Problem Statement:

We are creating a Database of all the Employees of Organisation:

Data/Fields - Name, Department

Create the logic - Java Application that can support a Form for Creating/ Updating/Deleting an Employee

1. Requirement Gathering
    1. Employee DB - Name, Department, ID(unique identifier)
    2. Create an Employee - Name(string), Department(String)
        1. Name is string - Error - Please

type a valid name
    2. Department is String - Sales, Marketing, Accounts, IT
        1. Or number - 1(Sales), 2(Marketing), 3(Account) or 4(IT)
    3. Update an employee - ID to find what I want to update
        1. Id must exist - Please type a valid ID
    4. Delete an employee
2. Schema for my application - Creating a basic E-R Diagram - Workflow -
3. Database - Creating one basic record -
4. Domain - Entity Class - Version Control
5. Service Layer, Controller Layer....

Login -

Authentication - Check if the user exists - Response back with role

User - id, username/email/phone, password(Hash value)

Authorisation - Based on the role we will give the access

Sales -

Marketing -

Admin - All visible

Encryption is reversible but not Hashing - Securest way to handle password

Password - Critical info - Create a Hash Value and save to my DB - Sign up

Authenticate - password from form -hash value - I will compare DB password

Status, creationDate, CreatedBy - Save this info with every step

Create a record -

Library - OAuth

Azure Active Directory App Registration

GetId - id= if match found - {
“name”:””
“Id”
“Age”
}

id=99 out of 15

Exception Handling - UI or create an API request - We will customise our error as per our need

Application - Developer made the app live
Any changes made are verified by Testing Team and Client/Organization

UAT/Testing Environment - Changes verified by org/client
Production Environment -

Testing of the application:

BEFORE THE TESTING:
1. As a developer, I will myself test my code:
    Check the functionality is working
2. Unit Testing -

test(int a, int b). ==> a+b

* a is number and b is string - 2*pink

* b is number and a is string - 2*pink

Create a function - Test all these scenarios

Mockito/JUnit

Unit Testing -

function. - input- a and b

output a+b

1. A is number , b in string - (5 + red)
2. A is string ,b is number

3. A and b are string/social character/

Test Cases - Executed - JUnit/Mockito

Test coverage - whole app - 100 files - 20

15%

Sign In Page- QA, Social Media, E-Commerce

emailAddress -

1. If pattern for email address is honoured
2. If the value is null
3. Special Character check

Password -

1. Cannot be null

Validations:

Integer

1. Min
2. Max

String

1. Size(min, max)

SUBMIT button

Check if email and password matches

Dependency Injection - Spring

Loosely Coupled -

Database - MySQL

application.properties
.java file - Connection string - username, password and url
If any change is made- change all my files having a DB connection

INVERSION OF CONTROL

```
class Employee{
Address address;
Employee(Address address){
    this.address = address
}
}
```

```
Public void setAddress(Address address){
    this.address = address
}
```

1. Constructor method
2. Setter

Components:
@Component

@Service - store class with business logic - Service Layer

@Repository - CRUD Operation - Handling Data Access Objects

Database Operations

@Controller - Indicate class as a Controller - Request/Response

API - Rest/Soap

@getMapping
@PostMapping

Application: Social Media, E-Commerce Website, IoT Application

1. Requirement Gathering
    1. I will note down all features and functionalities - SRS/Functional Document
2. Design the architecture - Language, Frameworks, SQL/NoSQL
3. Creating theUI for application - Images/pages for all screens

4. Development of the application -
    1. HTML/CSS/JavaScript - FrontEnd
    2. Java/Python/C# - Logic resides - Backend
5. Unit Testing - Test changes myself
6. Testing team reviews the changes made - Continuous Integration
7. Deployment - Deployed to Live Server - Continuous Deployment

Automation - Testing and Deployment

1. I will clone code from GitHub
2. I will install python SDK, Flask
3. Run the app and verify the change
4. I will deploy the code from GitHub to my production/live server

Script - git clone https://github.com/deepanshu/qa

sudo apt update

sudo apt install python3

Agents: Compute used for Automation -

4 steps execution

Self Hosted - I will do all 4 steps in a compute/system
MS Hosted - Microsoft Azure will complete 4 steps

Speed is increased - Multiple parallel jobs

No coding required - I signup for azure DevOps, create a project and configure
    SaaS

Azure Pass -
SkillPipe - Az-400 or Azure DevOps

CI and CD with out Windows Application

CI-CD - Azure DevOps

LUNCH BREAK

Session Start Time - 01:55 PM

Waiting for audience

Overview to Active Directory:

1. Create a user and verify by signing in
    1. Search for Azure Active Directory and click on it
    2. Create a new user by clicking New User > Create New User
    3. Please give the username and copy

the generated password and click create
4. Please open a private window and open portal.azure.com to sign with new user
5. First time I will be reseting the password for security

2. Share the access to virtual machine to user and verify
    1. Open azure portal from your our email on regular window
    2. Search virtual machine and select the linux vm
    3. Select Access Control (IAM) side menu
    4. Click add and then Add role Assignment
    5. Choose role and click on next
    6. Choose member and click Review+ Assign
    7. Review info and click Review+ Assign
    8. Review the virtual machine from New user private window

9. Kindly sign-out and sign in again if required

3. Share the whole resource group and verify
    1. Open azure portal from your email on regular window
    2. Search Resource Group and select the linux vm RG
    3. Select Access Control (IAM) side menu
    4. Click add and then Add role Assignment
    5. Choose role and click on next
    6. Choose member and click Review+ Assign
    7. Review info and click Review+ Assign
    8. Review the Resource Group from New user private window
    9. Kindly sign-out and sign in again if required

deep@anshu0105gmail.onmicrosoft.com
Vafu2191

2 browser window
    Private window - New user signed in
    Regular window - We are signed in

Users:
    1. New User - I create the user with my domain
    2. External User - a user already on Azure, we create a guest access

Domains -
    Organisation - Amazon, Apple
    Employee will have an email - deep@amazon.com, john@apple.com
Domains - Buy domain from GoDaddy, BigRock

Please complete **Jenkins Builds and Plugins**

Available on QA Community

CI and CD/CD

QA Community - Brief overview DevOps
- CI/CD Basics and Intermediate
    Only Jenkins

DevOps - CI/CD

Azure DevOps - Windows

CI-CD from Azure DevOps

Thank you for joining :)

Jenkins Job - Collection of steps - run one after another

Build my project
Create a text file - enter logs in it
Combining multiple files

Job - Project -

vi script.sh

Paste code from QA Community

Esc from keyboard

:wq (Save file and exit)

touch script.sh

nano script.sh

Permissions:

chmod 500 <key.pem>

./script.sh

Please checkout Infrastructure

Consistency Exercise

Available on QA Community

Lab 00- Setup: https://github.com/MicrosoftLearning/AZ400-DesigningandImplementingMicrosoftDevOpsSolutions/blob/master/Instructions/Labs/AZ400_M00_Validate_lab_environment.md

Lab 01- Azure DevOps Demo Generator: https://github.com/MicrosoftLearning/AZ400-DesigningandImplementingMicrosoftDevOpsSolutions/blob/master/Instructions/Labs/AZ400_M01_L01_Agile_Planning_and_Portfolio_Management_with_Azure_Boards.md

Sign In page:

Epic - Basic description of what I want to create - Login

Sign In

- Task for development with image containing - Team member

- Task for testing - Team member

Register

Forgot Password - Blank page - Bug

Reset Password- password@123 - Password@123

Not working or needs a fix - Bug/ Issue

Add to cart function -

Checkout - we will check for quantity and amount

BRANCH - cart branch

Any changes made to cart branch will not be reflected on any other branch/ main branch

Week1 -10 tasks - Roll out - Sprint - 1

week long

7

3 backlog

Sign In Page -

Username-

Password -

Please complete lab 00 and 01 Azure DevOps

Link in chat window

Thank you for joining :)

Script file and Executing the file

Installation of Jenkins

Creating a new file

Creating a zip

Virtual Machine - Azure

IaC - Create a Virtual Machine with my code

E-Commerce Application - 1 Virtual Machine

Easily handles 1000 concurrent users

50 GB Ram

Promotion - Users may increase

50,000 concurrent users

Scale my VM 50 times - 2500 GB RAM

LOAD

BALANCER

VM VM VM
VM VM VM
VM VM

Complex Architecture for VM can be managed by Load Balancer

1. Create VM from Azure Portal : 5-10 mins
2. Create VM from Azure Portal -
3. Create VM from Azure Portal -
4. Create VM from Azure Portal -
5. Create VM from Azure Portal -
6. Create VM from Azure Portal -
7. Create VM from Azure Portal -
8. Create VM from Azure Portal -
9. Create VM from Azure Portal -
10. Create VM from Azure Portal -
11. Create LOAD BALANCER from Azure Portal -

Orchestration and Management of

Complex architecture with Multiple containers in parallel - Kubernetes Service

AKS - Implementation by azure

CT CT CT CT CT
CT . . . . .

