CANONICAL

How Juju makes cloud and Devops easy
CloudOpen Japan 2014

Yaguang Tang

yaguang.tang@canonical.com
2014.05.20

Agenda

- Who am I?
- All about Devops
- Juju introduction
- Juju's internals
- Juju Charms

Who am I?

- Software Engineer at Canonical
- OpenStack Active Technical Contributer since Essex(2012)

Yaguang Tang

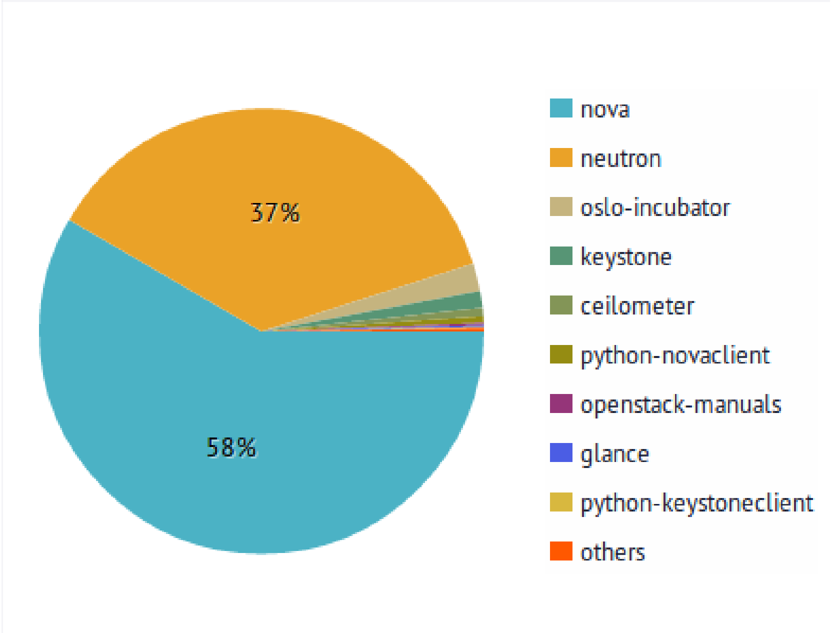
Company: Canonical [*how to change* ↗]
Launchpad: heut2008 ↗

Activity Log

Yaguang Tang (Canonical)
18 Jan 2014 14:36:35 in nova
Commit "Change RPC post_live_migration_at_destination from call to cast"
We don't expect post_live_migration_at_destination to return anything, so change call to cast.
Change-Id: Iccdb6e195d1d5d9b96e1372206823d414caae872 ↗
Commit date: 19 Dec 2013 09:44:27
+3 - 4

Yaguang Tang (Canonical)
15 Jan 2014 21:26:00 in nova
Commit "Optimize libvirt live migration workflow at source"

Contribution by modules



CANONICAL

What is DevOps?

- Rate of agile development and deployment requires deeper interaction between teams
- A melding of development, deployment, and QA principles, methods, and practices
- Fills the gap between developers and system administrators

CANONICAL

What drives DevOps?

- Speed of the deployment
- Continuous Integration, Automated Testing, etc.
- Fast change vs. Stability

