DevOps SetupGuide 1.06

# Table of Contents

1 IMPORTANT INSTRUCTIONS 3

2 Introduction 4

2.1 System Requirements 4

3 Windows 6

3.1 Installation of Java Development Kit 6

3.2 Setup Eclipse 9

3.3 Setup GitHub Web, Desktop & Git 10

3.4 Installing Docker 11

3.5 Setup Jenkins 13

3.6 Setup SonarQube 17

3.7 Setup JFrog Artifactory 19

3.8 Download & Install VMWare Workstation 21

3.9 Download Puppet 21

3.10 Download Nagios 22

3.11 Download Putty & Puttygen 22

3.12 AWS Setup 22

# IMPORTANT INSTRUCTIONS

WHEN YOU CREATE SETUP AS PER THE INSTRUCTIONS BELOW PLEASE MAKE SURE THAT YOU

1. INSTALL ONLY THE VERSIONS AS MENTIONED IN THE DOCUMENT. INSTALLING A WRONG VERSION MIGHT RESULT IN ISSUES AND THE SOFTWARE MAY NOT WORK.
2. FOLLOW THE INSTRUCTIONS CAREFULLY AND KEEP ALL FILE NAMES EXACTLY AS MENTIONED IN THE DOCUMENT.
3. ON MAC DO NOT PUT ANY SPACES IN THE FILE/FOLDER NAMES.

|  |
| --- |
| Please prepare the setup as mentioned in this document. Issues would be observed if we deploy a different environment (for instance - Different Windows version, Linux distro, Docker toolbox, etc.). Debugging these issues would not be possible in the training. |

# Introduction

Welcome to DevOps Course. This guide will help you to setup the environment for this hands-on course.

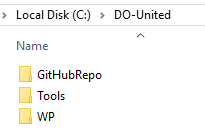
The environment setup is simple however it can become tricky if the instructions are not followed properly. Please also note that the setup requires downloads of large amount of data and can take some time. It is **STRONGLY**recommended that you finish the installation and AWS account creation before you come for the class.

Please retain all downloads in case you need to uninstall and install something again.

This document is valid for installations on Windows machines.

This guide is prepared based on 64-bit machine, if your machine is 32 bit, download and install the respective software versions accordingly.

Please download the corresponding software from respective website(s) as mentioned in the instructions below.Do keep all software in **Tools** folder as per the following folder structure:



The links provided in the document may have change after this document was published. Please try to find the appropriate link for corresponding software from respective website(s).

## System Requirements

|  |  |
| --- | --- |
| **Hardware** | **Windows** |
| MinimumRAM | 8 GB |
| Minimum CPU | Intel i5 |
| **Software** | **Windows** |
| OS | 64-bit version of Microsoft Windows 10(Pro or Enterprise version 14393& Above) |

Note: This guide has been created for below mentioned software versions

1. Java 10(jdk 10.0.1) (You can install any higher version as well)
2. Eclipse IDE for Java Developers(Photon)
3. GitHub Web, Desktop 1.3.3& Git 2.18
4. Docker 18.06.0
5. Jenkins 2.138.1
6. SonarQube 7.7
7. JFrog Artifactory 6.10
8. VMWare Workstation 14.0
9. Puppet 6.5
10. Nagios 5.5.2
11. Putty 0.70 &Puttygen 0.70

# Windows

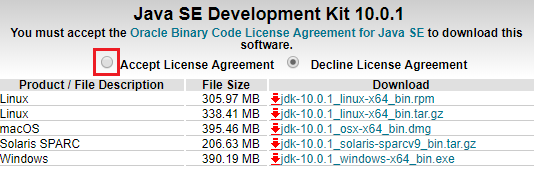
## Installation of Java Development Kit

#### Downloading the Java Development Kit

* Open an internet browser and go to

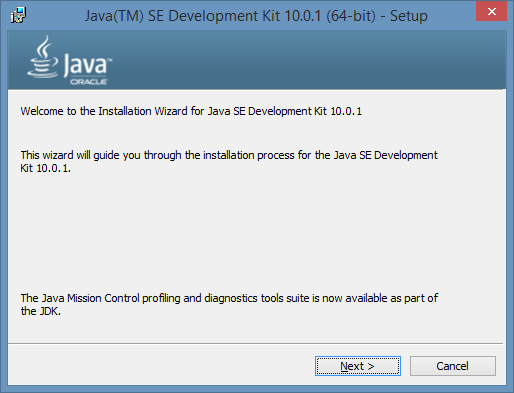
<http://www.oracle.com/technetwork/Java/Javase/downloads/index.html>

* Click the button “Download” below JDK
* Select “Accept License Agreement”
* Select the “Windows” installer. Choose the latest version available (>=10.0.1)



#### Installing the Java Development Kit

* Run the installer
* Follow the instructions from the wizard until the installation is completed

