Troubleshooting Your SUSE® Cloud

TUT6113

Paul Thompson
SUSE Technical Consultant

Dirk Müller
SUSE OpenStack Engineer

SUSE® Cloud ...

SUSE® Cloud Troubleshooting

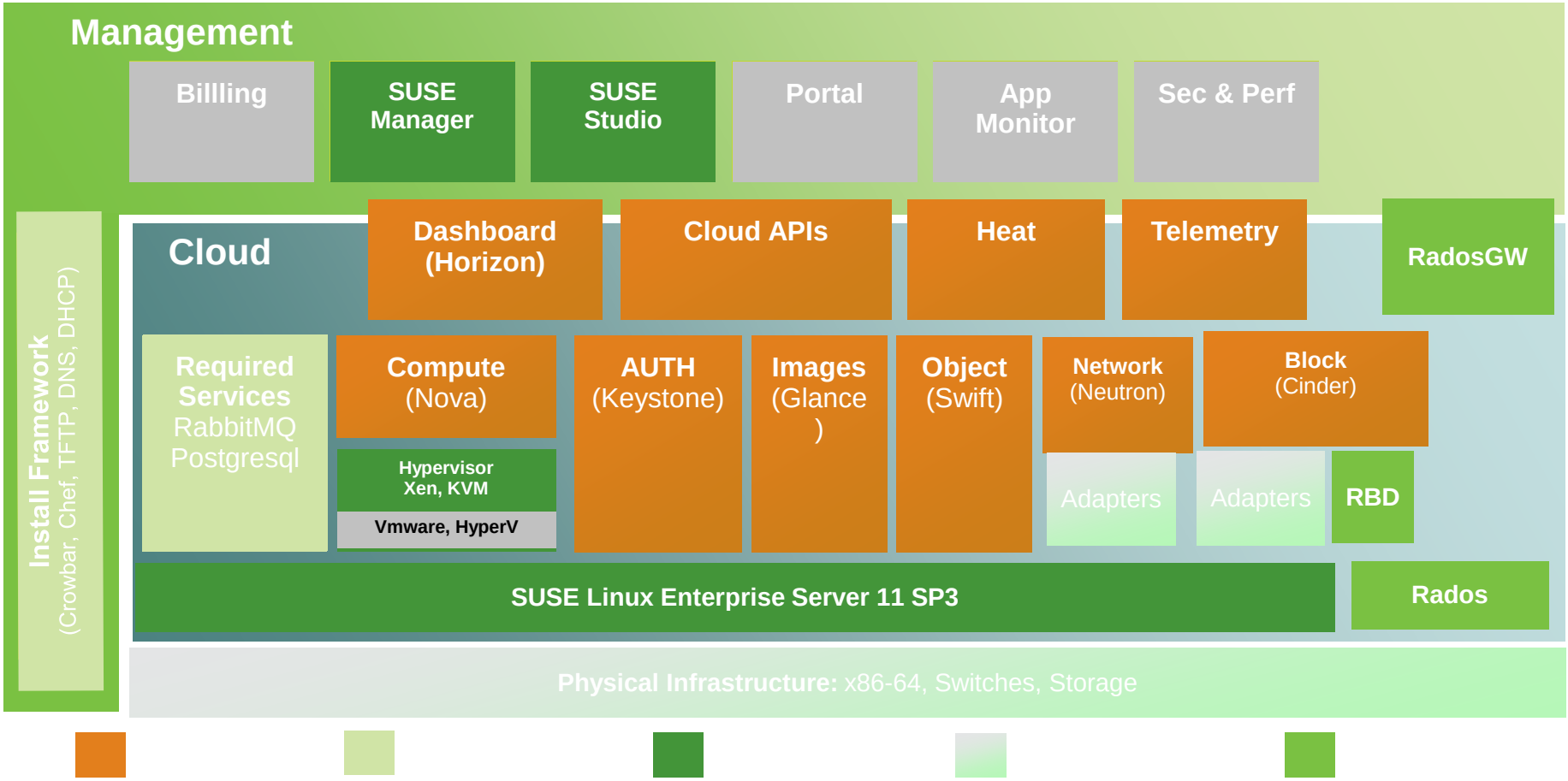
SUSE® Cloud

SUSE® Cloud

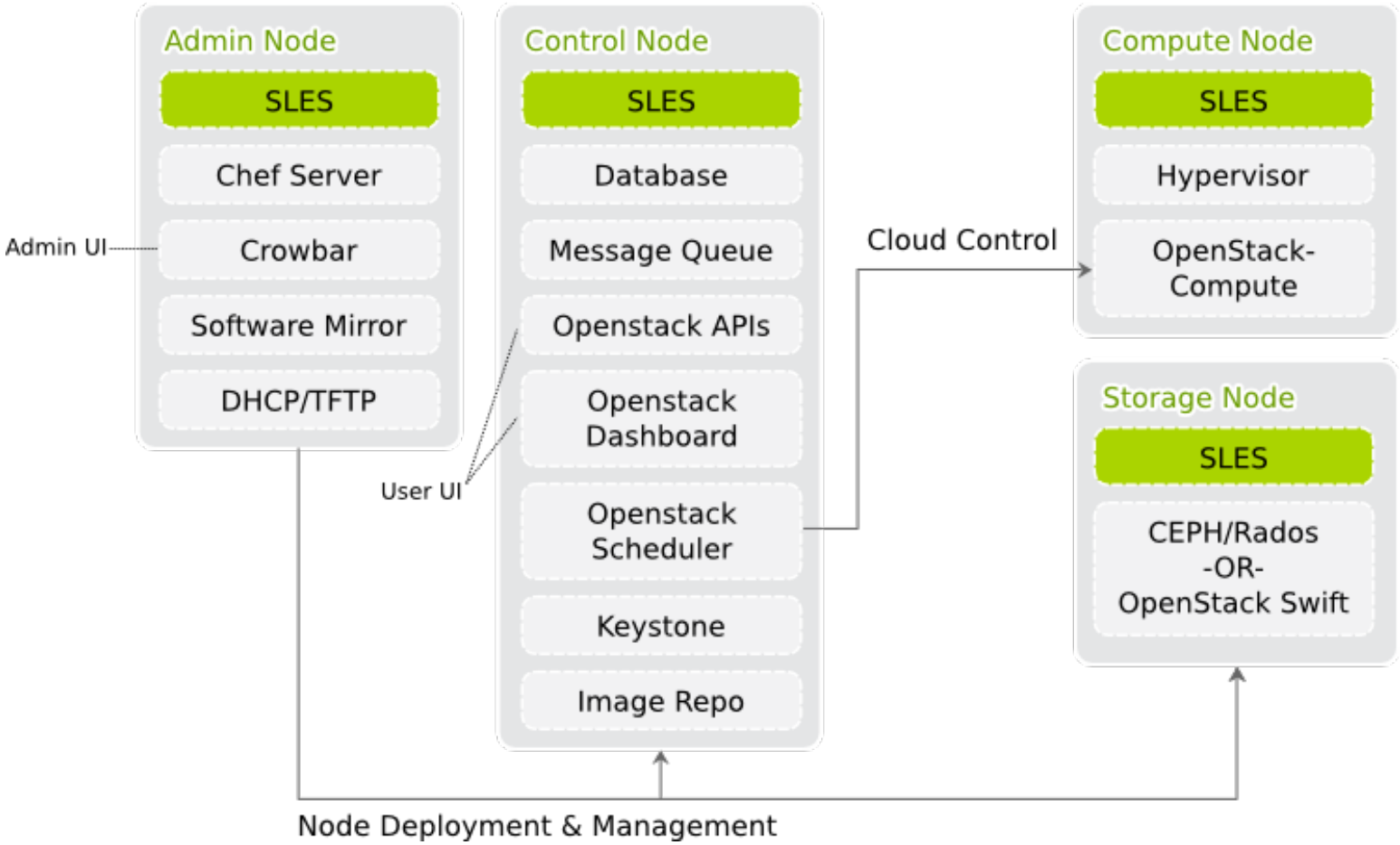
SUSE® Cloud

SUSE® Cloud Troubleshooting

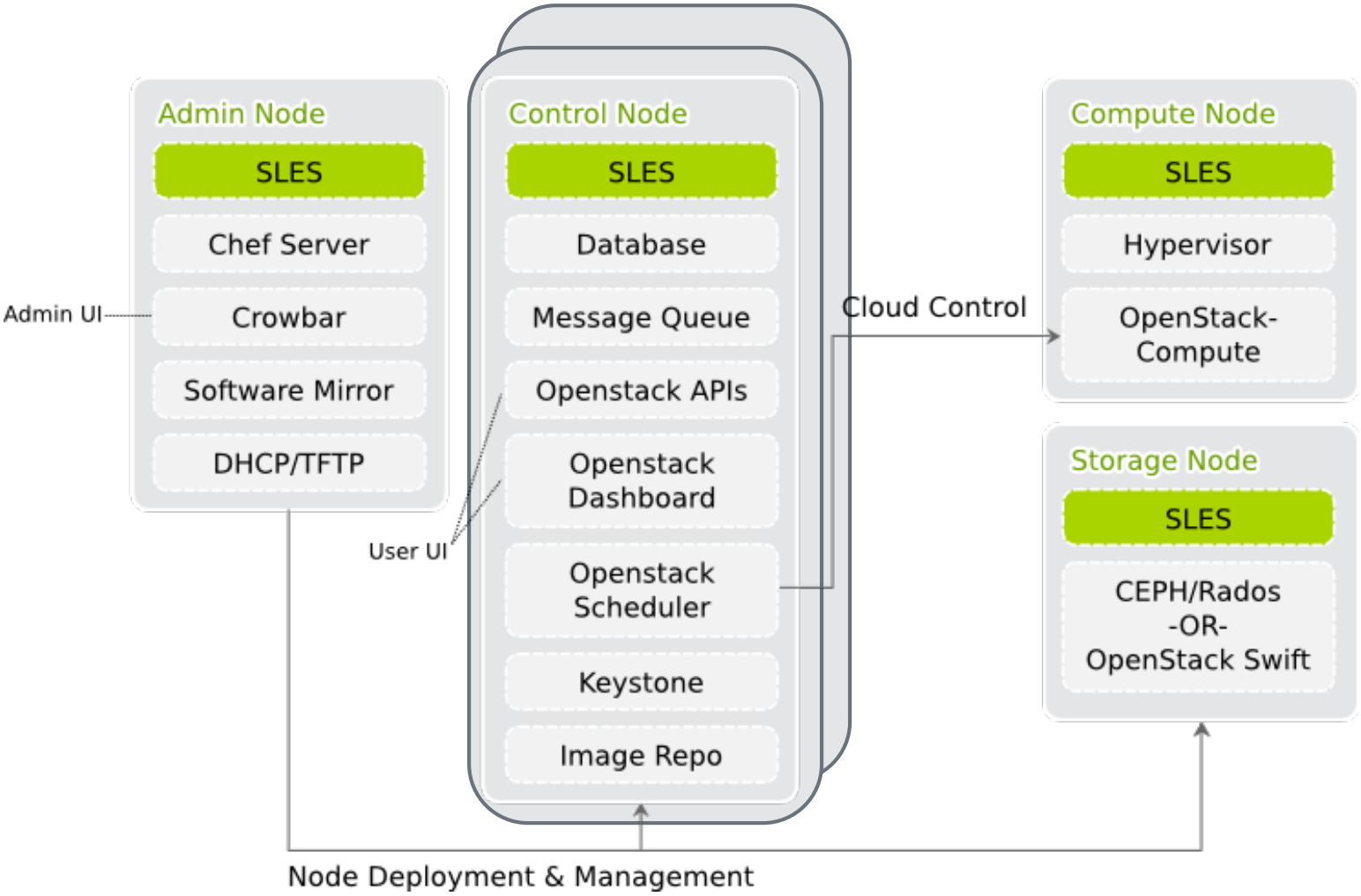
SUSE® Cloud Building Blocks



Non-HA SUSE® Cloud Installation



HA SUSE® Cloud Installation

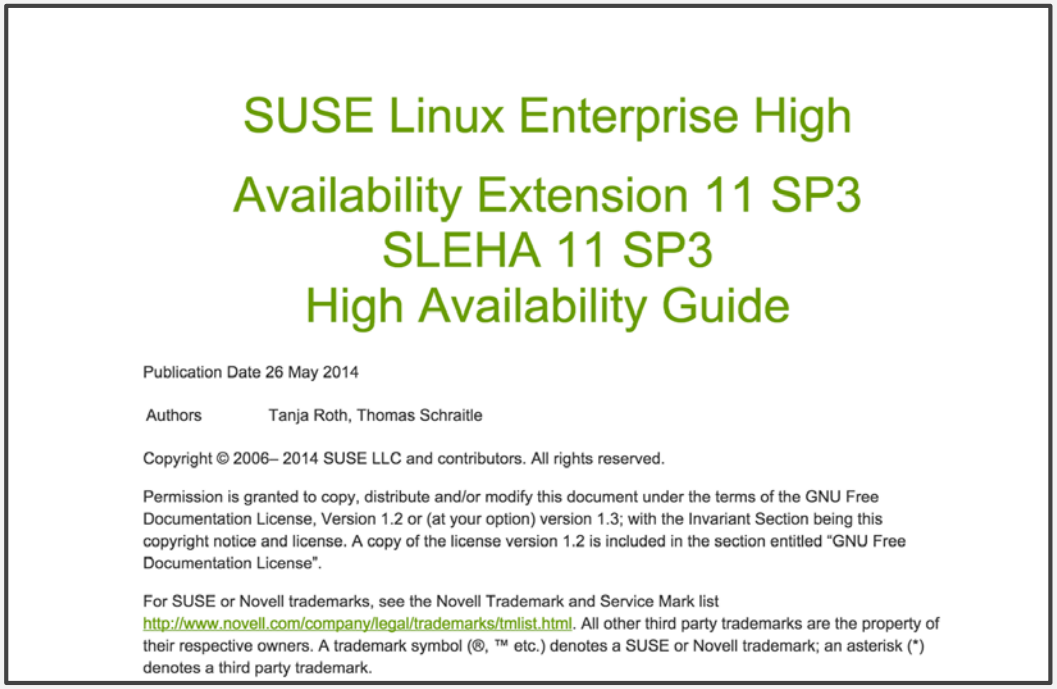


Just Enough HA for Troubleshooting

- `crm resource list`
- `crm_mon`
- `crm resource restart <X>`
- `crm resource cleanup <X>`

More About HA...

