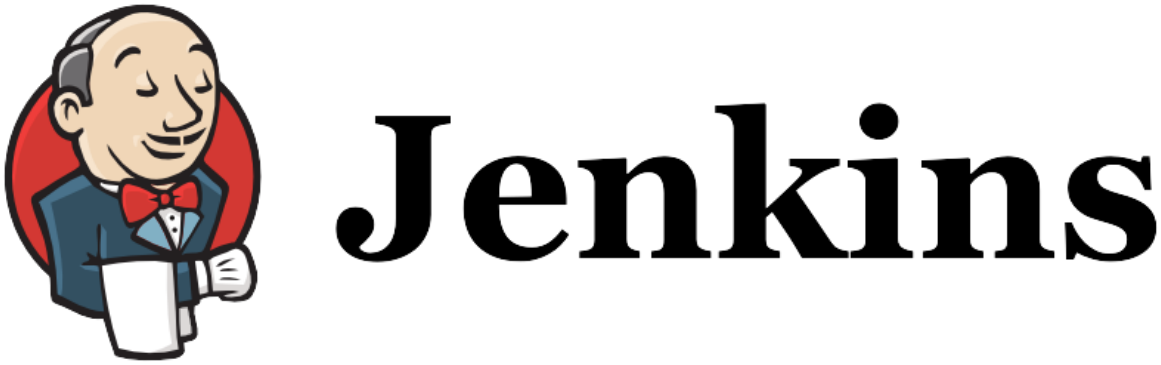
Install Jenkins software development automation server on GNU / Linux

Author : admin



As I have gone through a hiring procedures for **Dev Ops system administration (Senior System Engineer)** positions in the largest IT company in Belarus EPAM (A Global provider for software engineering and IT consulting.one of the **Dev Ops test tasks** to do was to automate installation of **Jenkins software development automation server on a** Virtual Machine running Linust Guest of choice (VMWare / VirtualBox) with **Ansible / Docker or Pure Shell Script** as I love simplicity I choose to do it via **Bash Shell script**.

As Dev Ops position is more and more transforming into a programmer job into the **New Age of non-sense Cloud Computing (I hate Clouds guys - I share Richard Stallman opinion that "Clouds are clouding your minds" !!!) ...** the need for environments such as Jenkins allowing multiple pseudo "sys admins" (mostly copy / paste new age coders) to write and build there programs in **Go Language / Python / Perl / Bash** with a single environment for **Continuous Integration (CI)** that could deploy and keep software versionings in ***GitHub / Mercurial / SVN*** is exponentionally raising.

Old School Computer Geeks would definitely be amazed (***reasonably***) on why on earth would someone need a **Web Based Java Crazy environment** that **Jenkins** is to **build a multiple language code and submit it to a source repository system**.
But as the people like to hype and make easy things harder adding more and more layers of complexity, the product is the new buziness hype terms **Continuous Integration / Continuous Integration** raise exponentially together with the softwares to do stuff.
As IT people are becoming more and more lazy and illitetelarete things like Jenkins is the next web development CI environment that is about to die in the coming 5 to 10 years.

Jenkins doesn't really cut the need for writting scrpits to make your application (**Makefiles**), for the

Business Corporate world it is heavily used nowadays, because it is **used to building projects using Web UI, running tests, doing static code analysis, and deploying**.

What makes Jenkins in terms of **IT architecture design solution** is that every project that it builds is build via its **Java Virtuam Machine Processor** backend (the server itself runs in the background of the OS runs it through Java VM as a **WAR file (Web Application Resource)** with the help of Few **Ruby and other scripted files**.
Besides that **JNS** has in the moment of writting more than 1400 years spanning across **platforms, UI, administration, source code management, and, most frequently, build management.**

Jenkins is either used as a standalone server or as a servlet in Java app servers such as Tomcat.

This is the short script I came up with in bash that when executed **installs Jenkins on the remote VM host server that ran Debian 9 Linux, the** install_jenkins.sh script is downloadable here**.**

```
#!/bin/bash
# Install jenkins and test whether it runs prints password on prompt or send via email
# if email variable is set Jenkins password will be set to your email of choice using mail command
# NOTE: bsd-mailx package should be installed in order for email sent to work and local machine should be running a properly configured
# relay SMTP
# Author: Georgi Georgiev
# hipo@pc-freak.net
email='hipo@mail.com';

add_repos_install_jenkins () {
apt-get install --yes -qq apt-transport-https git curl

wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

if [ "$(sed -n '/jenkins/p' /etc/apt/sources.list|wc -l)" -eq 0 ]; then
echo 'deb https://pkg.jenkins.io/debian binary/' >> /etc/apt/sources.list
fi

apt-get update -qq && apt-get install --yes -qq jenkins
}

check_j_install () {
if [ "$(dpkg --get-selections | cut -f1|grep -i jenkins)" ]; then echo 'succesfully installed';

else printf 'Problem in installing please check';
exit 1;
```

```
  fi

  }

  check_j_running_s_pass () {
if [ $(ps -e -o command|grep -i jenkins) ]; then
echo 'Jenkins process working.';
echo '... do more here if necessery with some more commands';
else
echo 'not working log to file' >> jenkins.log
exit 1;
fi

  JENKINS_PASSWORD=`cat /var/lib/jenkins/secrets/initialAdminPassword`;
echo "Jenkins Admin password is $JENKINS_PASSWORD" | tee -a
"jenkins_credentials.log";
if [ ! -z $email ]; then
echo $JENKINS_PASSWORD | mail -s "NEW Jenkins password" $email

fi

  }

  main () {
      add_repos_install_jenkins;
      check_j_install;
      check_j_running_s_pass;

  }

  main;
```