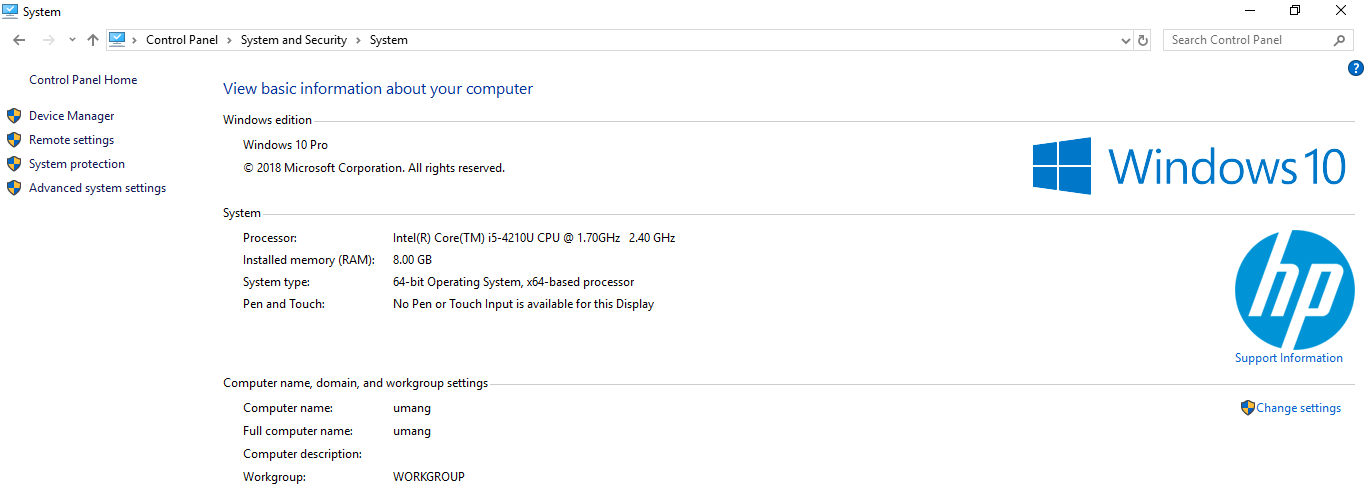


### Setting JAVA\_HOME

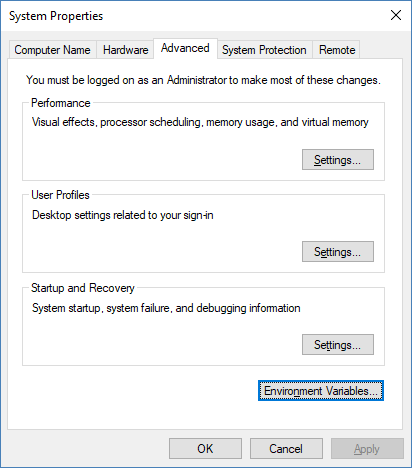
* To set JAVA\_HOME go to system properties.
  + Right click “My Computer” (or “This PC”) and select Properties.

On Windows 10 another way could be to go to Control Panel>System and Security > Click System

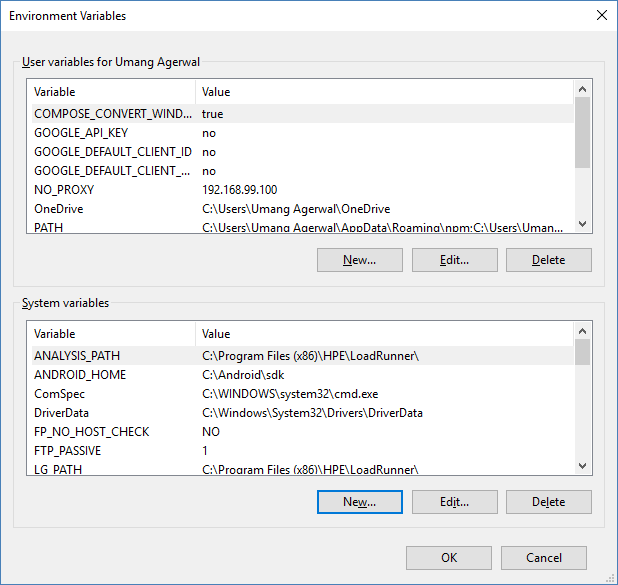
* + Click Advanced systems settings



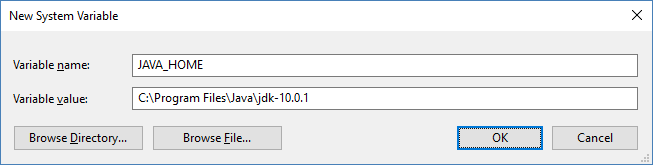
* + Click “Environment Variables” button

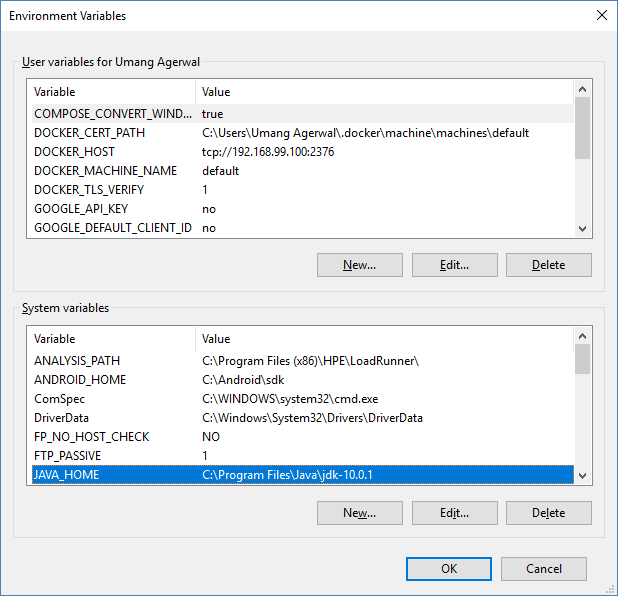


* Click on New button at the bottom of the screen for defining a new system variable



