**DevOps Setup**

**# Installing VirtualBox and Vagrant**: (please install the latest version of Vagrant. And because we will be using VirtualBox as our provider for the getting started with DevOps, please install that as well.)

1. Download the latest stable version of vagrant from <https://www.vagrantup.com/downloads.html>
2. Download the latest stable version of VirtualBox from   
   <https://www.virtualbox.org/>
3. Install vagrant and VirtualBox
4. Open command prompt and type vagrant -v to ensure proper install.

**# Getting Vagrant up and running with Linux:**

1. Create a folder any directory for vagrant through CMD  
   eg script:  
   >**E:**  
   >mkdir **Vagrant**  
   >cd **Vagrant**  
   >mkdir **ProjDevOps**  
   >cd **ProjDevOps**
2. Initialise vagrant using command “vagrant init”
3. Go to **ProjDevOps** folder and open the vagrantfile using command “notepad Vagrantfile”
4. Replace the contents with text below

|  |
| --- |
| # -\*- mode: ruby -\*-  # vi: set ft=ruby :  # All Vagrant configuration is done below. The "2" in Vagrant.configure  # configures the configuration version (we support older styles for  # backwards compatibility). Please don't change it unless you know what  # you're doing.  Vagrant.configure("2") do |config|  config.vm.box = " ubuntu/trusty64"  config.vm.hostname = "server-central"  config.vm.network "forwarded\_port", guest: 8585, host: 7575  config.vm.network "forwarded\_port", guest: 5666, host: 5666  config.vm.network "forwarded\_port", guest: 80, host: 1234  config.vm.network "private\_network", ip: "192.168.33.10"  config.vm.provider "virtualbox" do |vb|  #vb.gui = true  vb.memory = "2048"  end  end |

1. Type vagrant up in the command line being inside the **ProjDevOps** folder.
2. Wait for the virtualBox to install.

**#Accessing Vagrant through third party SSH access (WINSCP , PUTTY & PUTTYGEN):**

1. Install **Putty(for SSH terminal access) and PUTTYGEN** from <http://www.putty.org/> and **winSCP(for SSH folder access)** from [https://**winscp**.net/eng/download.php](https://winscp.net/eng/download.php)
2. Open PUTTYGEN
3. Click load and go to c:/users/yoursystemnameorusername/vagrant.d/insecure\_private\_key to load a private key and save it inside your vagrant folder for Linux installation by clicking Save Private key.
4. Open PUTTY to setup it for vagrant.
5. Enter **127.0.0.1** in **host name** and **2222** as **port** which is the default SSH port for vagrant, type **vagrant** in **Saved** **session** andpress **Save**
6. InCategories goto **SSH -> AUTH->Browse E:\Vagrant\OMNILCS\.vagrant\machines\default\virtualbox\vagrant\_private\_key.ppk**to set the private key saved earlier.
7. Goto sessions and Click save again ->Load vagrant and Open. This opens a linux terminal for installation of all the needed softwares.