https://www.suse.com/documentation/sle_ha/



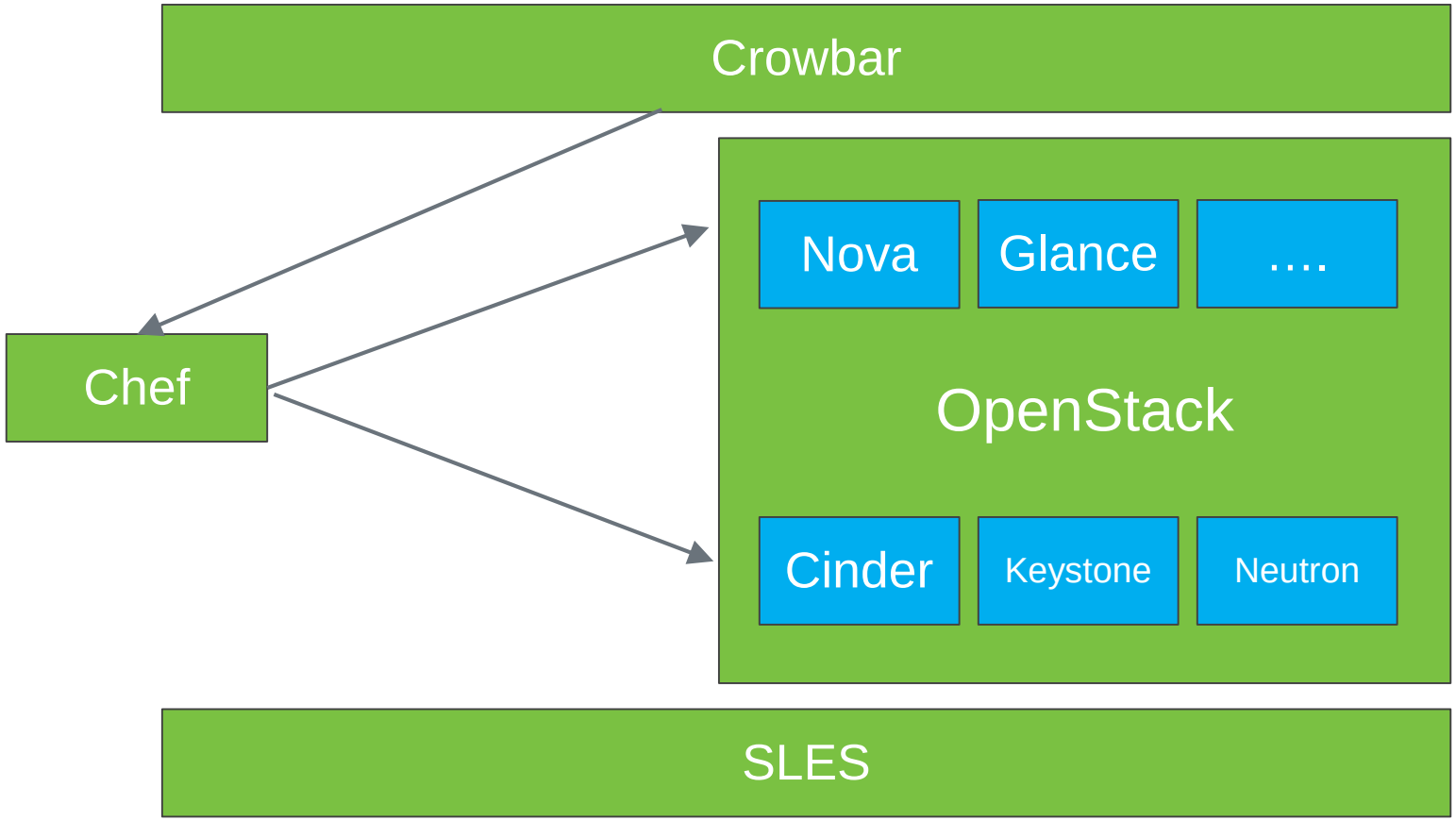
SUSE Linux Enterprise High

Availability Extension 11 SP3
SLEHA 11 SP3
High Availability Guide

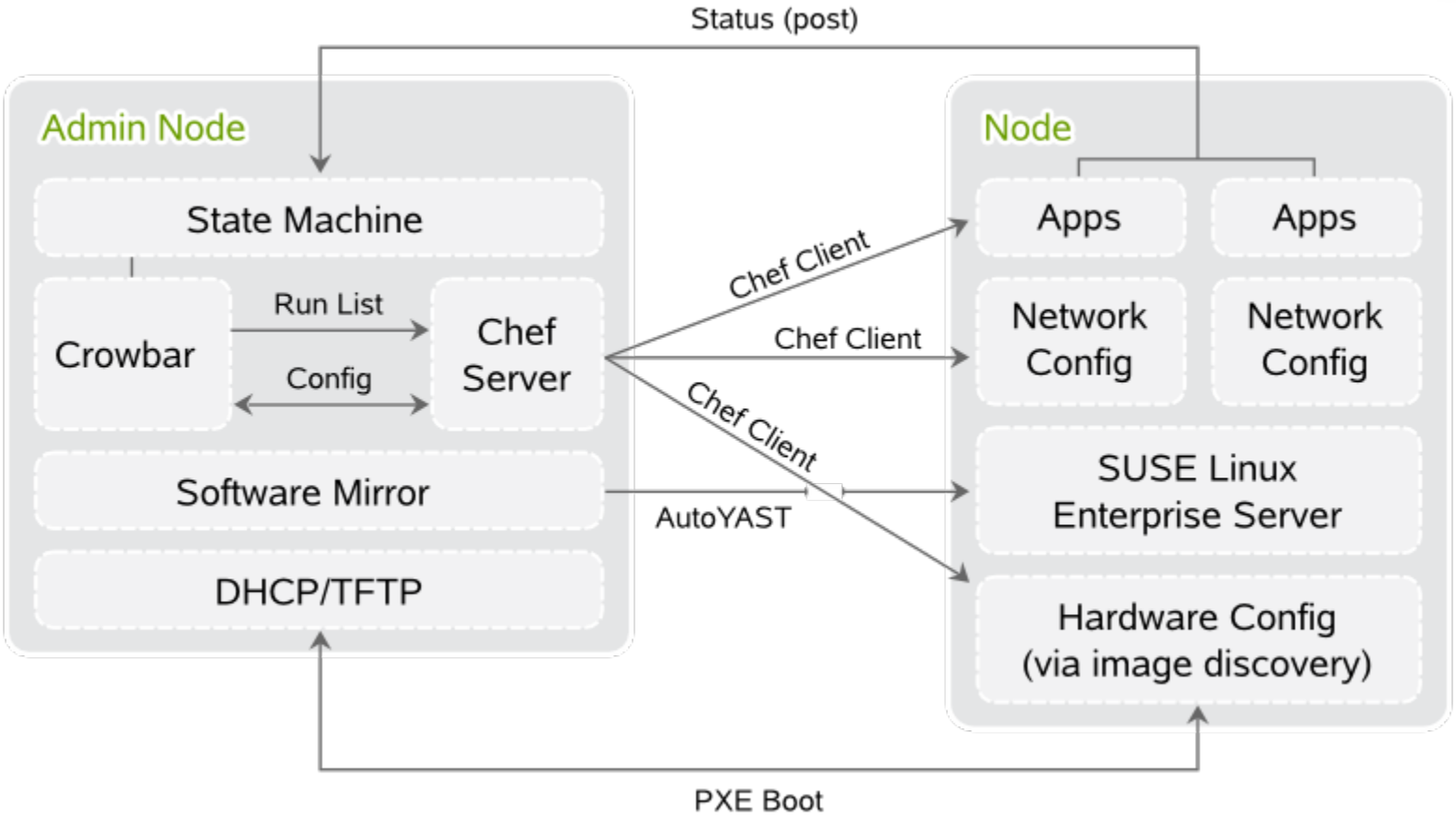
Publication Date 26 May 2014

Authors Tanja Roth, Thomas Schraitle

SUSE® Cloud Functional Blocks



Crowbar and Chef



Generic SLES® Troubleshooting

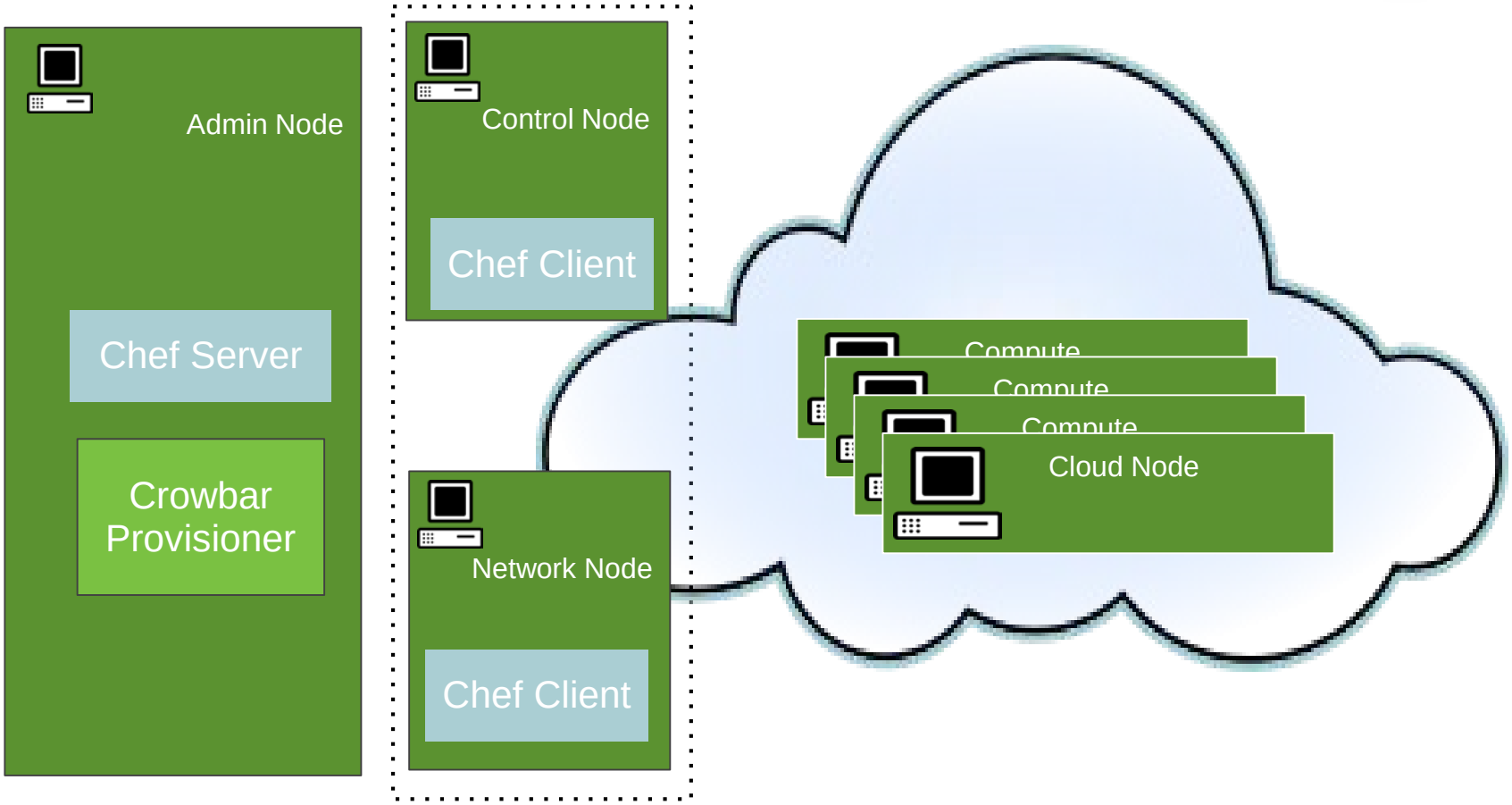
- All Nodes in SUSE® Cloud are SLES11 SP3
- Watch out for typical issues:
  - `dmesg` for hardware-related errors, OOM, interesting kernel messages
  - usual syslog targets, e.g. `/var/log/messages`
- Check general node health via:
  - `top, vmstat, uptime, pstree, free`
  - core files, zombies, etc

Supportconfig

- `supportconfig` can be run on any cloud node

- `supportutils-plugin-susecloud.rpm`
  - installed on all SUSE Cloud nodes automatically
  - collects precious cloud-specific information for further analysis

Typical Deployment Schema



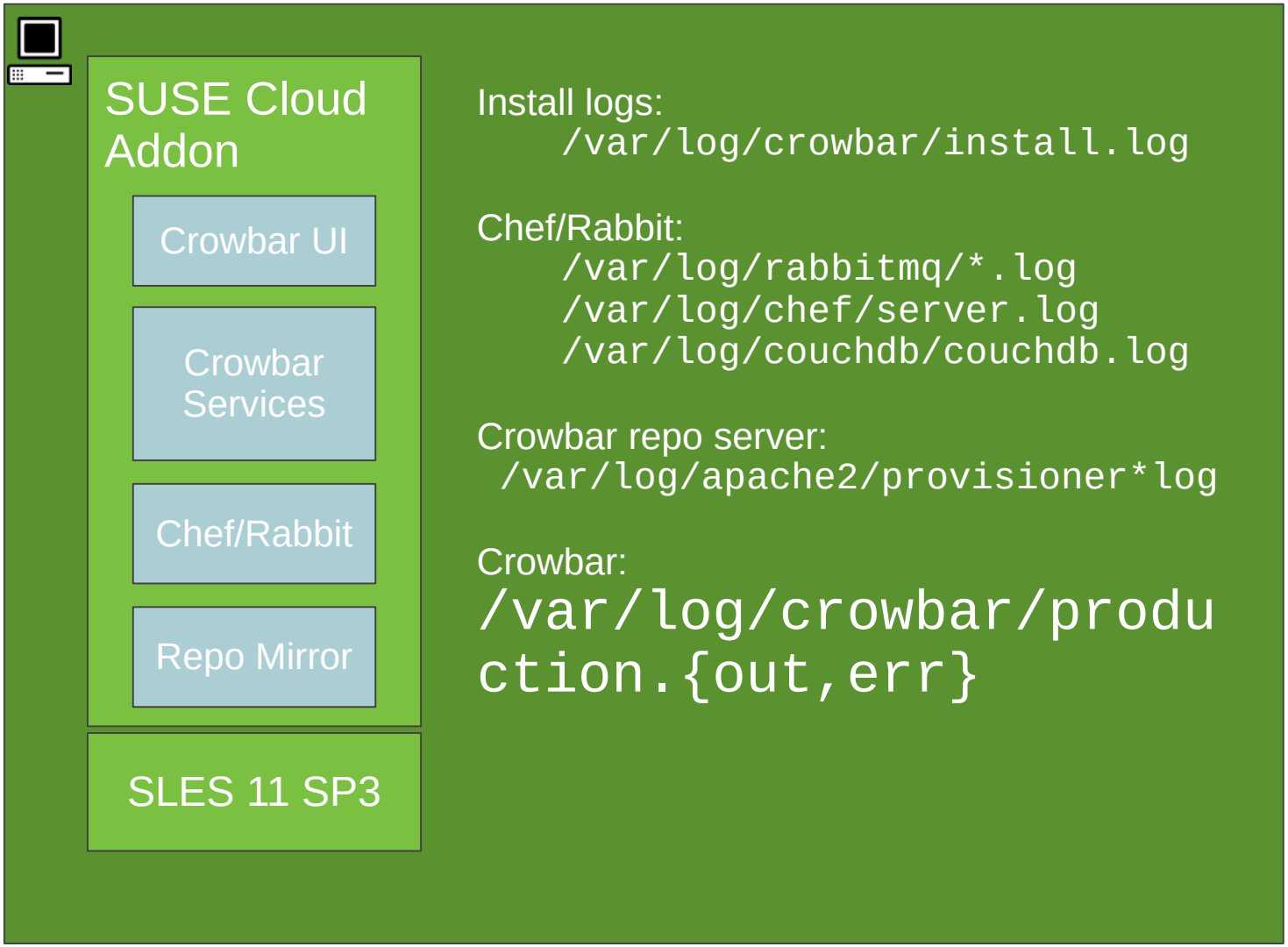
Cloud Install

```
screen install-suse-cloud --verbose

/var/log/crowbar/install.log

/var/log/crowbar/barclamp_install/*.log
```

SUSE® Cloud Admin Node



Chef

- Cloud uses Chef for almost everything:
  - All Cloud and SLES non-core packages
  - All config files are overwritten
