Deis Blog

12 NOV 2015 in **COREOS, VIRTUALBOX**

Get started with CoreOS and VirtualBox

Managing and scaling infrastructure is one of the biggest Enterprise concerns all over the world. The new wave of containerization led by Docker has made things a little easier. Applications can now be deployed on almost any system. You just need the appropriate Dockerfile and Docker installation.

But what if you have hundreds of containers and you need to scale them frequently? CoreOS (https://coreos.com/) helps you address this. Designed for security, consistency, and reliability, CoreOS is a Linux based operating system for clusters.

It is important to mention here that CoreOS doesn't ship a package manager. Rather, it comes with Docker pre-installed and CoreOS lets you use containers to manage your services instead of installing packages. So, for every service that you need (e.g. web server, database, cache, and so on) just create and use a container. CoreOS uses *systemd* (https://coreos.com/using-coreos/systemd/) and *fleet* (https://github.com/coreos/fleet) to manage the containers and connects the containers together using *etcd* (https://coreos.com/etcd/).

CoreOS is supported on almost any platform. That includes bare metal cloud providers like Amazon EC2, Azure, DigitalOcean, and Google Compute Engine. It also includes virtualization platforms like VMware, VirtualBox, and even cloud based operating systems like OpenStack.

In this post we will focus on installing CoreOS on your laptop (using VirtualBox) and we'll learn some CoreOS basics while we're at it. We'll look at systemd, fleet, and etcd in future posts.

Dependencies

VIRTUALBOX

One of the most well known virtualization tools, VirtualBox lets you install different operating systems on top of your current operating system. We will use VirtualBox to install CoreOS. If you don't have it installed already, download it from Virtualbox official website (https://www.virtualbox.org/wiki/Downloads) and follow the installation instructions.

COREOS ISO FILE

Download the CoreOS ISO file (https://coreos.com/os/docs/latest/booting-with-iso.html). We need it to mount on VirtualBox.

CLOUD-CONFIG FILE

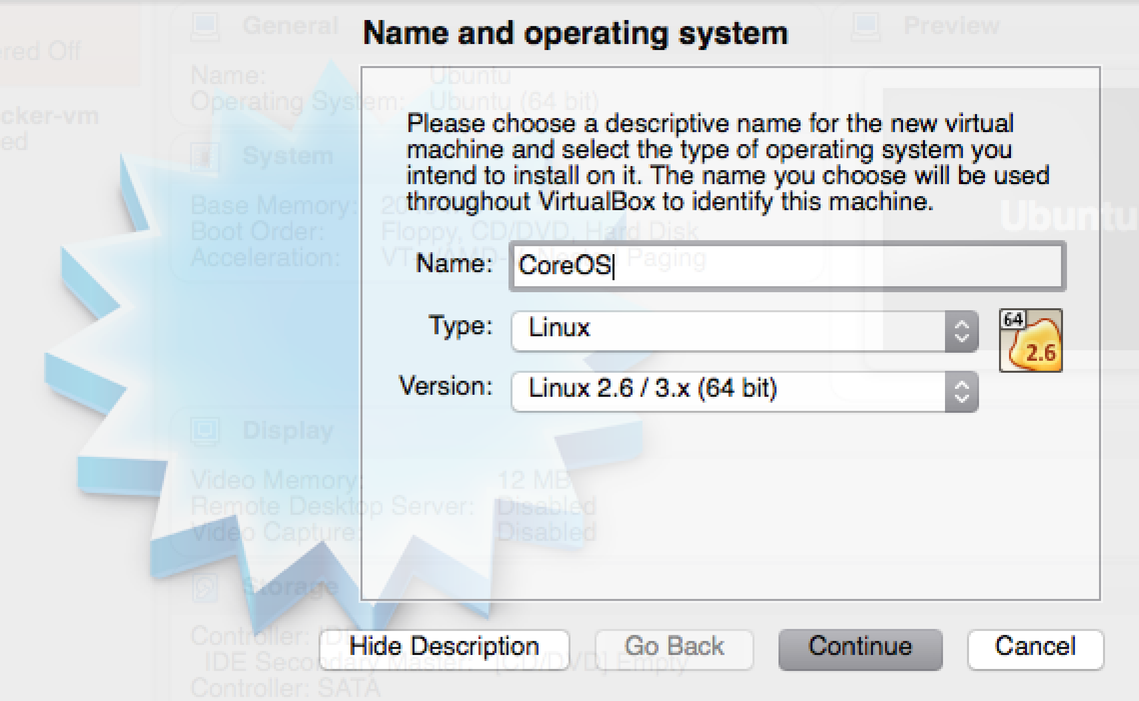
The cloud-config file (https://coreos.com/os/docs/latest/cloud-config.html) provides a way to customize various OS level items like user accounts, network configurations, systemd services, and so on. This file is processed every time the system boots up. We'll use this file to create the username and password during CoreOS installation. Note that the cloud-config file can only be created once we have live CoreOS running. So, we will create it during the installation process.

