

New Developer Workflow

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

Sign Up for Slack Account(s)

Most of the day-to-day communication happens via Slack.

Sign up for an account on <https://nationaldataservice.slack.com/messages>.

You can ask [Sara Lambert](#) for an invite if necessary.

Important Channels

- [#general](#) - please do not talk in general, to avoid bothering all users on the server
- [#random](#) - non work-related talk or potentially distracting content
- [#development](#) - general discussion towards NDS development
- [#nebula](#) - read-only forwarding from NCSA RocketChat about OpenStack Nebula
- [#notifications](#) - Update spam from GitHub / DockerHub / JIRA
- [#alerts](#) - NAGIOS alerts from production instances
- [#standup](#) - automated daily standup reports

Access OpenStack (aka 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 (for example, **key.pem**).

Allocate a development VM

Choose a project at the top-middle of the page (we tend to use NDS-hackathon, but your choice will depend on which project you can access).

Choose **Instances** from the left side navbar.

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

Be sure to select a flavor (aka "machine size").

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 from outside of Nebula via **141.142.xxx.xxx**
- Be sure to add **Permissions** to this VM:
 - remote SSH
 - remote HTTP
 - default

Create Necessary Accounts

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.

If you are an official member of NDS, 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 <https://hub.docker.com/>

If you are an official member of NDS, 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.

Introducing Leveraged Technologies

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/>

GitHub has even released a Desktop client!

Going forward, we will try to follow a "forking" 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.

For more about using Git to contribute to NDS, see [Git Workflows](#).

Enter: Docker

So you've got a shiny new CoreOS VM and you've figured out Git... let's see what it can do!

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.

For more about using Docker to contribute to NDS, see [Docker Workflows](#).

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/docs/getting-started-guides/docker/>

For more about using Kubernetes, see [Kubernetes Workflows](#).

Next Steps

Now that you're familiar with our technology stack, you might be interested in taking a look at some projects that utilize these technologies:

- <https://github.com/nds-org/ndslabs-clowder>
- <https://github.com/nds-org/ndslabs-dataverse>
- <https://github.com/nds-org/ndslabs-irods>

These projects build up the Docker images deployed by NDSLabs.

The NDS Labs spec for each service mimics that of the "Kubernetes Spec" and references the images built by the repos above:

- NDS Labs Service Specs: <https://github.com/nds-org/ndslabs-specs>
- The Kubernetes Spec: <http://kubernetes.io/docs/user-guide/pods/multi-container/#the-spec-schema>

These specs are then fed by the NDS Labs API Server into the etcd instance that is run by Kubernetes.

The API Server hosts a REST API that allows the AngularJS GUI to grab the specs and walk the user through a [Stack's](#) configuration using the details contained within its spec.

The code for the API / GUI can be found here: <https://github.com/nds-org/ndslabs>

Further documentation for the NDS Labs initiative can be found here:

- [WordPress: NDS Labs](#)
- [Official Confluence: NDS Labs](#)
- [OpenSource Confluence: NDS Labs](#)
- [GitHub: NDS Labs](#)

Congratulations!

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

Extra Credit

Useful Tutorials / Discussions:

Here are a few useful talks regarding the technologies utilized by NDS Labs.

Feel free to watch as many as you'd like, and / or suggest more in the comments of this page!

- [Jessie Frazelle's Whacky Docker Gymnastics](#)
- [Jerome P on Docker for devs](#)
- [Kelsey Hightower's series on Docker/Core/K8](#)
- [Designer of K8 great architectural overview](#)