CANONICAL

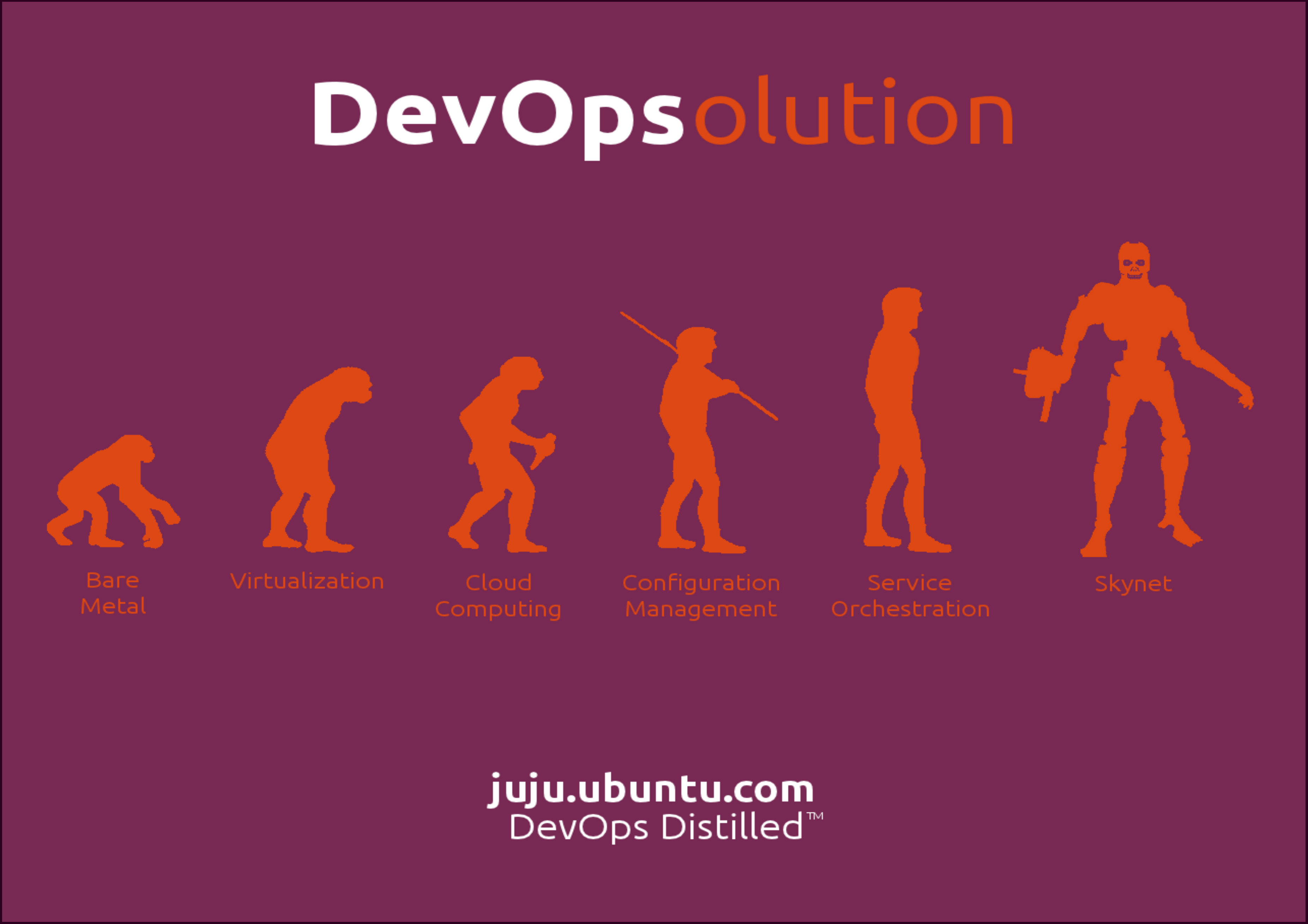
What does DevOps “deliver”?

- Fast repeatable server setup, consistent environment
- Abstract ops tasks to empower devs
- Smaller deployments empower ops
- Repeatable processes that let you scale out quickly

You've got the tools already

- Hardware
- Virtualization
- Platform (OS)
- Configuration Management

… need to tie that together into something whole.


DevOpsolution
Bare Metal
Virtualization
Cloud Computing
Configuration Management
Service Orchestration
Skynet
juju.ubuntu.com
DevOps Distilled™

CANONICAL

juju

CANONICAL

Automate your cloud infrastructure
Configure, manage, maintain,
deploy and scale efficiently with
best-practice Charms on any public,
private or hybrid cloud from a
powerful GUI or the command-line.

CANONICAL

So juju is ...

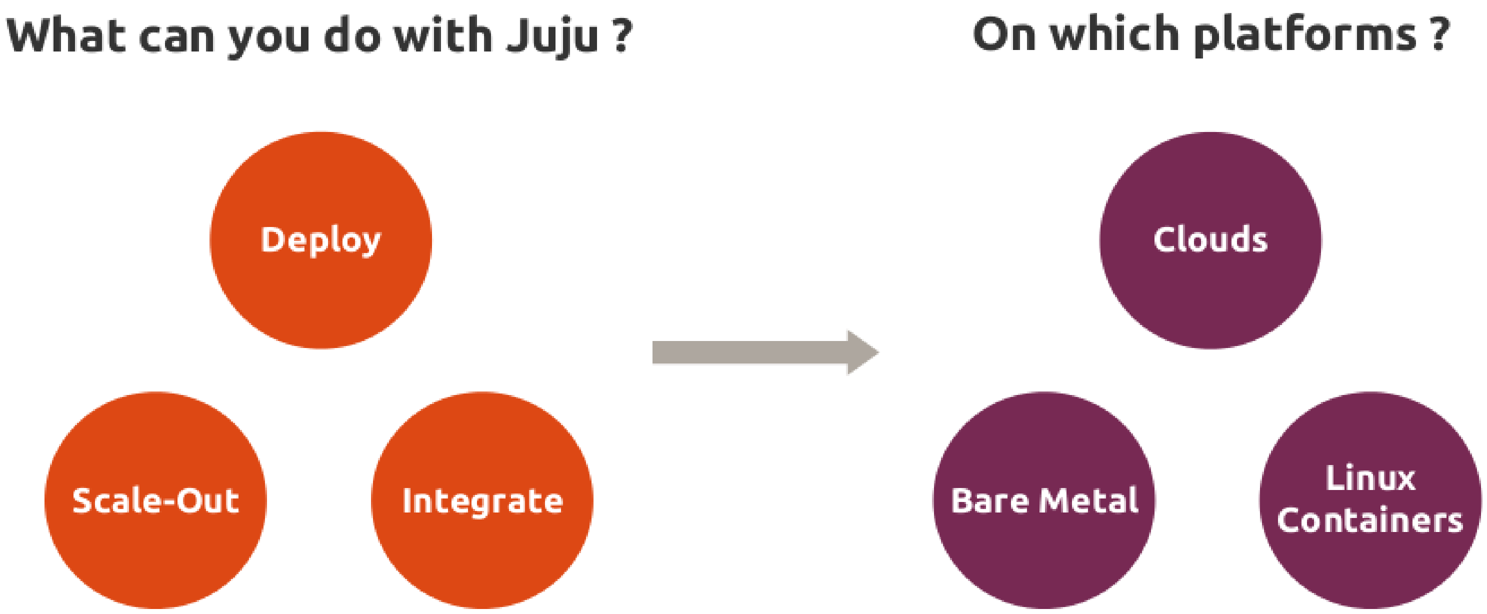
Like apt-get, but for sets of machines
Charms do all the work for you.
Connected to a Charm Store of community contributed charms that you can deploy.