Installation

Follow these steps to install CoreOS on a VirtualBox virtual machine.

CREATE COREOS VIRTUAL MACHINE

Start VirtualBox on your local system.



Name and operating system

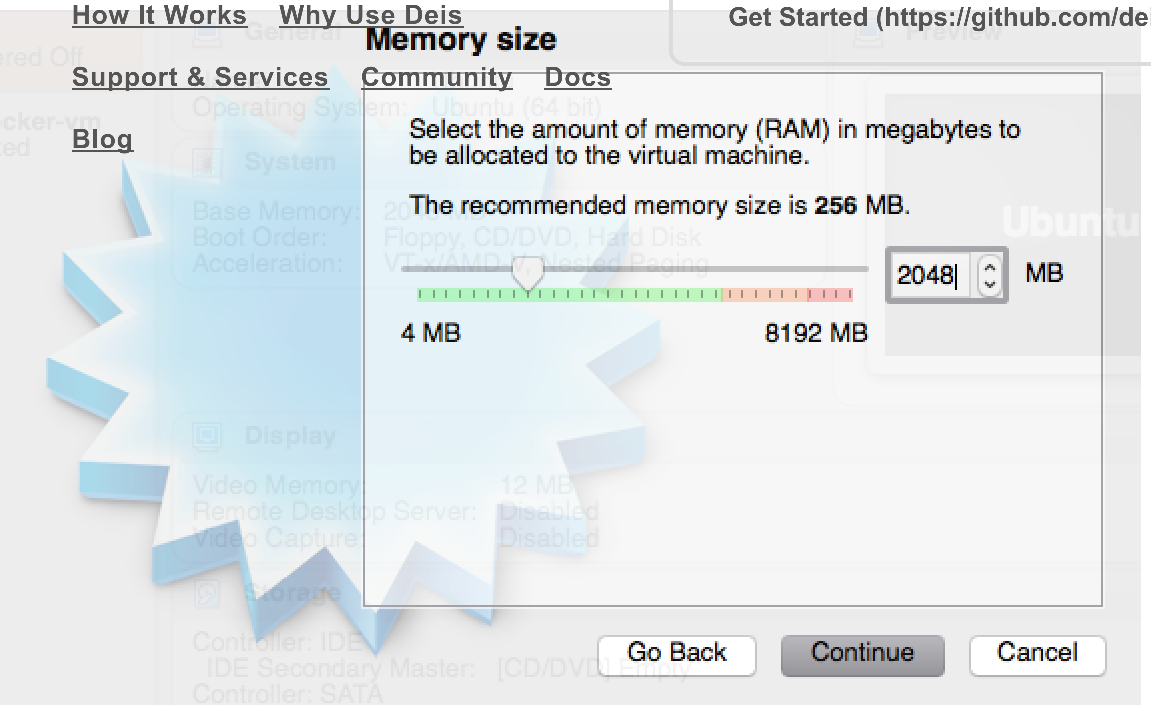
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: CoreOS

Type: Linux

Version: Linux 2.6 / 3.x (64 bit)

Click the *New* button to create a new virtual machine. Select the type as Linux an "Linux 2.6 / 3.x (64 Bit)".



Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **256** MB.

2048 MB

On the next page, set the memory size as 2 GB. Then set the hard drive file type as VDI (Virtual Disk Image) and the storage on physical hard drive as dynamically allocated.

