



CANONICAL

Service Orchestration for Cloud Environments with Juju

USENIX Configuration Management Summit
June 12, 2012

Jim Baker <jim.baker@canonical.com>

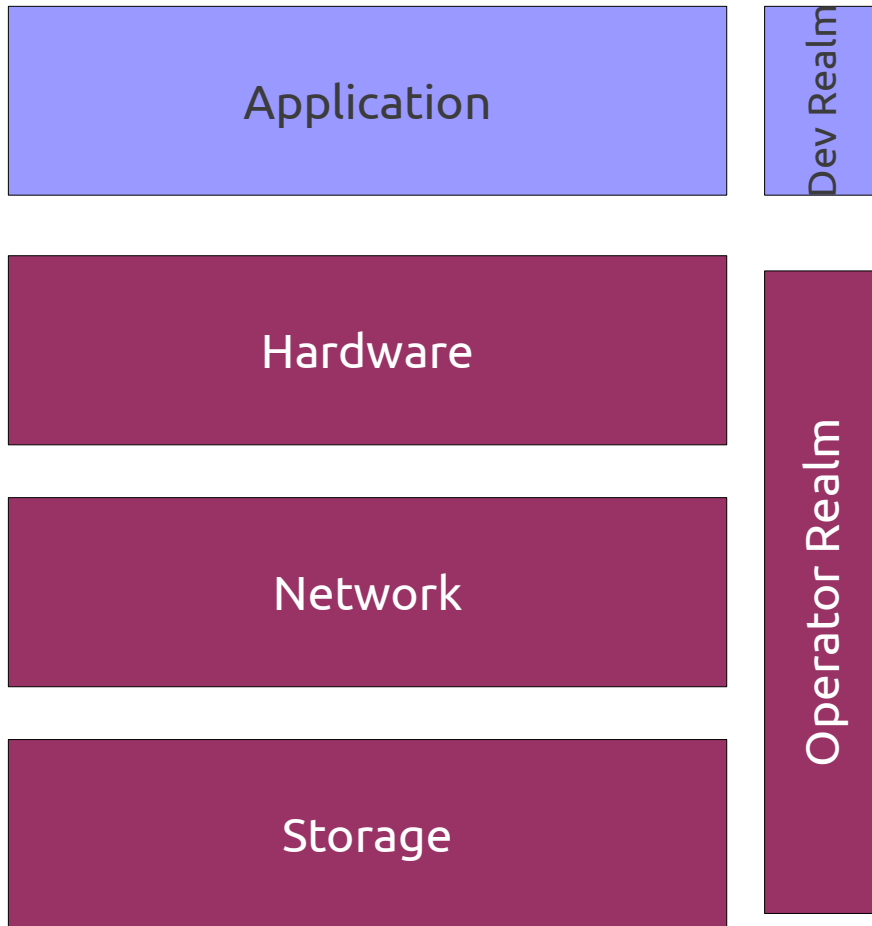
Ubuntu Server Team

Cloud deployments



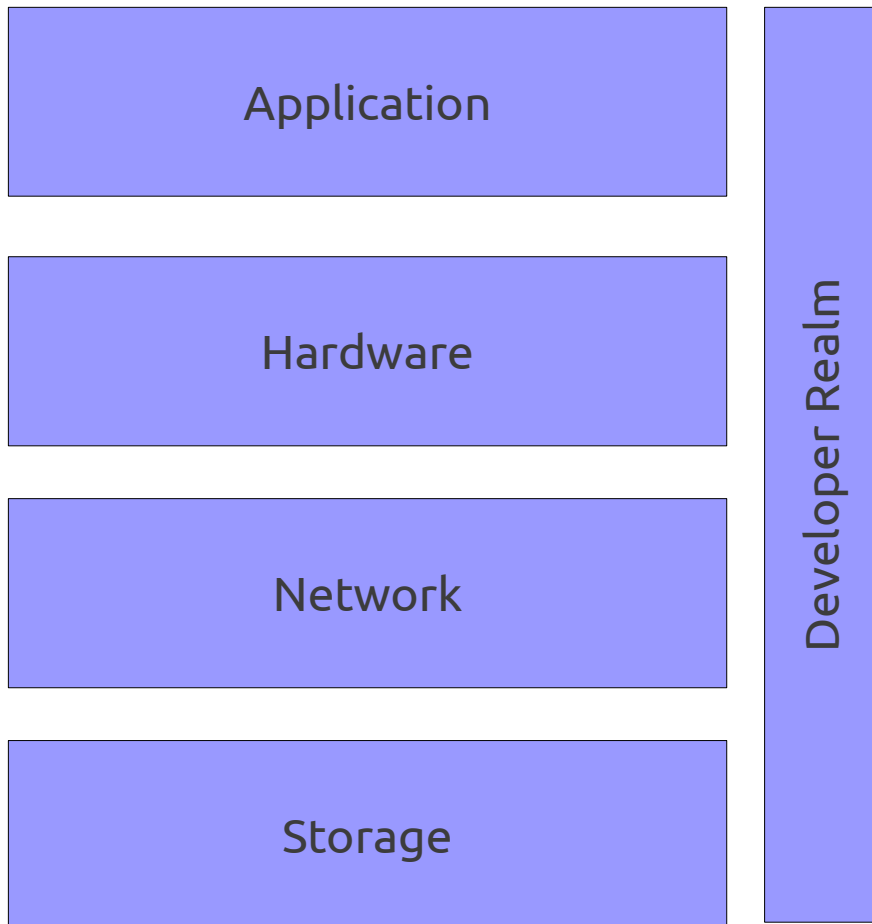
- Deploying in the cloud → the server is not the proper unit anymore
 - How the service scales horizontally is the complexity
 - Current configuration tools are clumsy as they still focus on servers
-
- Juju focuses on the services
 - Creates servers when and where needed
 - Hides the complexity of horizontal scalability in clean reusable charms

Old Habits



- Assume that the infra is resilient
- Assume that the hardware scales vertically
- Assume that SQL is the only DB method, and that it will grow with the app
- SPOF and scalability are a data center issue

New Habits



- Applications can scale horizontally
- Application can spawns additional instance machines based on needs
- Any instance can fail, but you can always launch another one
- Data can scale horizontally too if you use NoSQL databases or sharding
- Applications can run across data centers (availability zones)
- Application control their storage

APIs your application can use – AWS as an example



- Amazon's API offers control
 - The infrastructure (EC2)
 - Elemental/Object storage (S3)
 - Block storage (EBS)

 - Instances have access to meta-data and user-data
 - Can query about environment
 - Can learn about self
- Developers are now operators of the data center: **DevOps**



