Reynold Clinton

clintonr@meredith.com

Venus Framework Installation guide

Table of Contents

[Install Git for Mac 2](#_Toc98267930)

[SSH Creation 2](#_Toc98267931)

[Provide SSH Key to Bitbucket 2](#_Toc98267932)

[Install Amazon Corretto Java JDK 8 3](#_Toc98267933)

[Set the JAVA\_HOME variable 3](#_Toc98267934)

[Installing Maven manually 4](#_Toc98267935)

[Set the MAVEN\_HOME variable 4](#_Toc98267936)

[NVM Installation 4](#_Toc98267937)

[Yarn Installation 5](#_Toc98267938)

[Grunt Installation 5](#_Toc98267939)

[Clone Venus Project 5](#_Toc98267940)

[Download and Install IntelliJ IDE (CE Edition) 6](#_Toc98267941)

[Add settings.xml 6](#_Toc98267942)

# Install Git for Mac

* Paste the following command to install Homebrew. This will provide the functionality to install multiple applications just by using “brew” command

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

* Now Install Git using the command:

$ brew install git

* Install Git-GUI (Optional)

$ brew install git-gui

# SSH Creation

* Open terminal and navigate to SSH Directory

$ cd ~/.ssh

* Generate public/private rsa key pair using your Meredith email address that is used to set your Bitbucket account

$ ssh-keygen -t rsa -C "<your\_meredith\_email@meredith.com>"

* Enter the file to save the key, or press Enter for the default location. Default is .ssh/id\_rsa.
* Press enter twice when prompted, or supply a passphrase
* Verify identification and public key files were created like: .ssh/id\_rsa and .ssh/id\_rsa.pub just by entering $ ls command

Text

Description automatically generated

# Provide SSH Key to Bitbucket

* Login to <https://bitbucket.prod.aws.about.com/projects/AUT/repos/venus/browse>
* Navigate to Profile 🡪 Manage account 🡪 SSH Keys
* Click on “Add SSH Keys”
* To open the key file, switch back to terminal and enter the following using your own public key:

$ open -e .ssh/id\_rsa.pub

* Copy the entire key and paste it on the Bitbucket and click on “Add Key”
* The Key gets added to Bitbucket as shown below:

Graphical user interface, application

Description automatically generated

# Install Amazon Corretto Java JDK 8

* Download the Mac .pkg file from <https://docs.aws.amazon.com/corretto/latest/corretto-8-ug/downloads-list.html>

Text

Description automatically generated

* Double click the downloaded file to start the installation wizard. Follow the steps in the wizard.
* Once the wizard completes, Amazon Corretto 8 will be installed in /Library/Java/JavaVirtualMachines/.

You can run the following command in a terminal to get the complete installation path.

/usr/libexec/java\_home –verbose

# Set the JAVA\_HOME variable

* Open terminal and type:

nano ~/.zshenv

* Add the following environment variable to the end of the **~/.zshenv** file

export JAVA\_HOME=/Library/Java/JavaVirtualMachines/amazon-corretto-8.jdk/Contents/Home

* Save and exit nano (ctrl-x 🡪 enter option-Y to save the file 🡪 hit Enter)
* Source the file

source ~/.zshenv

* Enter the following command to print $JAVA\_HOME:

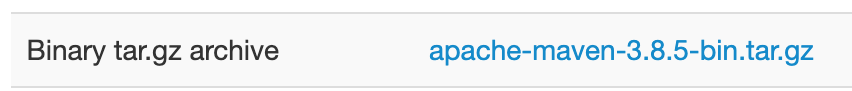
echo $JAVA\_HOME

* If we do see the below command, then we have successfully set your JAVA\_HOME environment variable

/Library/Java/JavaVirtualMachines/amazon-corretto-8.jdk/Contents/Home

# Installing Maven manually

* Download the file from <https://maven.apache.org/download.cgi>



* Type the following command to extract this Maven file. (First switch to the directory in the terminal where you have downloaded this file and then type the command):

tar xzvf apache-maven-3.8.5-bin.tar.gz

* Folder gets extracted to the default location if not move it to (/Users/{user}/ apache-maven-3.8.5)

# Set the MAVEN\_HOME variable

* Open terminal and type:

nano ~/.zshenv

* Add the following environment variable to the end of the **~/.zshenv** file

export MAVEN\_HOME=~/apache-maven-3.8.5

export PATH=$PATH:$MAVEN\_HOME/bin

* Save and exit nano (ctrl-x 🡪 enter option-Y to save the file 🡪 hit Enter)
* Source the file

source ~/.zshenv

* Enter the following command to print the $MAVEN\_HOME:

mvn -version

* If you do see the below command, then you have successfully set your MAVEN\_HOME environment variable

Text

Description automatically generated

# NVM Installation

* Enter any one of the following command in the terminal to install or update NVM.