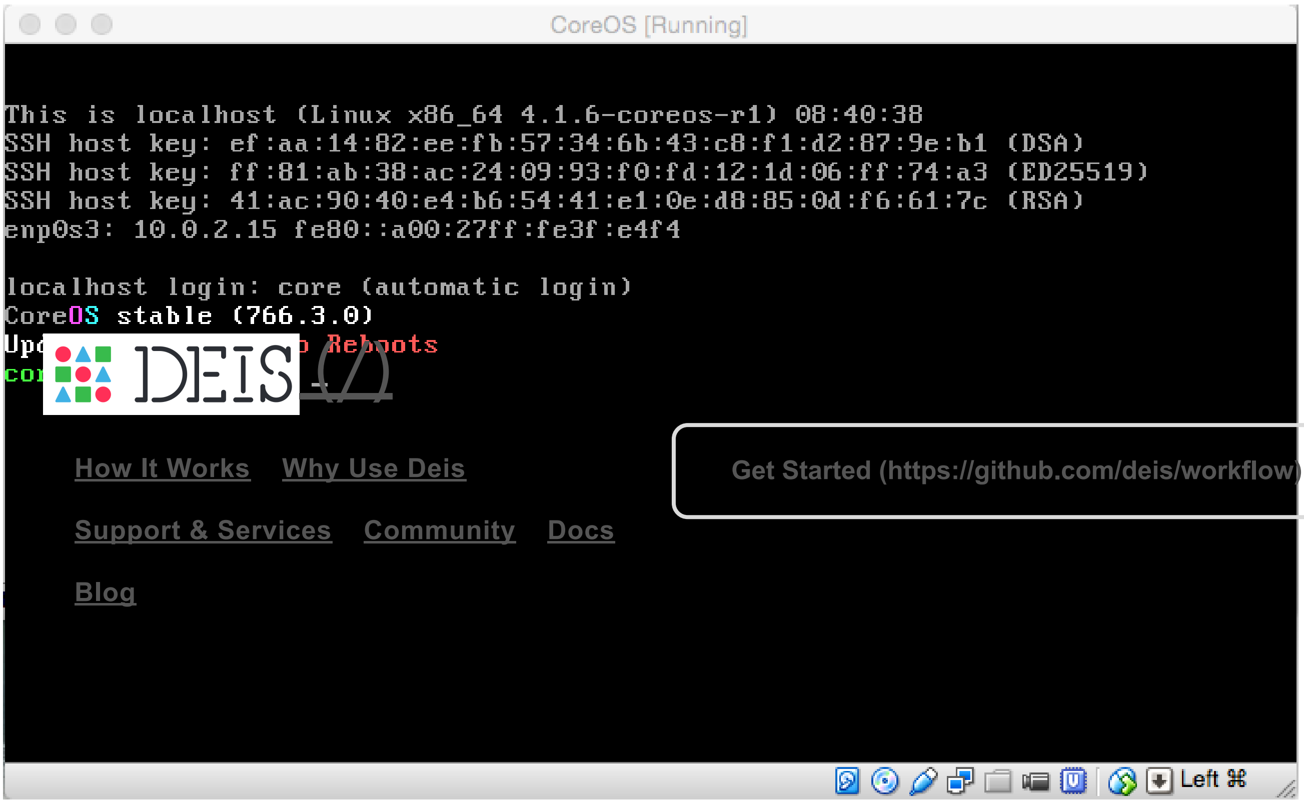
Finally, hit the *Create* button. This creates a virtual machine with the specifications we set.

MOUNT ISO

Next step is to mount the CoreOS ISO image we have.

Right click on the CoreOS virtual machine on the left hand panel of VirtualBox and select settings. Then click on the *Storage* tab, and associate the CoreOS ISO file to the IDE controller. Then boot up the CoreOS virtual machine.

You should see something like this:



```
This is localhost (Linux x86_64 4.1.6-coreos-r1) 08:40:38
SSH host key: ef:aa:14:82:ee:fb:57:34:6b:43:c8:f1:d2:87:9e:b1 (DSA)
SSH host key: ff:81:ab:38:ac:24:09:93:f0:fd:12:1d:06:ff:74:a3 (ED25519)
SSH host key: 41:ac:90:40:e4:b6:54:41:e1:0e:d8:85:0d:f6:61:7c (RSA)
enp0s3: 10.0.2.15 fe80::a00:27ff:fe3f:e4f4

localhost login: core (automatic login)
CoreOS stable (766.3.0)
Up              o Reboots
co
```

DEIS (/)

How It Works Why Use Deis

Support & Services Community Docs

Blog

Get Started (https://github.com/deis/workflow)

This means, right now, we have the live version of CoreOS running on VirtualBox, with the default `core` user logged in.