* + Set Java Home& Click OK button





### Setting Java in Path

* + Go to Path under System variables
  + Click Edit
  + Add new variable (Keep “**%JAVA\_HOME%\bin**” in it)
  + Click OK
  + To test that JAVA\_HOME has been set correctly type the following command in command prompt and observe the output:
    - echo %JAVA\_HOME%
    - Output should be something like C:\Program Files\Java\jdk-10.0.1
  + To test that Java path has been setup correctly type the following command in command prompt
    - java --version

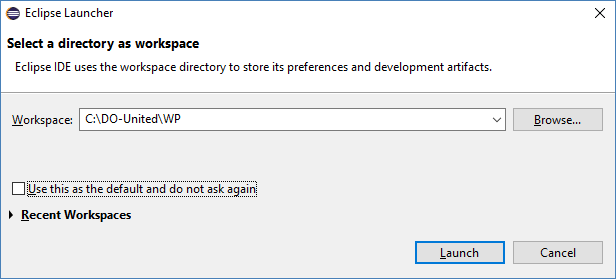
## Setup Eclipse

#### Downloading the Eclipse

* Download **Eclipse** from

<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/photon/R/eclipse-java-photon-R-win32-x86_64.zip&mirror_id=1272>

* Unzip the Eclipse folder
* Open the folder and launch the Eclipse by double clicking on eclipse.exe
* Workspace launcher opens with default workspace location
* Type in or browse C:\DO-United\WP



* Click Launch
* Click Workbench icon on right top section of window
* Eclipse home window will be launched

#### Setup Maven in Eclipse

* Go to Eclipse > Help > Install New Software > Add > Enter any meaningful name(e.g. Maven)
* Enter following URL under Location:

http://download.eclipse.org/technology/m2e/releases

* Click Add
* Select "Maven Integration for Eclipse" option
* Follow the instructions from the wizard until the installation is completed

Note: It may take some time to download the maven dependency

* Once it is done, click Finish

## Setup GitHub Web, Desktop & Git

#### **Setup GitHub Web**

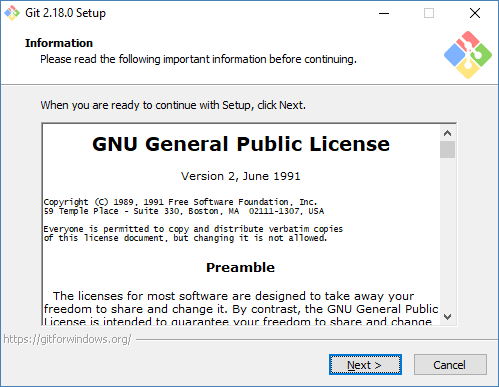
GitHub Web is applicable only if you are a participant and setting up your own machine. Each participants needs to have a GitHub account.

Launch below URL & create an account on GitHub

<https://github.com/>

#### **Setup Git**

* Download **Git** from <https://git-scm.com/downloads>
* Click “Download for Windows”
* Run the installer
* Follow the instructions from the wizard until the installation is completed



* Create a folder(e.g. git) in any drive in your machine(e.g. C:\)
* Run git-bash.exe from C:\Program Files\Git\ to launch GitBash
* Launch following commands in GitBash:
  + git --version

Note: It should show git version

* + cd /C/git
  + git config --global user.name "<your github username>"
  + git config --global user.email "<your email id>"

#### **Setup GitHub Desktop**

* Download **GitHub Desktop** from<https://desktop.github.com/>
* Click “Download for Windows (64bit)”
* Run the installer
* Follow the instructions from the wizard until the installation is completed



* Login to GitHub Desktop(File > Options > Accounts > Sign in)

## Installing Docker

**Pre-requisite:**Virtualization/Hyper-V is enabled on machine

* Steps of turning on (VT-x) on Windows machine with UEFI:

Go to "Change advanced startup options" in machine

Click "Restart" now under Advanced startup section

Click Troubleshoot

Click Advanced Options

Click UEFI Firmware Settings

Click Restart

Press F10 from keyboard (BIOS Setup)

Go to System Configuration

Press(F5/F6) from keyboard for enabling Virtualization Technology

Press F10 from keyboard to Save and Exit

Steps of turning on (VT-x) with BIOS – If you can’t find UEFI settings on your computer as in the steps listed above you may want to visit <http://www.sysprobs.com/disable-enable-virtualization-technology-bios>

To check Virtualization is enabled or not, follow below steps:

Way I --> Go to Task Manager > Performance window > CPU section > Virtualization: Enabled/Disabled.