To run the script on the remote VM server started for the purpose I **created a passwordless ssh key authentication** with

```
  ssh-keygen -t rsa
...
```

and

```
ssh-copy-id -i ~/.ssh/id_dsa.pub root@remote-vm-host.com
...
```

command **...** for more check out my previous article "**How to execute command to Multiple Servers / Establishing passwordless SSH key authentication on 50+ servers**"

Once the passwordless authentication was established to **remote Private Virtual Server** I've used scp command to upload my **install_jenkins.sh** script with:

```
# scp -v install_jenkins.sh root@remote-vm-host.com:/root/install_jenkins.sh
...
Sending file modes: C0644 726 install_jenkins.sh
Sink: C0644 726 install_jenkins.sh
install_jenkins.sh
...
```

Next to run the **install_jenkins.sh** on remote host I used remote SSH run command capability, the syntax goes like this:

```
ssh [USER-NAME]@[REMOTE-HOST] [command or script]
```

In that case the command I used was:

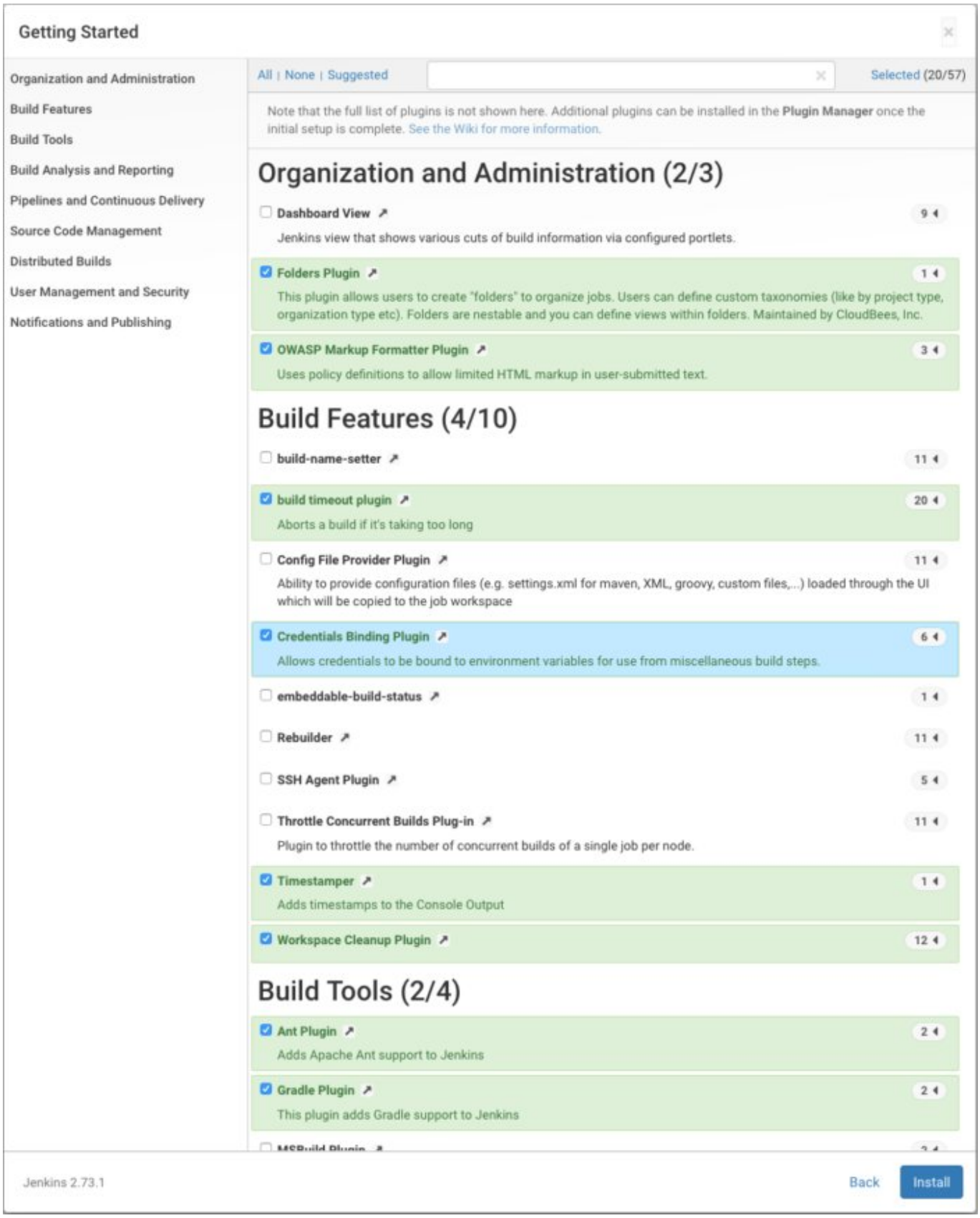
ssh root@remote-vm-host.com "chmod +x; /root/jenkins.sh"