There are two important things to note here.

Firstly, we still haven't installed the CoreOS standalone version. What we have now is similar to a live CD of the OS running.

Secondly, we haven't still created a user to login to the OS. The `core` user automatically logs in for the first time, but you must create a user to login when CoreOS boots up the next time.

So, let's first create a user account and password, using the cloud-config file.

CREATE CLOUD-CONFIG FILE

The cloud-config file uses YAML. We need follow the file format (https://coreos.com/os/docs/latest/cloud-config.html#file-format) to make sure CoreOS recognizes and processes the file.

Our first step is creating the header. It should either be `#cloud-config` (for processing as cloud-config file) or `#!` (for processing as shell script).

The header should be followed by an associative array with zero or more of the following keys: `coreos`, `ssh_authorized_keys`, `hostname`, `users`, `write_files`, and `manage_etc_hosts`.

We'll focus on the `users` entry for now.

The `users` entry has three sub-fields: `name`, `passwd` and `groups`.

These fields are self-explanatory, except for one catch: the `passwd` field holds the password hash and not the actual password. This is good for security, but adds another step for creating the cloud-config file.

You need to create the hash manually, and add it to the cloud-config file.

One option is to use the `openssl` utility (available with CoreOS) to create the hash.

Just run:

```
sudo openssl passwd -1
```

The system prompts you for password and then asks to verify it. Once done successfully, you get the hash for your password. You can then add this hash to the cloud-config file.

With all the entries properly added to the cloud-config file, it should look like this:

```
$ cat cloud_config.yml
#cloud-config
users:
- name: nitish
  passwd: $1$yxV9YDKT$s.fAj5dlFyrPwrH0xAQJy/
  groups:
  - sudo
  - docker
```

You can even use above text as the cloud-config for your set-up. Just remember to change the `passwd` field to a value that corresponds with your password.

We are now ready to install CoreOS.

Install CoreOS

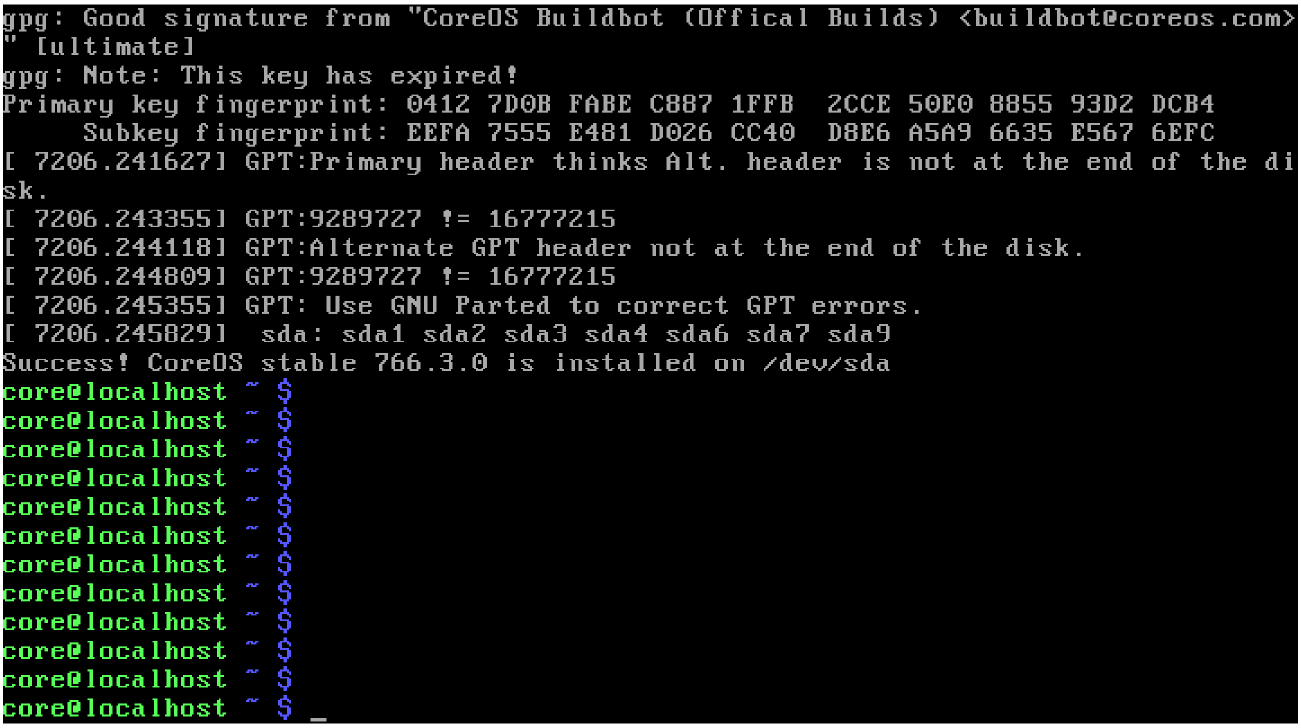
The final installation step is to download and install the CoreOS image on your VirtualBox.

