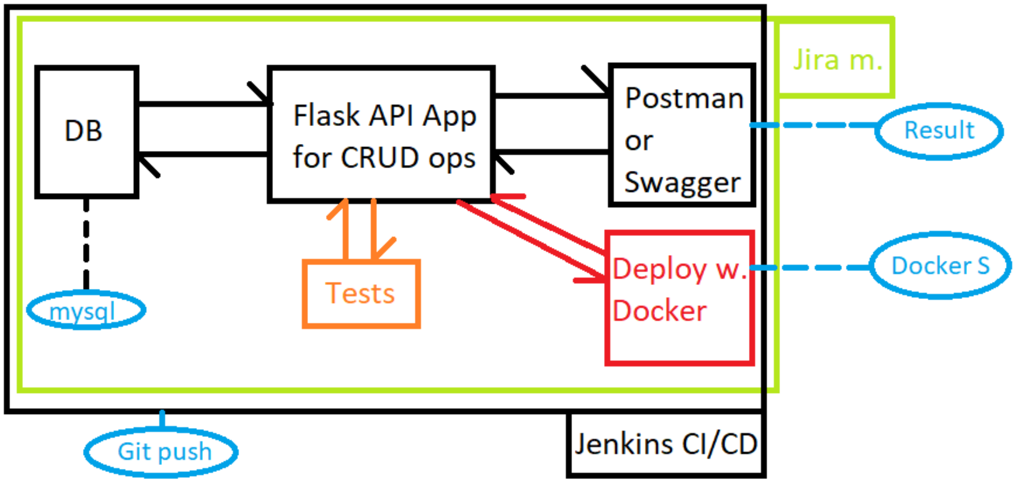
I uploaded the files on Github so I can access it via my VM and Jenkins pipeline. I also tried to connect the DB to AzureSQL but I couldn’t manage it as it returned more errors.



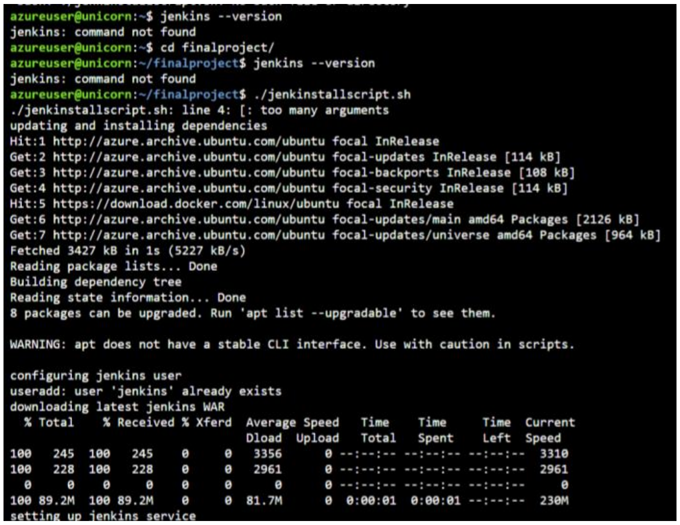
```
from flask_sqlalchemy import SQLAlchemy
from flask import Flask, render_template
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import DataRequired, Length, ValidationError

#Config
app = Flask(__name__)
💡p.config['SECRET_KEY'] = 'supersecret'
app.config['SQLALCHEMY_DATABASE_URI'] = "mssql+pyodbc:///?odbc_connect=%s" % params
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///mydata.db'
db = SQLAlchemy(app)

#Schema

class Leagues(db.Model):
    id = db.Column(db.Integer, primary_key= True)
    name = db.Column(db.String)
    country = db.Column(db.String)
    level = db.Column(db.String)
    teams = db.relationship('Teams', backref = 'all_teams')

class Teams(db.Model):
    id = db.Column(db.Integer, primary_key= True)
    name = db.Column(db.String)
    city = db.Column(db.String)
    manager = db.Column(db.String)
    leagues = db.Column(db.ForeignKey('leagues.id'))
    players = db.relationship('Players', backref = 'all_players')

class Players(db.Model):
    id = db.Column(db.Integer, primary_key= True)
    name = db.Column(db.String)
    position = db.Column(db.String)
    age = db.Column(db.String)
    teams = db.Column(db.ForeignKey('teams.id'))
```