CANONICAL

Juju Manages Services, not Machines

CANONICAL

Application deployment at radical velocity



CANONICAL

- Using Juju deploy your application on Cloud
- Joyent
- Amazon EC2
- HP Cloud
- Azure
- LXC containers
- Vagrant
- Your VPS running Ubuntu and openssh (Digital Ocean)

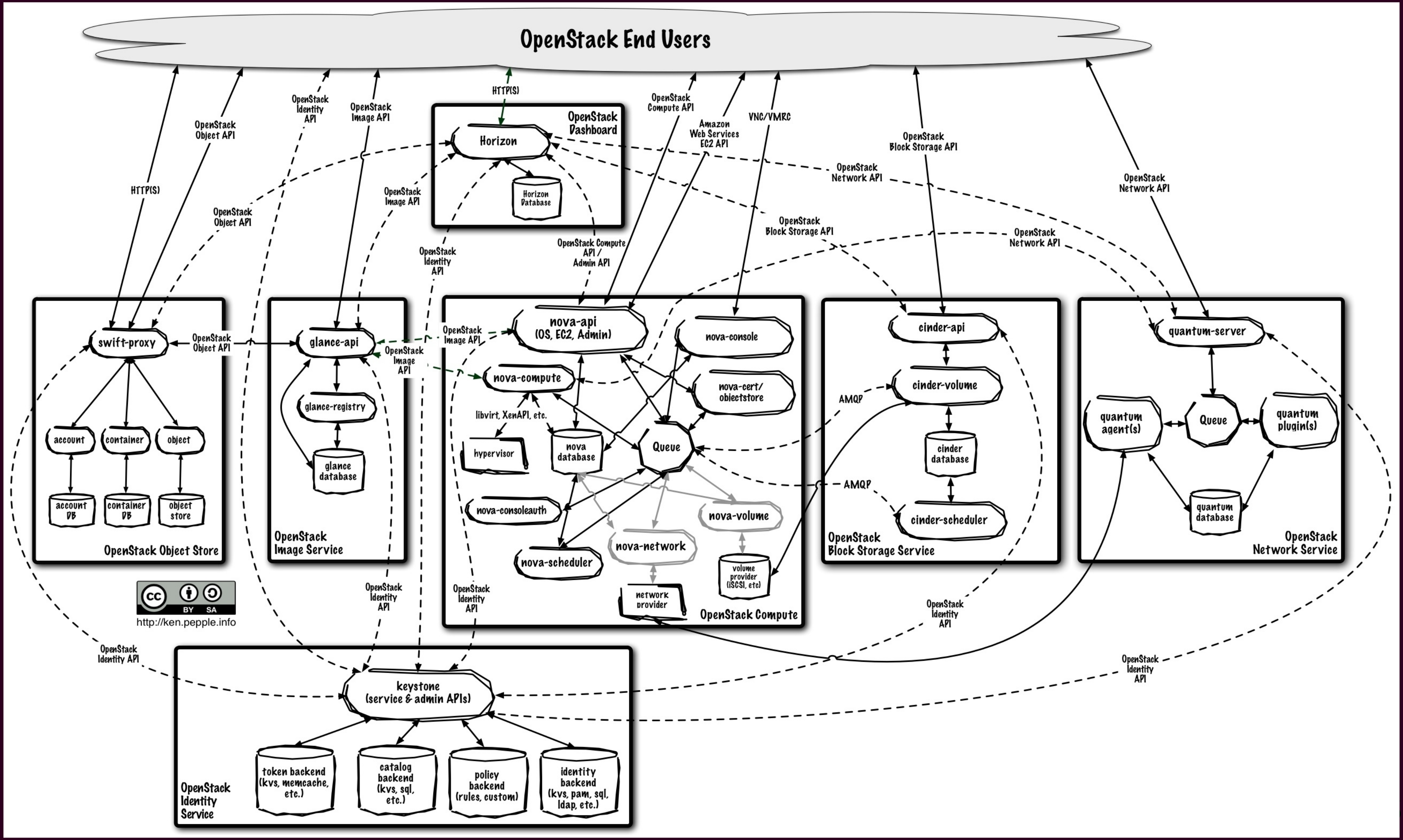
Juju provides service orchestration

- Juju focuses on managing the service unit you need to deliver a single solution, above simply configuring the machines or cloud instances needed to run them.