To do this, run:

```
sudo coreos-install -d /dev/sda -C stable -c cloud_config.yml
```

This will download and install the latest stable CoreOS release on your virtual machine.

It should look a bit like this:



```
gpg: Good signature from "CoreOS Buildbot (Offical Builds) <buildbot@coreos.com>
" [ultimate]
gpg: Note: This key has expired!
Primary key fingerprint: 0412 7D0B FABE C887 1FFB  2CCE 50E0 8855 93D2 DCB4
     Subkey fingerprint: EEFA 7555 E481 D026 CC40  D8E6 A5A9 6635 E567 6EFC
[ 7206.241627] GPT:Primary header thinks Alt. header is not at the end of the di
sk.
[ 7206.243355] GPT:9289727 != 16777215
[ 7206.244118] GPT:Alternate GPT header not at the end of the disk.
[ 7206.244809] GPT:9289727 != 16777215
[ 7206.245355] GPT: Use GNU Parted to correct GPT errors.
[ 7206.245829]  sda: sda1 sda2 sda3 sda4 sda6 sda7 sda9
Success! CoreOS stable 766.3.0 is installed on /dev/sda
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $
core@localhost ~ $ _
```

The success message at the end confirms the CoreOS installation.

CROSS CHECK THE INSTALLATION

Unmount the ISO image from the IDE controller and reboot the virtual machine. If everything’s right, the virtual machine should reboot and you should be able to login using the username and password set in the cloud-config file.

Conclusion

CoreOS (https://coreos.com/) offers a new approach towards packaging and deployment of software: everything is a container. Not only does this make the OS relatively lightweight, it makes it easy to deploy and scale in the cloud.

In this article we learnt how to install CoreOS on a virtual machine. Though the official minimum cluster size is 3 nodes, we installed it on one virtual machine so you can learn the CoreOS basics.

Along with the installation process, we also covered the basics of the cloud-config file (https://coreos.com/os/docs/latest/cloud-config.html), and how it can be used to configure core functionality like users, networking, and so on.

In future articles we’ll take a look at how to deploy your software on CoreOS using Docker and CoreOS features like systemd, fleet, and etcd.

Nitish Tiwari

Nitish is a technical writer, software developer, and startup and open source enthusiast. Apart from serving as a community moderator for opensource.com, he also writes for FOSS based magazines.

@tiwari_nitish (https://twitter.com/@tiwari_nitish)

Subscribe to our newsletter

(https://twitter.com/opendeis) (http://stackoverflow.com/qu

(/)

Privacy Policy (/policies/privacy)

Deis.com and the Deis project are maintained by Engine Yard (https://engineyard.com/)

(https://engineyard.com/)

(https://engineyard.com/)

(https://engineyard.com/)

(https://engineyard.com/)

(https://engineyard.com/)

(https://engineyard.com/)

Deis Community

Get Involved (/community)

Videos & Events (/community#videos-events)

Project Roadmap (https://deis.com/docs/workflow/installing-workflow/)

Workflow Docs

Installing Workflow (https://deis.com/docs/workflow/installing-workflow/)

Architecture (https://deis.com/docs/workflow/understanding-workflow/architecture/)

Services

Support Plans (/services#support)

Pro Services (/services)

Company

About (/about)

Careers (/careers)

Contact (/about#contact)