Price Service – Digital Ocean Jenkins set up with Github repository

Here are steps I used to build Price Service Jenkins server. It would be better to build an automated process as built for product service. I’ve detailed where I got hold of the required rpm files, but many of them are also now in the Product Service Github repositories.

1. Create server with Centos 5.8 (2 core with 2Gb memory fine).
2. Install Java 1.7 JDK

Download **jdk-7u55-linux-x64.rpm** from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Transfer it to server and do

sudo rpm -Uvh /path/to/binary/jdk-7u55-linux-x64.rpm

1. Install Gradle 1.10
2. # installs to /opt/gradle
3. # existing versions are not overwritten/deleted
4. # seamless upgrades/downgrades
5. # $GRADLE\_HOME points to latest \*installed\* (not released)
6. gradle\_version=1.11
7. wget -N http://services.gradle.org/distributions/gradle-${gradle\_version}-all.zip
8. sudo unzip -foq gradle-${gradle\_version}-all.zip -d /opt/gradle
9. sudo ln -sfn gradle-${gradle\_version} /opt/gradle/latest
10. sudo printf "export GRADLE\_HOME=/opt/gradle/latest\nexport PATH=\$PATH:\$GRADLE\_HOME/bin" > /etc/profile.d/gradle.sh
11. . /etc/profile.d/gradle.sh
12. # check installation
13. gradle -v
14. Install Git

Try

sudo yum install git

Most likely this will not work, if so install from RPMForge repository as follows:

wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el5.rf.x86\_64.rpm

sudo rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt

sudo rpm -K rpmforge-release-0.5.2-2.el5.rf.\*.rpm

sudo rpm -i rpmforge-release-0.5.2-2.el5.rf.\*.rpm

sudo yum install git-1.7.11.3-1.el5.rfx.x86\_64

Test for Git installation using:

git --version

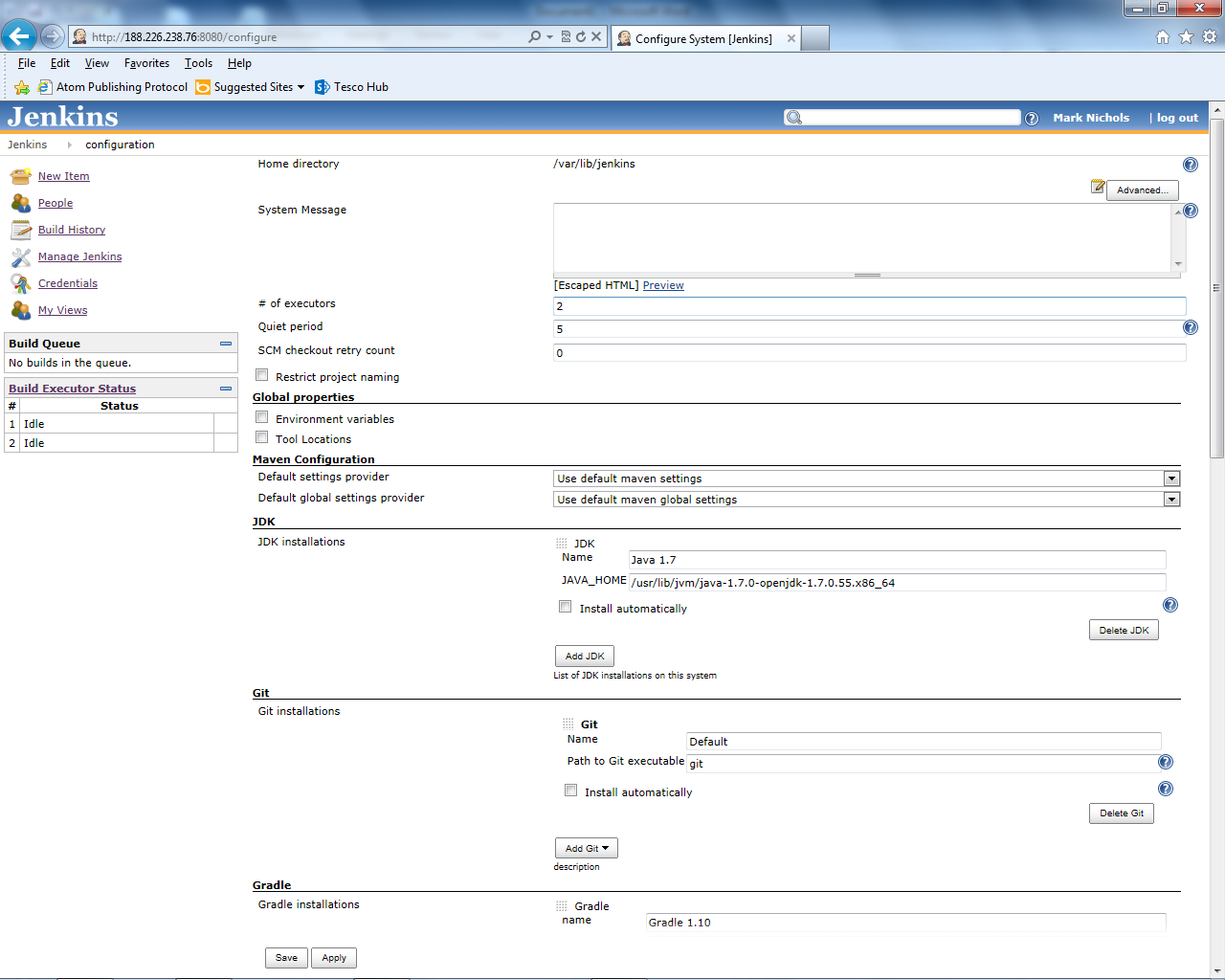
1. Install Jenkins

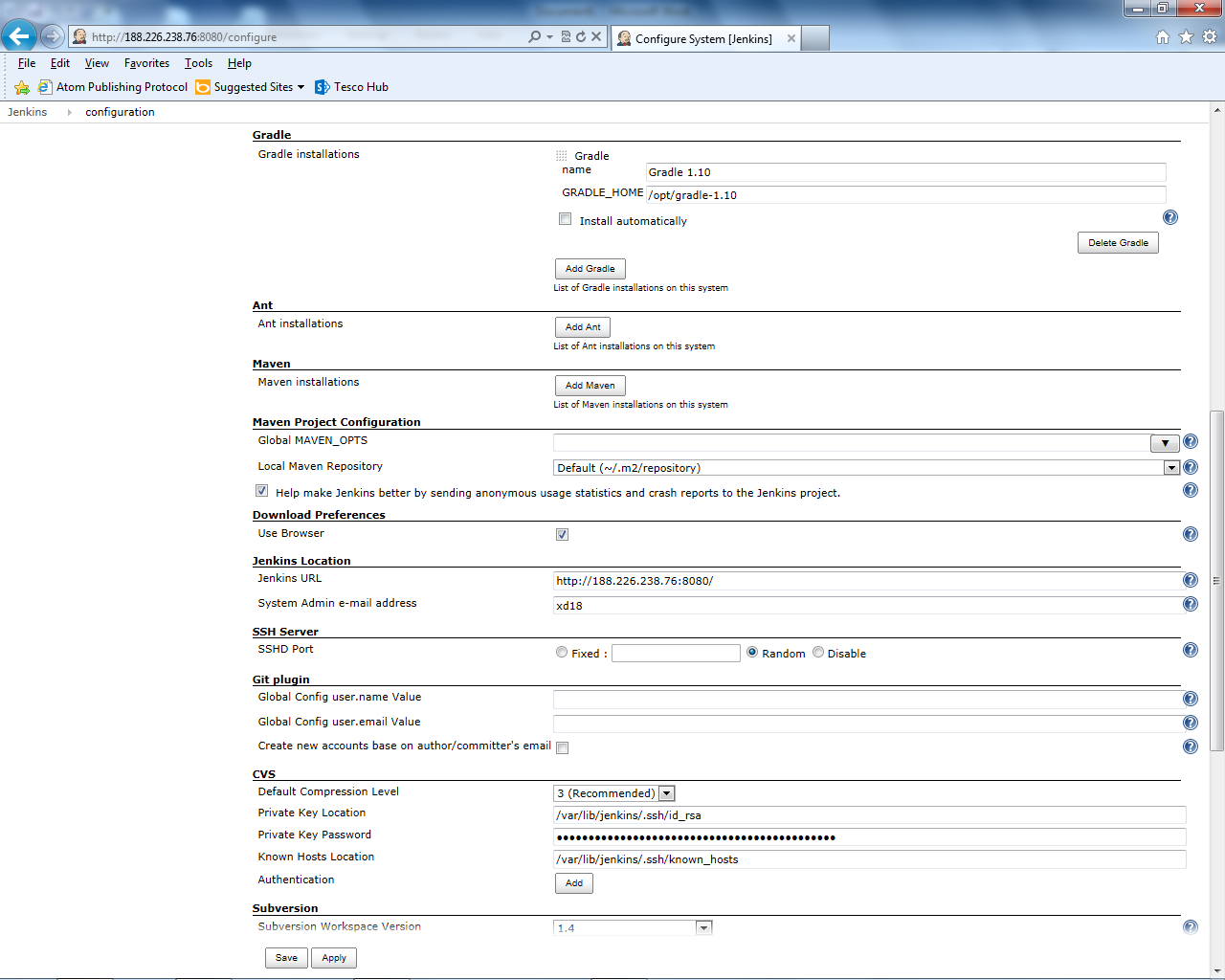
sudo wget -O /etc/yum.repos.d/jenkins.repo <http://pkg.jenkins-ci.org/redhat/jenkins.repo>