  - All daemons are started
  - Database tables are initialized

---

http://docs.getchef.com/chef_quick_overview.html

Admin Node: Using Chef

```
knife node list

knife node show <nodeid>

export EDITOR=/usr/bin/vim; \
knife node edit -a <nodeid> node
```

SUSE® Cloud Admin Node

- Populate `~root/.ssh/authorized_keys` prior install
- Barclamp install logs:

  `/var/log/crowbar/barclamp_install`
- Node discovery logs:

  `/var/log/crowbar/sledgehammer/d<macid>.<domain>.log`
- Syslog of crowbar installed nodes sent via rsyslog to:

  `/var/log/nodes/d<macid>.log`

Useful Tricks

- Root login to the Cloud installed nodes should be possible from admin node (even in discovery stage)
- If admin network is reachable:

~/.ssh/config:

```
host 192.168.124.*
    StrictHostKeyChecking no
    user root
```

SUSE® Cloud Admin Node

- If a proposal is applied, chef client logs are at:

  `/var/log/crowbar/chef-client/<macid>.<domain>.log`

- Useful crowbar commands:

  `crowbar machines help`

  `crowbar transition <node> <state>`

  `crowbar <barclamp> proposal list|show <name>`

  `crowbar <barclamp> proposal delete default`

Admin Node: Crowbar Services

- Nodes are deployed via PXE boot:

`/srv/tftpboot/discovery/pxelinux.cfg/*`

- Installed via AutoYaST; profile generated to:

`/srv/tftpboot/nodes/d<mac>.<domain>/autoyast.xml`

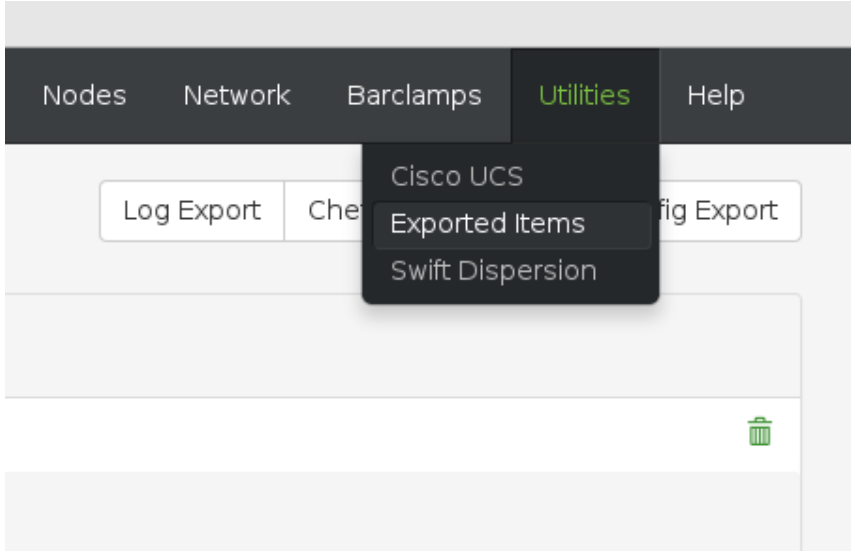
- Can delete & rerun chef-client on the admin node

- Can add useful settings to autoyast.xml:

```
<confirm config:type="boolean">true</confirm>
```

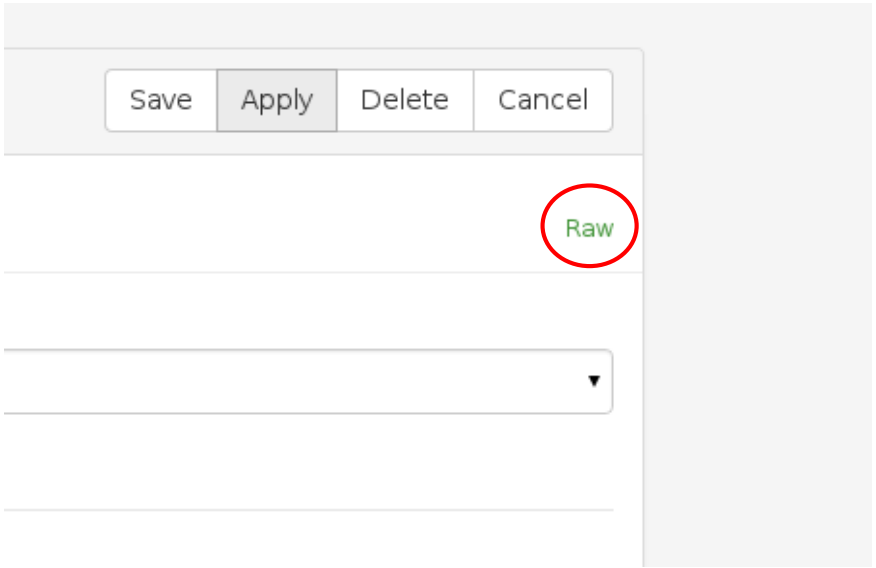
(don't forget to `chattr +i` the file)

Admin Node: Crowbar UI



Useful Export Page available in the Crowbar UI in order to export various log files from a customer installation

Admin Node: Crowbar UI



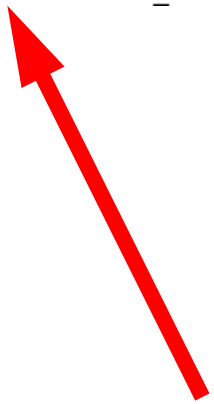
Raw settings in barclamp proposals allow access to "expert" (hidden) options

Most interesting are:

```
debug: true
verbose: true
```

Admin Node: Crowbar Gotchas

```
admin:~ # ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 LOCAL(0)        .LOCL.          10 l    6   16  377    0.000    0.000   0.001
*hermes.suse.de  131.188.3.222    2 u  190  256  377    0.339    0.305   0.067
```

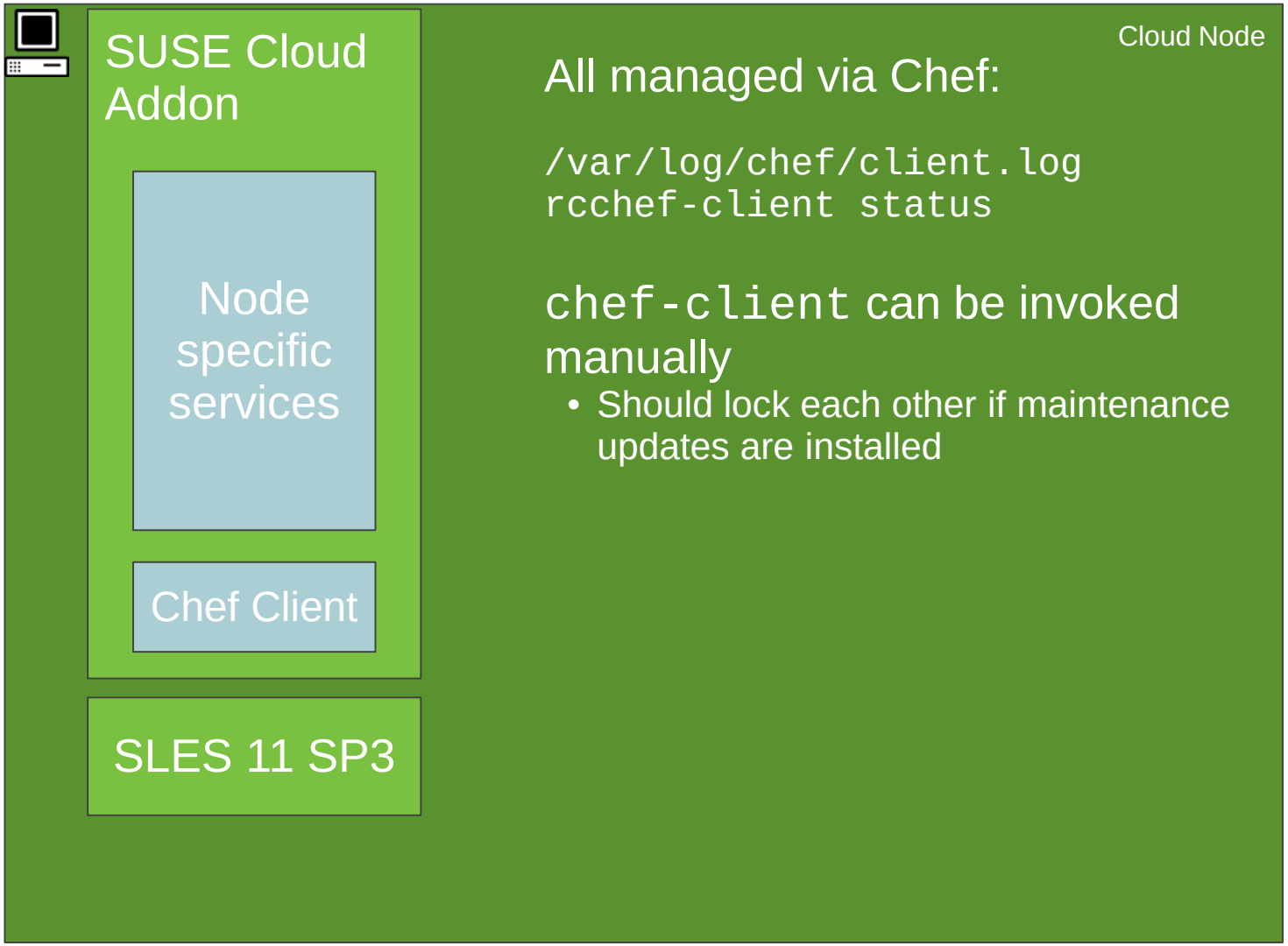


Admin Node: Crowbar Gotchas

- Be patient
  - Do not multiple transition nodes from one state to another
  - Do not apply proposals while a proposal is applying

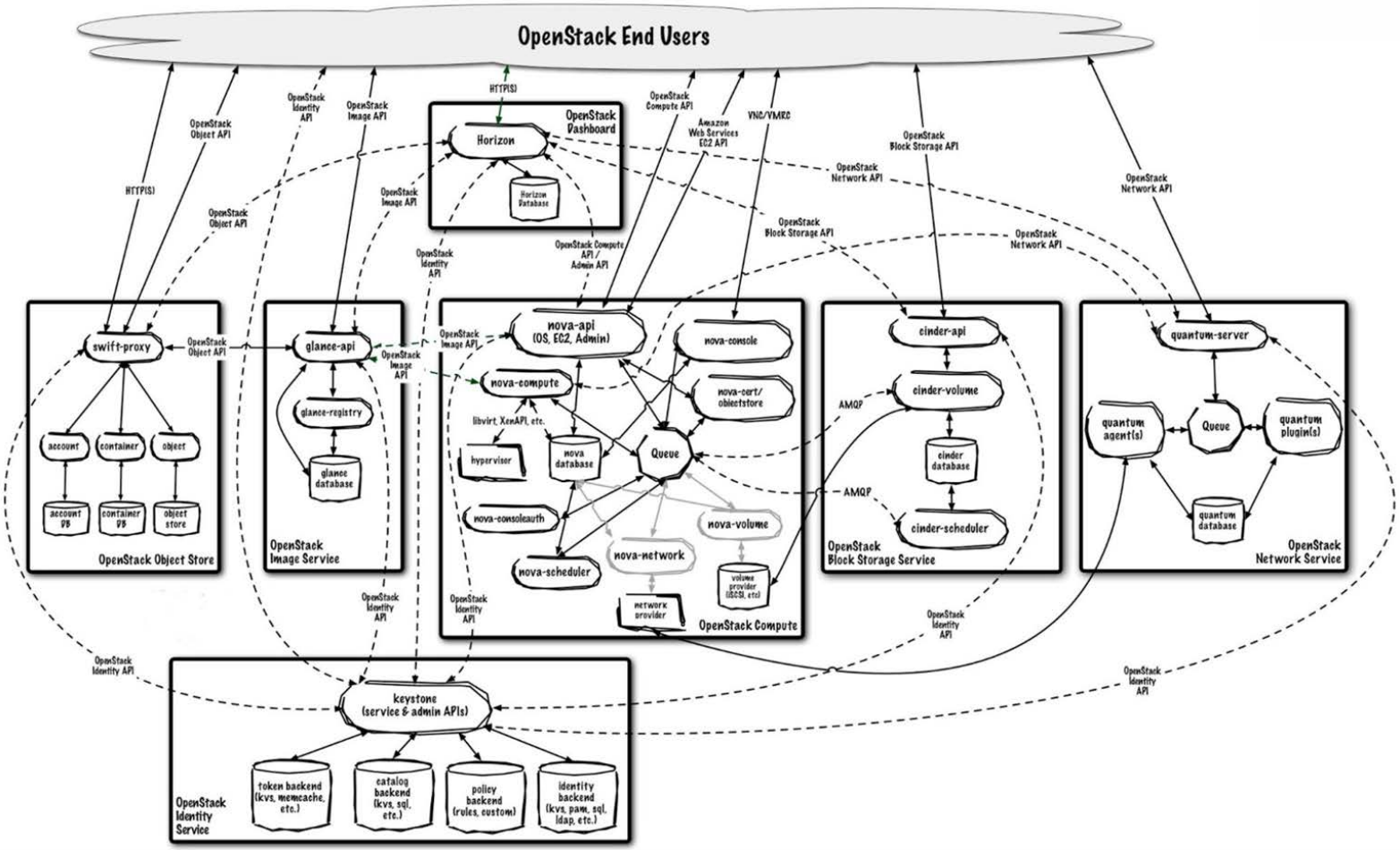
- Cloud nodes should boot from:
  1. Network
  2. First disk