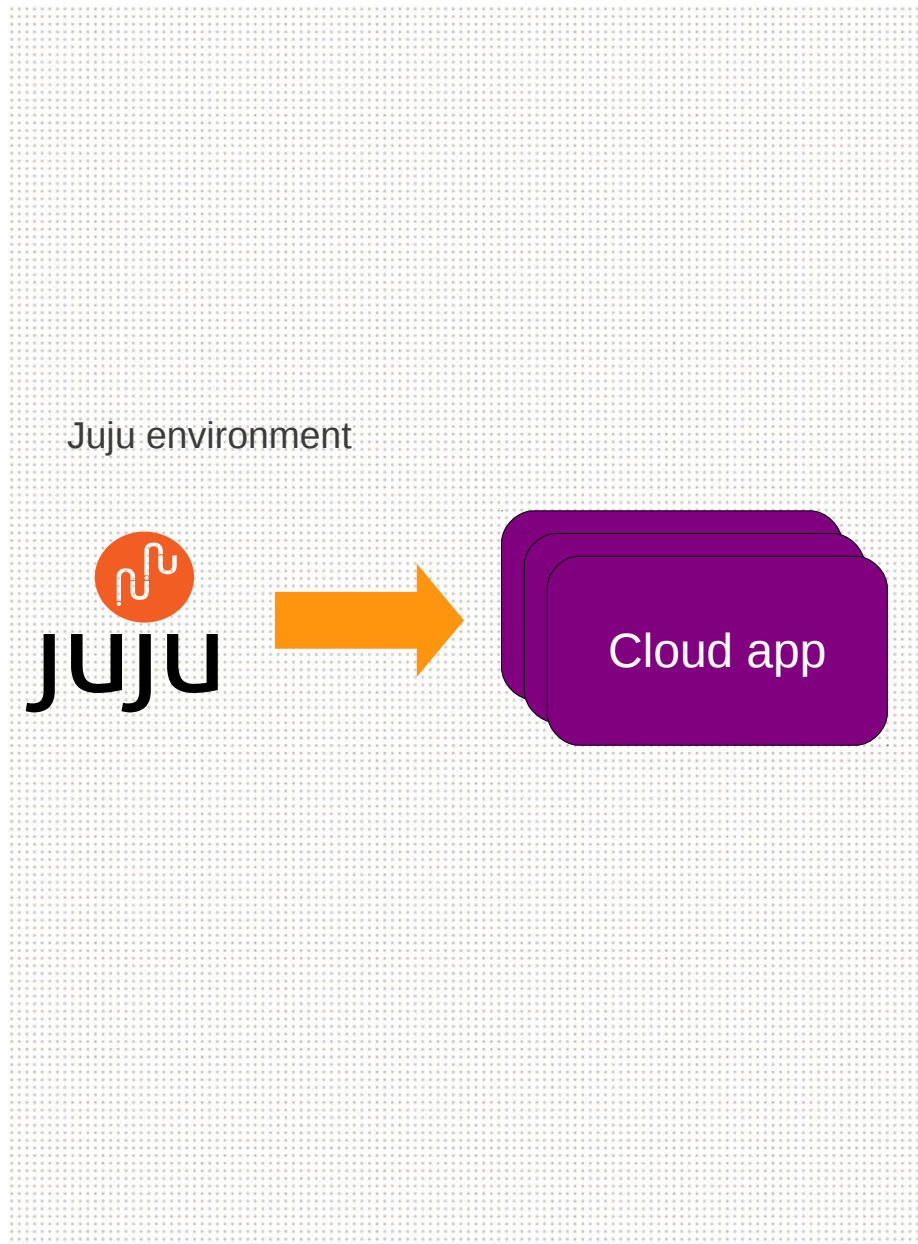
CANONICAL

But still not easy...
still need to configure, discover, scale
while applying best practices

