

# Setting Up Jenkins Pipeline to Deploy Docker Swarm

## DESCRIPTION

### Project objective:

You have to develop an environment for Docker networking.

### Background of the problem statement:

As you have worked on Docker containers previously, your manager has asked you to perform container scheduling over multiple hosts using Docker CLI and connect multiple hosts with Docker containers.

### You must use the following:

- Jenkins: To create a pipeline to deploy Docker Swarm
- Docker Swarm: To implement container networking
- Git: To connect and push files from the local system to GitHub
- GitHub: To store the Angular application.

### Following requirements should be met:

- A few of the source code should be tracked on GitHub repositories. You need to document the tracked files that are ignored during the final push to the GitHub repository.
- Submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document.
- The step-by-step process involved in completing this task should be documented.

## CODE

```
<!DOCTYPE html>

<html>

<head>

<title>Provisioning Test Page</title>

<link href="https://fonts.googleapis.com/css?family=Slabo+27px" rel="stylesheet">

<style type="text/css">

body {

    text-align:center;

    font-family: 'Slabo 27px', serif;
```

```

        height:100vh;
    }

    .vertical-center {

        position:relative;
        top:50%;
        transform: translateY(-50%);

    }

    img {
        width:100px;
    }
</style>
</head>
<body>
    <div class="vertical-center">
        <h1>Fibonacci Generator</h1>
        <p>The number at position <%= index %> is <%= value %></p>
        
    </div>
</body>
</html>

```

#### **HACKABLE:-**

```

<!DOCTYPE html>

<html>

<head>

```

```
<title>Provisioning Test Page</title>
```

```
<link href="https://fonts.googleapis.com/css?family=Slabo+27px" rel="stylesheet">
```

```
<style type="text/css">
```

```
body {
```

```
    text-align:center;
```

```
    font-family: 'Slabo 27px', serif;
```

```
    height:100vh;
```

```
}
```

```
.vertical-center {
```

```
    position:relative;
```

```
    top:50%;
```

```
    transform: translateY(-50%);
```

```
}
```

```
img {
```

```
    width:100px;
```

```
}
```

```
#command {
```

```
    width:50%;
```

```
    display: inline-block;
```

```
}
```

```
#stdout {
```

```
    width:50%;
```

```
    display: inline-block;
```

```
}
</style>
</head>
<body>
  <div class="vertical-center">

    
    <h1>Hackable: Code Injection</h1>

    <p>The following command was run on the server!</p>
    <code id="command">
      <%= command %>
    </code>
    <p>This was the result</p>
    <code id="stdout">
      <%= stdout %>
    </code>
  </div>
</body>
</html>
```

## **POSTS:-**

```
<!DOCTYPE html>
<html>
<head>
  <title>Provisioning Test Page</title>
  <link href="https://fonts.googleapis.com/css?family=Slabo+27px" rel="stylesheet">
  <style type="text/css">
    body {
```

```
font-family: 'Slabo 27px', serif;
height:100vh;
}
```

```
img {
width:100px;
}
```

```
.blog {
padding:50px;
}
```

```
.post {
padding:20px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="blog">
```

```

```

```
<h1>Recent Posts</h1>
```

```
<hr/>
```

```
<% posts.forEach(function(post){ %>
```

```
<div class="post">
```

```
<h3><%= post.title %></h3>
```

```
<p><%= post.body %></p>
```

```
</div>
```

```
<% }) %>
```

```
</div>
</body>
</html>
```

#### **JENKINSFILES:-**

```
pipeline {
    environment {
        registry = "naistangz/docker_automation"
        registryCredential = "docker hub"
        dockerImage = "
        PATH = "$PATH:/usr/local/bin"
    }

    agent {
        'docker'
    }

    stages {
        stage('Cloning our Git') {
            steps {
                git 'https://github.com/naistangz/Docker_Jenkins_Pipeline.git'
            }
        }

        stage('Building Docker Image') {
            steps {
                script {
                    dockerImage = docker.build registry + ":$BUILD_NUMBER"
                }
            }
        }
    }
}
```

```
stage('Deploying Docker Image to Dockerhub') {
    steps {
        script {
            docker.withRegistry("", registryCredential) {
                dockerImage.push()
            }
        }
    }
}

stage('Cleaning Up') {
    steps{
        sh "docker rmi $registry:$BUILD_NUMBER"
    }
}
}
```