

Having a local and isolated development sandbox environment is essential to practice DevOps automation and configuration. A local development sandbox will help you avoid some of these common DevOps problems.

- Stepping over each other in shared development environments
- Having different version of software or tools installed
- Taking too long to spin up and down virtual machines
- Having different configurations between development and production environments
- Not able to version control the DevOps tools and configurations.

With VirtualBox, Vagrant and Ansible, you can provision your local development environment the same way as production using the same software version and configurations.

Mac OS Environment

1. Launch the Terminal in the Applications/Utilities folder or on the Launchpad
2. Install the XCode command line tools

```
xcode-select --install
```

This will install the entire XCode development toolkit package to the following directory.

```
/Library/Developer/CommandLineTools/
```

3. Install Homebrew, a Mac OS X package manager.

```
/usr/bin/ruby -e "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

It will install the following script and directories:

```
==> This script will install:  
/usr/local/bin/brew  
/usr/local/share/doc/homebrew  
/usr/local/share/man/man1/brew.1  
/usr/local/share/zsh/site-functions/_brew  
/usr/local/etc/bash_completion.d/brew  
/usr/local/Homebrew  
==> The following new directories will be created:  
/usr/local/bin  
/usr/local/etc  
/usr/local/include  
/usr/local/lib  
/usr/local/sbin
```

```
/usr/local/share
/usr/local/var
/usr/local/opt
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var/homebrew
/usr/local/var/homebrew/linked
/usr/local/Cellar
/usr/local/Caskroom
/usr/local/Homebrew
/usr/local/Frameworks
```

4. Download the bootcamp project from GitHub to install the DevOps toolchain defined in the Brewfile. Assume the workspace directory is already created.

```
cd ~/workspace
git clone https://github.com/openecs/bootcamp.git
cd bootcamp
brew bundle
```

Note, for High Sierra 10.13 or above, VirtualBox installation will fail unless you allow loading new system extension(s) in System Preferences > Security & Privacy. Click the "Allow" button and reinstall with the following command. In this case you will need to manually re-install VirtualBox and Vagrant.

Note VirtualBox is not supported by Mac OS Mojave (10.14) Here are the possible workaround:

```
Try this first
sudo spctl --master-disable
sudo spctl --status
install virtualbox again
udo spctl --master-enable
```

I was able to get the kernel extensions to load the KEXTs without user approval by adding the VirtualBox Apple Developer Team ID in spctl kext-consent

I used the following command to get the Team ID:
CODE: SELECT ALL EXPAND VIEW
codesign -dv --verbose=4 /Applications/VirtualBox.app

Result:
CODE: SELECT ALL EXPAND VIEW
TeamIdentifier=VB5E2TV963

Turn on your Mac, then immediately press and hold Command-R to start up from macOS Recovery.
Select Disk Utility from the Utilities window, then click Continue.

```
From the Disk Utility sidebar, select the volume that you're using, then
choose File > Mount from the menu bar. (If the volume is already mounted,
this option is dimmed.)
Then enter your administrator password when prompted.
Quit Disk Utility.
Choose Terminal from the Utilities menu in the menu bar.
Type the command:
CODE: SELECT ALL    EXPAND VIEW
spctl kext-consent add VB5E2TV963
When done, choose Apple () menu > Restart.
```

```
brew cask install virtualbox
brew cask reinstall vagrant
```

5. With the toolchain installed, you can now provision your local development environment on VirtualBox.

```
cd ~/workspace/bootcamp
vagrant up
```

Windows Environment

1. Download and install Java 8

```
https://www.oracle.com/technetwork/java/javase/downloads/index.html
```

2. Install Gradle build tool for Java

```
https://docs.gradle.org/current/userguide/installation.html
```

3. Download and setup VirtualBox

```
https://www.virtualbox.org/wiki/Downloads
```

4. Install Git for Windows

```
https://git-scm.com/download/win
```

5. Install Vagrant

```
https://www.vagrantup.com/downloads.html
```

you must ensure that Hyper-V is not enabled on Windows. You can turn off the feature by running this Powershell command

```
Disable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V-All
```

6. Download the bootcamp project and provision the local development virtual machines (VMs). Open a Git for Windows command line window.

```
cd ~/workspace
git clone https://github.com/openecs/bootcamp.git
cd bootcamp
vagrant up
```

Connect to the Environment

You should see the three VMs, dev01, dev02, and ops01 created and running. To connect to a VM,

```
cd ~/workspace/bootcamp
vagrant ssh dev01
```

or

```
ssh vagrant@<vm name>
```

Destroy the Environment

You can destroy the whole environment, or just a VM

```
cd ~/workspace/bootcamp
vagrant destroy [<vm-name>]
```

Customize the Environment

You can customize your local development environment by changing the Vagrantfile provided in the downloaded repo. Then repeat setup step to spin up the environment again.

VM Lifecycle Management

```
vagrant up <vm name> --provision  
vagrant halt <vm name>  
vagrant suspend <vm name>  
vagrant destroy <vm name>
```