50-100 mins

SCRIPT that can create VM -

Single Sign-On:
I can signing at multiple places with same credentials
Sign to Gmail

Sign In to Google Engine

Sign in for outlook
Sign in to TeaMS
SIGN IN TO dynamics365

Plugins - Azure Active Directory
IAM - Role based access control

Continuous Integration and Continuous Deployment

All the modules/ QA Community Modules
Jenkins Installation
Linux - Ubuntu :
Azure Portal - Virtual Machine - Subscription?
Logging into VM -

CI-CD : Implementation/Application - DevOps

Windows - CI-CD - Azure DevOps

Create/Manage/Delete resources together - RG
    Application - Delete - VM and DB -

Multiple Domains - AWS , Amazon.in

4 applications -
    2 websites
    1 Mobile App
    1 Web Application

10+ resources - 1 Application
40+ resources -

One website and one sub domain -

Azure DevOps:

Azure Active Directory
Azure DevOps Organisation then creating a project

Application: Social Media, E-Commerce Website, IoT Application

1. Requirement Gathering
    1. I will note down all features and functionalities - SRS/Functional Document
2. Design the architecture - Language, Frameworks, SQL/NoSQL
3. Creating theUI for application - Images/pages for all screens
4. Development of the application -
    1. HTML/CSS/JavaScript - FrontEnd
    2. Java/Python/C# - Logic resides - Backend
5. Unit Testing - Test changes myself
6. Testing team reviews the changes made - Continuous Integration
7. Deployment - Deployed to Live Server

- Continuous Deployment

Pipeline - DevOps, Machine Learning, Data Engineering

Automating my tasks:
Taking a clone from the GitHub repo
Verify all functionality/code change
Deploy to production or live environment

MS Hosted Agents - Parallel Jobs-
Self Hosted Agents -

Function - a and b
output - a+b

Test cases:
1. A is number but b is red - red + 4
2. B is number and a is special

character

All my development will happen in local/ my own system
Testing - Unit testing
Testing team

Client - Approval

An application is working in my system - Next step is to make it available for all

Buy my servers and deploy the app - Public/private application

Deploy on Cloud Service Provider - VM / App Service (Pass)

E-Commerce Website -

Website for selling products

Portal for updating changes to website - Adding products, promotions , etc

Access, Security and Automation -

Local Laptop - Everything works

Devlopment server - VM for checking changes myself

Testing Server - Testing team will verify the changes - Another VM

Client - UAT - User Acceptance Testing - Another VM

Deployment Server - Live/Production Server

Environments - Application

Containers - Light weight PaaS
resource - App Service
1 vm where everything is deployed- Python SDK, SQL Server - App Running
Container image

App Service - Containers/code

IaaS, PaaS, SaaS

Please complete the tutorial for CI-CD Credentials
and check-out Pipeline Snippet Generator

Available on QA Community

Thank you for joining :)

IaaS vs PaaS and SaaS

Virtual Machine is IaaS
    Runtime or virtual environment

PaaS - App Service - Web App
    Lift and shift of my application - Lowest downtime

Php, python, java, .net

E-Commerce Website:

Products, payment
Application - products

Static Web App:
UI - Html/CSS/Javascript

QA-Community
CI-CD Basics
CI-CD Intermediate

Azure DevOps
DevOps Organisation and Project
Virtual Machine to create a GitHub repo
Using this repo we will apply continuous Deployment
Using this repo we will apply continuous Integation

Today :

1. App Service - qadp
2. GitHub Repo - qadevops050822

App Service - Docker - Example to check out the difference -

Today :

1. App Service - We will create one today - Continuous Deployment
2. GitHub Repo - qadevops050822

Continuous Deployment :

1. If anyone makes a change on GitHub
2. A new workflow will be started automatically
3. Check - Make a change in GitHub

Please check out Continuous Deployment

Function - (a,b)
Output - a+b

Test case scenarios -

1. A number , b string
2. B number, a special characters

Agent:

1. Downloading .NET
2. Clone repo
3. Run the compilation for the code

Development Server

- GitHub Integration
- CD
- GitHub URL - devqa
  - Branch - DEV

Testing/Staging
    GitHub Integration
    CD
    GitHub URL - stagingqa
        Branch - TEST

Production Server
    GitHub Integration
    CD
    GitHub URL - liveqa
        Branch - PRODUCTION

Please complete CI-CD with Azure DevOps

Once done please complete your bud activity for today.