- Juju exposes re-usable service units and well defined interfaces that allow you to quickly and organically adjust and scale solutions without repeating yourself.

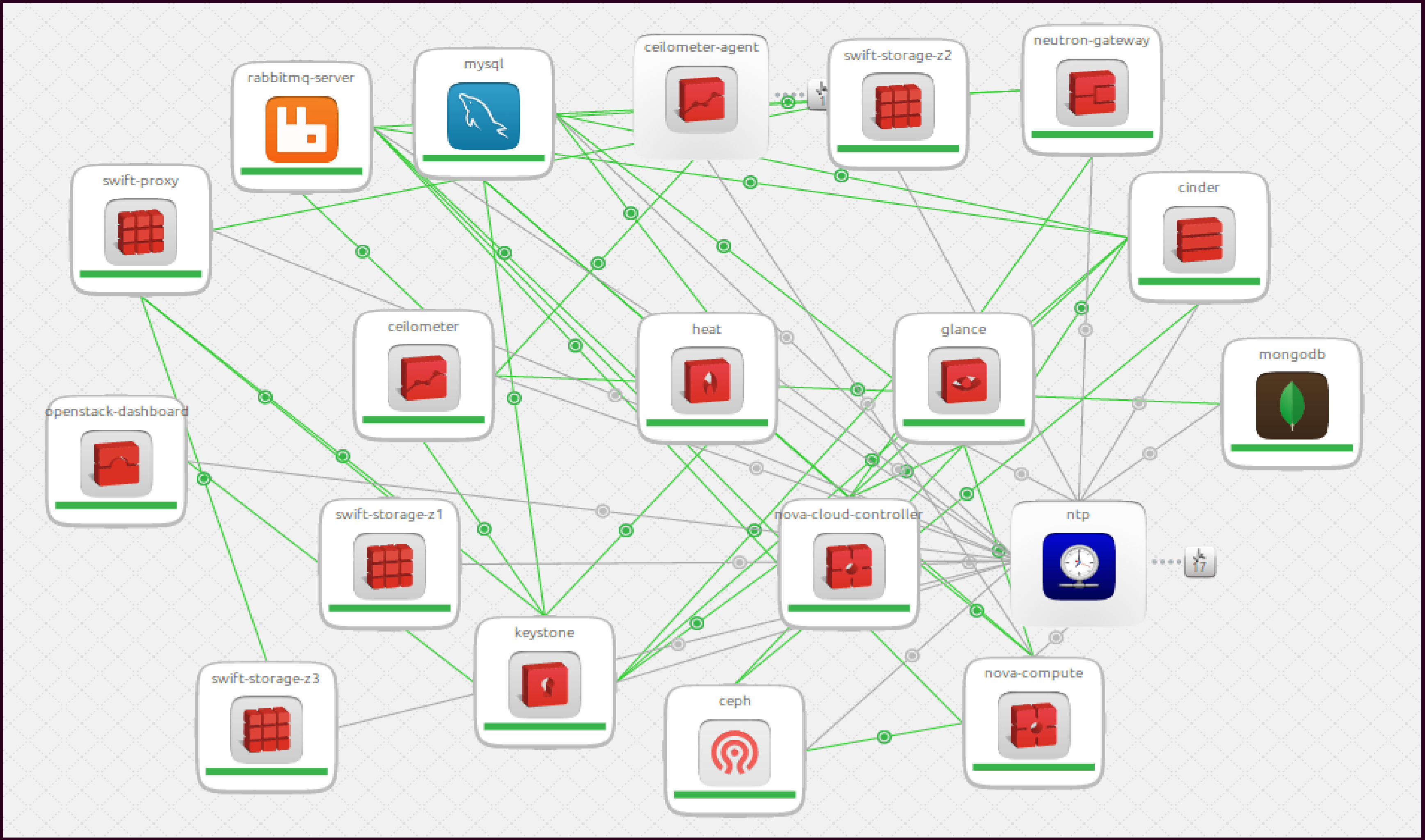
CANONICAL

CANONICAL OpenStack services



CANONICAL

Deployed by Juju



CANONICAL

Deploy your OpenStack cloud with Juju

```
juju deploy mysql
juju deploy rabbitmq-server
juju deploy --config=openstack.cfg keystone
juju deploy --config=openstack.cfg nova-cloud-controller
juju deploy --config=openstack.cfg nova-volume
juju deploy nova-compute
juju deploy glance
juju deploy openstack-dashboard
```