SUSE® Cloud Nodes

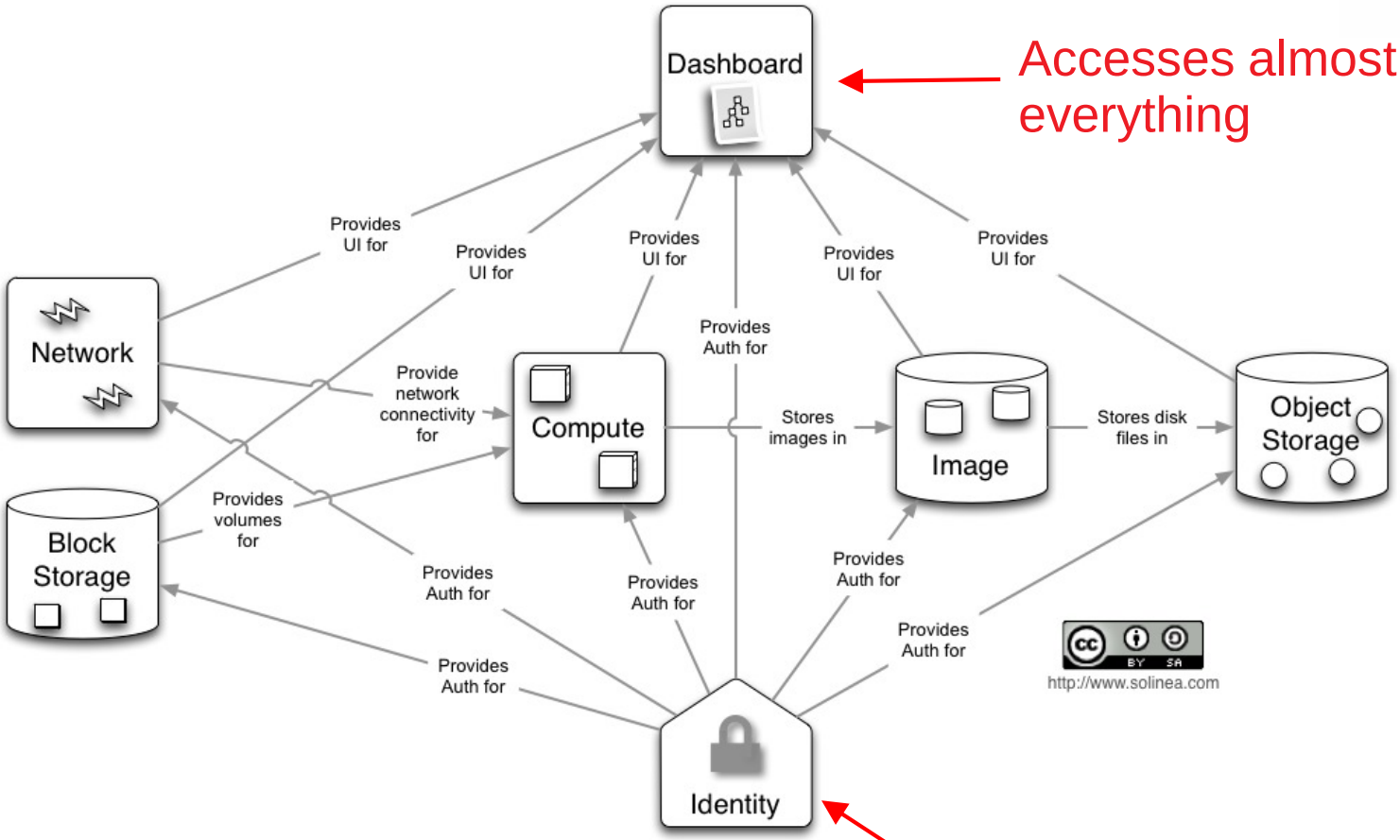


SUSE® Cloud Control Node

OpenStack Architecture Diagram



OpenStack Block diagram



OpenStack Architecture

- Typically each OpenStack component provides:
  - an API daemon / service
  - one or many backend daemons that do the actual work
  - command line client to access the API
  - `<proj>-manage client` for admin-only functionality
  - dashboard ("Horizon") Plugin providing a graphical view on the service
  - uses an SQL database for storing state

OpenStack Packaging Basics

- Packages are usually named:

  `openstack-<codename>`

  - usually a subpackage for each service (`-api`, `-scheduler`, etc)
  - log to `/var/log/<codename>/<service>.log`
  - each service has an init script:

```
dde-ad-be-ff-00-01:~# rcopenstack-glance-api status
Checking for service glance-api                 ...running
```

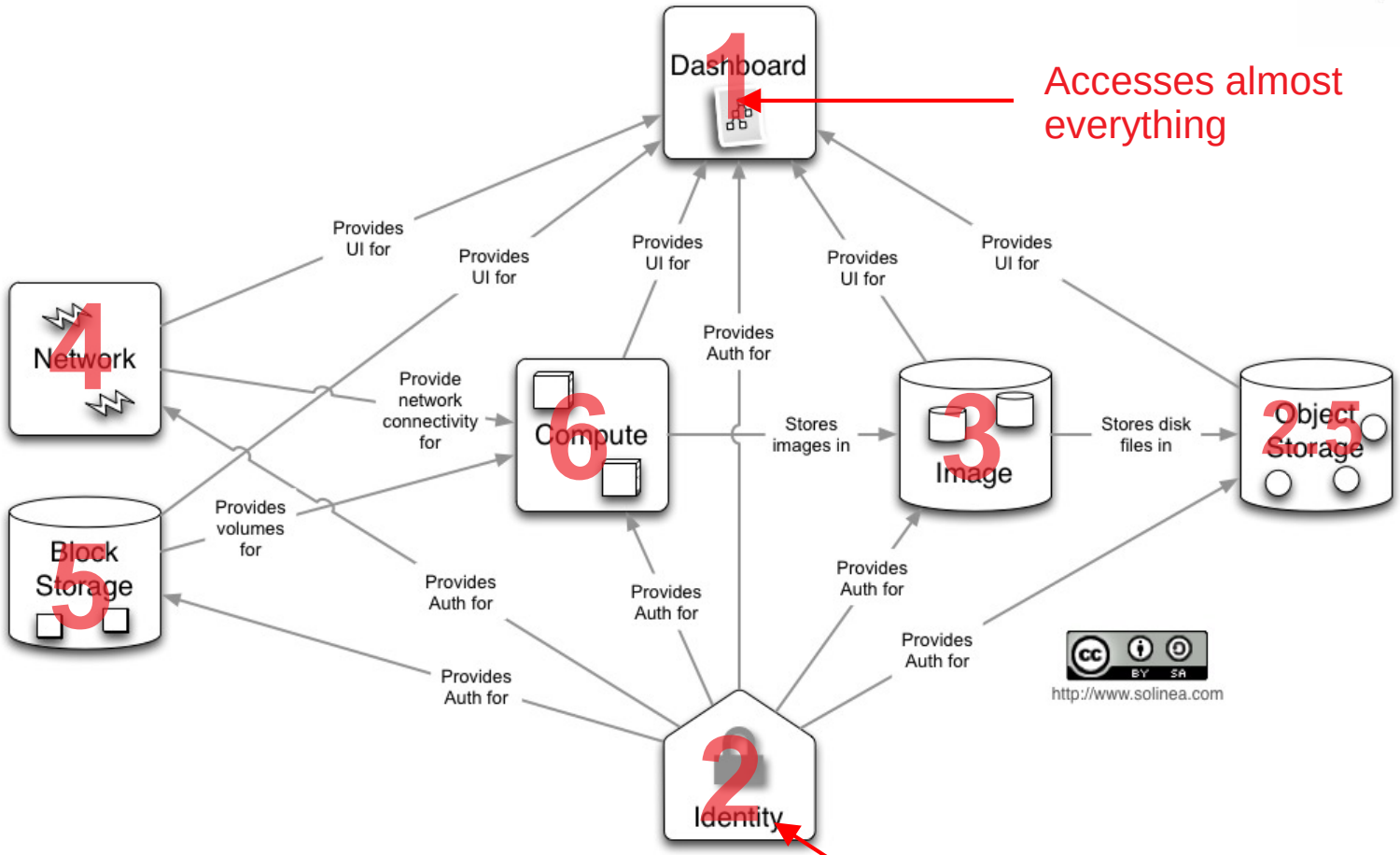
OpenStack Debugging Basics

- Log files often lack useful information without `verbose` enabled
- TRACEs of processes are not logged without verbose
- Many reasons for API error messages are not logged unless `debug` is turned on
- Debug is very verbose (>10GB per hour)

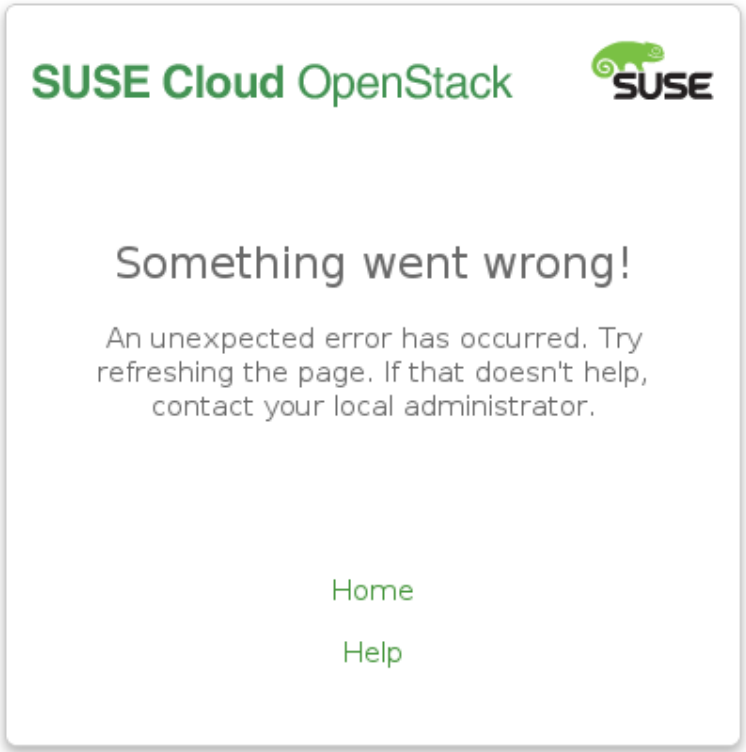
https://ask.openstack.org/

http://docs.openstack.org/icehouse/

OpenStack Architecture



OpenStack Dashboard: Horizon



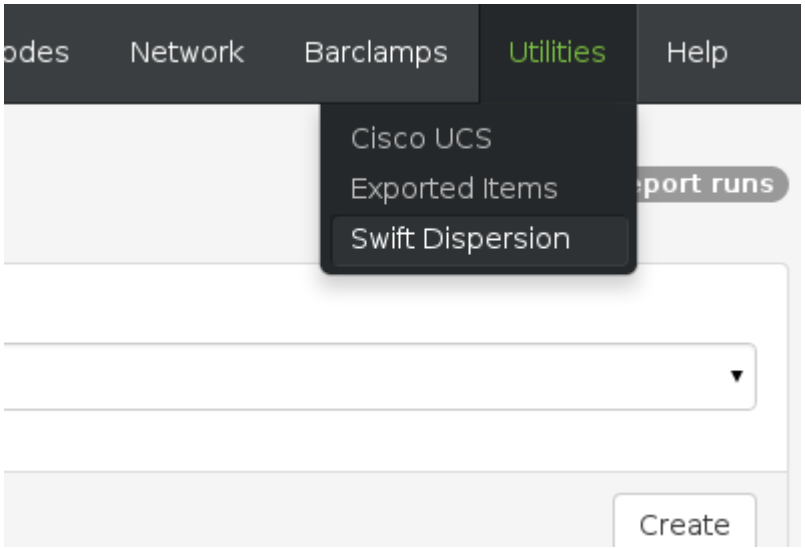
`/var/log/apache2/openstack-dashboard-error_log`