CI/CD Pipeline:

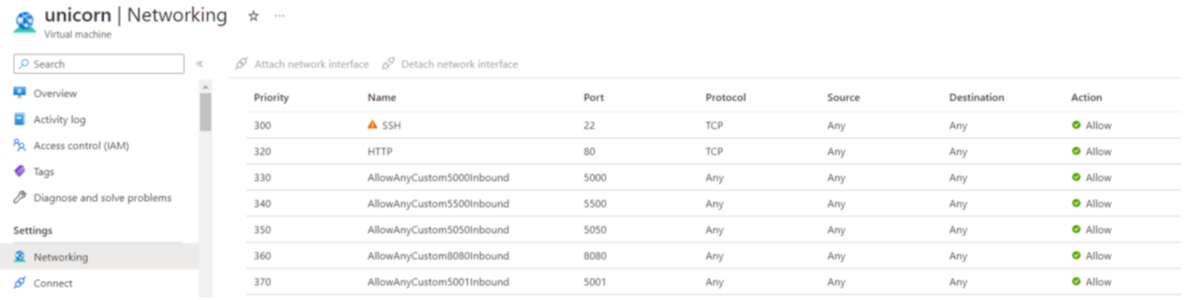
This is a diagram of what I am trying to do in the Jenkins CI/CD pipeline. Its going to have 3 stages. A build stage, deployment stage and an admin approval stage.



I then installed Jenkins on my Linux VM running on Azure and opened up the port 8080 for Jenkins and port 5000, 5001 for the containerized webapp.

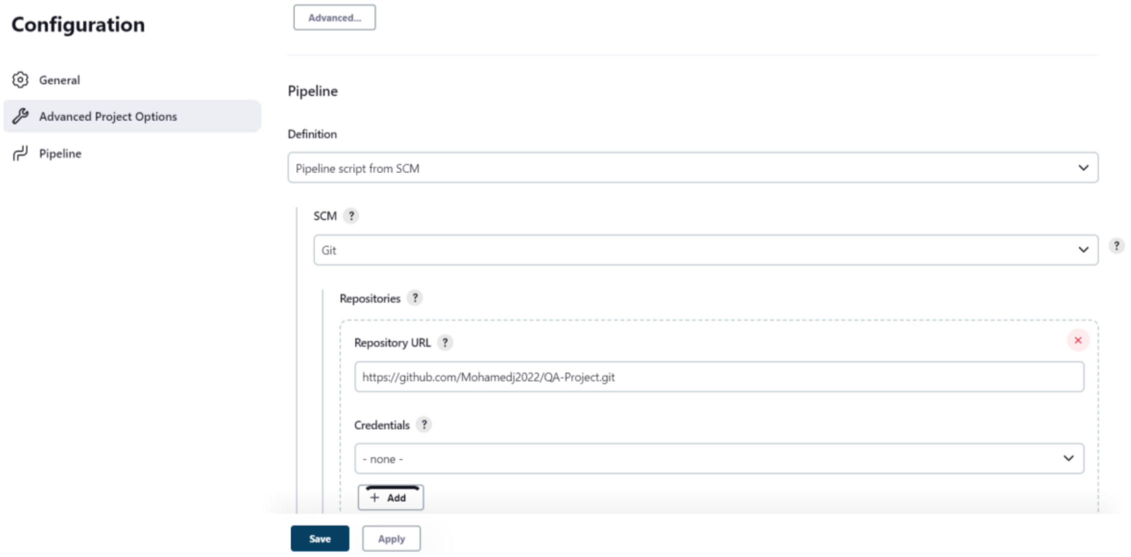


```
azureuser@unicorn:~$ jenkins --version
jenkins: command not found
azureuser@unicorn:~$ cd finalproject/
azureuser@unicorn:~/finalproject$ jenkins --version
jenkins: command not found
azureuser@unicorn:~/finalproject$ ./jenkinstallscript.sh
./jenkinstallscript.sh: line 4: [: too many arguments
updating and installing dependencies
Hit:1 http://azure.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:5 https://download.docker.com/linux/ubuntu focal InRelease
Get:6 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2126 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [964 kB]
Fetched 3427 kB in 1s (5227 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
8 packages can be upgraded. Run 'apt list --upgradable' to see them.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

configuring jenkins user
useradd: user 'jenkins' already exists
downloading latest jenkins WAR
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   245  100   245    0     0   3356      0 --:--:-- --:--:-- --:--:--  3310
100   228  100   228    0     0   2961      0 --:--:-- --:--:-- --:--:--  2961
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 89.2M  100 89.2M    0     0  81.7M      0  0:00:01  0:00:01 --:--:--  230M
setting up jenkins service
```