https://help.ubuntu.com/community/UbuntuCloudInfrastructure

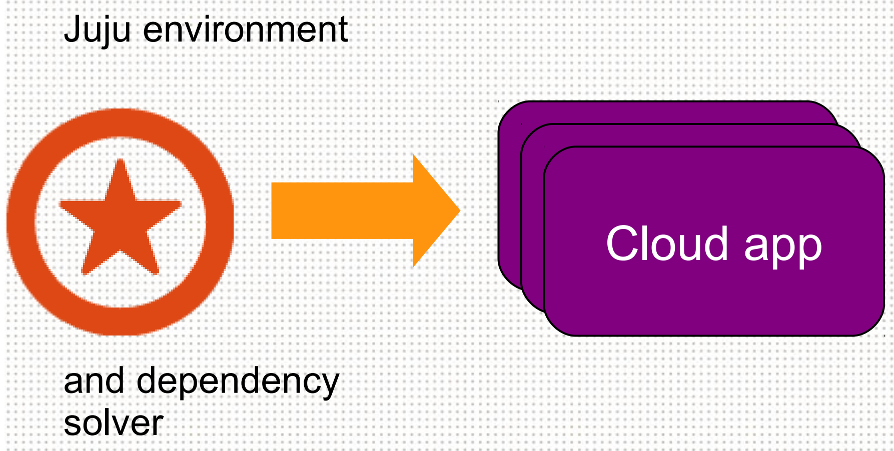
CANONICAL

- juju add-relation keystone mysql
- juju add-relation nova-cloud-controller mysql
- juju add-relation nova-cloud-controller rabbitmq-server
- juju add-relation nova-cloud-controller glance
- juju add-relation nova-cloud-controller keystone
- juju add-relation nova-volume nova-cloud-controller
- juju add-relation nova-volume mysql
- juju add-relation nova-volume rabbitmq-server
- juju add-relation nova-compute mysql
- juju add-relation nova-compute rabbitmq-server
- juju add-relation nova-compute glance