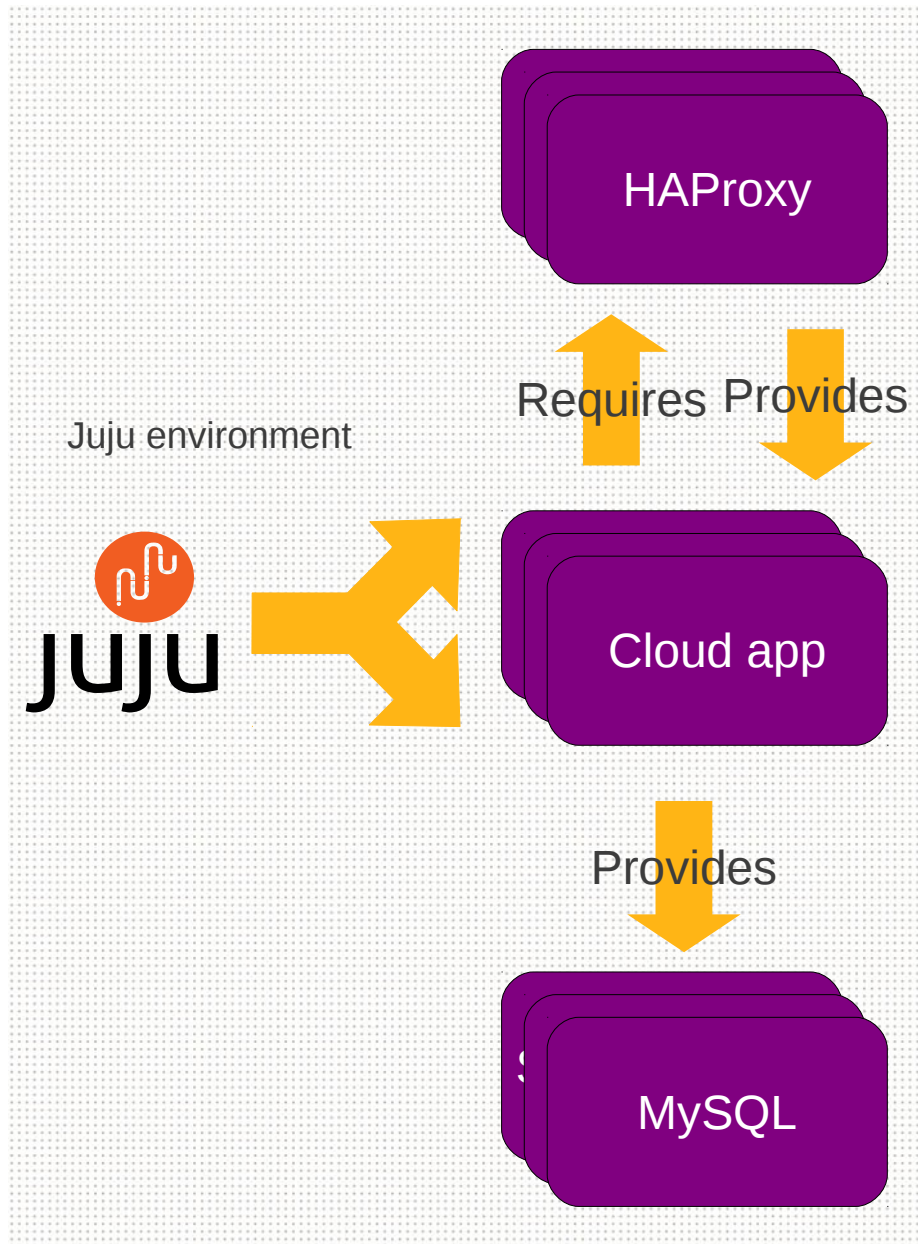


CANONICAL

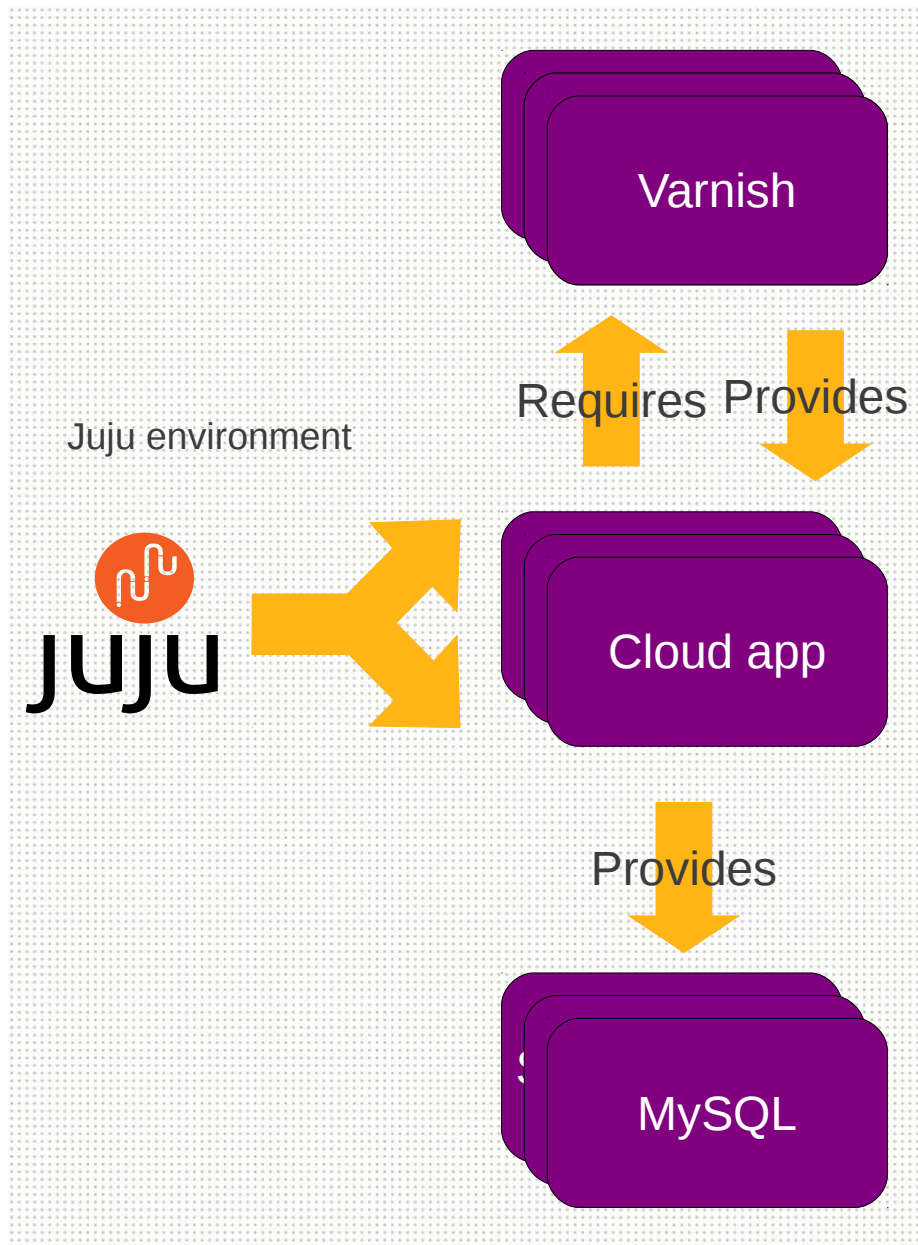