**#Installation of Apache Tomcat:**

|  |
| --- |
| The first thing you will want to do is update your apt-get package lists:   * sudo apt-get update   Now you are ready to install Tomcat. Run the following command to start the installation:   * sudo apt-get install tomcat7   Answer yes at the prompt to install tomcat. This will install Tomcat and its dependencies, such as Java, and it will also create the tomcat7 user. It also starts Tomcat with its default settings.  Let's make a quick change to the Java options that Tomcat uses when it starts. Open the Tomcat7 parameters file:   * sudo nano /etc/default/tomcat7   Find the JAVA\_OPTS line and replace it with the following. Feel free to change the Xmx and MaxPermSizevalues—these settings affect how much memory Tomcat will use:  /etc/default/tomcat7 — JAVA\_OPTS  JAVA\_OPTS="-Djava.security.egd=file:/dev/./urandom -Djava.awt.headless=true -Xmx512m -XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC"  Save and exit.  Now restart Tomcat with this command:   * sudo service tomcat7 restart   Tomcat is not completely set up yet, but you can access the default splash page by going to your domain or IP address followed by :8080 in a web browser:  Open in web browser:  http://server\_IP\_address:8080  You will see a splash page that says "It works!", in addition to other information. Now we will go deeper into the installation of Tomcat. Step Three - Installing Additional Packages Note: This section is not necessary if you are already familiar with Tomcat and you do not need to use the web management interface, documentation, or examples. If you are just getting into Tomcat for the first time, please continue.  With the following command, we will install the Tomcat online documentation, the web interface (manager webapp), and a few example webapps:   * sudo apt-get install tomcat7-docs tomcat7-admin tomcat7-examples   Answer yes at the prompt to install these packages. We will get into the usage and configuration of these tools in a later section. Next, we will install the Java Development Kit. Step Four - Install Java Development Kit (Optional) If you are planning on developing apps on this server, you will want to be sure to install the software in this section.  The Java Development Kit (JDK) enables us to develop Java applications to run in our Tomcat server. Running the following command will install openjdk-7-jdk:   * sudo apt-get install default-jdk   In addition to JDK, the Tomcat documentation suggests also installing Apache Ant, which is used to build Java applications, and a source control system, such as git. Let's install both of those with the following command:   * sudo apt-get install ant git   For more information about Apache Ant, refer to [the official manual](http://ant.apache.org/manual/index.html). For a tutorial on using git, refer to[DigitalCloud's Git Tutorial](https://www.digitalocean.com/community/articles/how-to-use-git-effectively). Step 5 - Configure Tomcat Web Management Interface In order to use the manager webapp installed in Step 3, we must add a login to our Tomcat server. We will do this by editing the tomcat-users.xml file:   * sudo nano /etc/tomcat7/tomcat-users.xml   This file is filled with comments which describe how to configure the file. You may want to delete all the comments between the following two lines, or you may leave them if you want to reference the examples:  tomcat-users.xml excerpt  <tomcat-users>  ...  </tomcat-users>  You will want to add a user who can access the manager-gui and admin-gui (the management interface that we installed in Step Three). You can do so by defining a user similar to the example below. Be sure to change the username and password to something secure:  tomcat-users.xml — Admin User  <tomcat-users>  <user username="admin" password="password" roles="manager-gui,admin-gui"/>  </tomcat-users>  Save and quit the tomcat-users.xml file. To put our changes into effect, restart the Tomcat service:   * sudo service tomcat7 restart  Step 6 - Access the Web Interface Now that we've configured an admin user, let's access the web management interface in a web browser:  Open in web browser:  http://server\_IP\_address:8080 |