CANONICAL

Juju's internals

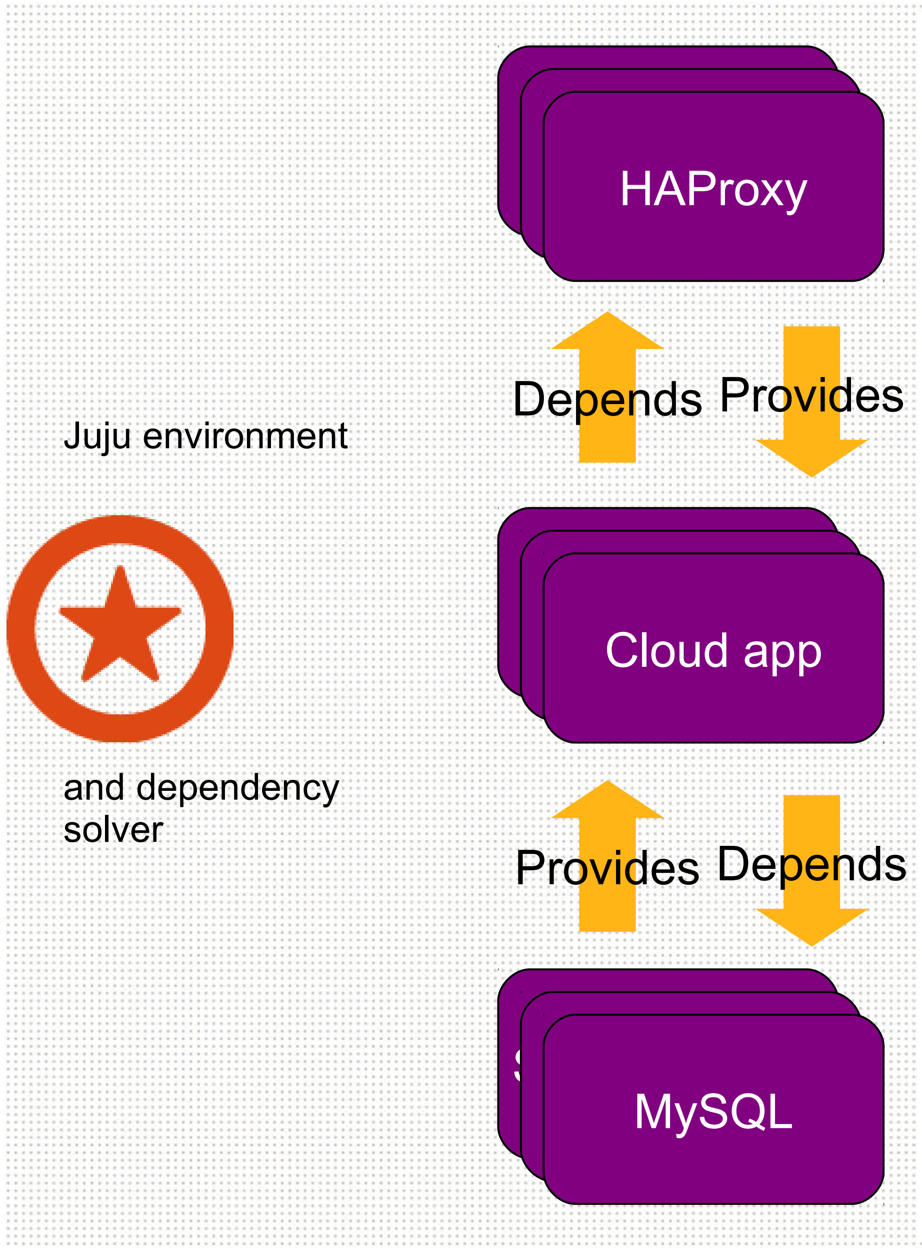
Juju



Juju treats individual services as atoms that are described as formulas and can be instantiated one or many times.

CANONICAL

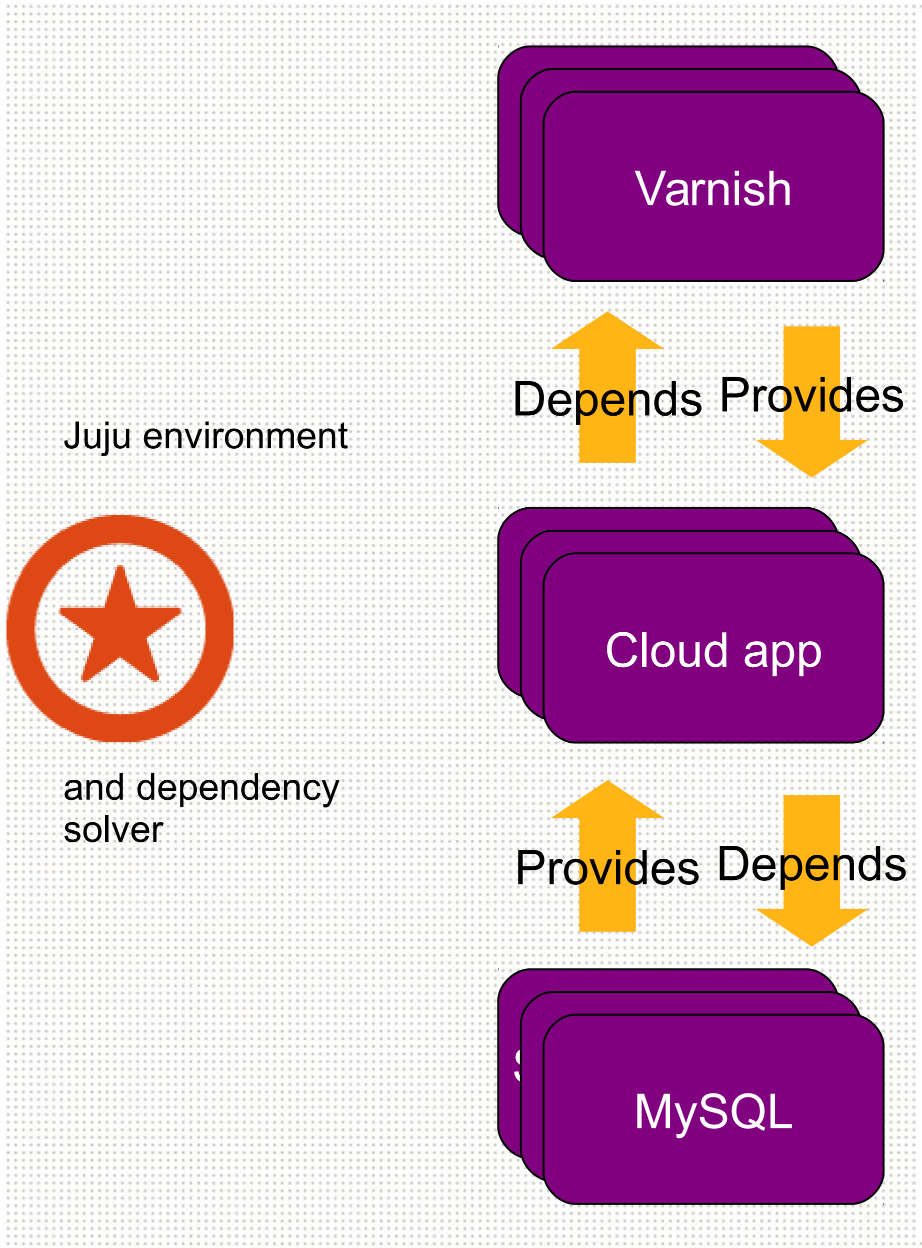
Juju



Each formula (or atom) define dependencies and/or provides.

