

Getting Your Hands Dirty

So you think you're ready to start touching some code? We'll see about that...

Create an NCSA OpenSource Account

Click the **Sign Up** button at the top right corner of this page.

Walk through the steps to create an account on NCSA OpenSource.

You will use this account to log into the many Atlassian products (JIRA, Confluence, Bamboo, HipChat, etc).

Be sure to let your supervisor know once you have created your OpenSource account, so you can be granted the necessary / appropriate permissions.

Extra Credit: Sign Into HipChat

We use HipChat to facilitate communication here at NCSA.

Visit <https://www.hipchat.com/downloads> and download the appropriate HipChat client for your environment.

You should then be able to add your account to HipChat, sign in, and join the **NDS** room.

NOTE: You will need to specify a server of *hipchat.ncsa.illinois.edu* and OpenSource credentials to log in.

Create a GitHub Account

If you already have a GitHub account, you can skip this step.

Navigate to <https://github.com/> and create an account.

Be sure to let your supervisor know once you have created your GitHub account, so you can be added to the **nds-org** organization as a contributor.

Create a Docker Hub Account

If you already have a Docker Hub account, you can skip this step.

Navigate to

Be sure to let your supervisor know once you have created your Docker Hub account, so you can be added to the **ndslabs** organization as a contributor.

Enter: OpenStack (Nebula)

The next few steps will involve familiarizing yourself with Nebula, our on-site OpenStack implementation.

Retrieve Your Nebula Password

Specify your AD username, then execute the following command to retrieve your Nebula password:

```
export AD_USERNAME=  
sudo ssh ${AD_USERNAME}@public-linux.ncsa.illinois.edu cat /afs/ncsa/projects/openstack/etc/users/${AD_USERNAME}  
/password
```

Create an SSH Keypair

Log into [Nebula](#) and expand the **Compute** dropdown.

Select **Access & Security**, navigate to the **Key Pairs** tab, and click the **Create Key Pair** button.

Walk through the wizard to generate an SSH key pair.

Your browser should automatically download a private key (a **.pem** file).

Allocate a development VM

At the top, choose the project: **NDSLabsDev**.

Create a new Instance using the image entitled **Core899**.

Be sure to select a size (which is paradoxically entitled "flavor").

If you are unsure which to use, you likely should be using "large" or "medium".

Networking Gotchas

- **DO NOT FORGET** to associate an **SSH key** for this machine, or else you will never be able to get into it
- Remember to associate a **Floating IP** with this VM, so you can access it externally via a **141.142.xxx.xxx** IP
- Be sure to add **Permissions** to this VM:
 - remote SSH
 - remote HTTP

Enter: CoreOS

Now that you've got a development environment set up, you should be able to SSH into it using the .pem file you downloaded from Nebula:

```
sudo ssh -i path/to/<cert_name>.pem core@<vm_ip_address>
```

You have now entered your CoreOS Development environment! Congratulations on completing the first step!

Read more about CoreOS and what it has to offer here: <https://coreos.com/using-coreos/>

Enter: Git

Git is a commonly-used source control management system in modern software engineering.

If you are unfamiliar with Git SCM, you can learn more about the basics here: <http://rogerdudler.github.io/git-guide/>

Currently, we follow a "feature branch" workflow. For comparisons of this workflow to others, see below:

- <https://www.atlassian.com/git/tutorials/comparing-workflows/>
- <https://www.atlassian.com/pt/git/workflows>

CoreOS should include an installation of Git. Feel free to play around and become familiar with Git in this environment.

Enter: Docker

So you've got a shiny new VM and you know Git... I bet you sure think you're sooOOoo cool.

Well, try walking through this tutorial and learning a bit about Docker: <https://docs.docker.com/linux/>

Docker is a containerization software that has been blowing up in popularity in recent years.

Upon completion, you should be familiar with the basics of docker, including how to push and pull images from Docker Hub.

Enter: Kubernetes

Now that you've gotten the hang of Docker's basics, try to extend your knowledge to Kubernetes, which wraps around Docker to support clustering of containers.

Check out this quick start guide to learn the basics of how Kubernetes works: <http://kubernetes.io/v1.1/docs/getting-started-guides/docker.html>

Congratulations!

If you have made it through all of the tutorials without hurting yourself or others, then you're doing great! 😊

You should now have all of the information that you need to get started working on NDS Labs!

Check out the documentation for that project here: [NDS Labs](#)