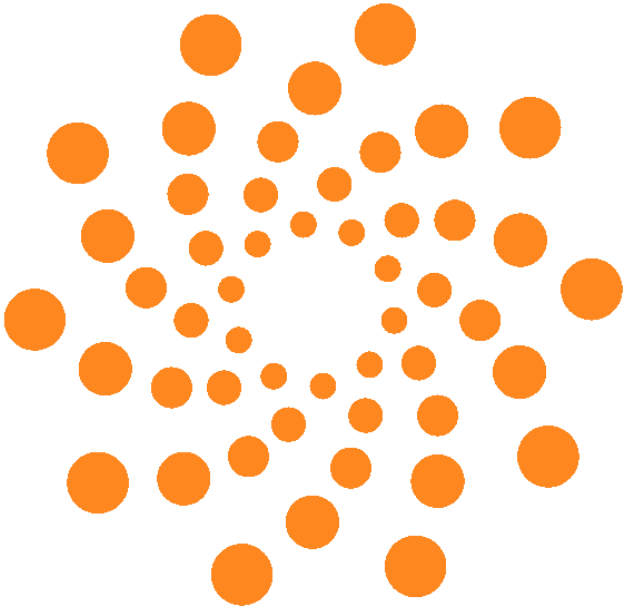
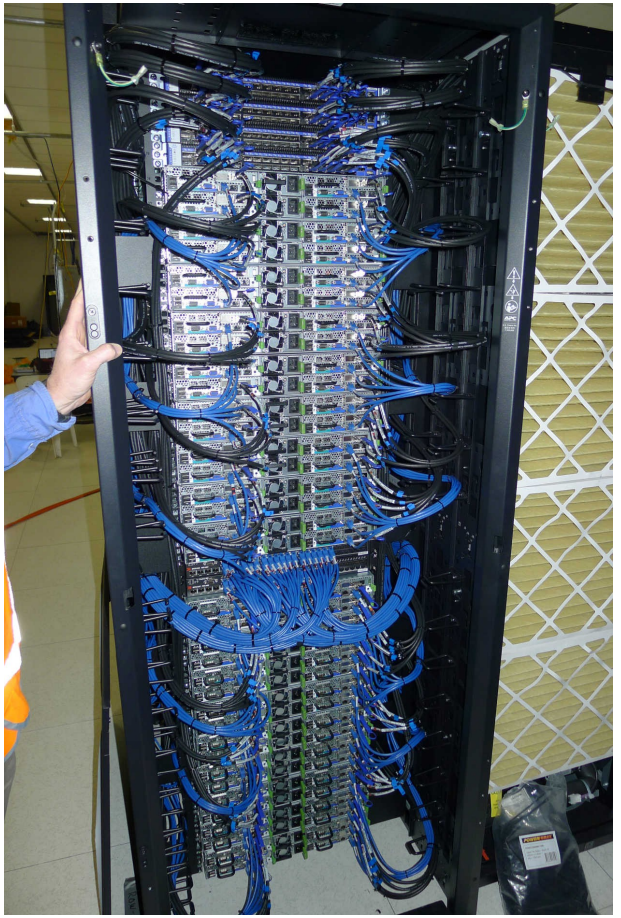
Lightweight PaaS on the NCI OpenStack Cloud



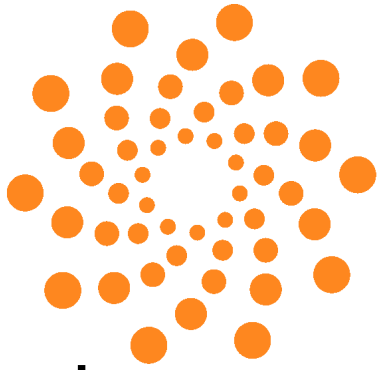
Kevin Pulo, Michael Chapman, Ben Evans
National Computational Infrastructure, ANU
kevin.pulo@anu.edu.au
http://nf.nci.org.au/

Who is NCI?

- National Computational Infrastructure
- Australia's peak academic HPC and Data Intensive science facility

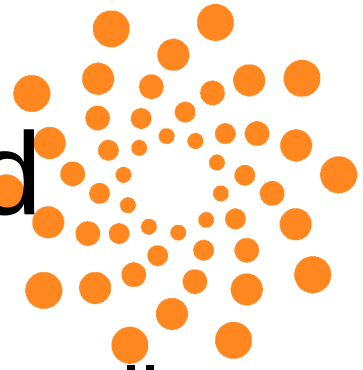


NCI Cloud Activities



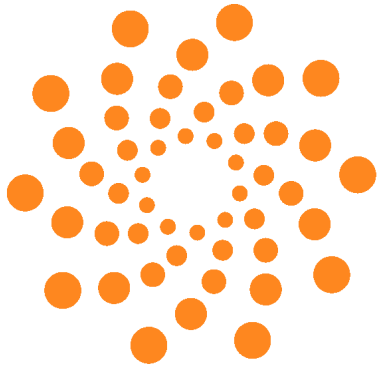
- Ad-hoc virtualised environments for researchers and projects
- High performance node of NeCTAR's National Research Cloud
  - OpenStack Essex (Folsom in progress)
  - Infiniband interconnect
  - Large memory nodes
  - GPU accelerators
- NCI Cloud