CANONICAL

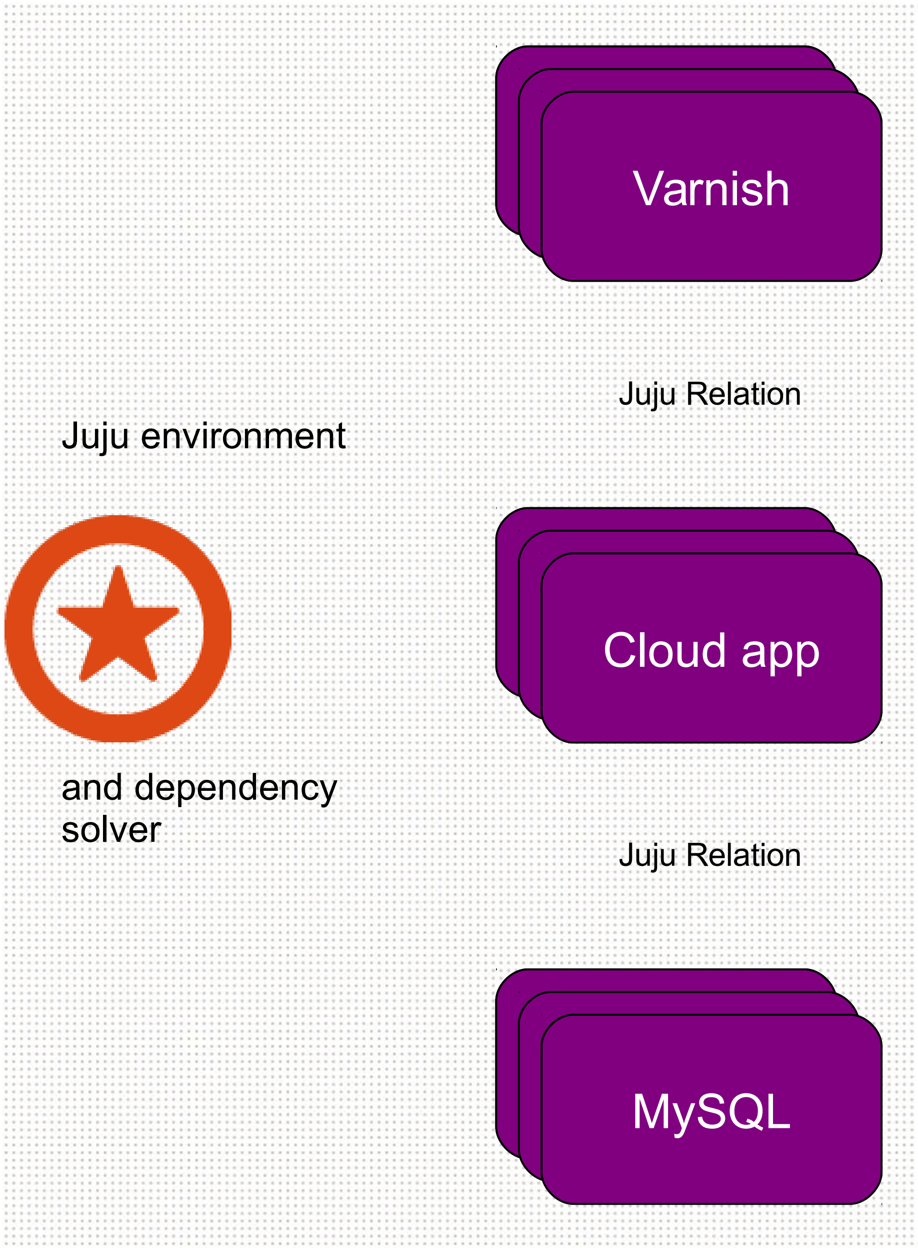
Juju



Multiple formulas can provide the same service and can be easily switched.