Priority	Name	Port	Protocol	Source	Destination	Action
300	⚠ SSH	22	TCP	Any	Any	✓ Allow
320	HTTP	80	TCP	Any	Any	✓ Allow
330	AllowAnyCustom5000Inbound	5000	Any	Any	Any	✓ Allow
340	AllowAnyCustom5500Inbound	5500	Any	Any	Any	✓ Allow
350	AllowAnyCustom5050Inbound	5050	Any	Any	Any	✓ Allow
360	AllowAnyCustom8080Inbound	8080	Any	Any	Any	✓ Allow
370	AllowAnyCustom5001Inbound	5001	Any	Any	Any	✓ Allow

I set up a pipeline using Jenkins and used it to link to my Github.



Pipeline

Definition: Pipeline script from SCM

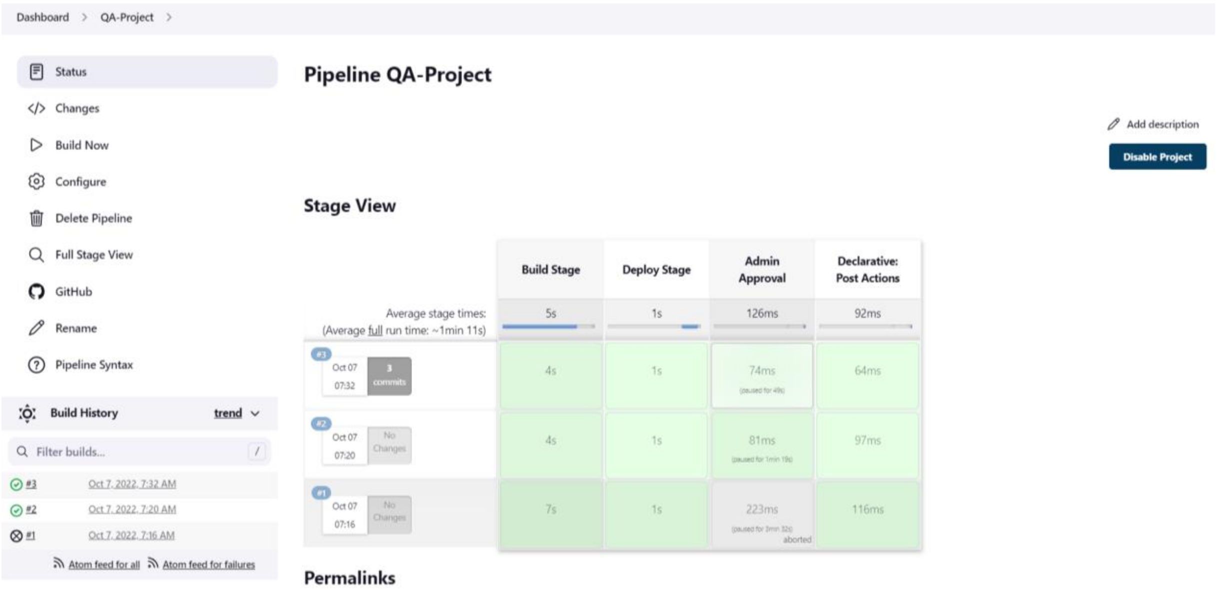
SCM: Git

Repositories