Juju is DevOps distilled



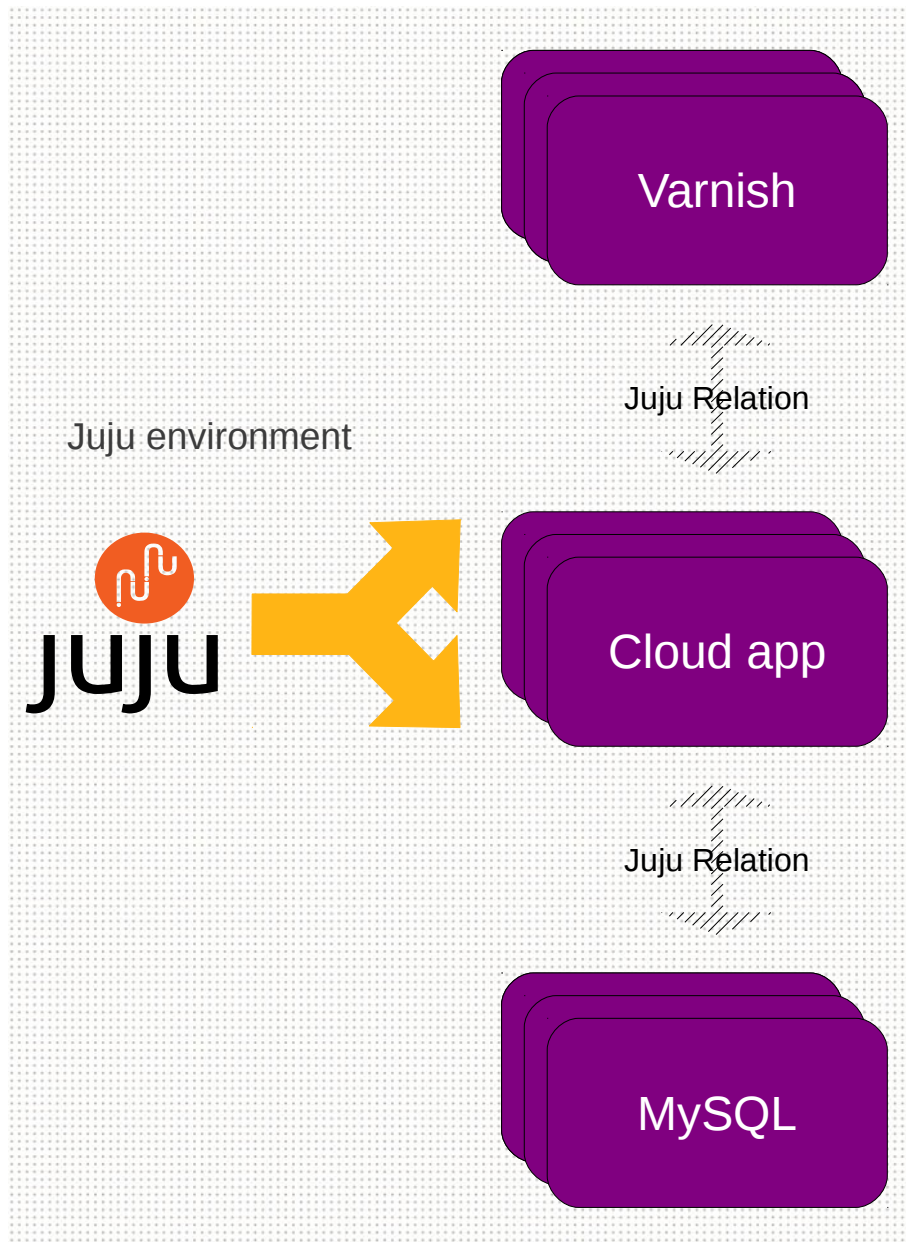
Juju treats individual services as atoms that are described as charms and can be instantiated one or many times.



