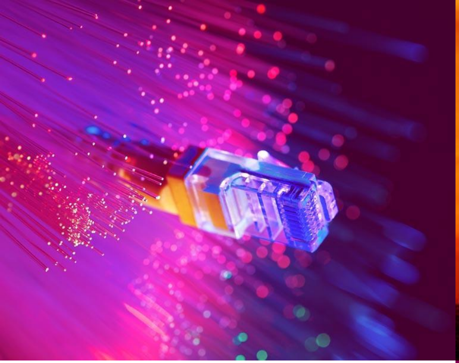
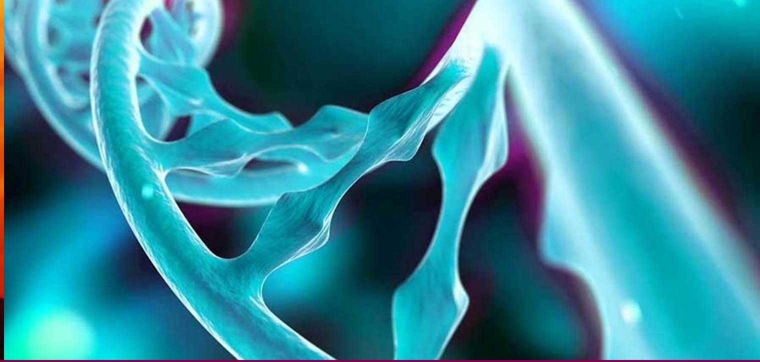




CSC OpenStack Clouds' admin and dev tooling

Jukka Nousiainen
Senior Systems Specialist @ CSCfi

CSC – Finnish research, education and public administration ICT knowledge centre

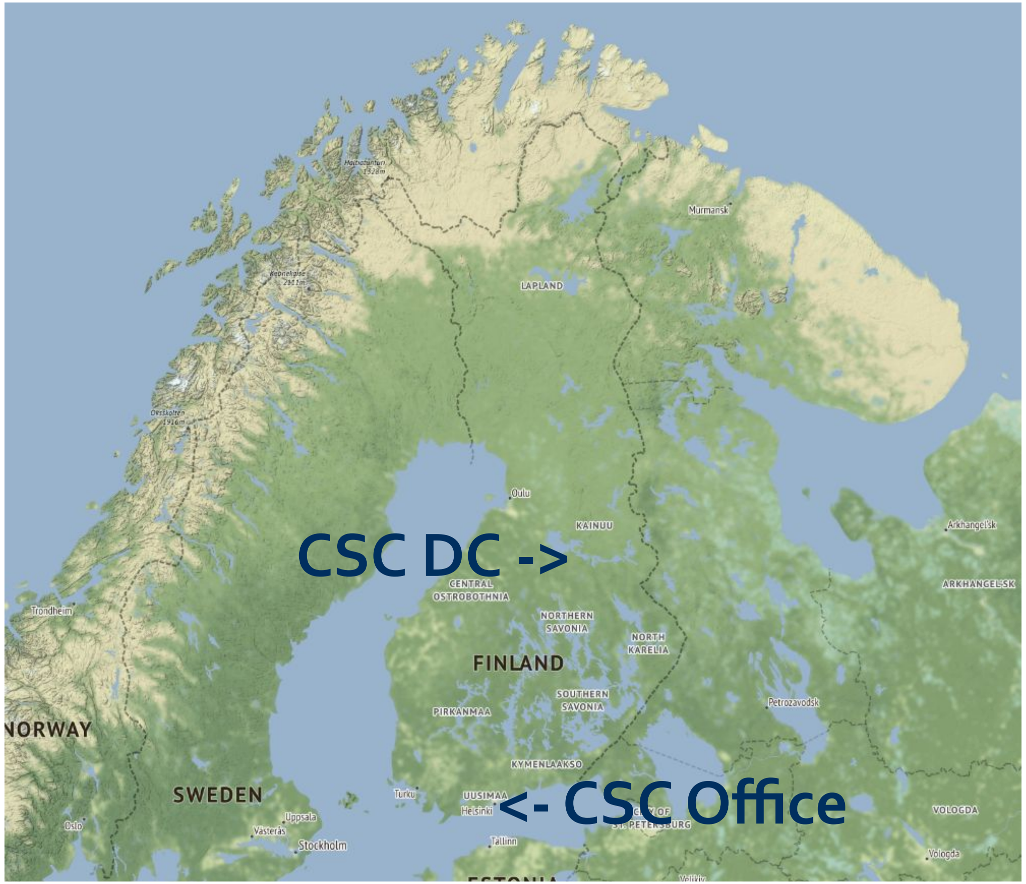
Really really really really quickly:

CSC Infrastructure Cloud ☁ in numbers

- **2** production OpenStack instances
- **400** servers (plus some buffer)
- **20k** HT CPUs
- **105 TB** RAM
- **1 PB** Ceph (and growing esp. with transition from L2 to routed)
- **1k** direct (OpenStack API) users
- Many, many end users

Really really really really quickly:

CSC Infrastructure Cloud ☁️ on the map



Paradigm, meet buzzword 🐝

"How"

"What"

State

Bootstrap

Paradigm, meet buzzword 🐝

“How”	OpenStack
“What”	Ceph
State	Puppet
Bootstrap	Ansible

Toolbox

Host/VM bootstrap, NIC/bond/routing	Compute nodes: Homegrown DHCP/PXE + Ansible Control plane: VMs + Ansible
Control Plane state	Puppet, RDO packages
Monitoring	NRPE, collectd, Gearman, custom scripts
User outreach	Email, surveys, custom scripts
Functionality testing	Rally, s3-tests
CI/CD	Travis, some experiments with Jenkins

Toolbox

Host/VM bootstrap, NIC/bond/routing	Compute nodes: **Homegrown DHCP/PXE (#2)** + Ansible Control plane: VMs + Ansible
Control Plane state	Puppet, RDO packages
Monitoring	NRPE, collectd, Gearman, **custom scripts (#1)**
User outreach	Email, surveys, **custom scripts (#3)**
Functionality testing	**Rally (#4)**, s3-tests
CI/CD	Travis, some experiments with **Jenkins (#5)**

Tool #1

Detecting PCI device ghosts

- GPU passthrough only - no VGPU yet
- After creating our first GPU flavors we noticed something odd in certain validation tests: Servers claiming to have GPU cards left could not accept new GPU VMs scheduled into them
- On closer inspection it was found that an extra GPU had slipped up to an existing VM in the server
- Even closer inspection revealed that this is a pattern which may occur when doing rebuilds for a PCI-device equipped VM
- Bug opened at:

https://bugs.launchpad.net/nova/+bug/1780441

- Not a libvirt issue but a nova side issue, so possible to track in the database

Tool #1

Detecting PCI device ghosts

```
# Assumes that column value for flavors with pci_aliases
# are colon separated like M10:1 or P100:4
SELECT my_uuids
FROM
(
  SELECT
     pcid.instance_uuid as my_uuids,
     count(pcid.instance_uuid) as actual_gpus,
     SUBSTRING_INDEX(flex.value, ':', -1) as flavor_gpus
  FROM
     nova.pci_devices pcid,
     nova.instances inst,
     nova_api.flavor_extra_specs flex
  WHERE
     pcid.status = "allocated" AND
     inst.uuid = pcid.instance_uuid AND
     inst.instance_type_id = flex.flavor_id AND
     flex.`key` LIKE "%alias"
  GROUP BY
     pcid.instance_uuid
  HAVING
     actual_gpus != flavor_gpus
)
AS T
```

Tool #2
Bootstrapping hosts with DHCP/PXE

This is the basic workflow when running a DHCP/PXE server with our Ansible role (link to follow):

1) Ensure VM’s ansible group has a kickstart profile
   a) For servers providing entirely new functionality, need to create a new profile
   b) otherwise just reuse old one
2) Touch /var/www/provision/reinstall/hostname
3) ILO reboot server
4) Once host has booted into the OS, we can continue with other parts of the configuration management chain

Tool #3 - Cloudmailer

- A happy user is a user who is aware of issues in the platform
- An unhappy user is unaware of what’s going on
- **Large outages:** Bulk email to list with all users of the environment
- **Single server outages/issues:** We use cloudmailer, a tool for contacting users about issues related to
  - One or multiple hypervisors
  - One or multiple VMs
  - One or multiple computing projects.
- At this time we don’t provide a status portal such as https://status.aalto.fi though that might be nice to build some day, too
  - Flagging individual hypervisors in a status portal is probably overkill anyway

Tool #3 - Cloudmailer - Notify mode

```
$ python cloudmailer.py -n -y hypervisors.txt -m "cPouta compute node - broken
DIMM" -t templates/mail_template.txt-notify-dimm
--I-am-sure-that-I-want-to-send-emails
Get All Servers
All Servers received
Get All Users
Get All Projects
All Projects received
Start requesting Project Role Assignments
Threads created
Role Assignments received
6 projects to send email to.
Are you sure that you want to send the emails? Required answer: "Yes I am
sure"Yes I am sure
Really sending emails to:
user1@university, user2@university…
```

Tool #3 - Cloudmailer - Scheduling mode

If livemigration is not possible for whatever reason, cloudmailer also has ability to schedule reboots based on anti-affinity server groups. Quorum is king!