- Get the exact URL it tries to access!
- Enable “debug” in Horizon barclamp
- Test components individually

OpenStack Identity: Keystone

- Needed to access all services
- Needed by all services for checking authorisation
- Use `keystone token-get` to validate credentials and test service availability

OpenStack Object Store: Swift



`swift stat`

- swift dispersion in Crowbar
- uses regular syslog for many messages:

`/var/log/messages`

`console`

- easiest to debug using `curl`

OpenStack Imaging: Glance

- To validate lifeness:

```
glance image-list

glance image-download <id> > /dev/null

glance image-show <id>
```

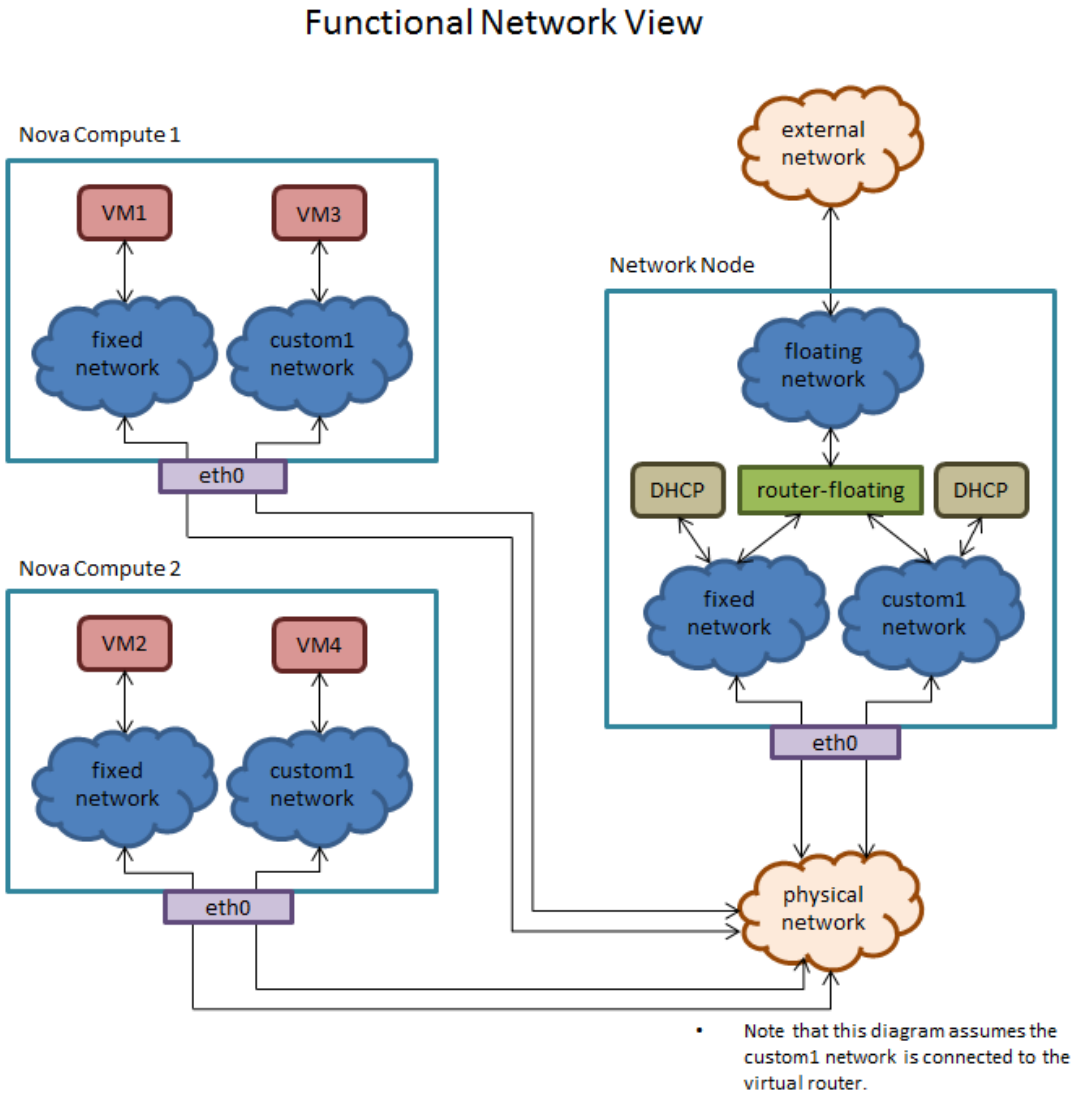
OpenStack Networking: Neutron

- Swiss Army knife for SDN

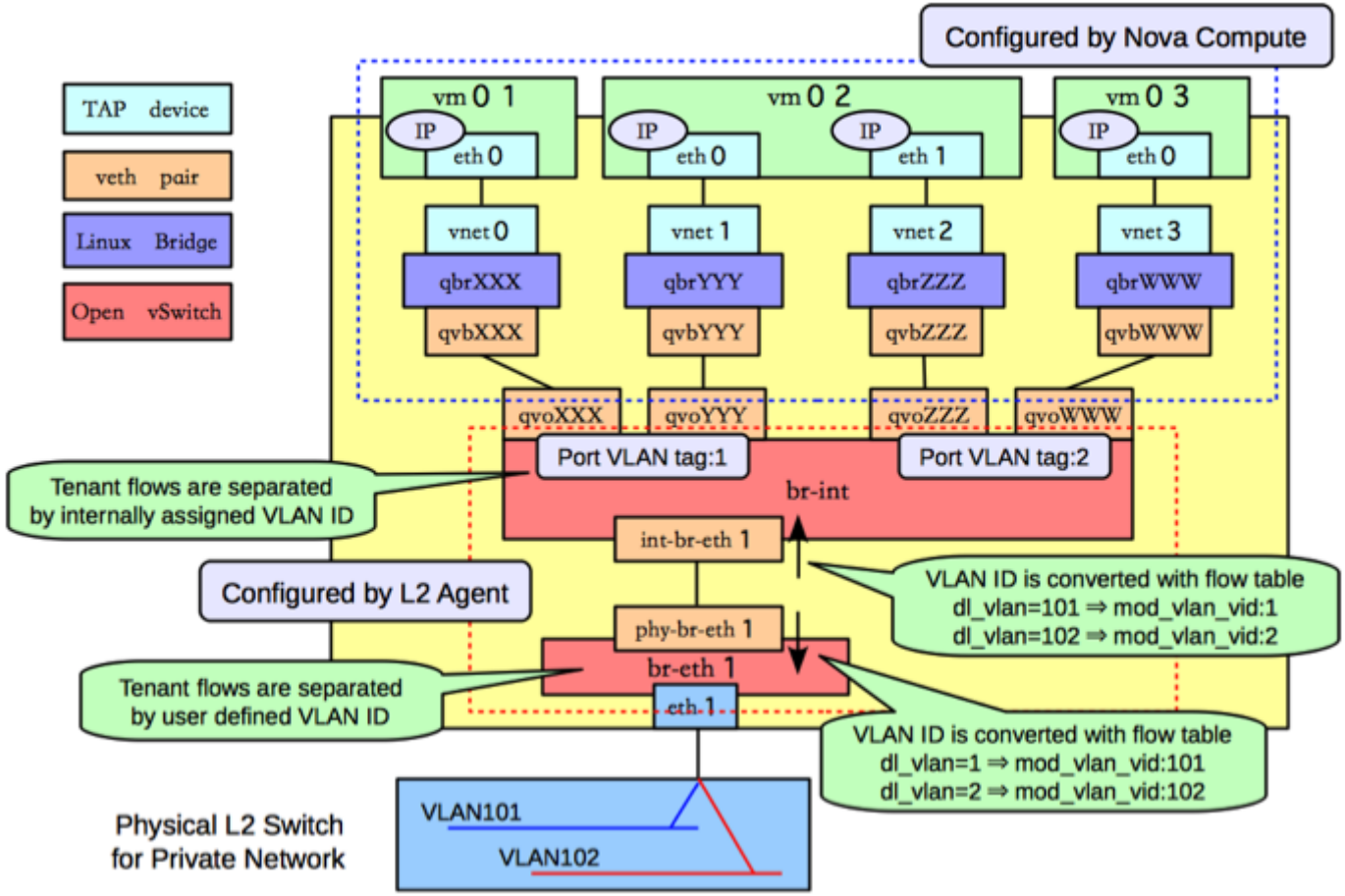
```
neutron agent-list
neutron net-list
neutron port-list
neutron router-list
```