Researchers and the Cloud



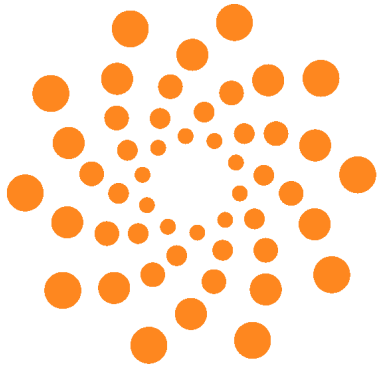
- Researchers are used to NCI's persistent, well manicured peak compute user environment
- Plain emphemeral IaaS would be too “raw”
- Augment our OpenStack deployment
  - Give users a structured admin/mgmt framework
  - Without increasing our support burden
- Range of requirements: Hosting services (web, database, code repos, login) thru to cloud-bursting pure HPC compute

Lightweight PaaS



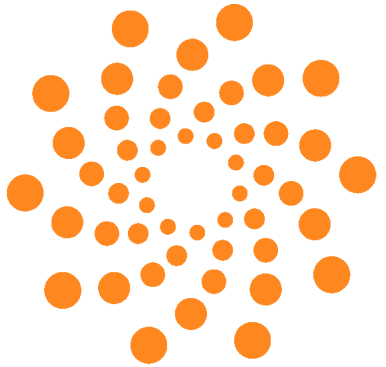
- Puppet for managing instances
- Git repositories for revision control
  - Researchers generally use svn (if anything)
- Integration with existing LDAP accounts

Features



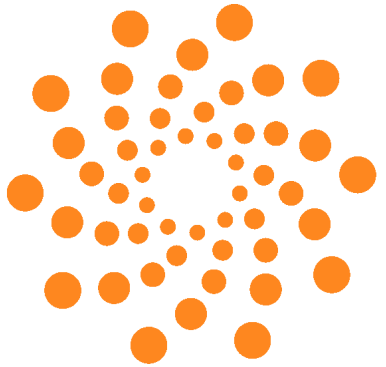
- Manage production services
- Spin up per-user development “clones”
- Integration with NCI user environment
- Collaborate with other tenants' Puppet configs
  - While keeping sensitive information private
- Receive updates from NCI
  - We don't control instances, but need them maintained

Main Components



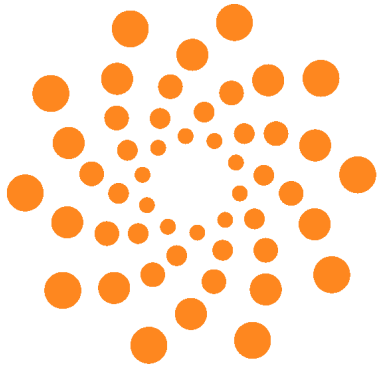
- Multi-tenant Puppet
- Git and Gitolite for Puppet configs
- Nova-boot wrapper and userdata helper
- Updating instances

Multi-tenant Puppet



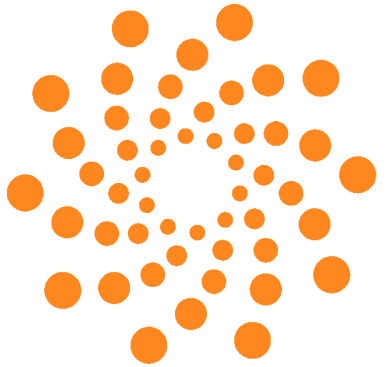
- Puppetmaster/agent isn't appropriate
  - Users befuddled by ssh keys, so Puppet keys ugh
- Discourage ad-hoc on-host config updates
- git hooks to automatically update running instances when commits are pushed
  - Immediate feedback on errors
- “Continuous deployment” model: System state is determined solely by repo contents

Git and Gitolite



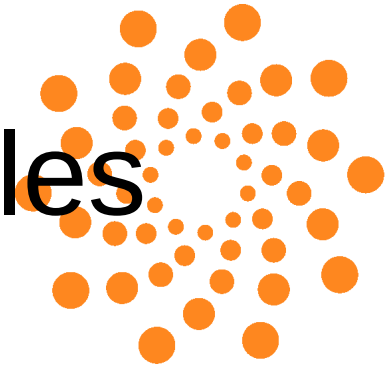
- Excellent access control layer for git
- https://github.com/sitaramc/gitolite
- Not using Gitolite's ssh key management
- Users login initially with LDAP password
- Then access self-service ssh key management system (plumbed into Gitolite)
- Fork the `nci/puppet` repo
- Gitolite has interface for managing permissions
- User branches starting with username