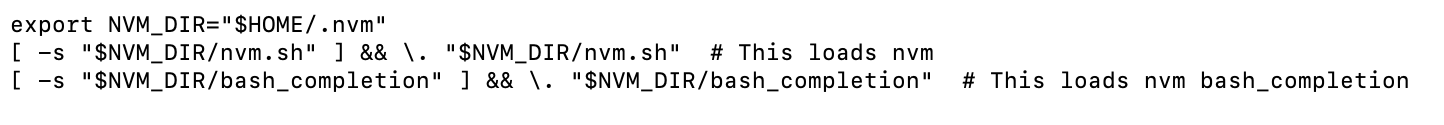
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash

or

wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash

* Running either of the above commands downloads a script and runs it. The script clones the nvm repository to ~/.nvm, and attempts to add the source lines to the correct profile file (~/.zshrc).
* You check any one of these files by again typing the command:

nano ~/.zshrc



* To verify that nvm has been installed, do:

command -v nvm

* Output will be – nvm
* Ensure that nvm was installed correctly with nvm --version, which should return the version of nvm installed.
* Install the version of Node.js you want:
  + Install the latest version with nvm install node
  + Use the latest version with nvm use node
  + Install the latest LTS version with nvm install --lts
  + Use the latest LTS version with nvm use --lts

# Yarn Installation

* Once we have nvm installed we can run the following both to install and upgrade Yarn:

npm install --global yarn

* Check that Yarn is installed by running:

yarn --version

# Grunt Installation

* When using nvm we do not need sudo to globally install a module with npm -g, so instead of doing sudo npm install -g grunt, do instead:

npm install -g grunt grunt-cli to install grunt on the current version of node.

# Clone Venus Project

* At first we must have an access to Dotdash Bitbucket repo <https://bitbucket.prod.aws.about.com/>
* Once the access is granted, go to this Bitbucket repo and click on “Clone”

Graphical user interface, text, application

Description automatically generated

* Copy the http URL <https://bitbucket.prod.aws.about.com/scm/aut/venus.git>
* In the terminal type the below command:

git clone https://bitbucket.prod.aws.about.com/scm/aut/venus.git

* Terminal will ask for the Bitbucket credentials
* If authorization is done, then it will download the Venus Project file in the default folder (/Users/{user}/Venus)

# Download and Install IntelliJ IDE (CE Edition)

* Open IntelliJ IDE and open this cloned Venus folder

File 🡪 Open 🡪 Venus

* Venus project will get imported into IntelliJ IDE

# Add settings.xml

* Right click on the Venus folder 🡪 maven 🡪 create ‘settings.xml’
* Append the following into this settings.xml file and Save the file

<mirrors>

<mirror>

<id>maven-public</id>

<name>About Maven Respository</name>

<url>http://nexus.prod.aws.about.com/repository/maven-public</url>

<mirrorOf>\*</mirrorOf>

</mirror>

</mirrors>

<profiles>

<profile>

<id>snapshots</id>

<repositories>

<repository>

<id>maven-snapshots</id>

<url>http://nexus.prod.aws.about.com/repository/maven-snapshots</url>

<releases><enabled>false</enabled></releases>

<snapshots><enabled>true</enabled></snapshots>

</repository>

</repositories>

</profile>

<profile>

<id>releases</id>

<repositories>

<repository>

<id>maven-releases</id>

<url>http://nexus.prod.aws.about.com/repository/maven-releases</url>

<releases><enabled>true</enabled></releases>

<snapshots><enabled>false</enabled></snapshots>

</repository>

</repositories>

</profile>

</profiles>

<activeProfiles>

<activeProfile>snapshots</activeProfile>

<activeProfile>releases</activeProfile>

</activeProfiles>

* Do a **mvn clean install** and Venus will have no errors there after.

Graphical user interface

Description automatically generated