

JupyterHub using Docker and Flocker on Openstack via Ansible and Vagrant

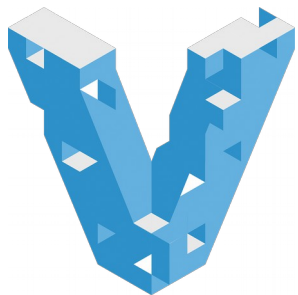
Ian Allison, James Colliander, Michael Lamoureux



Pacific Institute *for the*
Mathematical Sciences



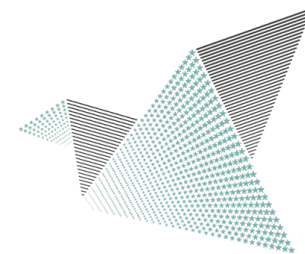
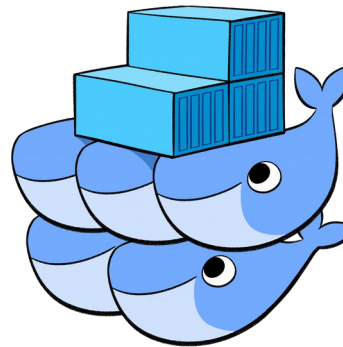
compute | **calcul**
canada | canada



VAGRANT



ANSIBLE



Flocker™
by ClusterHQ™

Who are we?

- Pacific Institute for the *Mathematical Sciences*
 - *10 sites at major western universities in Canada (+UW)*
 - *Broad mandate of promoting research and applications of the mathematical sciences*
 - *New director (James Colliander) has strong interests in Learning technologies*
 - *We want to make Jupyter notebooks available to all Canadian Researchers*
- Compute Canada
 - *Canada's national level HPC partner for resources and support*
 - *Providing us with resources (OpenStack)*

We've just signed an MOU, now we just have to actually build something...

What are we doing?

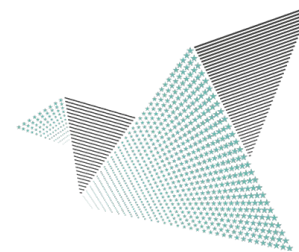
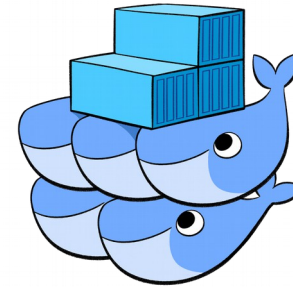
- At the moment, very traditional JupyterHub stuff
 - *Classes in Math, CS, Data Science*
 - *Casual use by researchers who show interest*
 - *Summer Students doing really cool things*
 - *Scoping out research focused hubs*
- Typical instances
 - *16 core / 64Gb VM*
 - *ZFS/NFS for storage*
 - *Docker spawner*
 - *Systemuser authenticator via LDAP at UBC (one of PIMS sites)*

What do we want to do?

- We want to offer similar services to orders of magnitudes more researchers
 - *Federated identity - Shibboleth for authentication*
 - *Also more research focused machines (HPC) for various groups.*
- At the moment we're experimenting with a larger single installation of JupyterHub
 - *How far can we push a single hub and where are the bottlenecks?*
 - *We're trying out Swarm and Flocker for this*
- Also looking at automated deployments for researchers - Hubs on demand

Development System

Using an OpenStack deployment from Compute Canada, we're looking at scaling



Flocker[™]
by ClusterHQ[™]

EBS, **Cinder**, GCE,
EMC...

Development System

We want to do this in a completely automated fashion

- **Vagrant** - “Reproducible development environments”
 - *Allows us to define and instantiate resources*
 - **Providers:** *OpenStack, Rackspace, AWS, ...*
 - **Provisioners:** *sh, ansible*
- **Ansible** - “Simple IT Automation”
 - *Can take generic image and build up entire infrastructure for Docker Swarm in minutes*
 - **Consul** - *distributed K/V store*
 - **Docker swarm** - *API similarities make transition simple*
 - *swarm nodes can be provisioned as needed (and automated)*
 - **Flocker** - “container data volume manage”
 - *maps common backends (EBS, GCE, Cinder) to containers*

vagrant up

Automatically:

- Provision volumes for e.g. docker devicemapper
- Defines all of the machines involved (currently 7)
- Brings them up and configures network, (ansible) groups etc.
- Triggers ansible provisioner to start configuring them

