Setting Up Jenkins Pipeline to Deploy Docker Swarm writeup

Setting up a Jenkins pipeline to deploy Docker Swarm and implement container networking can be a multi-step process. Below is a step-by-step guide to help you achieve this:

Step 1: Install and Configure Jenkins

1. Install Jenkins on a server or local machine.
2. Access Jenkins using your web browser and follow the setup wizard to complete the initial configuration.
3. Install necessary plugins: Go to "Manage Jenkins" > "Manage Plugins" > "Available" and search for and install the following plugins:
   - Docker Plugin: To integrate Docker with Jenkins.
   - GitHub Integration Plugin: To connect Jenkins with your GitHub repository.

Step 2: Set up Docker Swarm Environment

1. Set up multiple hosts that will participate in the Docker Swarm cluster. Install Docker on each of these hosts.
2. Initialize Docker Swarm on the manager node: Run `docker swarm init` on the designated manager node.
3. Join worker nodes to the Docker Swarm: After running `docker swarm init`, you will get a command with a token to join other nodes. Execute this command on the worker nodes to join them to the swarm.

Step 3: Create a GitHub Repository

1. Create a new GitHub repository to store your Angular application code.
2. Clone the repository to your local system: `git clone <repository_url>`.
3. Set up your Angular application in the cloned directory and commit the code.

Step 4: Configure Jenkins Job

1. Go to Jenkins Dashboard and click on "New Item" to create a new Jenkins pipeline job.
2. Choose the "Pipeline" option and configure the pipeline settings.
3. Under the "Pipeline" section, choose "Pipeline script from SCM" as the definition and select "Git" as SCM.
4. Provide your GitHub repository URL and set the appropriate credentials to access the repository.
5. Set up the Jenkinsfile: Create a Jenkinsfile in the root of your Angular application repository. The Jenkinsfile will define the stages and steps of your pipeline.

Step 5: Define Jenkins Pipeline Stages

Your Jenkinsfile should include the following stages:

1. **Build Stage**: In this stage, you will build the Docker image for your Angular application.
   - Use the Docker CLI commands in the Jenkinsfile to build the Docker image.

2. **Push Stage**: In this stage, you will push the built Docker image to a Docker registry (e.g., Docker Hub).
   - Use the Docker CLI commands to log in to the Docker registry and push the image.

3. **Deploy Stage**: In this stage, you will deploy the Docker Swarm service using the Docker image you pushed in the previous stage.
   - Use the Docker CLI commands to deploy the service on the Docker Swarm cluster.

Step 6: Configure Docker Swarm Networking

1. Define an overlay network in Docker Swarm using the following command:

```
docker network create --driver overlay <network_name>
```

2. Configure your Docker Swarm service to use the overlay network you created in the previous step.

Step 7: Configure Jenkins Global Credentials

1. In Jenkins, navigate to "Manage Jenkins" > "Manage Credentials."
2. Add a new credential of type "Username with password" for your Docker Hub account so that Jenkins can log in and push the Docker image.

Step 8: Save and Run the Jenkins Job

1. Save the Jenkins pipeline configuration.
2. Click on "Build Now" to run the pipeline job and trigger the deployment process.

The Jenkins pipeline will now execute the defined stages: build the Docker image, push it to the registry, and deploy the Docker Swarm service.

Remember to document each step of the process, including any troubleshooting or configurations required. This documentation should include the Jenkinsfile, Dockerfile (for building the Docker image), and any other relevant configuration files. Also, document any files that are ignored during the final push to the GitHub repository (e.g., sensitive files or temporary files).

Finally, share the GitHub repository link containing all the documentation and code with your manager for review and assessment.