sudo rpm --import <http://pkg.jenkins-ci.org/redhat/jenkins-ci.org.key>

sudo yum install jenkins

1. Go to Jenkins login page and create admin user
2. Go to Manage Jenkins / Manage Plugins and add Github, Git, Github OAuth, and Gradle plugins (Adding Github and Git plugins will also add some other dependent Git plugins)
3. Configure JDK path and Gradle path in Jenkins like below





1. Create id\_rsa keys in Jenkins .ssh folder (with no passphrase) (jenkins user .ssh folder is in /var/lib/Jenkins).

Create keys as follows:

sudo su -s /bin/bash Jenkins

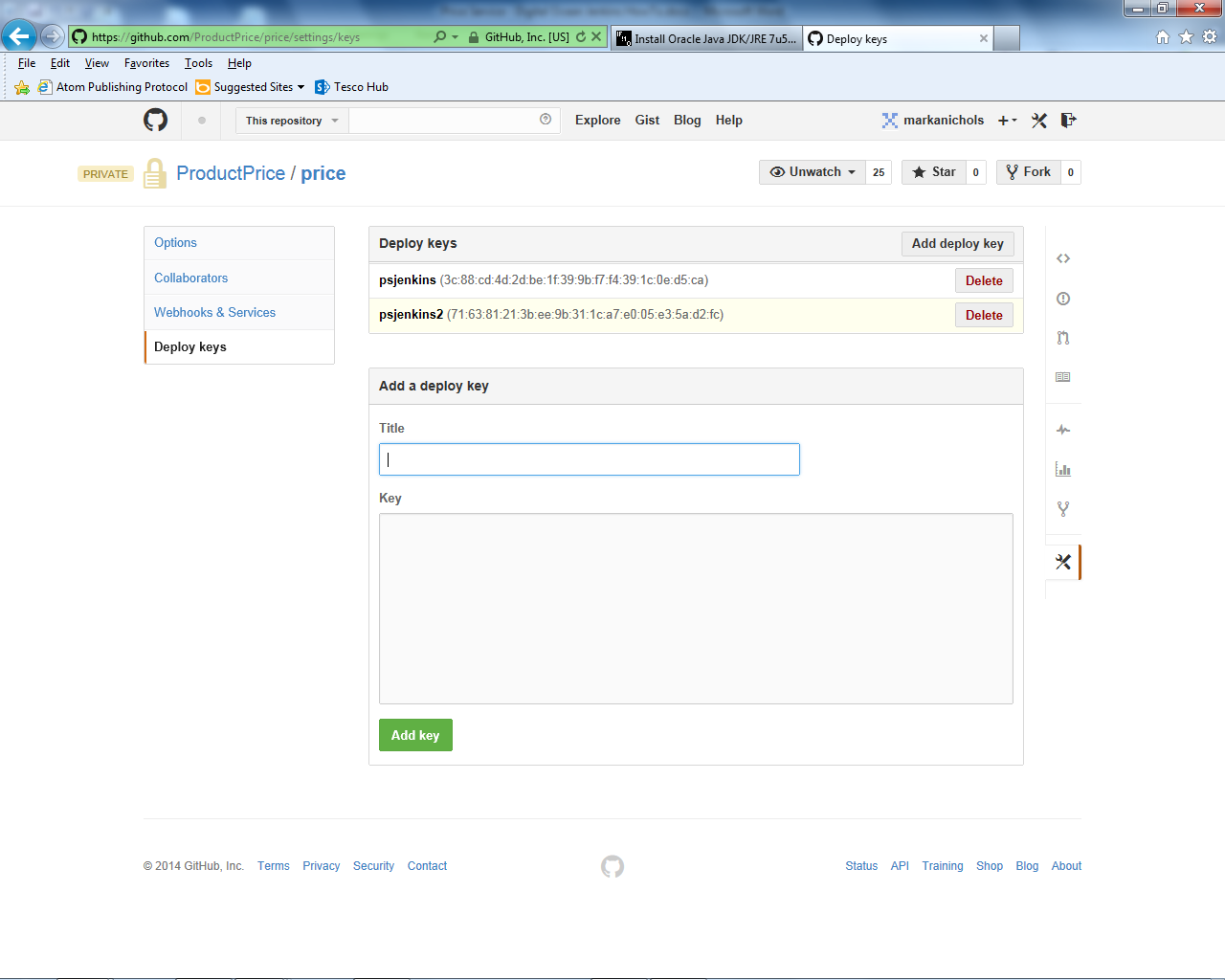
cd .ssh

ssh-keygen –t rsa

Accept the default filename id\_rsa, and do not enter a passphrase (just press enter when asked for passphrase, and again when asked to confirm).

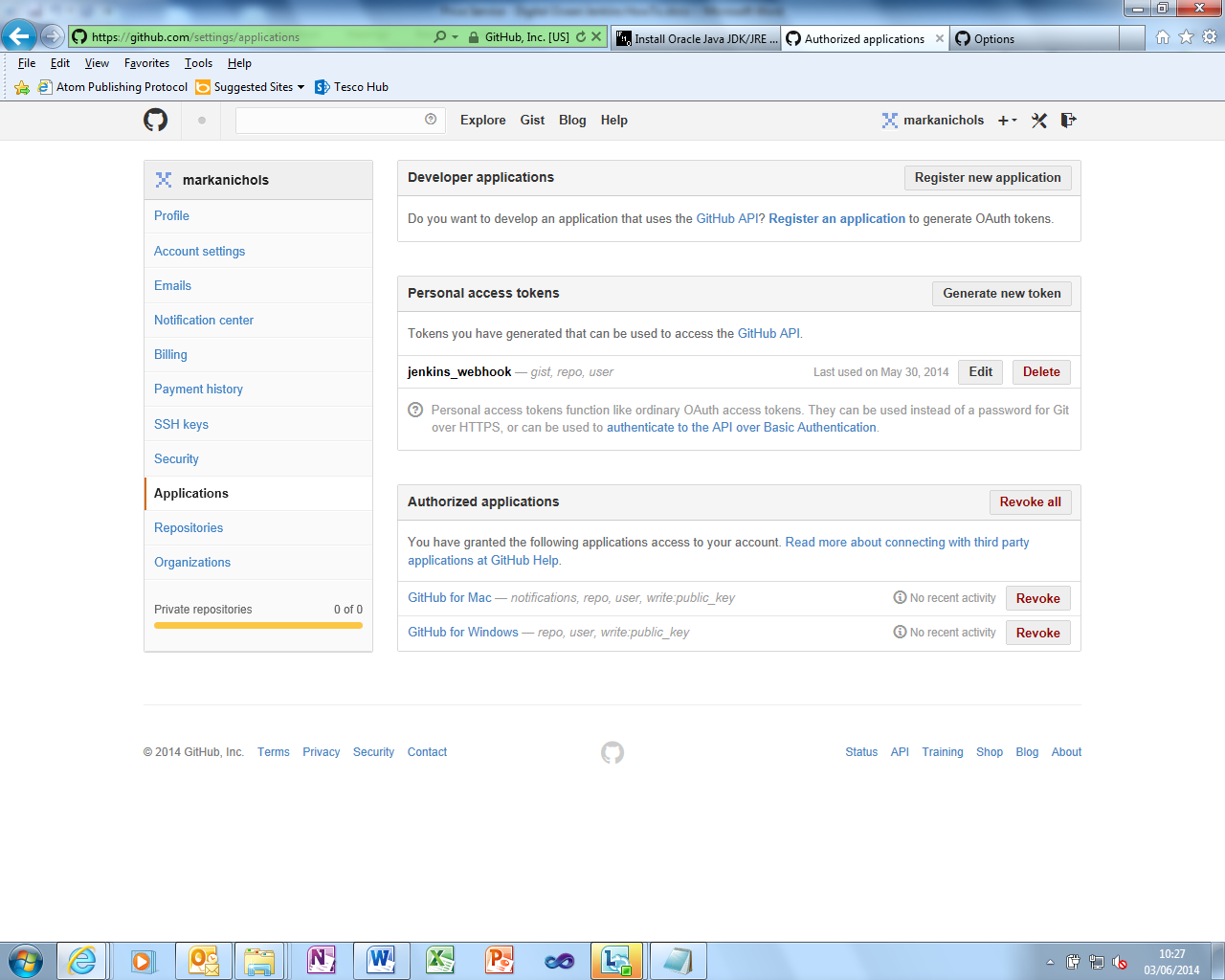
1. Set up id\_rsa key as Global Credential in Jenkins (Manage Jenkins/Manage Credentials)
2. Add id\_rsa.pub public key to keys in the Github repository