Day 1 9:00	Day 1 11:00	Day 1 14:00	Day 1 16:00	Day 2 9:00	Day 2 11:00	Day 2 14:00	Day 2 16:00	Day 3 9:00	Day 3 11:00
SLURM	SLURM	SLURM	SLURM	SLURM	SLURM	SLURM	SLURM	SLURM	SLURM
k8s	k8s	k8s	k8s	k8s	k8s	🟩			
SQL	SQL	SQL	SQL	SQL	🟪	🟥			
k8s	k8s	k8s	HAproxy	Haproxy	🟦	🟥			
SQL	SQL	SQL	🟥	⬛	🟧				

Tool #4

OpenStack Rally and Tempest

- Rally provides facilities for performance, functionality and many other type of testing
- Tempest is a “test kit” focusing on specifically on scenarios and other types of functionality testing
- As part of Glenna work, CSC has developed Ansible roles to
  - Install Rally and Tempest
  - Configure Tempest scenarios
- Ourselves we use these for “nightly builds” across all environments as well as spot testing if certain functionality changes in development

Tool #4

OpenStack Rally and Tempest

```
  cpouta-devel:
        region: kaj-devel
        endpoint: pouta-devel.csc.fi:5001/v3
        adminuser: rally-admin
        adminpassword: "{{ vault_rally_devel_admin_password }}"
        admin_project_name: rally
        admin_project_domain: service
        admin_domain_name: service
        admin_project_domain_name: service
        username: rally
        userpassword: "{{ vault_rally_devel_password }}"
        user_project_name: rally
        user_domain_name: service
        user_project_domain_name: service
        network_name: rally-network
        tempest_skip_tests: "{{ tempest_skip_tests_devtest_envs }}"
        tempest_extra_settings: "{{ rally_tempest_extra_settings }}"
        configure_tempest_cron: true
        tempest_manual_forloop_cleanup: true
        tempest_cron_run_hour: 0
configure_tempest: true
tempest_source: "https://github.com/CSCfi/tempest"
tempest_version: "csc_master"
tempest_test_flavor: standard.tiny
tempest_second_test_flavor: standard.small
rally_tempest_swift_operator_role: object_store_user
rally_tempest_swift_discoverability: "False"
```

Tool #5

Polte (CI/)CD

- Polte ~ "A burning sensation"
- Famously started because during an internal hackathon someone said to someone else that this can not be done
- Builds a Heat Stack (Heat equals VM Orchestration in OpenStack)
- Configures into that Heat Stack
  - LDAP server
  - Puppet server
  - Squid proxy
  - OpenStack Control Plane
  - Dummy data plane (no nested VLAN)
  - Dummy hypervisors (no nested virt)
  - Ceph cluster
  - Rally

