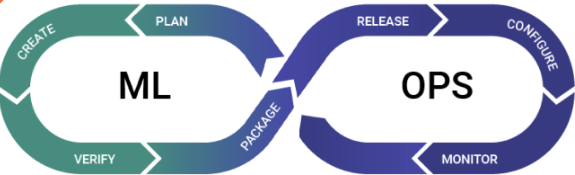
**AI 510 Artificial Intelligence of Cloud Computing**

**HOS03A Installation to Jenkins**

08/30/2022 Developed by Yared Shewarade

12/28/2022 Reviewed by Alekhya Malla

School of Technology and Computing (STC) @City University of Seattle (CityU)



**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to learn:

* Introduction to MLOps
* Introduction to Jenkins
* Jenkins Installation using Docker
* Jenkins Installation using other methods

**Resources**

* Deza, A., & Gift, N. (2021). *practical mlops*. O'Reilly Media, Inc.
* Treveil. (2021). *Introducing MLOps : how to scale machine learning in the enterprise (1st edition)*. O’Reilly.
* Microsoft(n.d), *Introduction to machine learning operations (MLOps)*, <https://docs.microsoft.com/en-us/learn/paths/introduction-machine-learn-operations/>
* Dingare, P. P. (2022). *CI/CD pipeline using jenkins unleashed: Solutions while setting up CI/CD processes*. Apress L. P.
* Smart, J. F. (2011;2014;). *Jenkins: The definitive guide* (1st ed.). O'Reilly Media.
* Gift, A. (2019). *Introduction to jenkins for DevOps* (1st ed.). Addison-Wesley Professional.
* Jenkins(n.d), Installing Jenkins, <https://www.jenkins.io/doc/book/installing/>
* Jenkins(n.d), Installing Docker, <https://www.jenkins.io/doc/book/installing/docker/>

1. **Introduction to MLOps**

Machine Learning Operations (MLOps) is an engineering discipline that aims to unify ML systems development (Dev) and ML system operations (Ops) in order to standardize and streamline the continuous delivery of high-performing models in production. It is the system development lifecycle and ensures high-quality software is continuously developed, delivered, and maintained in production.

MLOps can help organizations improve the quality of their machine learning models by providing a more reliable and automated way to develop, test, deploy, and manage them. It can also help organizations automate the process of training and tuning machine learning, which can save time and resources.

The core principles of MLOps are:

1. **Continuous Integration** (**CI**) is a process of integrating code changes from multiple developers in a single project. The goal is to create the code and verify the quality of both the code and the model by automated testing.
2. **Continuous Delivery** (**CD**) involves the steps that need to be taken to deploy a model to production, preferably automating as much as possible.
3. **Infrastructure as Code** (**IaC)** is the infrastructure that runs the machine learning models is treated as code, which can be versioned, audited, and automated.
4. **Monitoring and Logging** is the performance of machine learning models is monitored in production, and logs are collected to help with debugging and performance tuning.

MLOps involves the following steps:

* Data Extraction
* Data Analysis
* Data Preparation
* Model Training
* Model Evaluation
* Model Validation
* Serving & Monitoring

1. **Introduction to Jenkins**

Jenkins is an open-source continuous integration/continuous delivery and deployment (CI/CD) automation software DevOps tool written in the Java programming language. It supports the complete development life cycle of software from building, testing, documenting the software, deploying, and other stages of the development life cycle.

1. **Jenkins Installation**

**Note: Please have Python version as 3.8 above, you can check your Python version by running “python” command in PowerShell or Terminal. If upgrade is needed please follow the Python installation guide in this link:** <https://cityuseattle.github.io/docs/environment/python/>

1. Install java and Git on your laptop if you don’t have it already installed.
2. Jenkins released two types of versions. Long-term support (LTS) a which is a stable one and Regular release (Weekly). We will install the Stable one (LTS)
3. Check the [Hardware and Software requirement](https://www.jenkins.io/doc/book/installing/linux/#prerequisites) for your Operating System before jenkins installation

**3.1 Using Docker**

Docker is an open-source containerization platform used for developing, shipping, and running applications in lightweight virtualized environments called containers. It is used as a software development platform for developing distributed applications to separate from the infrastructure and to deliver the software quickly.

Follow the requirement and detail installation [here](https://docs.docker.com/engine/install/).