Way 2 --> Control Panel > Programs > Programs and Features > Turn Windows features on or off > Hyper-V option: Selected/Deselected.

#### Downloading Docker

* Launch below URL & create an account on Docker

<https://store.docker.com/signup>

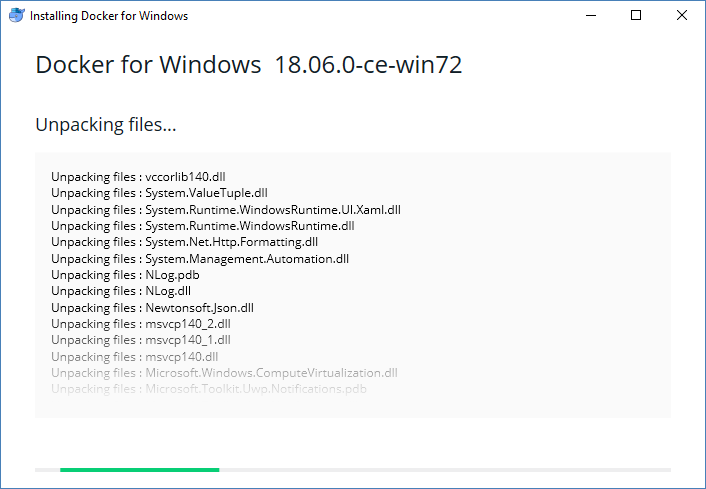
* Launch below URL

<https://docs.docker.com/docker-for-windows/install/>

* Click "Download from Docker Hub"
* Click "Please Login To Download"
* Login to Docker Store
* Click "Get Docker"

#### Installing the Docker

* Run the installer(Docker for Windows Installer.exe)
* Follow the instructions from the wizard until the installation is completed



* Launch "Docker for Windows" software, if not done already(it may take some time to be up)
* You can see Whale icon in notification area, it means Docker is up & running successfully
* Launch cmd (with administrative privileges) /Power Shell and runa command **docker --version**
* You should see output <<Docker version 18.06.0-ce, build 0ffa825>>
* Docker is now installed and running
* Run the following command
  + **docker run hello-world**
  + Once the download is complete, after a message that local image was not found, proceed to the next step

## Setup Jenkins

#### Installing Jenkins in Docker using devopstraining image

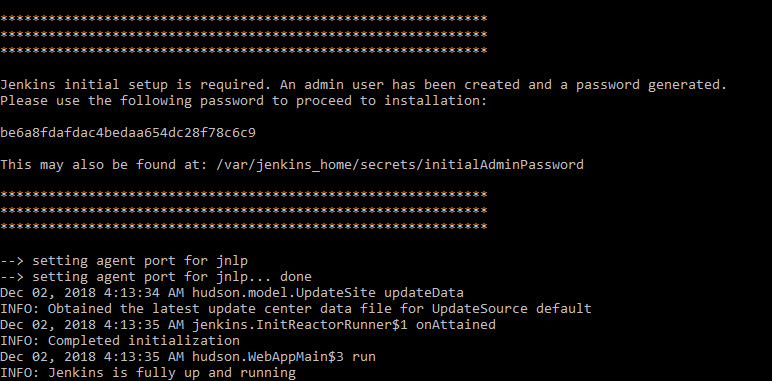
* Create **devopstraining** folder under **C:\Users\<your username>** location
* Launch following command in cmd

***For Windows:***

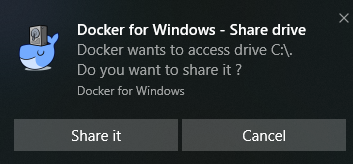
*docker run --rm -u root -p 8089:8089 -p 8080:8080 -v jenkins-data:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock -v c:\\"%HOMEPATH%"\devopstraining:/home veritycorp/devops*

***For Mac:***

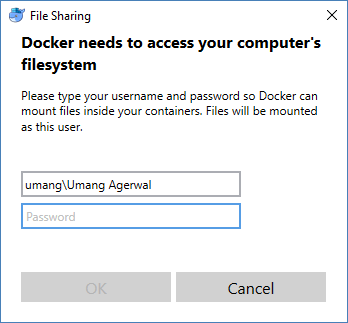
*docker run --rm -u root -p 8089:8089-p 8080:8080 -v jenkins-data:/var/jenkins\_home -v /var/run/docker.sock:/var/run/docker.sock -v $HOME:/home veritycorp/devops*

****

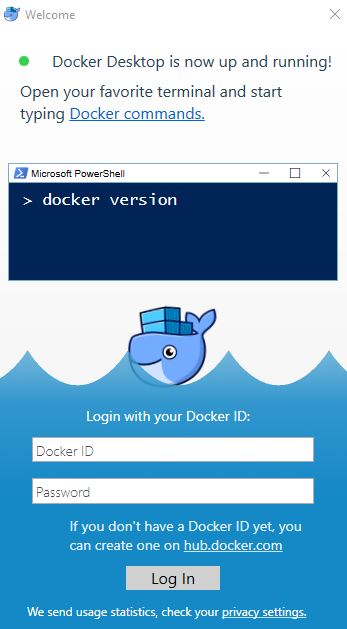
* You will get a pop-up UI requesting for sharing of C drive