Next I launched **Firefox browser** and accessed **http://localhost:8080** on the VM host and used the long password generated from the script by command:

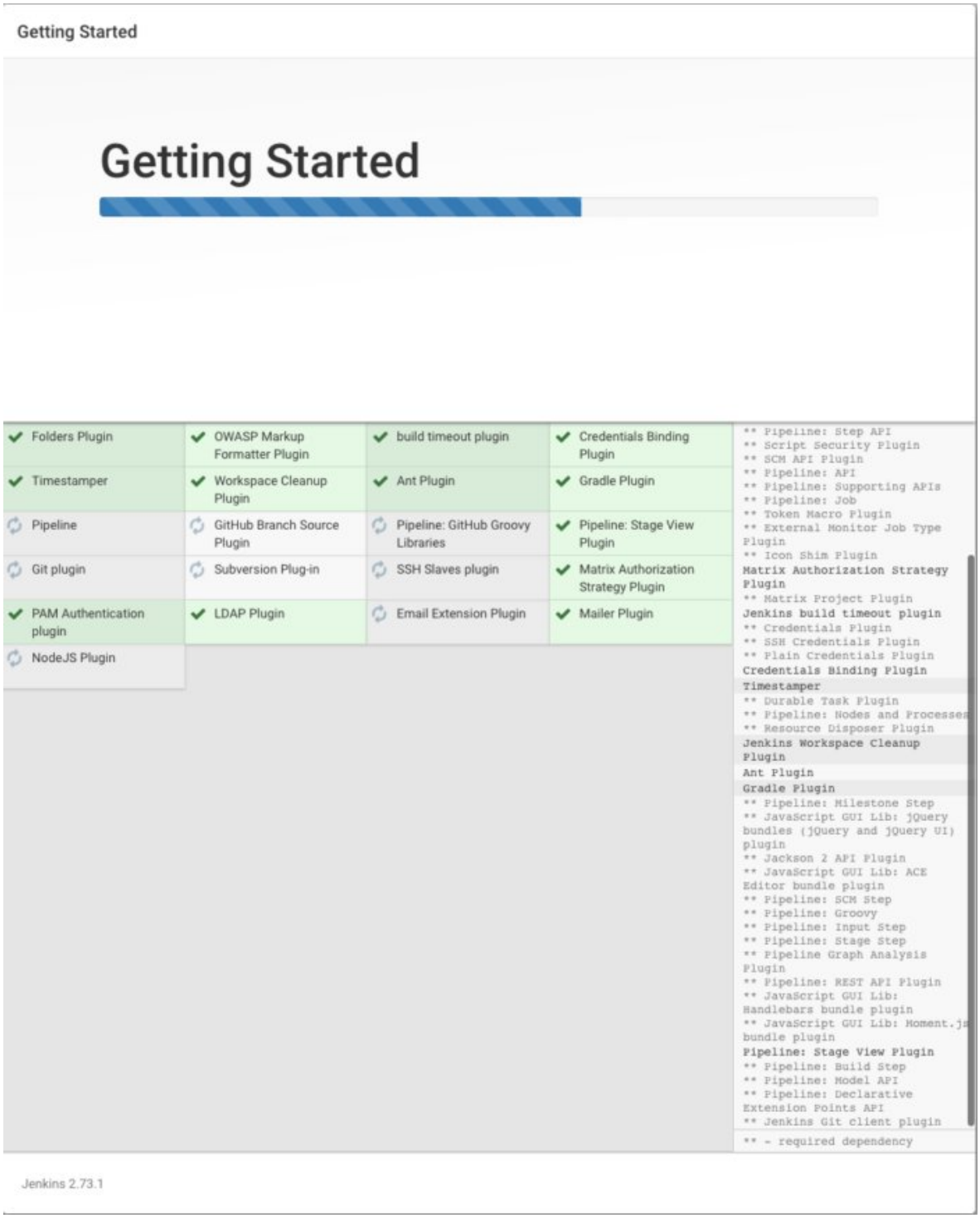
JENKINS_PASSWORD=`cat cat /var/lib/jenkins/secrets/initialAdminPassword`;

echo $JENKINS_PASSWORD

In the process of **Initial Jenkins setup** I selected the GitHub plugins necessery for me to **connect Jenkins with GitHub WebHooks (for that perhaps I will write another article when I have time).**