Copy the contents of file id\_rsa.pub created in step 21, and paste into new key at this screen in Github repository (accessed from Product/Price repository using Settings on rh side of screen, then Deploy Keys/Add deploy key)

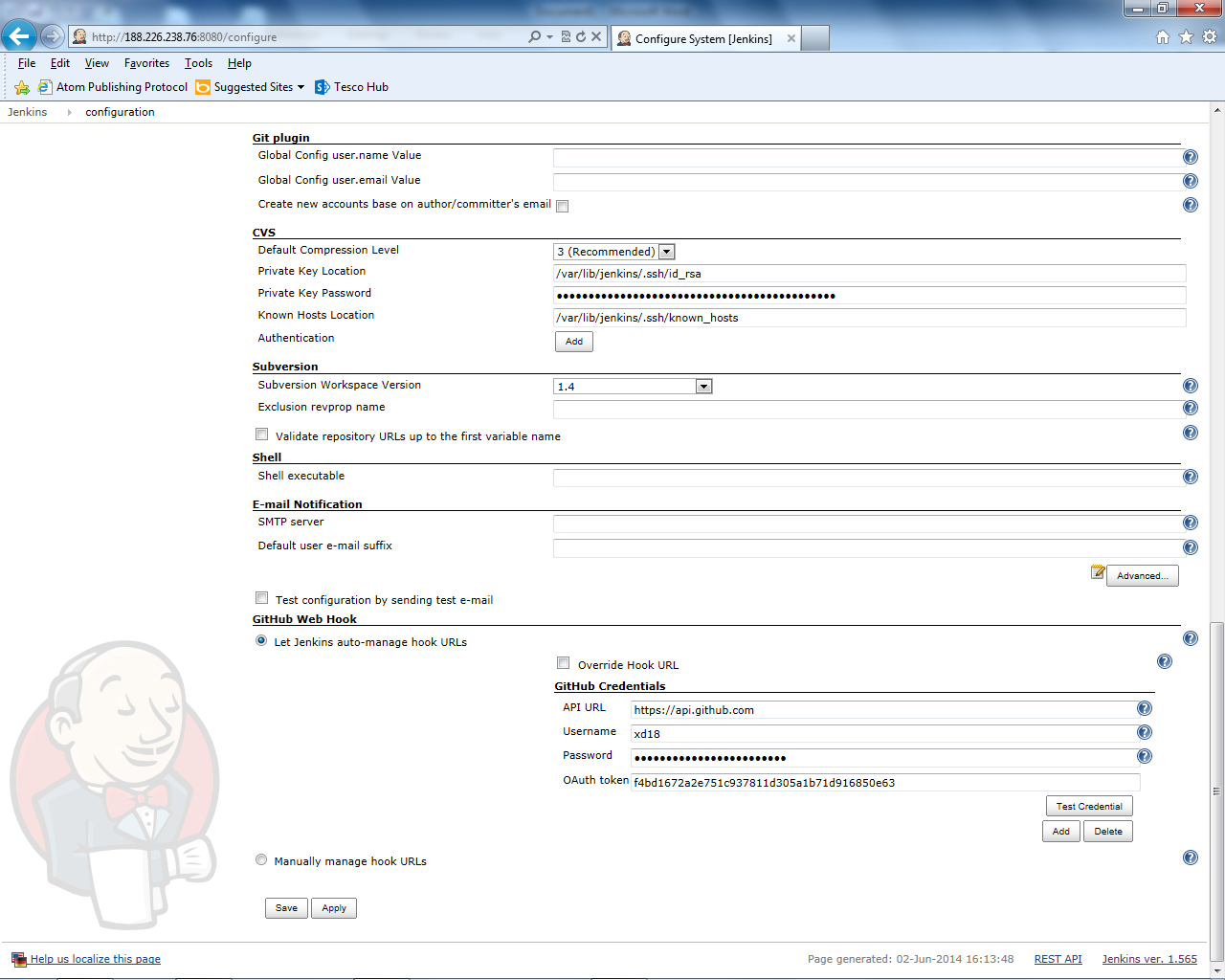