1. **Install and run Docker Desktop on Mac**

**Note:- Jump to section V if we have already installed Docker on HOS02, otherwise follow all steps**

1. Download Docker from [here](https://docs.docker.com/desktop/install/mac-install/) based on your Mac chip and follow the steps.

Icon

Description automatically generated

1. Double-click Docker.dmg to open the installer and drag the Icon to the Application folder

Diagram

Description automatically generated

1. Double-click the Docker.app in the Applications folder to start Docker

A picture containing table

Description automatically generated

1. Docker open and you can continue from here

Graphical user interface, website

Description automatically generated

1. **Install and run Docker on Window**

**Note:- Jump to section V if we have already installed Docker on HOS02, otherwise follow all steps**

1. Download Docker from [here](https://docs.docker.com/desktop/install/windows-install/) for your Window

Graphical user interface, text, application, chat or text message

Description automatically generated

1. Double-click Docker Desktop Installer.exe to run the installer.

Graphical user interface, application

Description automatically generated

(Optional) If Windows Docker desktop takes long time after default installation/restart:

Start button > type "powershell" > click "Windows PowerShell"

type following:

wsl --update

wsl --list --online

wsl --install -d Ubuntu-20.04

Close everything and restart Docker desktop again

1. Jenkins integration with GitHub

**Note: - We need Docker Hub, and GitHub account. Please create docker Hub account just like GitHub.**

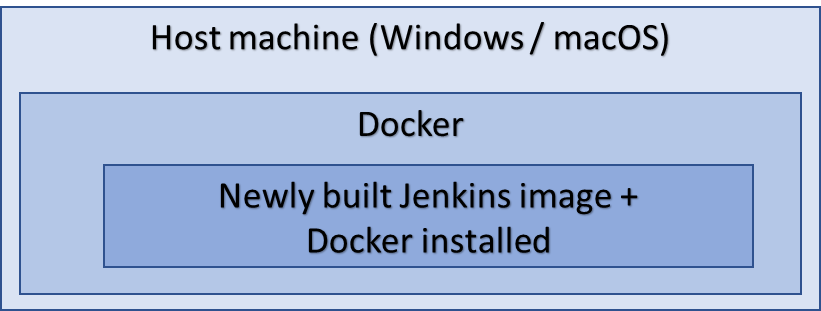
Jenkin is a free, open-source, and a powerful application that enables continuous integration and continuous delivery of projects on any platform. It can be integrated with a variety of testing and deployment technologies, and remote repository like GitHub.

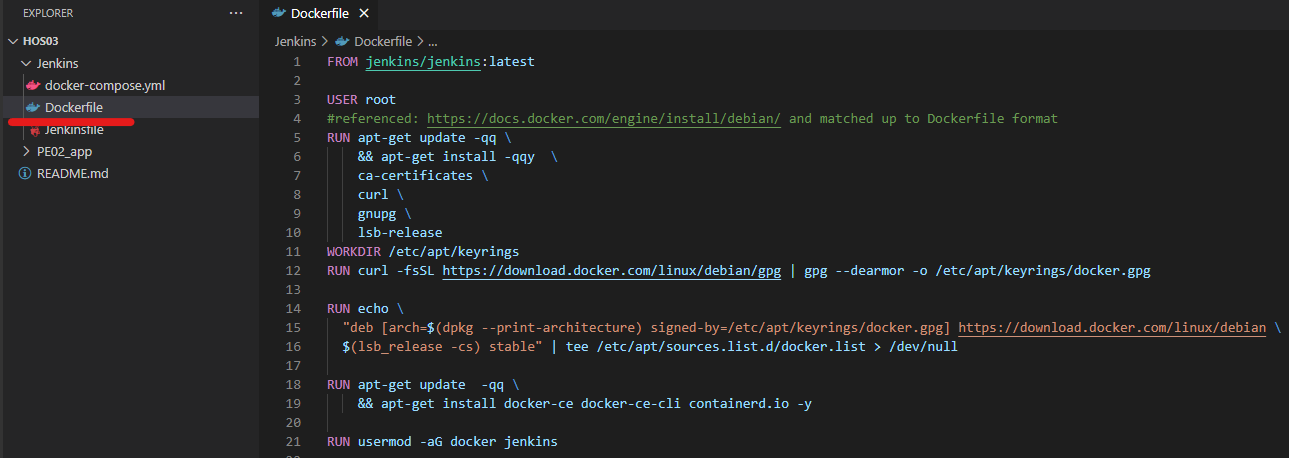
* 1. First let’s create a GitHub repository in your GitHub account. Click on “New repository” and create new repository called “**jenkinsPractice**”.