Repository URL: https://github.com/Mohamedj2022/QA-Project.git

Credentials: - none -

I used Jenkins to deploy the containerised webapp and it worked.



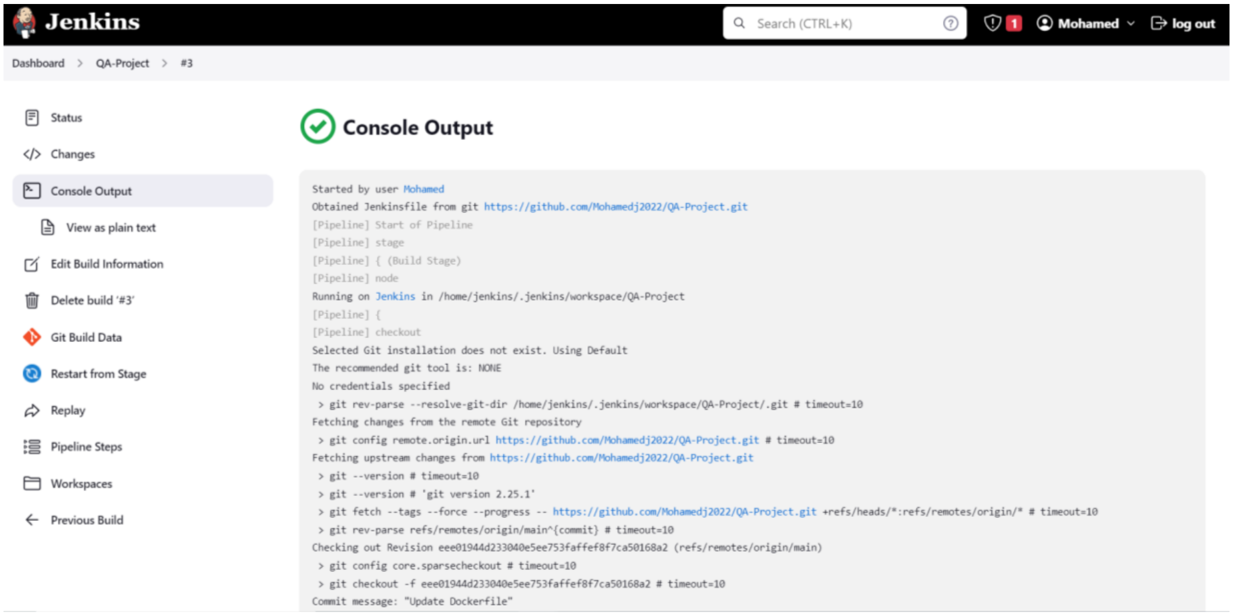
#3 Oct 7, 2022, 7:32 AM
#2 Oct 7, 2022, 7:20 AM
#1 Oct 7, 2022, 7:16 AM

Pipeline QA-Project

Stage View

	Build Stage	Deploy Stage	Admin Approval	Declarative: Post Actions
Average stage times: (Average full run time: ~1min 11s)	5s	1s	126ms	92ms
#3 Oct 07 07:32 3 commits	4s	1s	74ms (paused for 49s)	64ms
#2 Oct 07 07:20 No Changes	4s	1s	81ms (paused for 1min 19s)	97ms
#1 Oct 07 07:16 No Changes	7s	1s	223ms (paused for 3min 32s) aborted	116ms