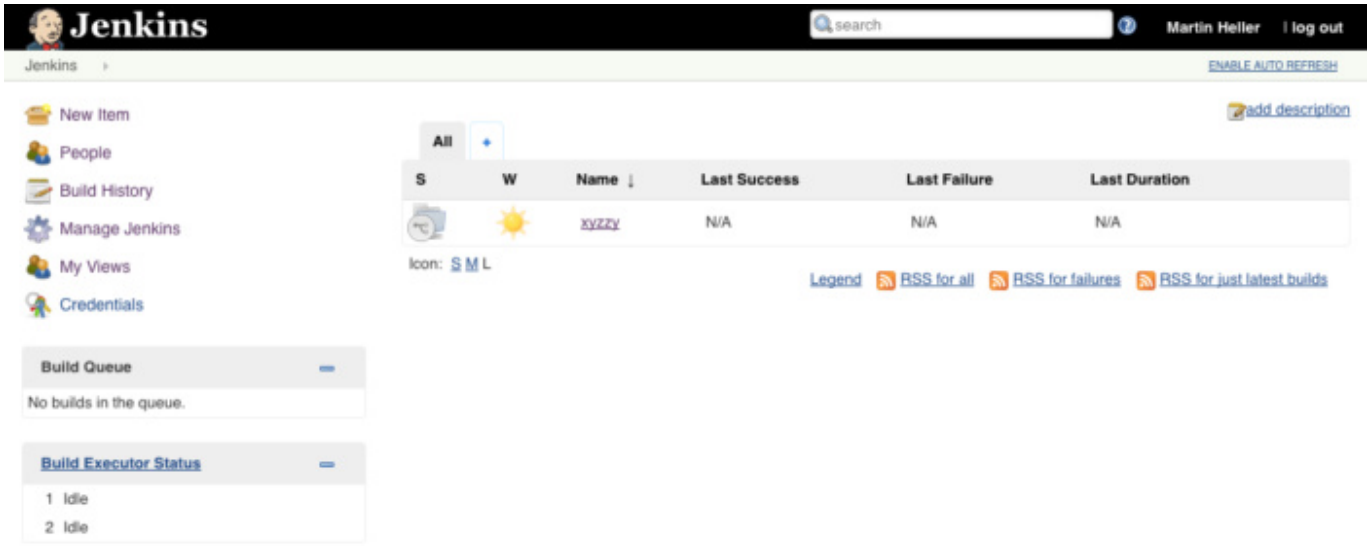


Jenkins Getting Started Initial Screen



Jenkins Selecting Plugins Screen

Once **successfully set-up Jenkins Initial Project creation / Configuration (Control Panel) screen looks like so**



There was also a task to create a simple print Jenkins and **Shell environment** variables with a sample **Jenkins Free Style Project.**
Following the **New Item** menu and setting it to **Execute Shell commands** as a **Build** Parameters, the parameters set for the **Jenkins Free Style Project to Print the Environment Varibles** were like so:

```
#!/bin/sh

echo “Jenskins Environment Variables”

echo "BUILD_NUMBER" :: $BUILD_NUMBER

echo "BUILD_ID" :: $BUILD_ID
```

```
  echo "BUILD_DISPLAY_NAME" :: $BUILD_DISPLAY_NAME

  echo "JOB_NAME" :: $JOB_NAME

  echo "JOB_BASE_NAME" :: $JOB_BASE_NAME

  echo "BUILD_TAG" :: $BUILD_TAG

  echo "EXECUTOR_NUMBER" :: $EXECUTOR_NUMBER

  echo "NODE_NAME" :: $NODE_NAME

  echo "NODE_LABELS" :: $NODE_LABELS

  echo "WORKSPACE" :: $WORKSPACE

  echo "JENKINS_HOME" :: $JENKINS_HOME

  echo "JENKINS_URL" :: $JENKINS_URL

  echo "BUILD_URL" ::$BUILD_URL

  echo "JOB_URL" :: $JOB_URL

  echo “===Linux Shell Variables ===”
env
```

Well, it wasn't really a rocket science was it?! :)

That's all folks, see you soon.