Note:- 1. You can create a new flask app **OR** you can copy the created app from HOS02 into this jenkinsPractice repository.

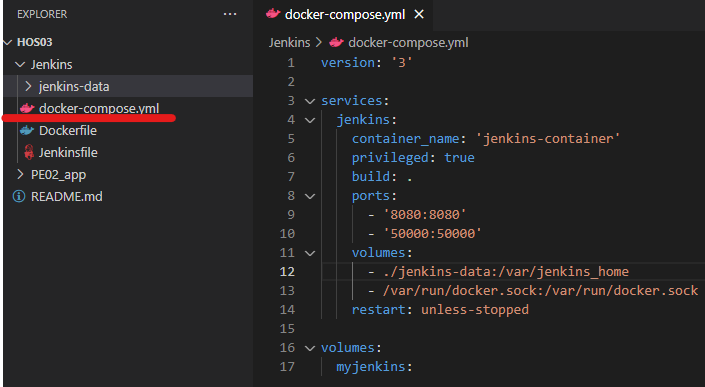
Graphical user interface, text, application, email, website

Description automatically generated

* 1. Jenkins provides an official docker image. However, to create a pipeline with the ‘docker build’ command, the image needs to include docker as shown below. Based on Jenkins official image, we will create a new image with docker installed.  
       
     During the process, we will use docker-compose command, which triggers the ‘docker build’ command and starts the newly built image. Open terminal/ cmd or any editor in HOS03 folder (your working directory), and create a Dockerfile. Type the following to create a Jenkins Docker Image (**Note: Dockerfile is provided, you can copy and paste it here or you can create your own**)



* 1. Next, create a docker-compose.yml file in HOS03 folder to make easy the process of Docker image creation. Type the following code inside yml file.(**Note: docker-compse.yml is provided, you can copy and paste it here or you can create your own**)



Now, the Docker volume called myjenkins is created to /var/Jenkins\_home directory inside the Docker container. In addition, looking at volumes content, we can see that Jenkins server data (Jenkins-data) is stored in the current working folder(HOS03/Jenkins/jenkins-data). If it needs to be at a certain path, you can change it to some other directory within your machine.

* 1. Type the following code to run the Created Docker image in your directory

Graphical user interface, text

Description automatically generated

Go to the bottom of the running Docker image to get the password. We can get the admin password from /var/Jenkins\_home/initialAdminPassword

Text

Description automatically generated with medium confidence

* 1. Next, run <http://localhost:8080> into your browser to set up Jenkins after the above Jenkins file is executed.

Graphical user interface, text, application

Description automatically generated

* 1. Next, Install the suggested Jenkins Plugins using the following steps

Application

Description automatically generated with medium confidence

* 1. Next, wait for the completion all the installation. It will take some time depending on the number of plugins.­­

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

* 1. Next create admin user and password if you want to, otherwise you can skip it

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

* 1. The next step is creating new Jenkins Pipeline. Click on the New Item and follow the stapes

Graphical user interface, text, application

Description automatically generated

* 1. Type “Jenkins\_Project” on the Enter an Item name, and the select Pipeline from the list

Graphical user interface, text, application

Description automatically generated

* 1. Select Poll SCM from the list under Build Triggers, and set a schedule as \* \* \* \* \* which will poll the SCM repository **every minutes**. Note that webhook can be configured in GitHub if the Jenkins server domain is exposed and accessible by the public.

Graphical user interface, application, Teams

Description automatically generated

* 1. Next, select “***Pipeline script from SCM***” option in the Pipeline section. Select ***Git***, insert ***URL*** of the SCM repository. It is optional to add credentials.

