**How to Automate Docker Swarm Service deployment using Jenkins**

## How to Automate service deployment to Docker Swarm using Jenkins

### Introduction

Jenkins is a wonderful tool for continuous integration and continuous deployment. The plethora of plugins available makes it really powerful. In this tutorial, I will show you how to use Jenkins to automate swarm deployment.

### How to do it

To do a Docker Swarm deployment all you need is a docker-compose file which will contain the references to docker images along with the configuration settings like port, network name, labels, constraints etc and to run this file you need to execute a command called “docker stack deployment”

So all we need to do is to send this docker-compose.yml file over SSH to Manager node and execute the command  docker stack deploy remotely. Let’s see how to achieve this.

### Jenkins plugin – Publish over SSH

We need to install a Jenkins plugins “Publish over SSH”, this plugin will allows us to

* Sends files over SSH(SFTP)
* Execute commands on a remote server

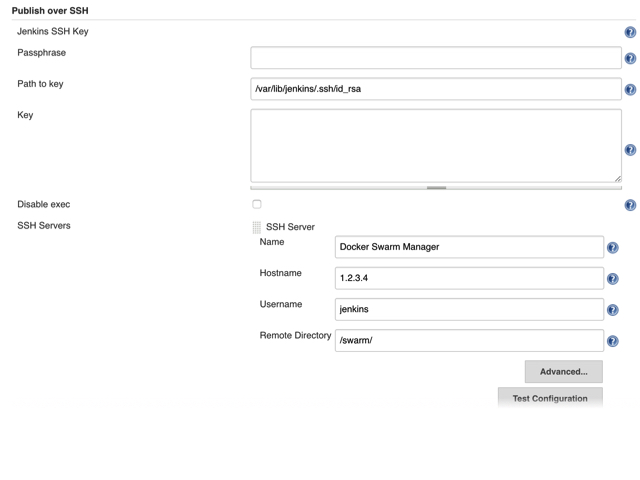
To add this plugin you need to go to Jenkins -> Manage Jenkins -> Manage Plugins -> Available and search for “Publish  
Over SSH”. Install this plugin and restart Jenkins.

### Adding Remote Hosts

Navigate to Jenkins -> Manage Jenkins -> Configure System and scroll down until you find Publish over SSH.

Since we need to execute the docker stack deploy command on any manager node, we need to connect to a manager node in Docker Swarm. This plugin offers various ways to connect to remote hosts, I prefer SSH public/private key value pair. Keys can be generated with ssh\_keygen. The private key must be kept on Jenkins server and the public key must be stored on the manager node.

Click on Test configuration and see if the connection is successful.  
Have a look at the screenshot below ( please note that Remote Directory contains a swarm directory if it doesn’t exist either leave Remote Directory as blank or create a directory on manager node, in the next step we will use this directory to publish docker-compose file)



You can add multiple manager nodes if you want.

### Configuring the Jenkins Job

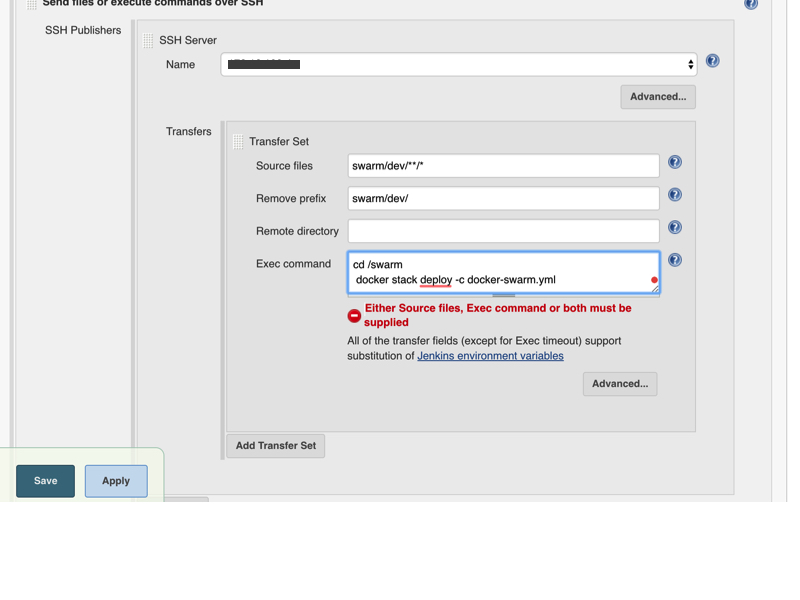
In this step, we need to tell Jenkins from where to get our docker-compose file and how to transfer using SSH to remote server and execute subsequently.

* + Under Source Code Management and add the repository(GIT/SVN) where you have checked-in or stored the docker-compose file.
  + Under Build section Select “Add Build Step” -> “Send files or execute commands over SSH”

Now Under SSH server Select the manager node where you want to publish/send the docker-compose file. In the Transfers Set block,

* In the Source, files enter the path for the docker-compose file. This folder is relative to your Jenkins Workspace so if say docker compose file exists in directory structure as swarm/dev/docker-compose.yml it can be written as swarm/dev/\*\*/\*
* In the Remove Prefix, enter the path that should not be created on the remote server.
* Now In exec Command, enter the command as shown below ( Please note that cd /swarm is only needed in if remote the configuration you have added swarm as a remote directory) cd /swarm docker stack deploy -c docker-compose.yml

This is how my configuration looks like for reference.



### Run the Jenkins Job

Now run this Jenkins Job using Build now and then check in Console Output to see the output of remote server. Hopefully, it will run fine !!