- There's no `neutron-manage`

Basic Network Layout

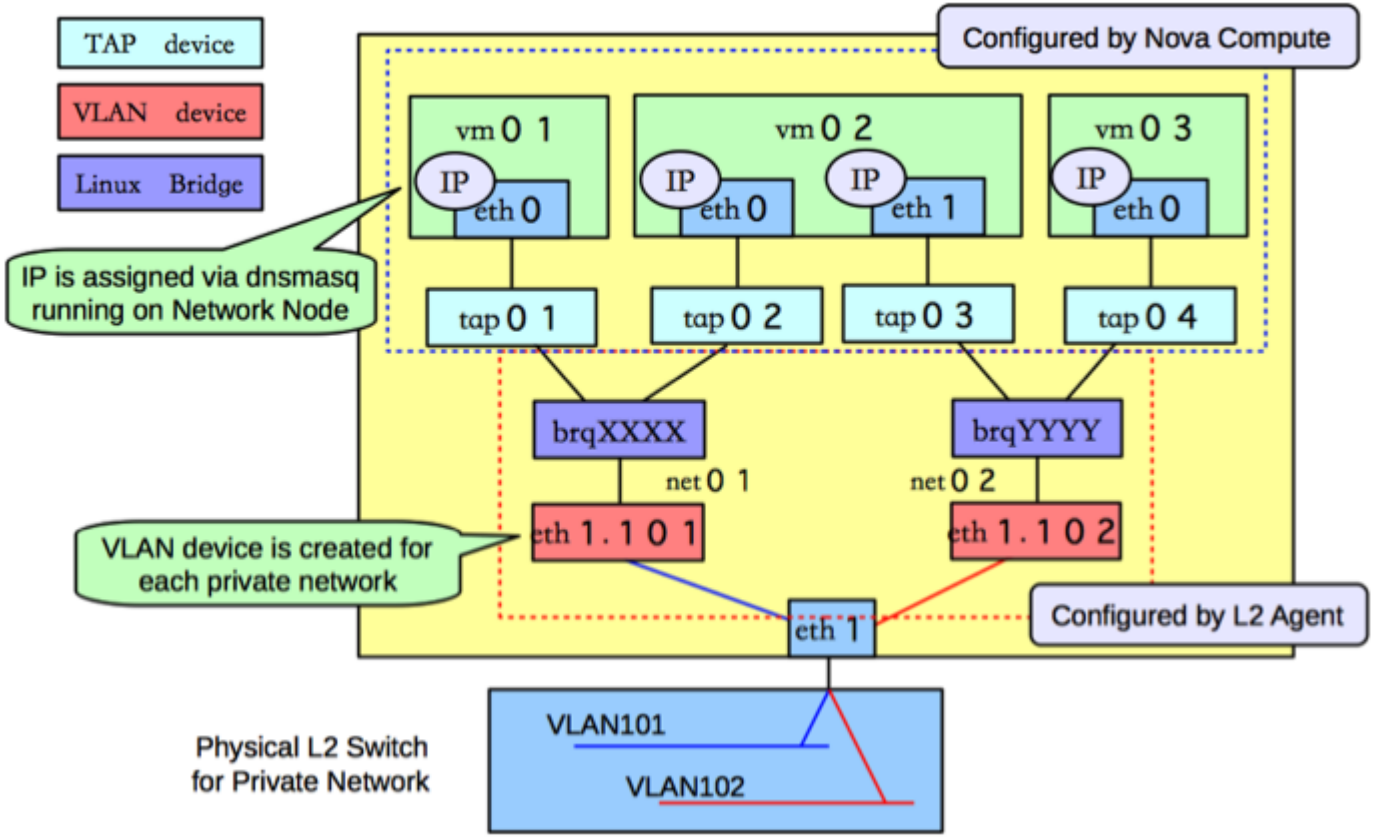


Networking with OVS: Compute Node



http://docs.openstack.org/havana/config-reference/content/under_the_hood_openvswitch.html

Networking with LB: Compute Node



Neutron Troubleshooting

Neutron uses IP Networking Namespaces on the Network node for routing overlapping networks

```
neutron net-list
ip netns list
ip netns exec qrouter-<id> bash
    ping.. arping.. ip ro.. curl ..
```

OpenStack Compute: Nova

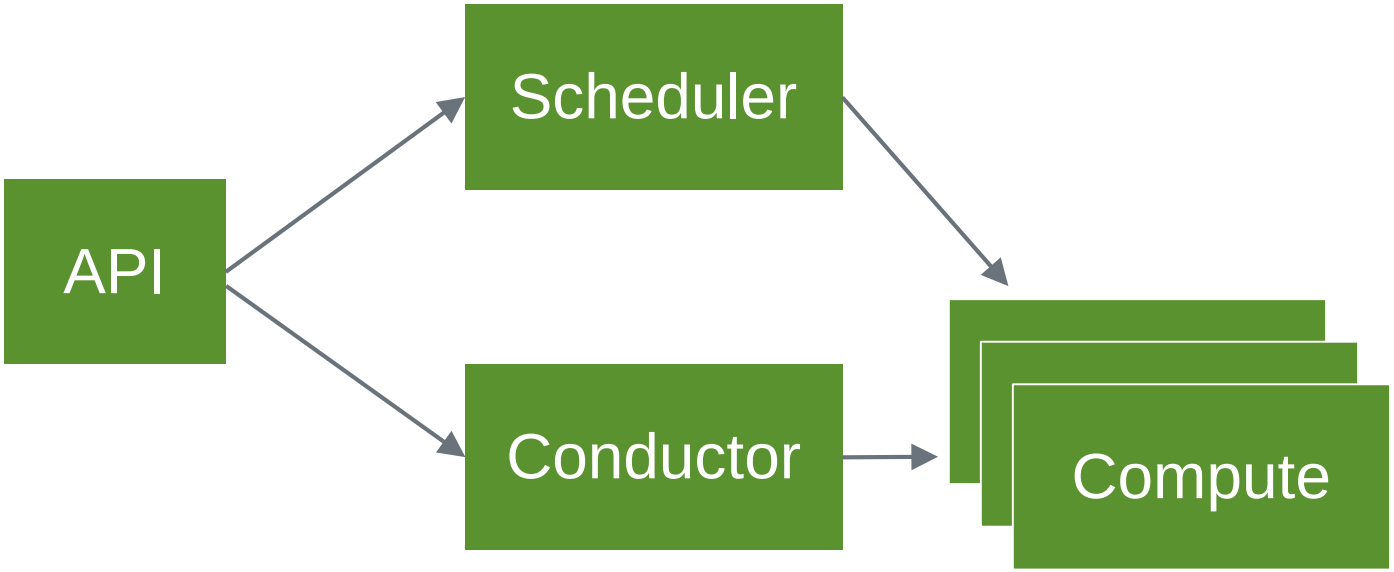
```
nova-manage service list
```

```
Binary           Host                 Zone             Status     State
nova-cert        dde-ad-be-ff-00-01   internal         enabled    :-)
nova-scheduler   dde-ad-be-ff-00-01   internal         enabled    :-)
nova-conductor   dde-ad-be-ff-00-01   internal         enabled    :-)
nova-compute     dde-ad-be-ff-1e-01   nova             enabled    :-)
nova-consoleauth dde-ad-be-ff-00-01   internal         enabled    :-)
nova-compute     dde-ad-be-ff-1e-02   nova             enabled    :-)
```

`nova show <id>` with admin privileges shows compute node

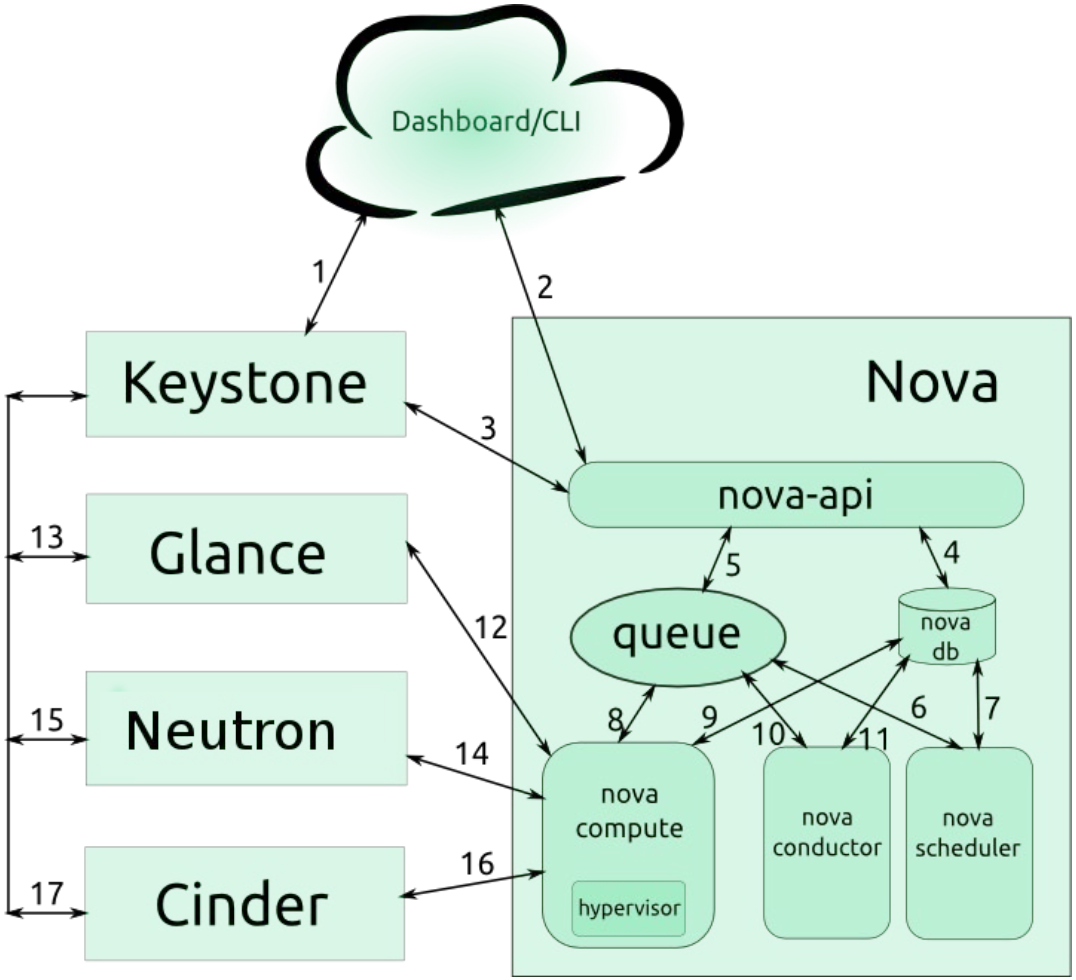
`virsh list` or `virsh dumpxml` can be used to analyze state of VM

Nova Overview



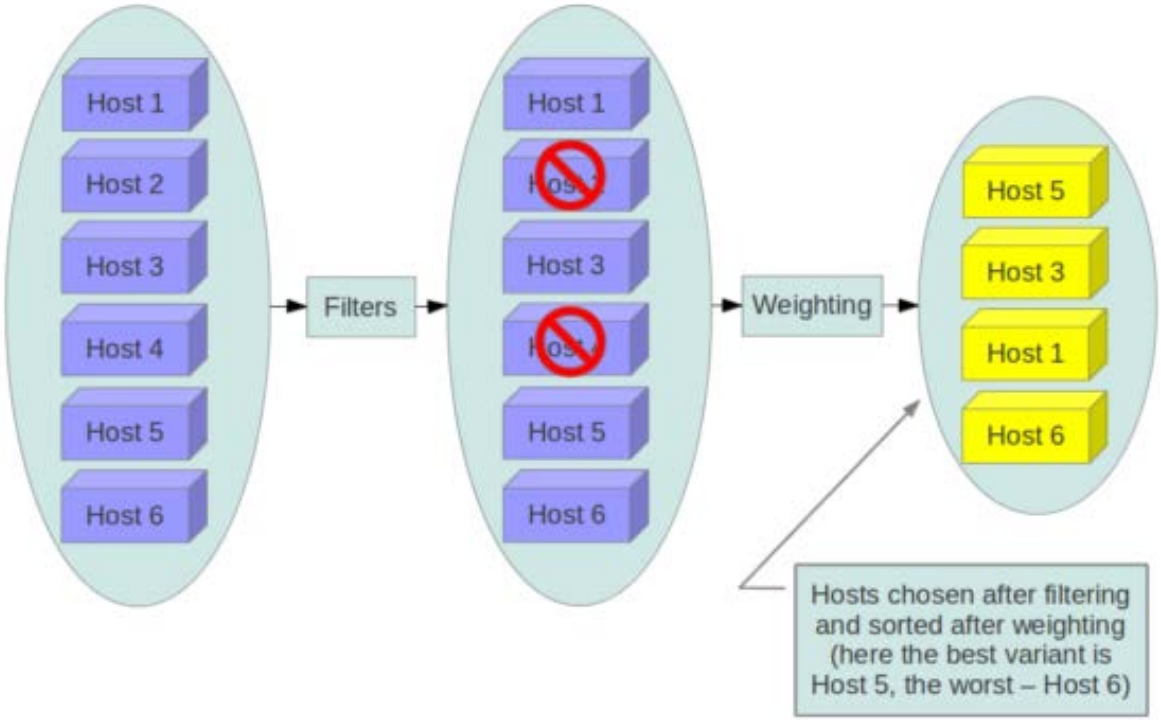
"Launches" go to Scheduler; rest to Conductor