CANONICAL

Juju



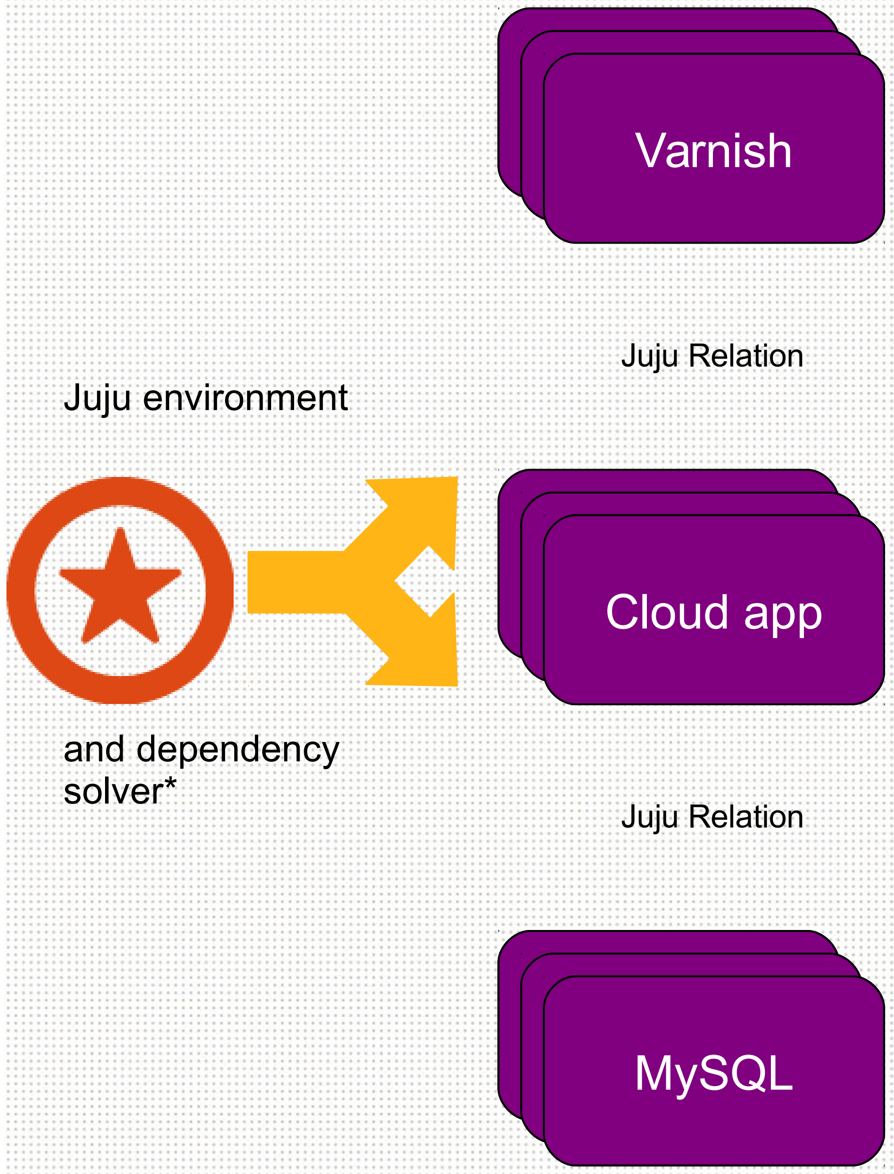
Juju maintains the relations between the services so that you don't need to care about the elasticity of your environment.

Relations are to formulas what bounds are to atoms.

Services are loosely coupled but highly cohesive.

CANONICAL

Juju



Juju delivers service focused management through their life-cycle

- Offers the same simple rules to components of you infra as we do already for packages on your servers: dependencies, provides
- Adds the notion of dynamic relations between components
- To provide you with simple automated elasticity that is easy to expand
- Working on your bare metal servers (through Orchestra*) as easily as on your favourite clouds (AWS, OpenStack*, ...)

*coming soon

CANONICAL

- Charms
- Scalable application services defined

- Charms give Juju its power. They encapsulate application configurations, define how services are deployed, how they connect to other services and are scaled. Charms are easily shared and there are 100s of Charms already rated and reviewed in our Charm store.

Charm

- Reusable, codified best-practice.
- Distilled deployment expertise.
- Communication via interfaces.
- Doesn't require foreknowledge of who will use them or how

Relations

- A high-level interface describing the interactions between services
- Services have `provides` and `requires` interfaces
- Juju models the relationship between services, not machines

CANONICAL

Scaling services

```
$ juju bootstrap
$ juju deploy hadoop-master
$ juju deploy hadoop-slave
$ juju add-relation hadoop-master hadoop-slave
$ juju add-unit hadoop-slave
$ juju add-unit hadoop-slave
```

CANONICAL

- Inside a Charm
- Charms define how services integrate and how their service units react to events in the distributed environment, as orchestrated by Juju. Charms can be written in any language that runs on Ubuntu. To pull it altogether, you just need to create a simple metadata.yaml file that defines the Charms' metadata, interfaces, hooks and requires.

Charm your application

1. Create new Charms based on templates
2. Use bash, python, perl, php or just about any other language supported on Ubuntu to write your Charm
3. Reuse any Puppet or Chef script you have
4. Develop store and track on Launchpad, github or your own repository

CANONICAL

CANONICAL

Thanks!

- juju.ubuntu.com
- github.com/juju
- #juju on Freenode