Public/Private Repos



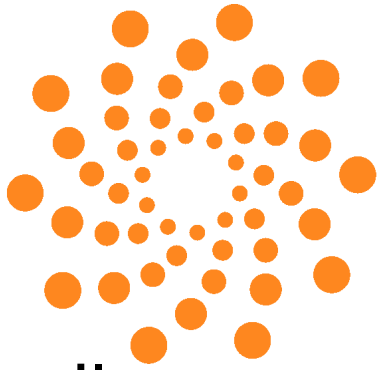
- Repos live under `p/tenant/*` namespace
  - Publicly readable, read/write by tenant members
- Repos under `p/tenant/private/*` namespace
  - Only read/write by tenant members
- `p/tenant/puppet` repo has submodule named `private`, at `p/tenant/private/puppet`
  - Stores passwords, private keys, etc
- Symlink into private repo, set Puppet variables
- NCI admins have full access

Typical `nci/puppet` Modules



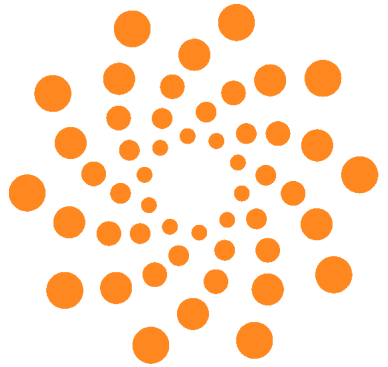
- LDAP client access
- Login access control via pam_access
- Sudo support
- Various software like Apache, Tomcat, Thredds, Python and CPAN modules
- Access to NFS filesystems
  - Including shared HPC filesystems
  - Including dealing with root_squash
  - Including for /home, /var, … in lieu of volumes/cinder
- Planned access to custom built peak HPC software tree

tools/nova-boot



- Eases the burden of bootstrapping a node to ~nil
  - Repo sanity checks (private repo correctly in place, keys generated and in place, git user/email defined, etc)
  - Primes `/root/.ssh/id_rsa` with private key that can clone repo (and checks that key works)
  - Primes `/root/.ssh/known_hosts` with repo hostkey
  - Propagates git config `user.name` and `user.email`
  - Set timezone/time, hostname, nameserver, floating IP (if requested)
  - Install git and puppet
  - Trigger initial puppet run

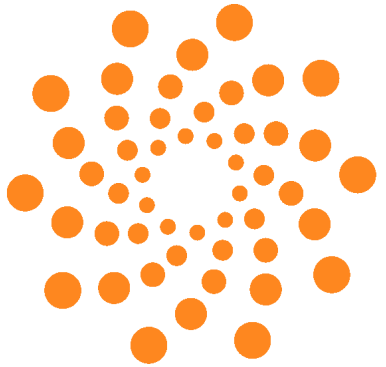
tools/nova-boot



Example usage:

```
cd puppet
tools/nova-boot
  --name fqdn.nci.org.au
  --repo git@repos.nci.org.au:p/tenant/puppet
  --branch username/some_branch
  --ip 192.43.239.xxx
  --
  --image centos-6-20121101
  --flavor 2
  --key-name my_key
  --security-groups open
```

Updating Instances

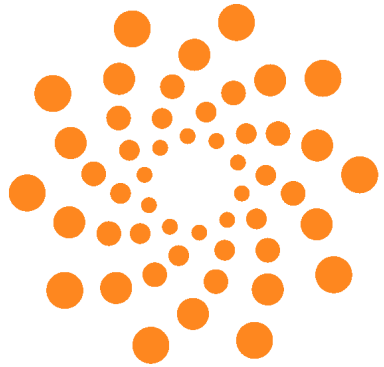


- Currently, users must manually do:
  - `git push`
  - `ssh root@instance puppet-update`

  Which effectively does:

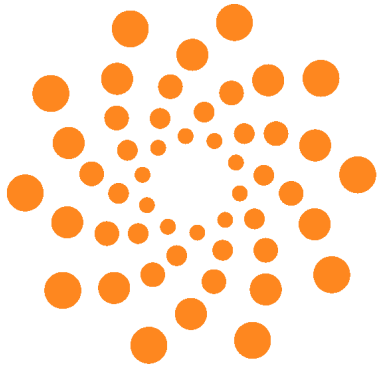
```
cd /etc/puppet
git pull
git submodule update
puppet apply
/etc/puppet/manifests/site.pp
```

Instance Auto-updating



- Client uses repo private key to ssh to repo machine
- Runs special command indicating the repo and branch to subscribe to (via Unix domain socket)
- git hook (pre/post-receive) notifies all subscribed instances that there is a new HEAD (via Unix domain socket)
- Small, simple daemon listens on the UDSs and relays the messages back and forth

Thank you



- Questions?
- Comments?