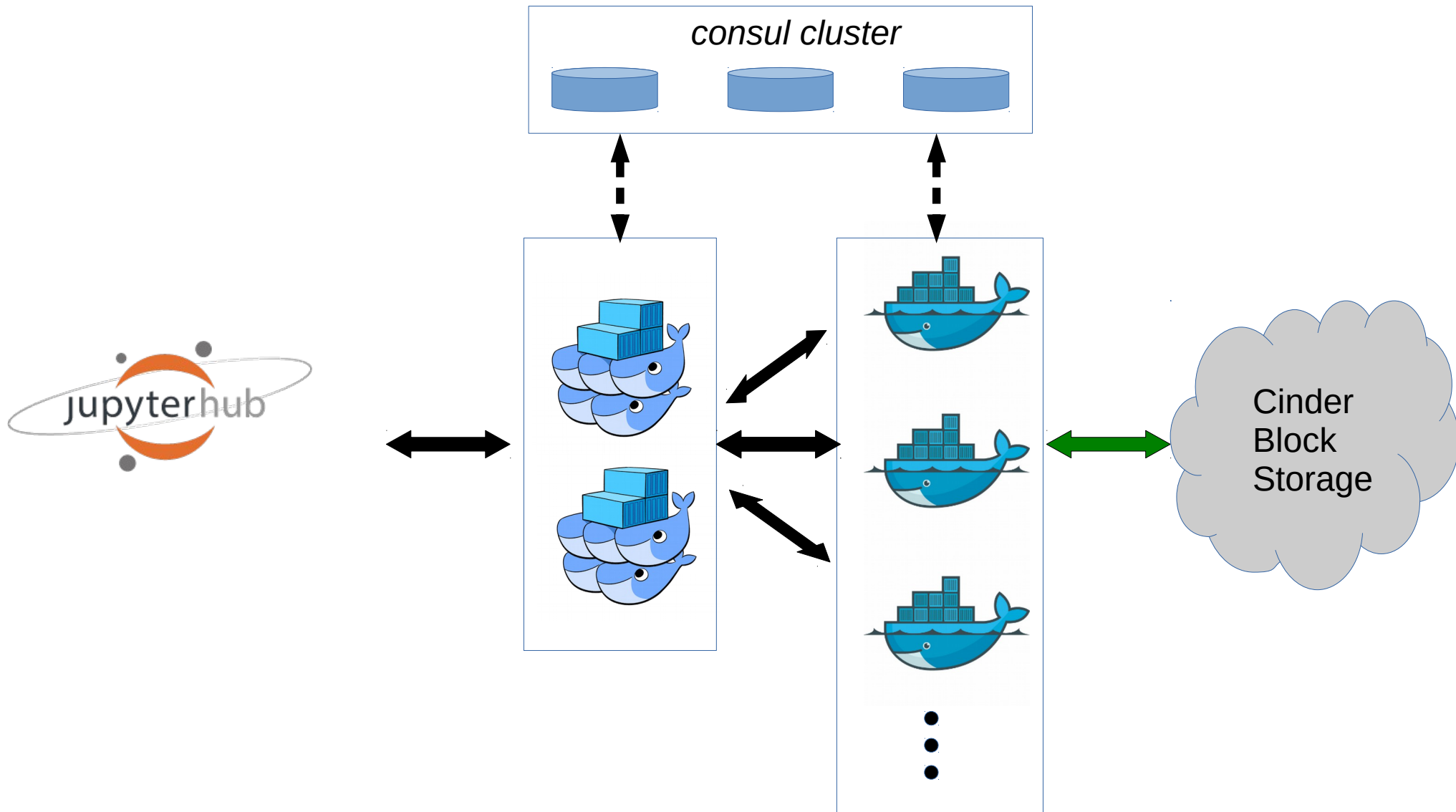
```
:playbook => "#{dir}/playbooks/docker/main.yml",
:security_groups => ["general"],
:groups => "docker,docker-swarm-node",
:docker_volume => 'docker-n1-lvm' },
{ :name => "docker-n2",
  :ip => "192.168.82.42",
  :playbook => "#{dir}/playbooks/docker/main.yml",
  :security_groups => ["general"],
  :groups => "docker,docker-swarm-node",
  :docker_volume => 'docker-n2-lvm' }
]
boxes.each_with_index do |opts, index|
  config.vm.define opts[:name] do |box|
    #box.vm.hostname = opts[:hostname]
    box.vm.provider "openstack" do |os|
      os.server_name = opts[:hostname]
      os.security_groups = opts[:security_groups]
      os.metadata = {'groups' => opts[:groups]}
      os.networks = [
        {
          id: '04dd241e-a28f-4448-989e-dc6be6a2e4b9',
```