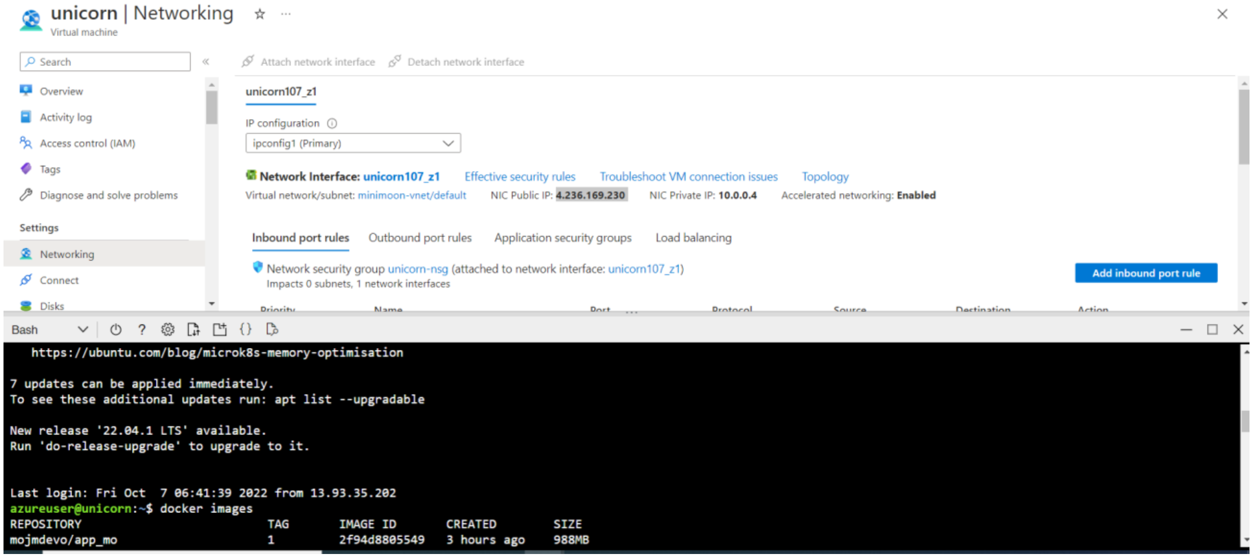
Permalinks



Console Output

```
Started by user Mohamed
Obtained Jenkinsfile from git https://github.com/Mohamedj2022/QA-Project.git
[Pipeline] Start of Pipeline
[Pipeline] stage
[Pipeline] { (Build Stage)
[Pipeline] node
Running on Jenkins in /home/jenkins/.jenkins/workspace/QA-Project
[Pipeline] {
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /home/jenkins/.jenkins/workspace/QA-Project/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Mohamedj2022/QA-Project.git # timeout=10
Fetching upstream changes from https://github.com/Mohamedj2022/QA-Project.git
 > git --version # timeout=10
 > git --version # 'git version 2.25.1'
 > git fetch --tags --force --progress -- https://github.com/Mohamedj2022/QA-Project.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision eee01944d233040e5ee753faffef8f7ca50168a2 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f eee01944d233040e5ee753faffef8f7ca50168a2 # timeout=10
Commit message: "Update Dockerfile"
```

I checked to see if my container deployed on my VM and then pushed it onto my Dockerhub.