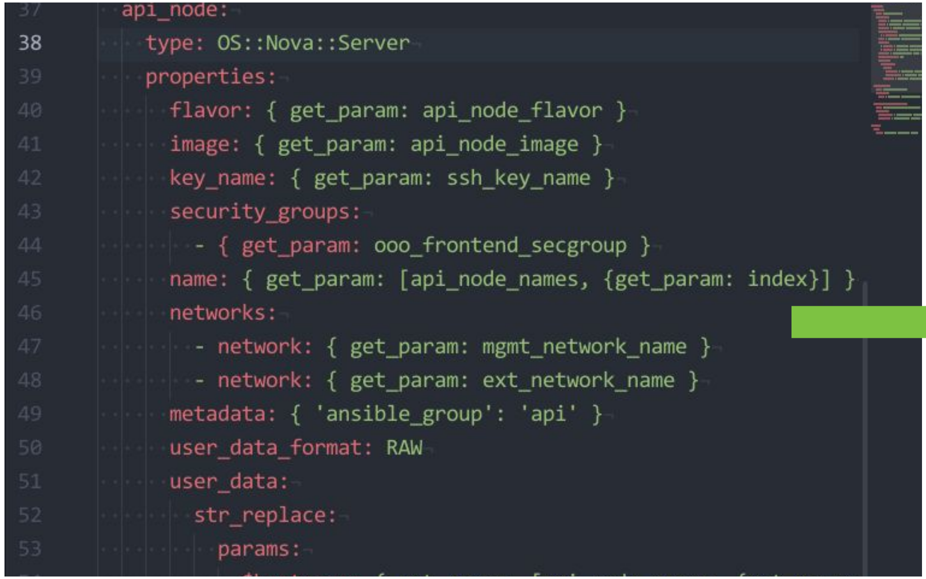
Tool #5

Polte (CI/)CD

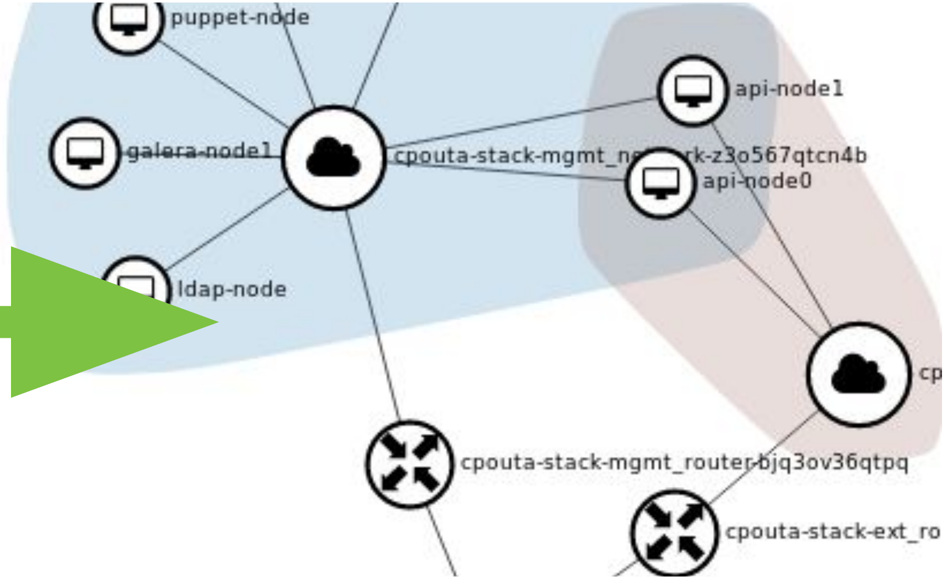
- Yes, container based testing would be much faster. But this can also be thought of as a learning experiment with Heat
- Currently using Jenkins but kind of without Jenkins
  - Zero lines of groovy :-)
  - Bash wrappers for Ansible :---)
- Jenkinsfile (multibranch pipeline) takes care of parallelization for the builds, otherwise they'd take many hours
- Build notifications pushed to Ops Flowdock
- Destroy stack after successful build unless otherwise configured
- All this is extra to basic linters, unit tests etc. run by Travis on each commit (on most of our stuff)

Tool #5
Polte - Continuous Development

Something like this **builds** Something like this



```
  api_node:
    type: OS::Nova::Server
    properties:
      flavor: { get_param: api_node_flavor }
      image: { get_param: api_node_image }
      key_name: { get_param: ssh_key_name }
      security_groups:
        - { get_param: ooo_frontend_secgroup }
      name: { get_param: [api_node_names, {get_param: index}] }
      networks:
        - network: { get_param: mgmt_network_name }
        - network: { get_param: ext_network_name }
      metadata: { 'ansible_group': 'api' }
      user_data_format: RAW
      user_data:
        str_replace:
          params:
```