Ansible

- Runs a provider dependent playbook (OpenStack)
- Take care of housekeeping tasks
 - Create a CA, issue certificates,
- Bring up a consul cluster (K/V store for swarm)
- Bring up docker swarm managers and nodes
- Bring up flocker agents and control nodes
- *... this is where we are ...*
- **Bring up a JupyterHub instance**

A lot of the hard work for the next steps has already been done:
[jupyterhub/jupyterhub-deploy-teaching](https://github.com/jupyterhub/jupyterhub-deploy-teaching)

Development Architecture



Development Architecture

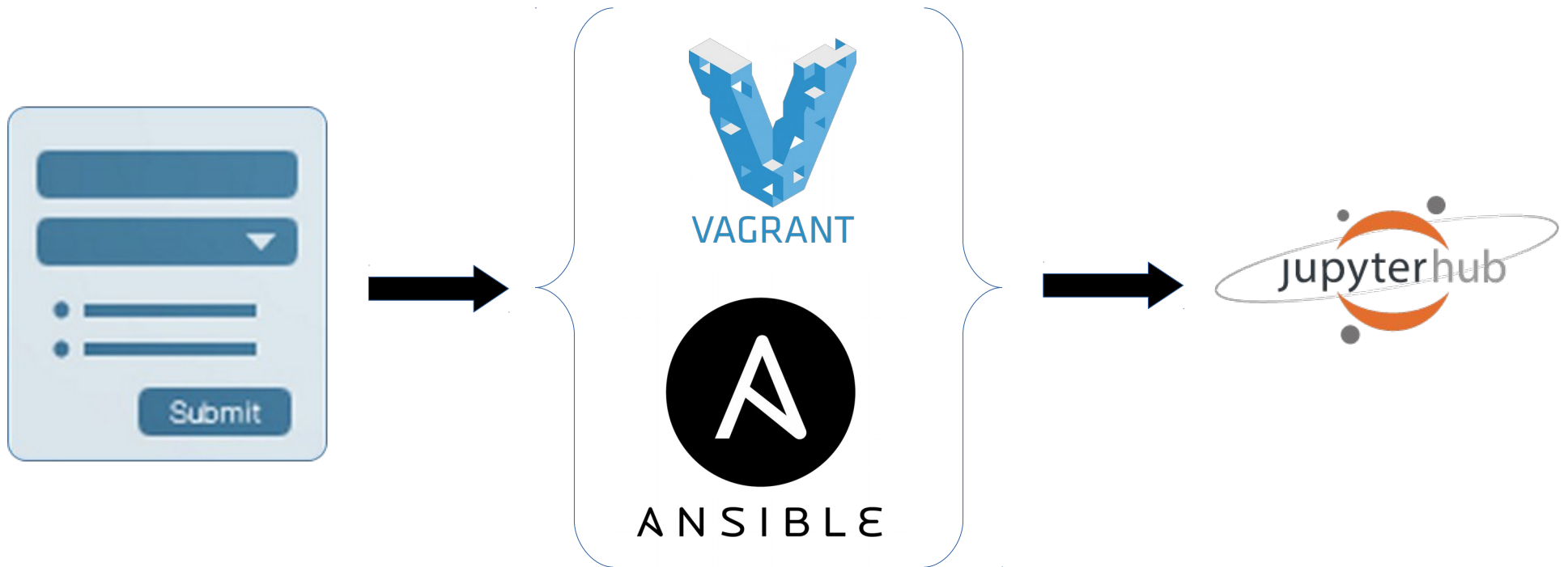
That should give us a nice testbed for pushing our larger installs, but we also have a bunch of open questions which we are looking at...

- We're looking at Shibboleth for authentication (we need to cross institution boundaries)
 - *We did the “proxy your proxy” thing for a while, but will probably want to do it natively*
- Will probably flatten file ownership for simplicity, so we're looking at securing docker e.g. volume labeling with SELinux alongside Flocker

We recognize that we're not going to build one massive JupyterHub to solve all of our needs, but we're learning useful automation tricks...

Automated Deployments

Honestly though, a lot of this is just learning the automation tools so that we can ultimately deploy “plain vanilla JupyterHubs”



Automated Deployments

- We'd like to make it trivial for our researchers to create hubs to suit their needs
 - *Vanilla instances similar to what we have*
 - *Larger installations for research groups*
 - *Make it provider agnostic (AWS, GCE, ...)*
- We also have an focus on training
 - *Part of our mandate*
 - *Make these tools and techniques part of the everyday vocabulary of our researchers*

Contact Us

We're very eager to learn of people doing similar things or wanting to work alongside us

- director@pims.math.ca
- mikel@pims.math.ca
- iana@pims.math.ca

<https://github.com/pimsmath>