unicorn107_z1

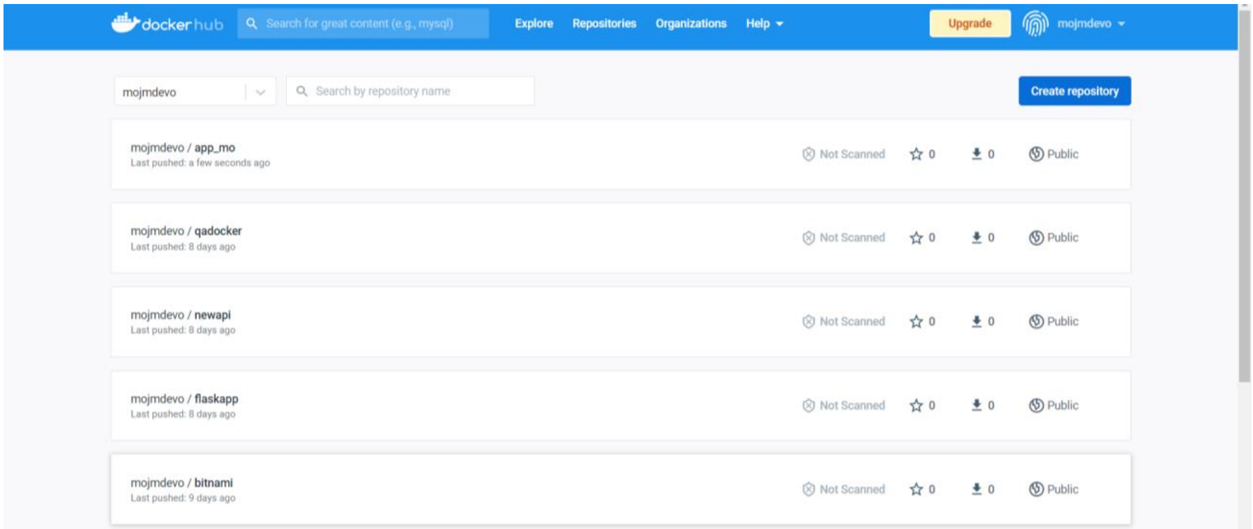
IP configuration: ipconfig1 (Primary)

Network Interface: unicorn107_z1 Effective security rules Troubleshoot VM connection issues Topology

Virtual network/subnet: minimoon-vnet/default NIC Public IP: 4.236.169.230 NIC Private IP: 10.0.0.4 Accelerated networking: Enabled

Network security group unicorn-nsg (attached to network interface: unicorn107_z1)
Impacts 0 subnets, 1 network interfaces

```
   https://ubuntu.com/blog/microk8s-memory-optimisation

7 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Oct  7 06:41:39 2022 from 13.93.35.202
azureuser@unicorn:~$ docker images
REPOSITORY                         TAG       IMAGE ID       CREATED       SIZE
mojmdevo/app_mo                    1         2f94d8805549   3 hours ago   988MB
```



mojmdevo / app_mo
Last pushed: a few seconds ago
Not Scanned ☆ 0 ⬇ 0 Public

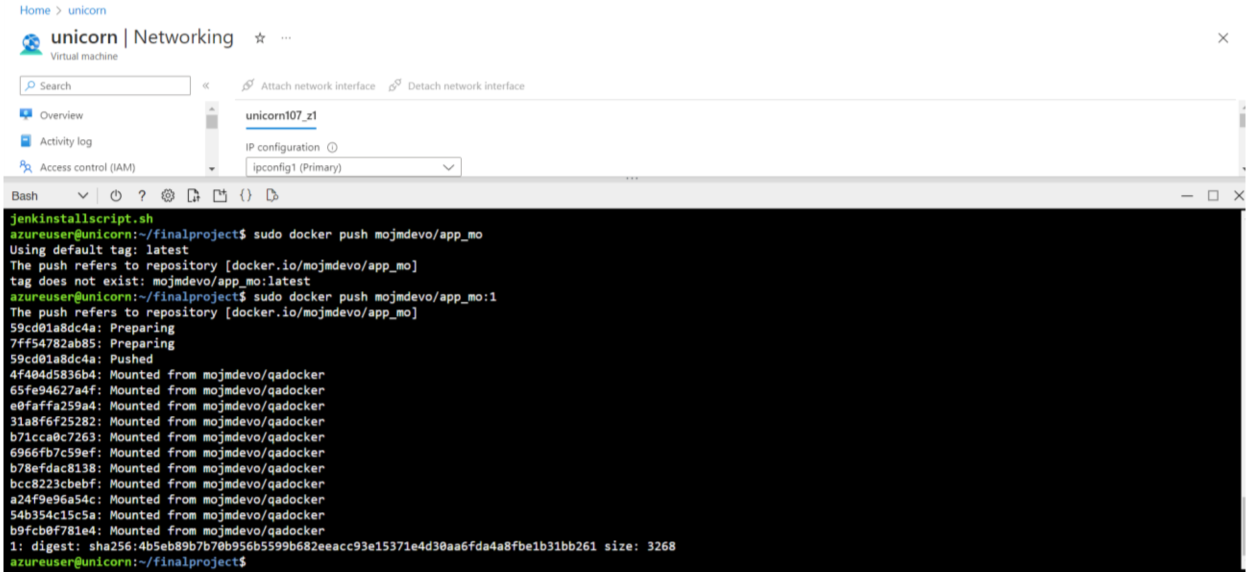
mojmdevo / qadocker
Last pushed: 8 days ago
Not Scanned ☆ 0 ⬇ 0 Public