**# Installing Oracle 11gR2 Express Edition:**

|  |
| --- |
| The final release version of Oracle 11gR2 Express Edition can be downloaded for free at<http://otn.oracle.com/database/express-edition/downloads>. The software should automatically downloaded into the "Downloads" folder of your home directory.   Enter the following commands to unpack the installer:  cd ~/Downloads  unzip oracle-xe-11.2.0-1.0.x86\_64.rpm.zip  rm oracle-xe-11.2.0-1.0.x86\_64.rpm.zip  The Debian Linux based package management of Ubuntu is not compatible with the Red Hat package manager. The Oracle installer needs to be converted using the following commands:  cd ~/Downloads/Disk1  sudo alien --to-deb --scripts oracle-xe-11.2.0-1.0.x86\_64.rpm  (This may take a few minutes)  rm oracle-xe-11.2.0-1.0.x86\_64.rpm  The following needs to be set for compatibility:  sudo ln -s /usr/bin/awk /bin/awk  sudo mkdir /var/lock/subsys  Ubuntu uses different tools to manage services and system startup scripts. The "chkconfig" tool required by the Oracle installer is not available in Ubuntu. The following will create a file to simulate the "chkconfig" tool.   Login as root:  sudo su -  Copy & paste the following **directly** into the command prompt to create a file:  cat > /sbin/chkconfig <<-EOF  #!/bin/bash  # Oracle 11gR2 XE installer chkconfig hack **for** Debian based Linux (by dude)  # Only run once.  echo "Simulating /sbin/chkconfig..."  **if** [[ ! \`tail -n1 /etc/init.d/oracle-xe | grep INIT\` ]]; then  cat >> /etc/init.d/oracle-xe <<-EOM  #  ### BEGIN INIT INFO  # Provides: OracleXE  # Required-Start: \\\$remote\_fs \\\$syslog  # Required-Stop: \\\$remote\_fs \\\$syslog  # Default-Start: 2 3 4 5  # Default-Stop: 0 1 6  # Short-Description: Oracle 11g Express Edition  ### END INIT INFO  EOM  fi  update-rc.d oracle-xe defaults 80 01  EOF  Exit root:  exit  Set execute privileges:  sudo chmod 755 /sbin/chkconfig  Install Oracle 11gR2 Express Edition entering the following commands:  cd ~/Downloads/Disk1  sudo dpkg --install ./oracle-xe\_11.2.0-2\_amd64.deb  (This may take a couple of minutes)  Run the configuration script to create (clone) the database and follow the screen. Accept the default answers, including "y" to startup the database automatically, or modify as required.   sudo /etc/init.d/oracle-xe configure  (This can take a few minutes - the installation completed successfully.)  To verify success, the procedure should end showing:  Starting Oracle Net Listener...Done  Configuring database...Done  Starting Oracle Database 11g Express Edition instance...Done  Installation completed successfully.  Set a password for the Oracle account:  sudo passwd oracle   9) Post-Installation In order to use sqlplus and other tools, the Oracle account requires specific environment variables. The following will set these variables automatically at every Oracle login:  Login as the Oracle user:  su - oracle  Copy the default account skeleton files and add the Oracle env script to .profile:  cp /etc/skel/.bash\_logout ./  cp /etc/skel/.bashrc ./  cp /etc/skel/.profile ./  echo "" >>./.profile  echo '. /u01/app/oracle/product/11.2.0/xe/bin/oracle\_env.sh' >>./.profile  By default, the Oracle Database XE graphical user interface is only available at the local server, but not remotely. The following will enable remote logins: Login as the Oracle user:  su - oracle  Login as SYSDBA and execute the following:  sqlplus / as sysdba  SQL> EXEC DBMS\_XDB.SETLISTENERLOCALACCESS(FALSE);  exit  See <http://download.oracle.com/docs/cd/E17781_01/admin.112/e18585/toc.htm> for more information. a) Unity desktop configurations The Oracle XE menu under the previous Gnome Classic desktop shows several useful scripts to backup the database, start and stop the database, etc. Under the Unity based desktop this menu is not available. You can either switch to the Gnome Classic desktop as outlined in chapter 2, or perform the following steps to modify and copy the scripts as outlined below. The start and stop database scripts will also be modified to perform a progress feedback.  Login as user root:  sudo su -  Convert desktop files:  cd /usr/share/applications  sed -i 's/Categories.\*/Categories=Database;Office;Development;/g' oraclexe\*  sed -i 's/MultipleArgs/X-MultipleArgs/g' oraclexe\*  sed -i 's/MimeType.\*/MimeType=application\/x-database/g' oraclexe\*  sed -i 's/.png//g' oraclexe\*  sed -i 's/Terminal=false/Terminal=true/g' oraclexe-startdb.desktop  sed -i 's/Terminal=false/Terminal=true/g' oraclexe-stopdb.desktop  Exit root:  exit  Login as user Oracle:  su - oracle  Modify database start and stop scripts:  cd /u01/app/oracle/product/11.2.0/xe/config/scripts  cp startdb.sh start.sh\_orig  cp stopdb.sh stopdb.sh\_orig  sed -i 's/>.\*//g' startdb.sh  sed -i 's/>.\*//g' stopdb.sh  You will need SYSDBA privileges and set Oracle environment variables in order to use your regular user account.   Login to your regular user account:  su - dude  Enter the folowing command:  sudo usermod -a -G dba dude  Then update your profile to automatically set the necessary Oracle environment variables:  echo "" >>./.profile  echo '. /u01/app/oracle/product/11.2.0/xe/bin/oracle\_env.sh' >>./.profile  Update your Desktop folder to contain useful Oracle XE scripts:  cp /usr/share/applications/oraclexe\* ~/Desktop  chmod 750 ~/Desktop/oraclexe\*  To verify success re-login and try "sqlplus":  su - oracle  sqlplus / as sysdba |