BUD ACTIVITY:

Submission - Please upload :

1. Snapshot of what we are doing.(ex snapshot of tutorial steps)
    1. Mention the task at hand
2. Snapshot of Input including code
3. Snapshot of Successful output

MANDATORY: Please fill the feedback Link in chat window and below:

URL: https://evaluation.qa.com/
Course Code: DFE2CLOUC14
Course Pin: 4738823-6
DP-080 - Learn basic SQL
DP-900 - Fundamental - Data Engineering, Databases, SQL or No SQL
DP-300- Advanced SQL Database on Azure(Databases, SQL)

Thank you for joining :)

GitHub - COde Repo -
    Azure DevOps - Replace with Azure Repo
    Jenkins

Continuous Deployment vs Continuous Delivery

Continuous Integration: Automated Testing
Continuous Deployment : As soon as a change is made on my Repo
    A new build deploy workflow will be created automatically

We test, verify and then deploy our code

No code verified will reach my repo

Continuous Delivery - Continuous Deployment + Approval workflow

Creation of a pipeline - Azure DevOps

Approval Workflow : Any change made on GitHub :

-> a new request will be sent to Reviewer/Approver to approve the changes

: changes are okay : Testing and then deployment

: changes not okay : A request sent to developer to update changes

JIRA - Create tasks

Trello - Like JIRA or DevOps Boards or Github Boards

MANDATORY: Please fill the feedback

Link in chat window and below:

URL: https://evaluation.qa.com/
Course Code: DFE2CLOUC14
Course Pin: 4738823-6

Deepanshu

Deep or Anshu

Discussion on Project:

1. Expectations from us
2. Concepts to be included
3. What are the pointers to get started

Attempt a problem statement similar to the project

Today:

Docker

Overview
Installation
Getting started

1. Create an Application:
    1. Virtual Machine - Python flask application - IaaS
        1. We first install all dependencies on our VM - Python SDK, Flask , SQLAlchemy
        2. Then running the server - 5000 port is open and in use
        3. Our app will be available on IP: 5000
    2. App Service - PaaS - Lift and Shift
        1. Code - All installations are done for me
        2. Docker - Containerisation - Light weight platform which has my whole app including all

dependencies

3. In a docker I will have the following:
    1. Code is available - Application code to run the application
    2. Python SDK Installation - install python - Runtime
    3. Operating System - Linux Ubuntu

Light Weight Resource -
Manage multiple containers -
Complex Architecture -
Application will be light weight and Highly Available

	VM	Container
Space	I will install in VM disk	Include the image for container
Time	Repeat the steps in VM	Use the same image and

save time

Difference Same exactly
php,java, python Image does it all
Diff IP for diff machine
Registry is created for reference

In my system its available as the application
I create an Image out of it and save it as a registry

Application -python -flask
app.py and rest of the files
I ran the server with flask

With Docker : Not just by me but accessible to anyone who wants access to app
Image with docker - Local - Like Code

Make code available through GitHub,
Make image available through registry

BUILD - Check or compile my server
Push - Creating a registry
Pull - accessing a created registry
    Development server
    Staging/testing server- my service is stable and no issues
    Production/live Server
Registry:
    Secure Connection
        http://20.225.86.138:5000/
            https://20.225.86.138:5000/
        https://portal.azure.com/

HTTP to HTTPS - Secure Connection - SSL- Secure Socket Layer/TLS Transport Layer Security

1 VM is already working,

Now we will create another VM - Same

VN, Same RG

Please complete tutorial for Docker Registry

Available on QA- Community

BREAK

Start Time - 03:30 PM

qavmdock - registry-vm
Qavmdock2 - external-vm

```
sudo apt-get update
sudo apt install curl -y
curl https://get.docker.com | sudo bash
sudo usermod -aG docker $(whoami)
```

Registry - Publicly - Docker - hello-world

Registry - Private - register-vm

Registrered my Registry on docker

I delete from my system - Locally

Registry on docker still exists

Application: Security:

1. Data Security: Crucial info such as passwords, CIN, SSN, Credit card info, Mobile, email

Data at rest (database or its backup) or Data in motion (sending data from frontend to backend) - I encrypt info before the data is sent to backend