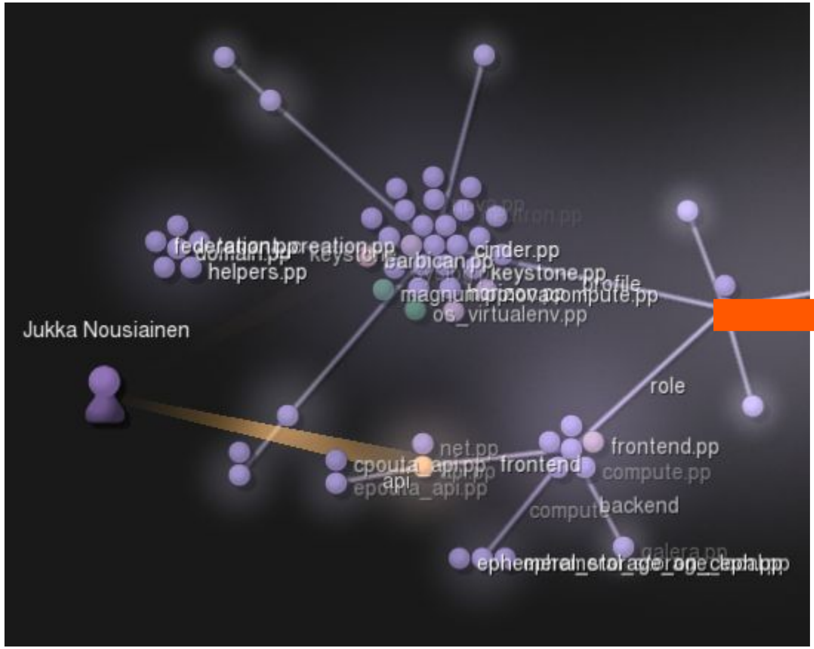


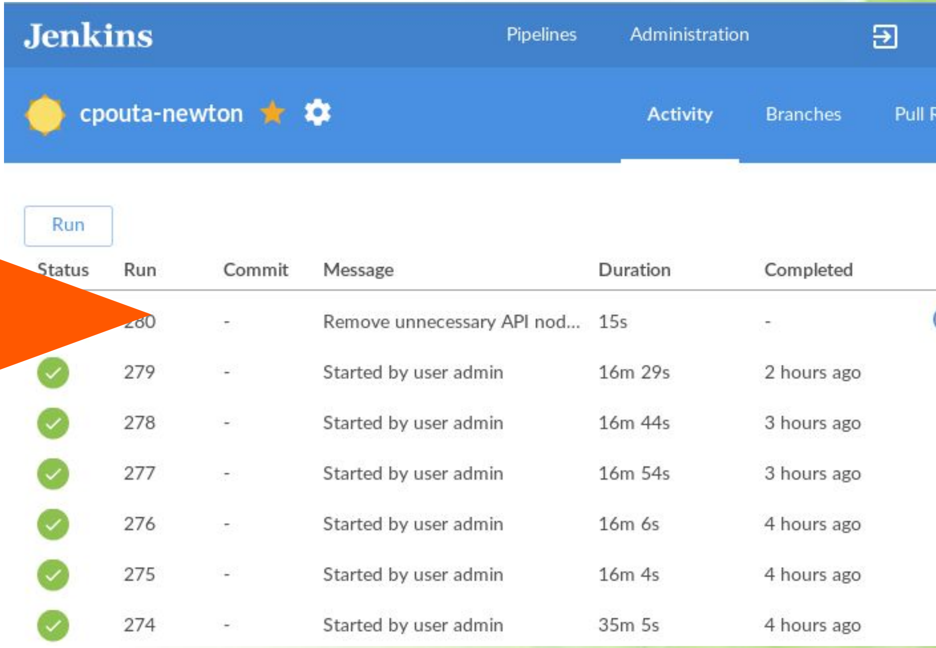
Tool #5
Polte - Continuous Development

Let’s see it in action!

Tool #5
Polte - Continuous Development

Something like this triggers Something like this




Jenkins Pipelines Administration

cpouta-newton

Activity Branches Pull R

Run

Status	Run	Commit	Message	Duration	Completed
	280	-	Remove unnecessary API nod...	15s	-
✓	279	-	Started by user admin	16m 29s	2 hours ago
✓	278	-	Started by user admin	16m 44s	3 hours ago
✓	277	-	Started by user admin	16m 54s	3 hours ago
✓	276	-	Started by user admin	16m 6s	4 hours ago
✓	275	-	Started by user admin	16m 4s	4 hours ago
✓	274	-	Started by user admin	35m 5s	4 hours ago

Links

- Tool #1 SQL PCI ghosts https://github.com/CSCfi/puppet-opsviewagent/blob/master/files/nrpe/find_pci_ghosts.sql
- Tool #2 DHCP/PXE role https://github.com/CSCfi/ansible-role-dhcp-kickstart/
- Tool #3 Cloudmailer https://github.com/CSCfi/cccp-ops-contrib
- Tool #4 Rally
  - https://github.com/CSCfi/ansible-role-rally
  - https://github.com/CSCfi/ansible-role-rally-scenarios
- Tool #5 Polte https://github.com/CSCfi/polte
- All in all we have about ~100 repos with the term "ansible-role" - already these are worth checking out if you're doing anything at all with Ansible! https://github.com/CSCfi/

Thank you!

Jukka Nousiainen

jukka.nousiainen@csc.fi

@junousi