**#Installing Nagios in linux**

|  |
| --- |
| Step 1: Install Apache The Apache web server is currently the most popular web server in the world, which makes it a great default choice for hosting a website.  We can install Apache easily using Ubuntu's package manager, apt. A package manager allows us to install most software pain-free from a repository maintained by Ubuntu. You can learn more about [how to use apt](https://www.digitalocean.com/community/articles/how-to-manage-packages-in-ubuntu-and-debian-with-apt-get-apt-cache) here.  For our purposes, we can get started by typing these commands:  sudo apt-get update  sudo apt-get install apache2  Since we are using a sudo command, these operations get executed with root privileges. It will ask you for your regular user's password to verify your intentions.  Afterwards, your web server is installed.  You can do a spot check right away to verify that everything went as planned by visiting your server's public IP address in your web browser (see the note under the next heading to find out what your public IP address is if you do not have this information already):  http://your\_server\_IP\_address Install Nagios 4 This section will cover how to install Nagios 4 on your monitoring server. You only need to complete this section once. Create Nagios User and Group We must create a user and group that will run the Nagios process. Create a "nagios" user and "nagcmd" group, then add the user to the group with these commands:   * sudo useradd nagios * sudo groupadd nagcmd * sudo usermod -a -G nagcmd nagios  Install Build Dependencies Because we are building Nagios Core from source, we must install a few development libraries that will allow us to complete the build. While we're at it, we will also install apache2-utils, which will be used to set up the Nagios web interface.  First, update your apt-get package lists:   * sudo apt-get update   Then install the required packages:   * sudo apt-get install build-essential libgd2-xpm-dev openssl libssl-dev xinetd apache2-utils unzip   Let's install Nagios now. Install Nagios Core Download the source code for the latest stable release of Nagios Core. Go to the [Nagios downloads page](http://www.nagios.org/download/core-stay-informed), and click the **Skip to download** link below the form. Copy the link address for the latest stable release so you can download it to your Nagios server.  At the time of this writing, the latest stable release is Nagios 4.1.1. Download it to your home directory with curl:  cd ~  curl -L -O https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz  Extract the Nagios archive with this command:   * tar xvf nagios-\*.tar.gz   Then change to the extracted directory:   * cd nagios-\*   Before building Nagios, we must configure it. If you want to configure it to use postfix (which you can install with apt-get), add --with-mail=/usr/sbin/sendmail to the following command:   * ./configure --with-nagios-group=nagios --with-command-group=nagcmd   Now compile Nagios with this command:   * make all   Now we can run these make commands to install Nagios, init scripts, and sample configuration files:   * sudo make install * sudo make install-commandmode * sudo make install-init * sudo make install-config * sudo /usr/bin/install -c -m 644 sample-config/httpd.conf /etc/apache2/sites-available/nagios.conf   In order to issue external commands via the web interface to Nagios, we must add the web server user,www-data, to the nagcmd group:   * sudo usermod -G nagcmd www-data  Install Nagios Plugins Find the latest release of Nagios Plugins here: [Nagios Plugins Download](http://nagios-plugins.org/download/?C=M;O=D). Copy the link address for the latest version, and copy the link address so you can download it to your Nagios server.  At the time of this writing, the latest version is Nagios Plugins 2.1.1. Download it to your home directory with curl:  cd ~  curl -L -O http://nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz  Extract Nagios Plugins archive with this command:   * tar xvf nagios-plugins-\*.tar.gz   Then change to the extracted directory:   * cd nagios-plugins-\*   Before building Nagios Plugins, we must configure it. Use this command:   * ./configure --with-nagios-user=nagios --with-nagios-group=nagios --with-openssl   Now compile Nagios Plugins with this command:   * make   Then install it with this command:   * sudo make install  Install NRPE Find the source code for the latest stable release of NRPE at the [NRPE downloads page](http://sourceforge.net/projects/nagios/files/nrpe-2.x/). Download the latest version to your Nagios server.  