2. Network Security:

20.35.33.66:8080 - E-Commerce Website - Public

Portal for managing the Website - Adding Products, Promo codes
This app is only for my organization -

Amazon.com - Add to cart products and checkout

Private app to manage the app - Will not accessible to us

Who can access the app

How you can access the app

No access for us to actually add additional products or promos - 100%

VM for app, database , ai service - same Virtual Network

Whitelisting IP - Virtual Network - Firewalls

NSG - Deals with Inbound and outbound port - 8080, 80, 5000

Firewall - Additional layer of security before app is accessed

Acts as a wall - Policies - IP that will be access -

No access for any other IP

App - UI

Static Website - No backend - only

HTML/CSS

UI - Front-end - HTML/JS/CSS
    JS - Create an API Call to my logic or backend

Sign in page -
    Username/email
    Password

Def function(username , password){
// logic
// check
        if username exists - DB if in users table the specified username exists and password matches
            yes - Send a response back with the role of the user
            no - error message - the users doesn't exist or password not correct

}

Sending to my backend - Python/Java/

C# code is having API where I can make a call

App
Front end(8080) -> Backend -> DB
nginx Server

Front end <- Backend <- DB

Python code :
app.py
templates/index.html
templates/css/app.css
templates/js/app.js
init.py
routes.py

Flask - HTTP server

User table with name field

SELECT role FROM USERS where name = ‘deepanshu’;
Return a result - user exists
Nothing is returned - user doesn’t exists

1. Front end is in one app(30.64.74.64:80), backend(30.64.74.64:5000) is in another -
   URL - 30.64.74.64:5000/api/users
   Data - data{‘username’:’deepanshu’, ‘password’:’mountain’}

2. Frontend and backend is in same app
   1.

E-Commerce. Website - Monolithic App - 10000 users
1. Signin - 100 concurrent users -
2. Product category -

3. Product page - 700 users - requests at the same time
4. Add to cart -
5. Checkout - 100 users
6. Payment
7. Thank you page - 100 users
8. Post order management

I will scale my VM X 10 times

LOAD BALANCER

VM VM VM VM VM
VM VM VM VM

Microservice - into small modules

4 App -

1St app 2md App
3rd app 4th app
1 2, 3, 4
5, 6 7, 8

Why?

1. Scale them separately

2. High availability

Nginx - Docker1
Front-end - HTML/CSS/JS - Docker2 - 32.523.62.63:80
Backend - Python - Docker3 - 32.523.62.45:80
Database - DB Container -

```
Stages {
    Stage{
        Step{
        }
    }
}
```

Build project and save a copy to execute

Python/Docker

1. Create our files - app.py,

Dockerfile, .dockerignore

I would want to build my code - Dependency check

I am installing a new extension - build

Dependency that it only works on restart

Divide my build for any dependency

Application

1. Server or hosting
2. Coding to create my app
3. Compile my app
    1. Build tool and compiler
4. I will create the exact same virtual environment - Java 8, python 3.7, - Runtime env
5. Share with team for testing and deployment

Python code

folder is ready with app.py and html files

Flask - NO

I

Python Application

app.py
routes.py
init.py
templates/index/html
.
.
.
passwords.txt
Saving my sql username and password
API Key for Email Service - Sendgrid

Very crucial details - Sendgrid Api Key

Sendgrid account was Disabled
API Key was saved and stored publicly
Crawlers which are made to identify if a key has been leaked or available privately

.gitignore
.dockerignore - Any data that doesn't

need to be available/on cloud will never reach
.txt and fetch data from the same.

Publicly also means it cannot GitHub public repo

Please complete tutorial for Docker-Compose CLI
and Configuration

Available on QA Community

MANDATORY - Please complete Bud activities

The Bud Learner Tasks are:

- Self-Reflection -Bud activity you are doing currently
- Evaluation - 15 questions survey on Bud

Please input this accurately to the form:
Trainer Name - Deepanshu Pasrija
Cohort Number & specialism

Course Code - DFE2CLOUC14
Wave 2, Cohort 14

Thank you for joining :)

From scratch:

1. Create a Ubuntu 20.04 VM
2. Login with ssh
3. Install docker
4. Resume the bind-mount tutorial

Docker
Docker-compose - 3 modules

Demo till 3 pm
Then hands-on practice

LOAD BALANCER

DOCKER COMPOSE -
NGINX-1 NGINX-2
NGINX-3

Docker - Production Use cases

Application - E-commerce - 500 concurrent users (users accessing at the same time)

1 VM - 32 GB 4vcpu

Consider in this we come across a promotion(sale): 50000 -12 am

Front-end Back-end
Database

VM VM VM VM
VM VM VM VM

Testing/UAT/Basic App

Complex Infrastructure - 10+ Docker Containers
Orchestration - Manage the multiple

containers

Parallel Processing - Docker Swarm/ Kubernetes - AKS/GKS

Help in managing/scaling/removing a sing container in Swarm

Manager/server/master Node - Managing the working of worker nodes
Worker Node - Will be processing requests

But what if I am using a Docker?