Each charm defines its relationships: provides and/or requires



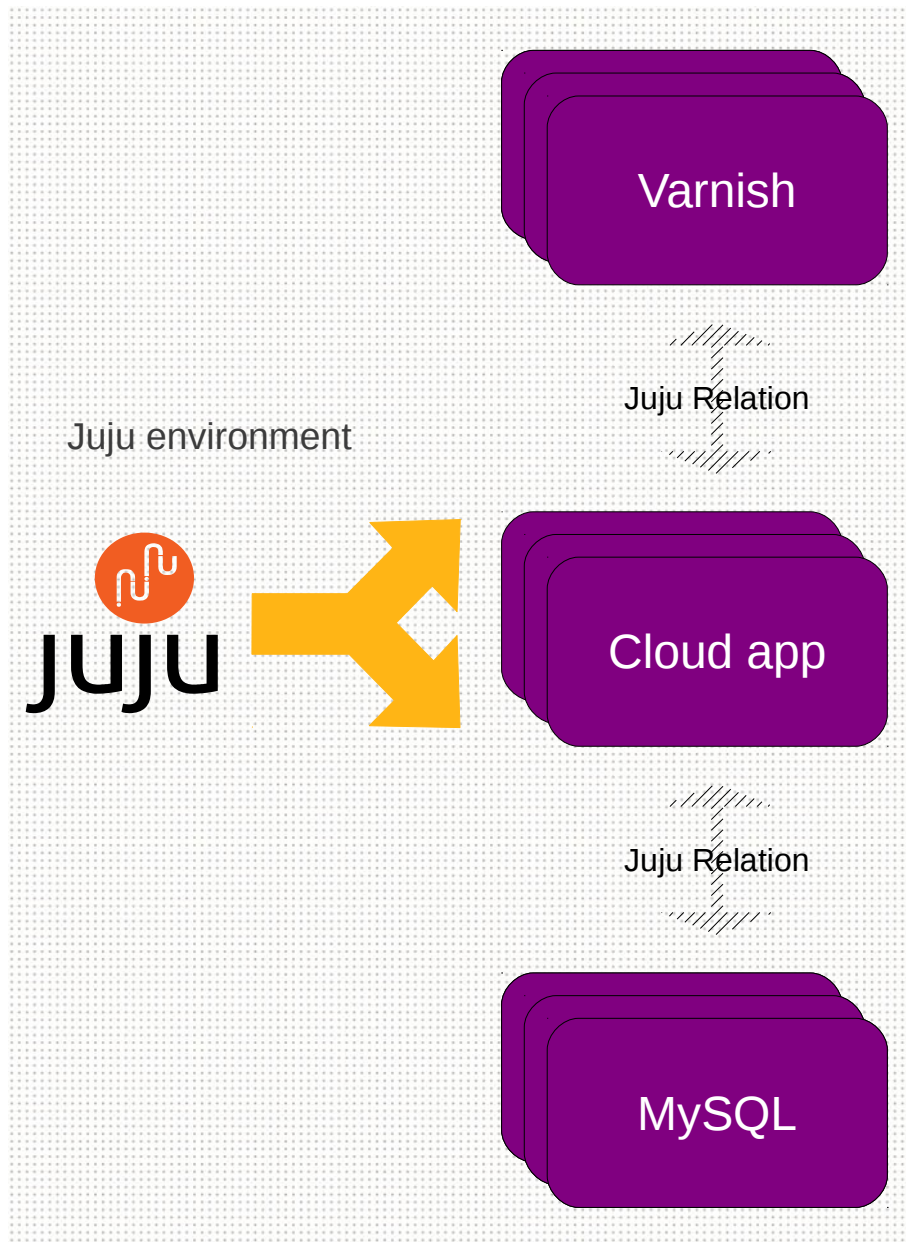
Multiple charms can provide the same service and can be easily switched.



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

Units of services negotiate as either the scaling changes (membership) or settings change.

Services are loosely coupled but highly cohesive.



Juju delivers service focused management through their life-cycle

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

Next steps



-
- All development is public
 - Communication is open

Join Us

- IRC: **#juju** **#juju-dev** and **#ubuntu-server** on irc.freenode.net
- Launchpad: **<https://launchpad.net/juju>**
- Web: **<http://cloud.ubuntu.com>**
<https://juju.ubuntu.com/>
<http://jujucharms.com/>