1. Create Oauth key for user in Github.

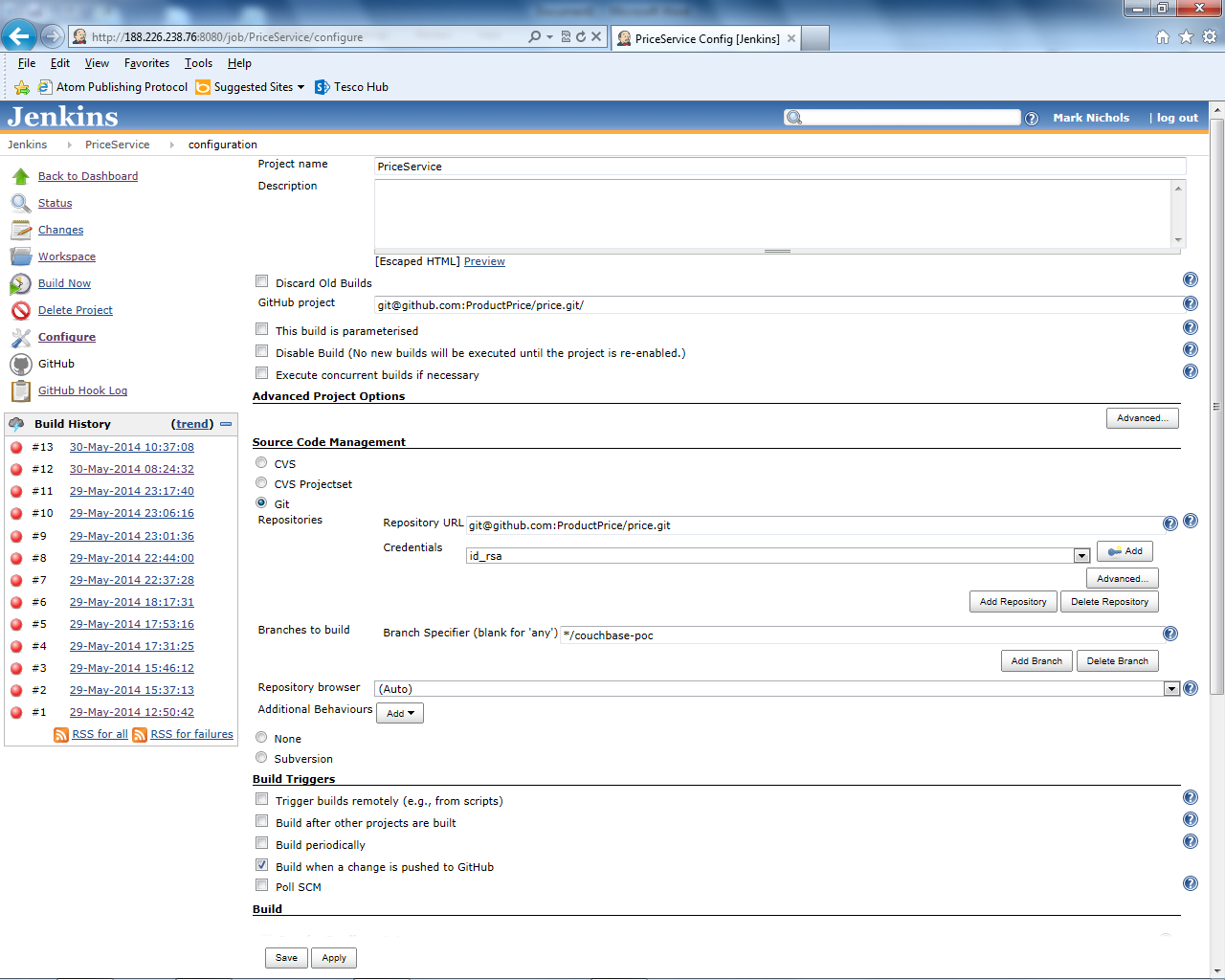
Set this up from the github user you plan to use, through user setting (top RH of screen), then Applications. You will see screen like below. Create a new Personal Access key, and copy the generated string.