Managing multiple docker-container working in parallel

1 app - Welcome page for nginx

Docker Docker

Docker
(App) (app)
(app)

Docker Compose

Flask-app
Web server
DB

Container for DB is not accessible/
reachable or is deleted

We create a volume outside the

container
Mount -
    Bind
    Volume

Git clone https://gitlab.com/qacdevops/duo-task.git
Docker build
Docker run
Curl localhost

GitHub - Used to store our code

1. Project

we also have a gitignore.txt consist of files or path I do not want on my server

Mounting - Raise it

Storage - Mount a storage

Docker - Bind -Mount

Help me to move data -

Host
Container
if I
make a change
Change also reflected

Project Discussion
Docker Swarm
5 modules - 4 labs at least 2 labs
How to get started for project

1. Documentation - Functionality Document
2. Create a basic design - Page by page
3. Database- Schema
4. Application
    1. UI - HTML/CSS/Js

Any App -
Football Team management
Sign in page
two input fields, one button to sign-in

create team,players

Start - tomorrow
TOTAL 3 DAYS
End - Thursday EOD - Deadline

FOCUS - Completing a basic project - Beautification, Functionality
Project Management -

1. Document - 3-4 hours
2. Database-schema - 1 hour
3. Application - html/css - 2 pages - 6 hours -

Git checkout https://github.com/username/org/repo.git

VM - Same vm also has your code - GitHub -
Windows/Linux - HTML page - Html,css,js
Flask Application - Python code - API

Request/Call
CI-CD - Jenkins -
Get started with Docker

Flow of the Application - User request - on UI, connected to our backend - query the db
Docker Swarm - One. Manager node and one worker - VM

Create 2 linux-VM with same virtual network - Same Resource Group

1VM : swarm-master
2VM : swarm-worker

HTTP-80 + HTTPS + SSH

BREAK

Start Time - 03:45 PM

sudo apt-get update
sudo apt install curl -y
curl https://get.docker.com | sudo bash
sudo usermod -aG docker $(whoami)

Please check-out Docker Swarm Management

Available on QA-Community

Load Balancing - Send a request from frontend to backend -Both are different containers

Load balance my nodes too - External Load balancer - Manage traffic across nodes

2 nodes - 1 tutorial
    1 node - Backend, frontend, database
- Application
    2 node -

ELB - 5 -
1. Node3
2. Node4
3. Node2
4. node1
5. node5

1. Documentation
    1. Readme.md - What is my app and its usage
    2. Creating ERD - Schema -> Tables and test data

3. Development -
    1. HTML/CSS - https://startbootstrap.com/themes or search for any template
    2. Python/JS - https://qa-community.co.uk/~/_/learning/flask/flask--validators-in-forms#setup
4. Connect to GitHub - https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup
5. Create repo - https://git-scm.com/book/en/v2/Git-Basics-Getting-a-Git-Repository
6. Jenkins - Install Jenkins and create CI-CD Pipeline
7. Docker - Create registry - https://qa-community.co.uk/~/_/learning/docker/docker--registry
8. Pull registry in Docker Swarm - Create service with 1 master and 1 worker
9. Video creation of working Application, CI-CD, Docker Swarm

10. Gathering all info to GitHub - Submit Git Repo

1st try to create - End goal -

Team Management- CRUD for team and players

- Sign-in
- Create team/player page
- Read
- Update
- Delete

Table

Name	team	status	ACTION BUTTONS
John	londonFC	active	edit

delete

HTML/CSS/JS - Python Backend

Projects - Individual

No help for code - Help you with direction - HTML/

I can help you with documentation

Employees
Department

Github repo is created with all code working

Good Morning Everyone

Welcome to DFE : Docker Training

Start Time - 10:15 AM

Waiting for audience

```
echo "Hello from the Jenkins job named: ${JOB_NAME}"
touch 1.txt 2.txt 3.txt 4.txt 5.txt
```

```
zip archive.zip *.txt
```