* Click **Share it**
* When you get the following UI

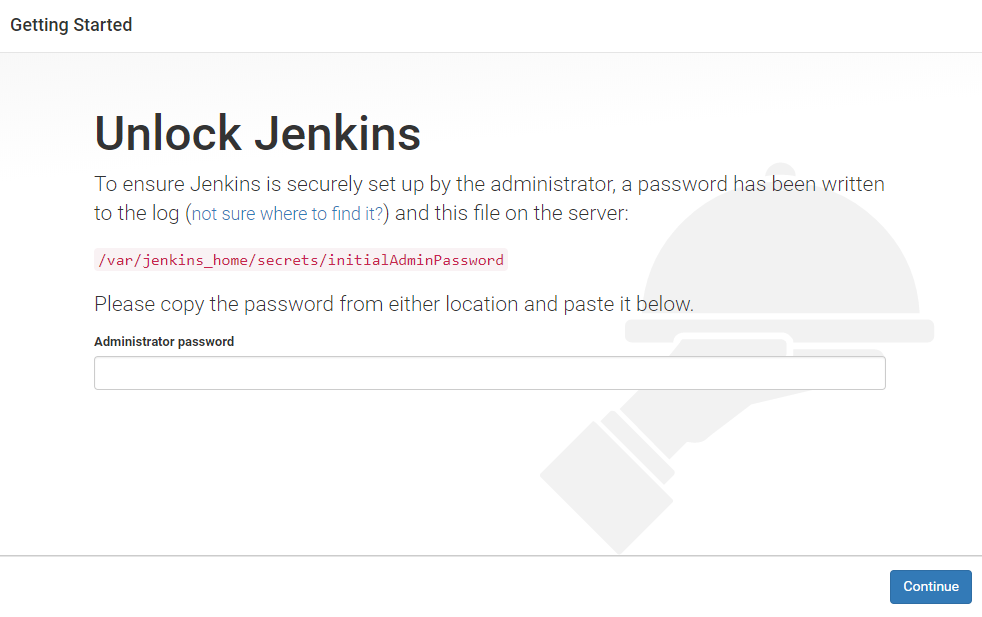


* Use your machine's username & password to give access to docker of your machine
* Login to Docker

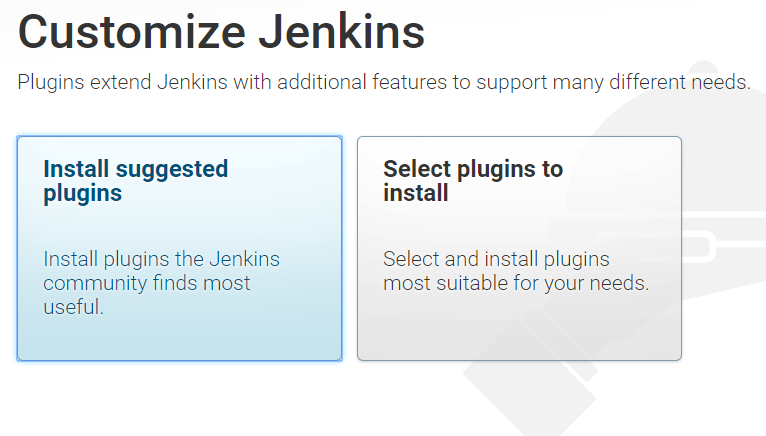


Note: Login credentialtobeused are the same that you chose while creating account in Docker. Use Docker ID and not the email id for login

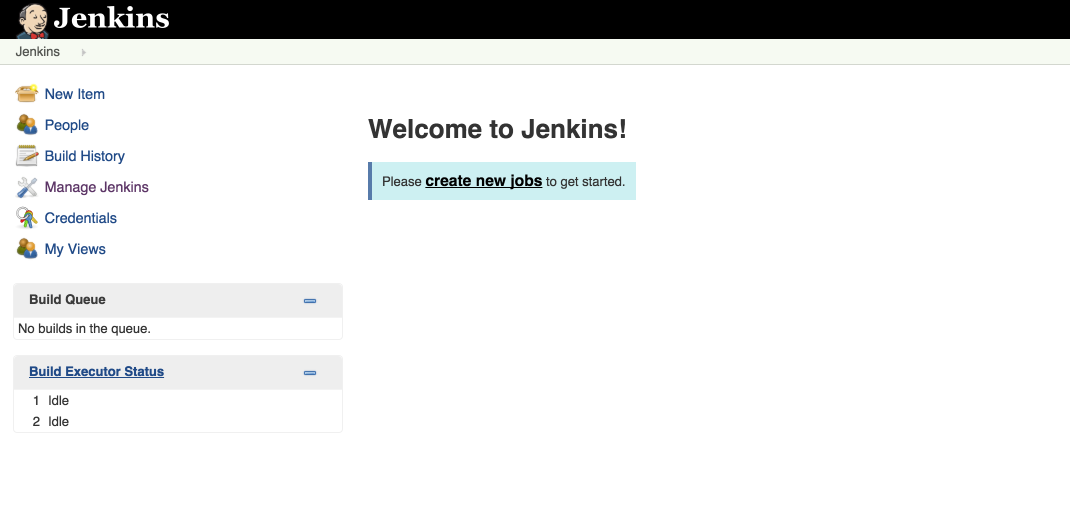
* After the 2 sets of asterisks appear in the terminal/command prompt window, browse to **http://localhost:8080** and wait until the Unlock Jenkins page appears
* From the terminal/command prompt window again, copy the automatically-generated alphanumeric password (between the 2 sets of asterisks)
* Click Continue