At the time of this writing, the latest release is 2.15. Download it to your home directory with curl:   * cd ~ * curl -L -O http://downloads.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz   Extract the NRPE archive with this command:   * tar xvf nrpe-\*.tar.gz   Then change to the extracted directory:   * cd nrpe-\*   Configure NRPE with these commands:   * ./configure --enable-command-args --with-nagios-user=nagios --with-nagios-group=nagios --with-ssl=/usr/bin/openssl --with-ssl-lib=/usr/lib/x86\_64-linux-gnu   Now build and install NRPE and its xinetd startup script with these commands:   * make all * sudo make install * sudo make install-xinetd * sudo make install-daemon-config   Open the xinetd startup script in an editor:   * sudo vi /etc/xinetd.d/nrpe   Modify the only\_from line by adding the private IP address of the your Nagios server to the end (substitute in the actual IP address of your server):  only\_from = 127.0.0.1 10.132.224.168  Save and exit. Only the Nagios server will be allowed to communicate with NRPE.  Restart the xinetd service to start NRPE:   * sudo service xinetd restart   Now that Nagios 4 is installed, we need to configure it. Configure Nagios Now let's perform the initial Nagios configuration. You only need to perform this section once, on your Nagios server. Organize Nagios Configuration Open the main Nagios configuration file in your favorite text editor. We'll use vi to edit the file:  sudo vi /usr/local/nagios/etc/nagios.cfg  Now find an uncomment this line by deleting the #:  #cfg\_dir=/usr/local/nagios/etc/servers  Save and exit.  Now create the directory that will store the configuration file for each server that you will monitor:  sudo mkdir /usr/local/nagios/etc/servers Configure Nagios Contacts Open the Nagios contacts configuration in your favorite text editor. We'll use vi to edit the file:  sudo vi /usr/local/nagios/etc/objects/contacts.cfg  Find the email directive, and replace its value (the highlighted part) with your own email address:  email nagios@localhost ; <<\*\*\*\*\* CHANGE THIS TO YOUR EMAIL ADDRESS \*\*\*\*\*\*  Save and exit. Configure check\_nrpe Command Let's add a new command to our Nagios configuration:   * sudo vi /usr/local/nagios/etc/objects/commands.cfg   Add the following to the end of the file:  define command{  command\_name check\_nrpe  command\_line $USER1$/check\_nrpe -H $HOSTADDRESS$ -c $ARG1$  }  Save and exit. This allows you to use the check\_nrpe command in your Nagios service definitions. Configure Apache Enable the Apache rewrite and cgi modules:  sudo a2enmod rewrite  sudo a2enmod cgi  Use htpasswd to create an admin user, called "nagiosadmin", that can access the Nagios web interface:  sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin  Enter a password at the prompt. Remember this password, as you will need it to access the Nagios web interface.  **Note:** If you create a user that is not named "nagiosadmin", you will need to edit/usr/local/nagios/etc/cgi.cfg and change all the "nagiosadmin" references to the user you created.  Now create a symbolic link of nagios.conf to the sites-enabled directory:  sudo ln -s /etc/apache2/sites-available/nagios.conf /etc/apache2/sites-enabled/  Nagios is ready to be started. Let's do that, and restart Apache:  sudo service nagios start  sudo service apache2 restart  To enable Nagios to start on server boot, run this command:  sudo ln -s /etc/init.d/nagios /etc/rcS.d/S99nagios Optional: Restrict Access by IP Address If you want to restrict the IP addresses that can access the Nagios web interface, you will want to edit the Apache configuration file:  sudo vi /etc/apache2/sites-available/nagios.conf  Find and comment the following two lines by adding # symbols in front of them:  Order allow,deny  Allow from all  Then uncomment the following lines, by deleting the # symbols, and add the IP addresses or ranges (space delimited) that you want to allow to in the Allow from line:  # Order deny,allow  # Deny from all  # Allow from 127.0.0.1  As these lines will appear twice in the configuration file, so you will need to perform these steps once more.  Save and exit.  Now restart Apache to put the change into effect:  sudo service nagios restart  sudo service apache2 restart  Nagios is now running, so let's try and log in. Accessing the Nagios Web Interface Open your favorite web browser, and go to your Nagios server (substitute the IP address or hostname for the highlighted part):  http://nagios\_server\_public\_ip/nagios  Because we configured Apache to use htpasswd, you must enter the login credentials that you created earlier. We used "nagiosadmin" as the username |