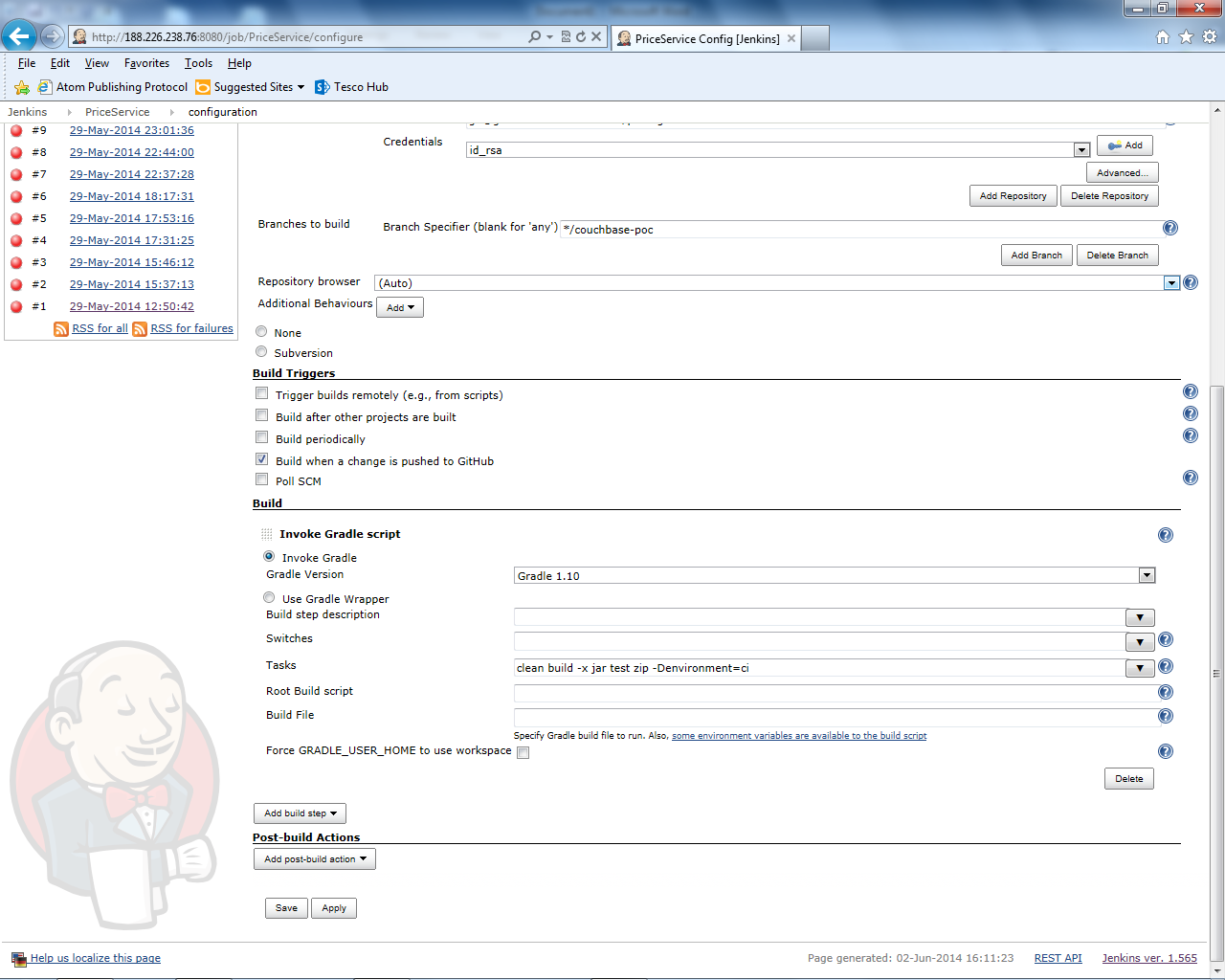


1. Add user name and Oauth keystring for login to Github in Jenkins configuration like below



1. Create build configuration in Jenkins like below





1. Run build once manually in Jenkins. This will automatically install hooks in Github which will trigger builds when you push to Githib. You can verify this has worked by checking in Github Repository Settings/Webhooks – you should a service created for Jenkins (Github Plugin) like below