mojmdevo / newapi
Last pushed: 8 days ago
Not Scanned ☆ 0 ⬇ 0 Public

mojmdevo / flaskapp
Last pushed: 8 days ago
Not Scanned ☆ 0 ⬇ 0 Public

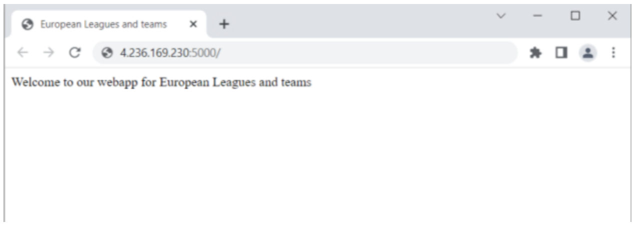
mojmdevo / bitnami
Last pushed: 9 days ago
Not Scanned ☆ 0 ⬇ 0 Public

I used the docker push command to get my image into Dockerhub



```
jenkinstallscript.sh
azureuser@unicorn:~/finalproject$ sudo docker push mojmdevo/app_mo
Using default tag: latest
The push refers to repository [docker.io/mojmdevo/app_mo]
tag does not exist: mojmdevo/app_mo:latest
azureuser@unicorn:~/finalproject$ sudo docker push mojmdevo/app_mo:1
The push refers to repository [docker.io/mojmdevo/app_mo]
59cd01a8dc4a: Preparing
7ff54782ab85: Preparing
59cd01a8dc4a: Pushed
4f404d5836b4: Mounted from mojmdevo/qadocker
65fe94627a4f: Mounted from mojmdevo/qadocker
e0faffa259a4: Mounted from mojmdevo/qadocker
31a8f6f25282: Mounted from mojmdevo/qadocker
b71cca0c7263: Mounted from mojmdevo/qadocker
6966fb7c59ef: Mounted from mojmdevo/qadocker
b78efdac8138: Mounted from mojmdevo/qadocker
bcc8223cbebf: Mounted from mojmdevo/qadocker
a24f9e96a54c: Mounted from mojmdevo/qadocker
54b354c15c5a: Mounted from mojmdevo/qadocker
b9fcb0f781e4: Mounted from mojmdevo/qadocker
1: digest: sha256:4b5eb89b7b70b956b5599b682eeacc93e15371e4d30aa6fda4a8fbe1b31bb261 size: 3268
azureuser@unicorn:~/finalproject$
```

Simplified working webapp deployed by Jenkins pipeline



Welcome to our webapp for European Leagues and teams

Future updates:

1. Get the DB into the AzureSQL database and interact with the webapp
2. Use better user input in the Flask application
3. Add a testing stage using pytest or Postman before it goes to the Admin Approval stage