Nova Booting VM Workflow



Nova: Scheduling a VM

- Nova scheduler tries to select a matching compute node for the VM



Nova Scheduler

Typical errors:

- `No suitable compute node can be found`
- `All suitable compute nodes failed to launch the VM with the required settings`
  - nova-manage logs errors

```
INFO nova.filters [req-299bb909-49bc-4124-
8b88-732797250cf5
c24689acd6294eb8bbd14121f68d5b44
acea50152da04249a047a52e6b02a2ef] Filter
RamFilter returned 0 hosts
```

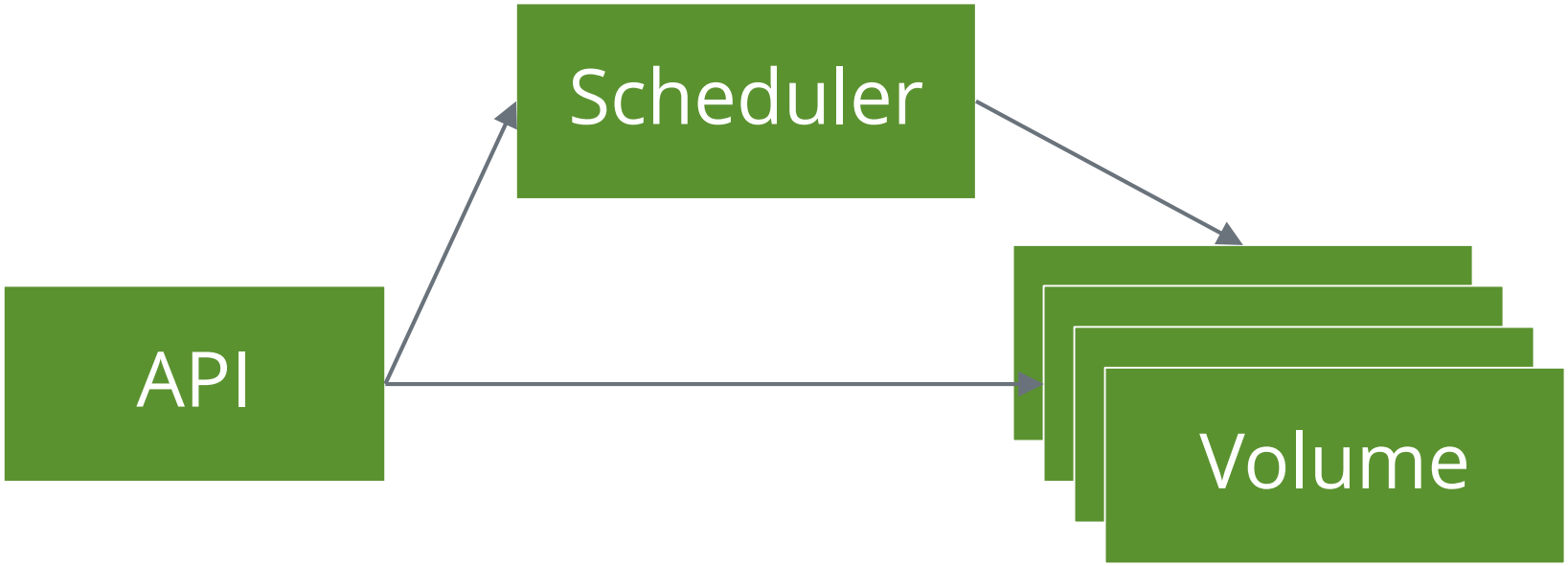
Nova Scheduler

Typical errors:

- No suitable compute node can be found
- All suitable compute nodes failed to launch the VM with the required settings
  - nova-manage logs errors

```
INFO nova.filters [req-299bb909-49bc-4124-8b88-
732797250cf5 c24689acd6294eb8bbd14121f68d5b44
acea50152da04249a047a52e6b02a2ef] Filter
RamFilter returned 0 hosts
```

OpenStack Volumes: Cinder



OpenStack Cinder: Volumes

Similar syntax to Nova:

```
cinder-manage service list

cinder-manage logs errors

cinder-manage host list
```

`cinder list|show` (with admin privs) shows volume-host

Troubleshooting Cloud-Init

- OpenStack Services like Heat or Nova depend on cloud-init
  - sets host name, ssh keys, resizes disks, launches custom scripts on boot
- Heat uses scripts to launch cfntools
- use `curl` on the metadata server inside the VM

```
/var/lib/cloud/

/var/log/cloud-init.log
```

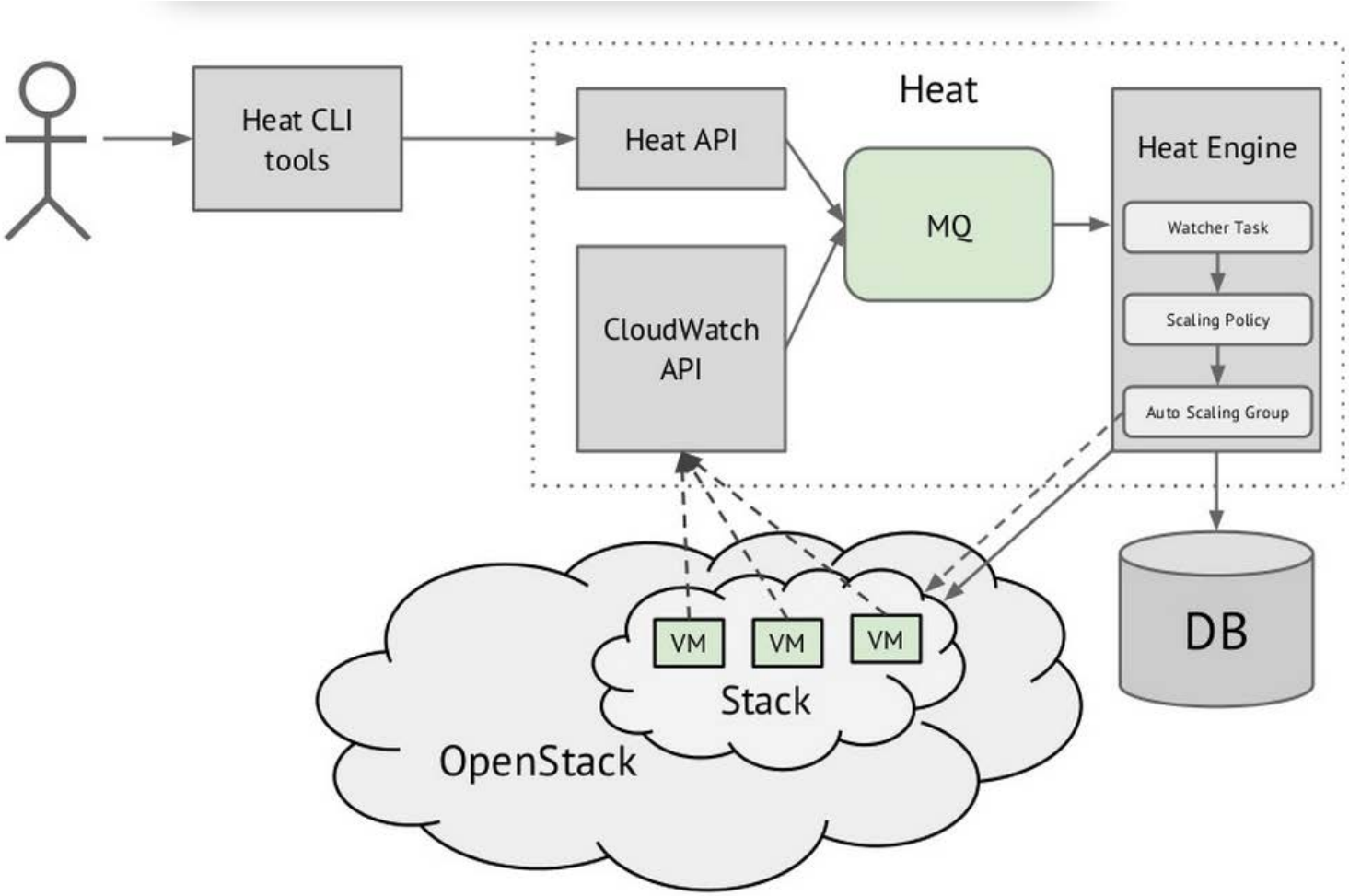
Q&A

- http://ask.openstack.org/
- http://docs.openstack.org/
- https://www.suse.com/documentation/suse-cloud4/

Thank you

Bonus Material

OpenStack Orchestration: Heat



OpenStack Orchestration: Heat

- Uses Nova, Cinder, Neutron to assemble complete stacks of resources

```
heat stack-list

heat resource-list|show <stack>

heat event-list|show <stack>
```

- Usually necessary to query the actual OpenStack service for further information

OpenStack Imaging: Glance

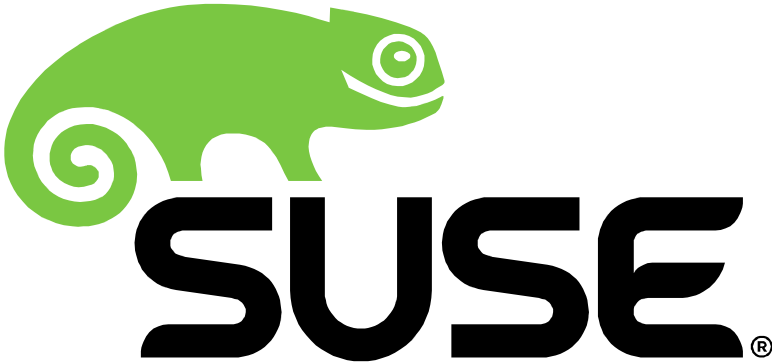
- Usually issues are in the configured glance backend itself (e.g. RBD, swift, filesystem) so debugging concentrates on those
- Filesytem:

```
/var/lib/glance/images
```

- RBD:

```
ceph -w
rbd -p <pool> ls
```

SUSE® Cloud


SUSE®
We adapt. You succeed.