* Click "Install suggested plugins"



* Once the plugins are installed, Click "Continue"
* Click "Continue as admin"
* Click "Save and Finish"
* Click "Start using Jenkins", it will take you to Jenkins UI dashboard



## Setup SonarQube

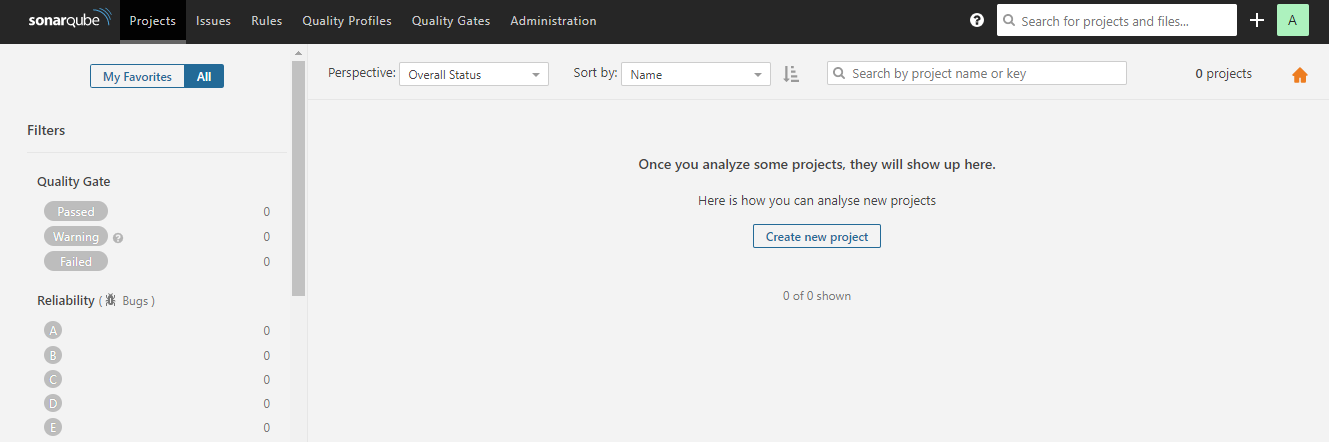
* Open the browser and go to

<https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.7.zip>

* Unzip sonarqube folder in C drive(if required give all access)
* Go to C:\sonarqube-7.7\bin\windows-x86-64\
* Run StartSonar.bat

(Note: It may take few minutes to be up)

* Launch **http://localhost:9000** on browser
* Login with U/N --> admin & P/W -->admin
* You will navigate to SonarQube dashboard



* Go to Administration > Security > Users > Update token icon > Enter token name(e.g. SonarQube Token)>Click Generate
* It will generate the token(you will need to use this token in Jenkins later)

Setup SonarQube in Jenkins

* Launch Jenkins

**Setup SonarQube scanner plugin in Jenkins:**

* Go to Manage Jenkins > Manage Plugins > Available Plugins > Search "SonarQube Scanner" plugin >Select the plugin >Install the plugin without restart

(Note: It may take few minutes to be installed)

* Back to Dashboard

**Setup SonarQube server in Jenkins:**

* Go to Manage Jenkins > Configure System > SonarQube servers section > Add SonarQube > Enter following details:

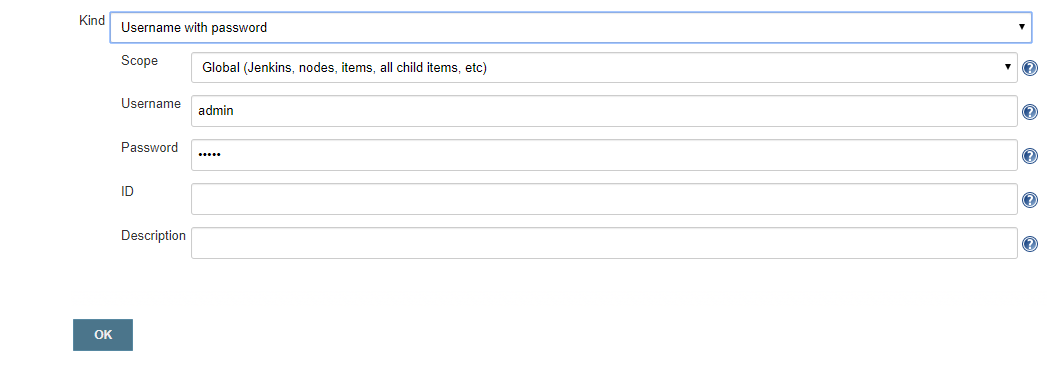
*Name - <any meaningful name>(e.g. My SonarQube Server)*