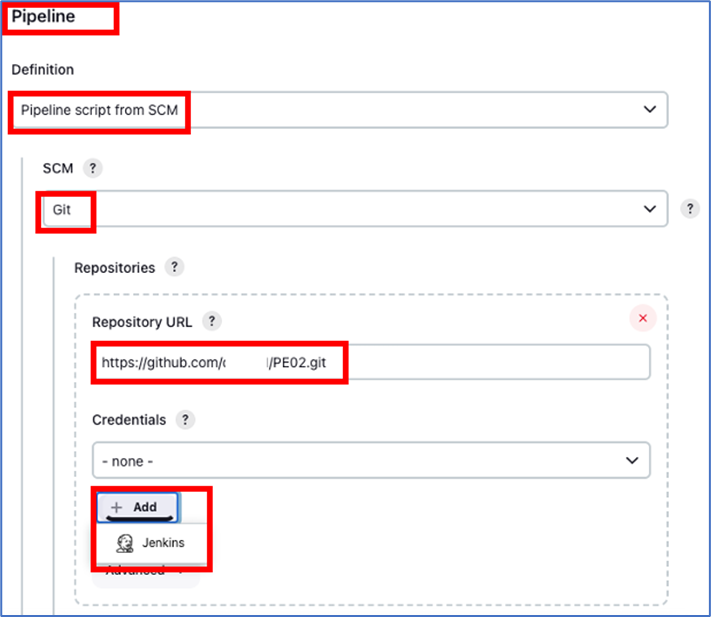
Note: You need to add your own repository. You may need to get authentication if you add GitHub repository of third party like City University GitHub repository.

Click save button to save it.

Graphical user interface, text, application

Description automatically generated

* 1. Click “+ Add”/”Jenkins” button. Complete creating credentials using Username and Password for docker hub(https://hub.docker.com/). Note that the ID is used Jenkinsfile in step “O” line number 4.



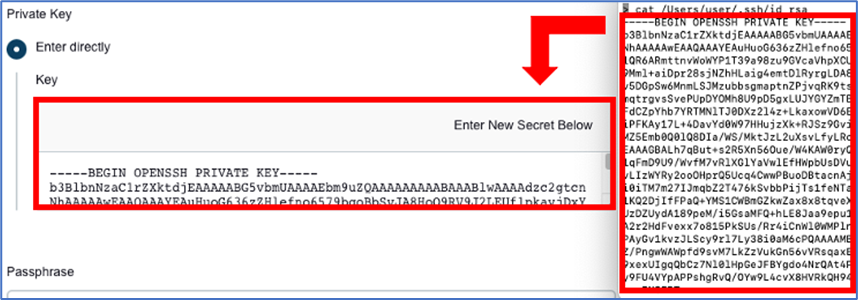
Graphical user interface, application

Description automatically generated

In addition, GitHub credential is needed for Jenkins to communicate with GitHub without warning.

* Click “ + Add” / ”Jenkins” button again to add SSH credential
* Select Kind: "SSH Username with private key", ID: something unique, Username: your git account user name, Private Key: click "Enter directly" and paste the private key
* Note that you should already have GitHub key pairs since you have been using GitHub. For example, from terminal you can execute, “cat /Users/{User dir}/.ssh/id\_rsa” to get the private key contents.
* The ID will be used for step “O” Jenkinsfile line number 11.
* click "Add" to close popup





* 1. Next, go to *Manage Jenkins -> Plugin Manager* to install “CloudBees Docker **Build and Publish”** and “Docker Pipeline” to build and push the Docker image to Docker Hub. You can type in search space if you can’t see it in the list. Note that there are several plugins that have CloudBees as prefix. Make sure you select one with “Build and Publish”.

Graphical user interface, text, application

Description automatically generated

* 1. Created “Jenkinfile” and type the following code to add a Jenkinsfile script once the above plugin has been installed.

Text

Description automatically generated with medium confidence

* 1. The last step is testing whether it works or not. **Commit changes** to the GitHub (jenkinspractice app) to test and see if Jenkins can access the GitHub repository and **Jenkinsfile**. If it is successful in the completion of the job, we can see the Docker image in your Docker Hub repository and in Jenkins as follow.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email, website

Description automatically generated

* Note: This is not mandatory but if you desire to clean up, you need to do the followings:
* Stop the docker-compose shell by giving the interrupt command (ctrl+c) and execute the below commands:
  + docker stop jenkins-container
  + docker rm jenkins-container
  + docker images #this will list images with tags/versions
  + docker rmi {jenkins image name}:{version of the image}
* Jenkins server data (Jenkins-data) is stored in the current working folder(HOS03/Jenkins/**jenkins-data**). Delete the folder.

**HOS submission instructions:**

1. Please install the GitHub Desktop: <https://cityuseattle.github.io/docs/git/github_desktop/>

2. Clone, organize, and submit your work through GitHub Desktop: <https://cityuseattle.github.io/docs/hoporhos>