*Server URL - <your ip address with port 9000>(e.g. http://192.168.101.47:9000)*

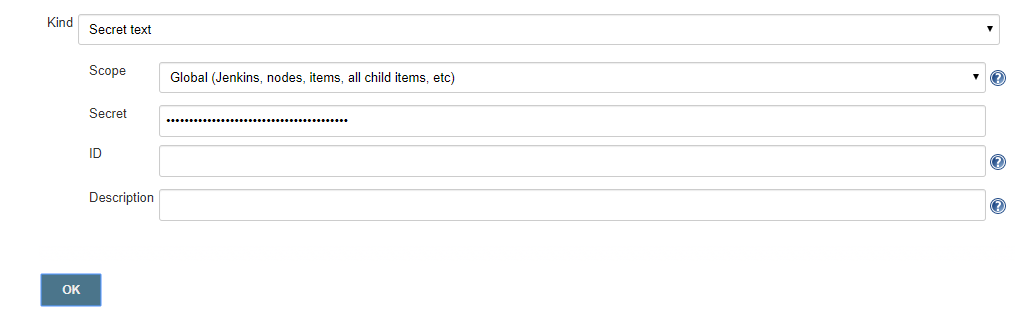
*Server authentication token - <token id that you got from SonarQube web portal>*

Note: If you don't find a field to write the server authentication tokenthen add a token at Jenkin's credentials level by following steps. After that come back to this screen and now you should be able to enter the token you created.

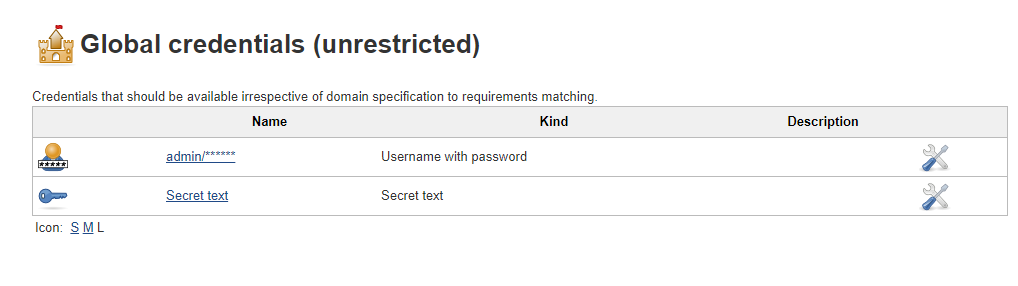
* + (If not done already)Go to Jenkins dashboard > Credentials >Stores scoped to Jenkins section > Click global >Click Add Credentials>Setyour Jenkins's Username with password credentials> OK



* + Click Add Credentials >Set Secret text(Enter your SonarQube Key) > OK



* + Your both credentials will be added in to list



* Select "Enable injection of SonarQube server configuration as build environment variable"
* Apply & Save
* Back to Dashboard

**Setup SonarQube Scanner in Jenkins:**

* Go to Manage Jenkins > Global Tool Configuration > SonarQube Scanner section> Add SonarQube Scanner > Enter following details:

Name - <any meaningful name>(e.g. sonar-scanner)

* Select install automatically checkbox
* Apply & Save
* Back to Dashboard

## Setup JFrog Artifactory

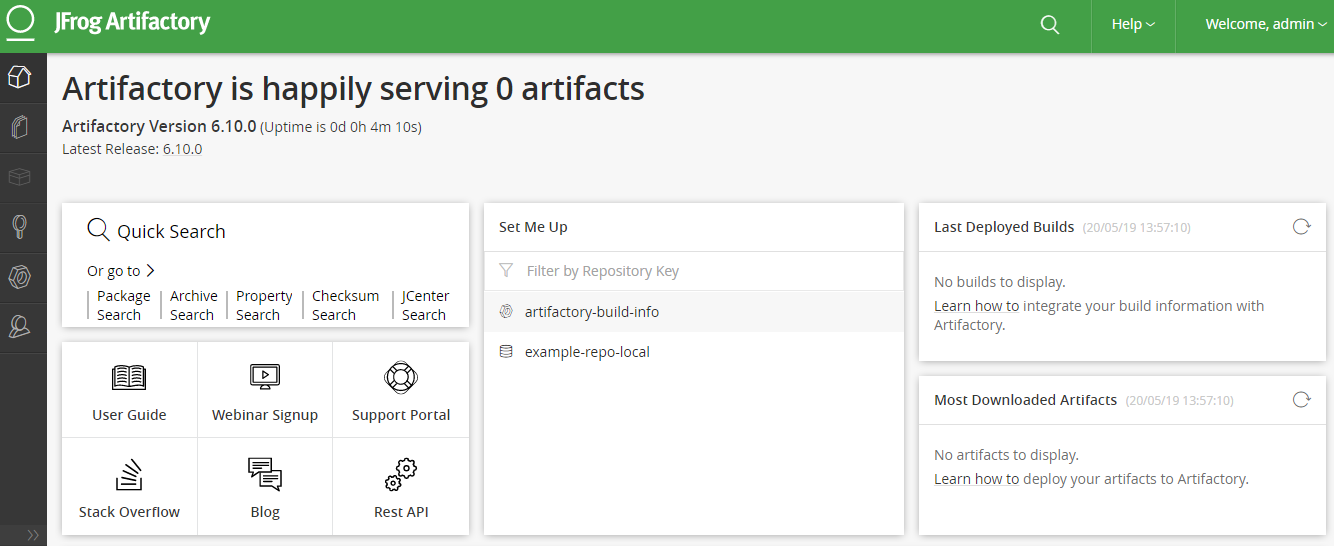
* Open the browser and go to

<https://api.bintray.com/content/jfrog/artifactory/jfrog-artifactory-oss-$latest.zip;bt_package=jfrog-artifactory-oss-zip>

* Unzip artifactory folder in C drive(if required give all access)
* Go to C:\artifactory-oss-6.10.0\bin\
* Run artifactory.bat

(Note: It may take few minutes to be up)

* Launch http://localhost:8081 on browser
* Login with U/N --> admin & P/W -->password
* You will navigate to Artifactory dashboard



Setup Artifactory in Jenkins

* Launch Jenkins

**Setup Artifactory plugin in Jenkins:**

* Go to Manage Jenkins > Manage Plugins > Available Plugins > Search "Artifactory" plugin > Select the plugin > Install the plugin without restart

(Note: it may take few minutes to be installed)

* Back to Dashboard

**Setup Artifactory server in Jenkins:**

* Go to Manage Jenkins > Configure System >Artifactory section > Add Artifactory> Enter following details:

***Artifactory Server ID*** *- <any meaningful name>(e.g. My ART)*

***Server URL*** *- <your ip address with port 8081>(e.g. http://192.168.101.47:8081/artifactory)*

***Default Deployer Credentials*** *- <U/N & P/W details that you have used for Artifactory web login>*

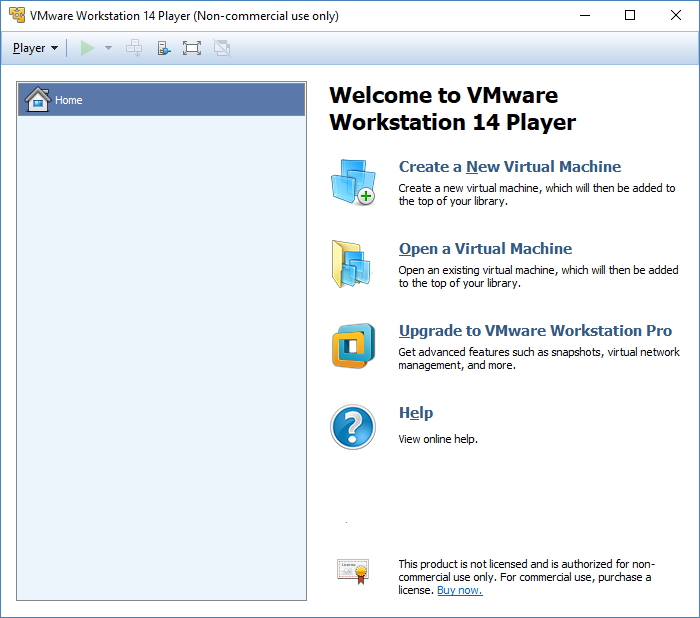
* Click Test Connection to verify Artifactory is connected to Jenkins successfully or not
* Apply & Save
* Back to Dashboard

## Download & Install VMWare Workstation

* Open the browser and go to

<https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html>

* Click "Download Now"
* Once download is finished, run the installer to install VMWare Workstation
* Follow the instructions from the wizard until the installation is completed



## Download Puppet

**Pre-requisite:**Virtualization/Hyper-V is disabled on machine

* Open the browser and go to <https://puppet.com/download-learning-vm>
* Enter your email id at right section & Click “Download Learning VM”
* Click "Download now“
* OVA file will be downloaded
* Once download is finished, unzip the ova file

## Download Nagios

**Pre-requisite:**Virtualization/Hyper-V is disabled on machine

* Open the browser and go to <https://www.nagios.org/downloads/nagios-core/>
* Click Nagios XI Download
* Click "Download Now" for Windows
* Click "Download Now" for Open Virtualization Format
* Skip the form
* OVA file will be downloaded

## Download Putty & Puttygen

* Open an internet browser and go to

<https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe>

* putty.exe will be downloaded
* Open an internet browser and go to

<https://the.earth.li/~sgtatham/putty/latest/w64/puttygen.exe>

* puttygen.exe will be downloaded

## AWS Setup

* Open an internet browser and go to

<https://aws.amazon.com/free/>

* Click the button "Create a Free Account"
* Create an free account on AWS

Note: You will need to provide your debit/credit card details while creating account on AWS. There will be call verification by AWS on your contact no. AWS may take 24 hours to approve your account, if not please contact to AWS customer support team

* Once account is approved, login to AWS and make sure you are able to login successfully

<https://console.aws.amazon.com/ec2/>

* You will navigate on AWS Dashboard

