What is Juju?

Why Juju?

Web services can be hard, particularly at scale. Juju provides tools

and a language (set of abstractions) that allow you to reason about

and manage your infrastructure from a service-oriented perspective,

Rather than reasoning about configuration management, create

reference implementations of complex systems that you can discuss

in a natural way. Decompose big problems for improved quality and

flexibility. Deploy many parts of your stack and let Juju configure

* charm store and ecosystem—a ready-made portfolio of many of

* public charms published on the store are vetted by Canonical

* CI of published charms provided gratis by Canonical

instead of a machine-oriented one.

(and re-configure) them to talk to each other.

the services and solutions you rely on

* only OSI-approved licenses

* charms are easily bundled

* easy-to-use CLI

and metal

as production

Service component deployment, configuration, and coordination,

encapsulated in charms. Think of it as "apt-get for the web".

environment

relation

service

machine

cloud instance

- Juju server



Montréal

Python and Juju

* charm helpers (http://pythonhosted.org/charmhelpers/)

* juju quickstart (plugin)

* gui support code (incl. testing)

* deployer (http://pythonhosted.org/juju-deployer/)

* Python API client (http://python-jujuclient.readthedocs.org/en/latest/)

* amulet (https://jujucharms.com/docs/1.20/tools-amulet)

* bundletester (https://github.com/juju-solutions/bundletester)

* CI for charm store

* CI for juju-core

* originally written in Python

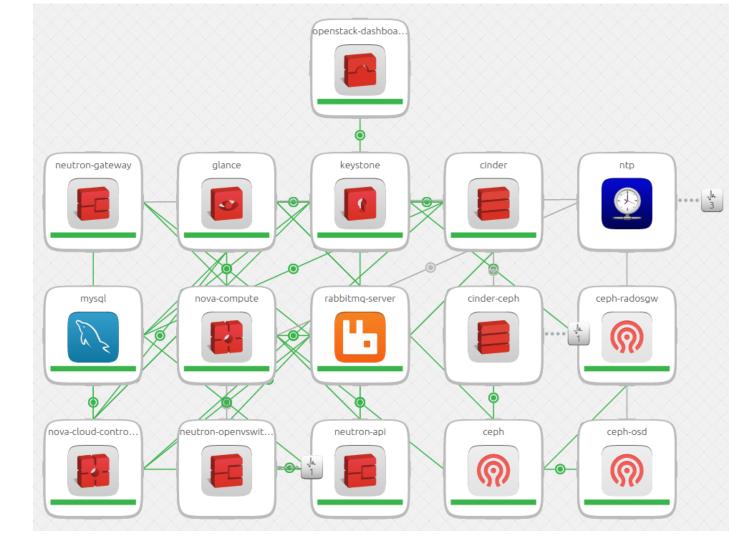
Charms

- * The key abstraction in Juju
- * Models a service using a declarative syntax

http://goo.gl/JpyXV1

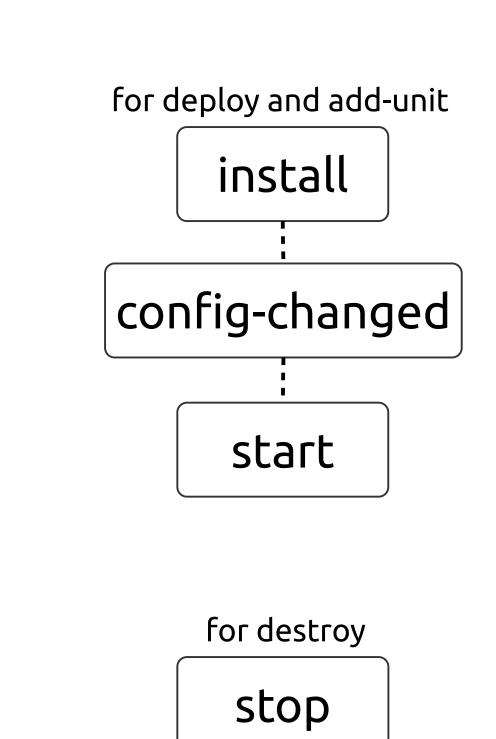
- * Encapsulates and abstracts away ops knowledge about a service
- * Describes how to interact with other services

OpenStack



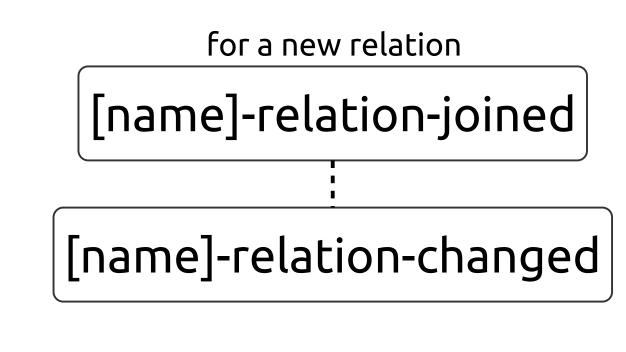
Charm Hooks

Executables called by Juju with access to special env vars and tools.



for charm upgrade	
config-changed	
i upgrade-charm	
upgrade-chaim	

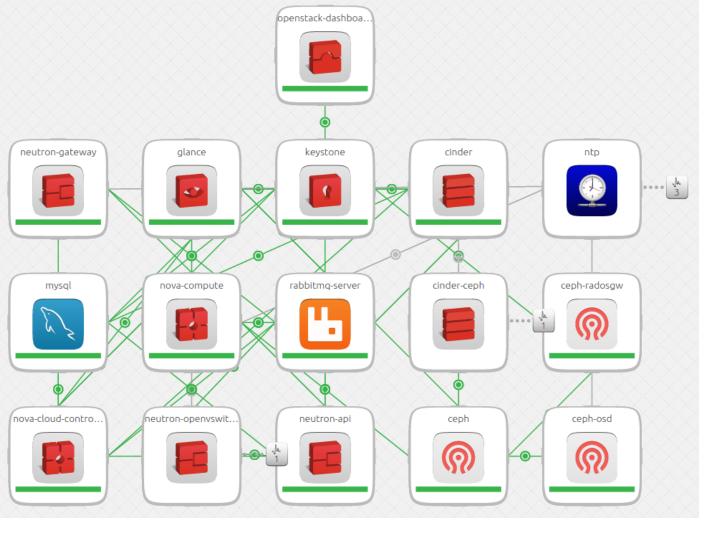




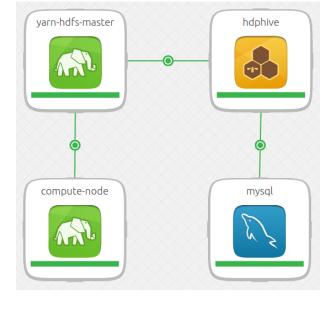
for updated remote relation settings [name]-relation-changed

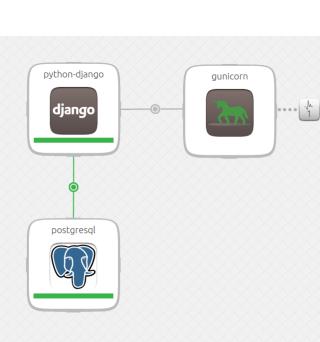
for remote relation ending [name]-relation-departed

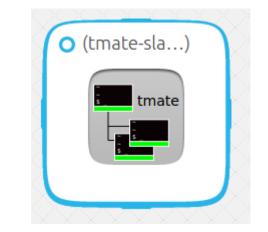
for relation finally ended [name]-relation-broken

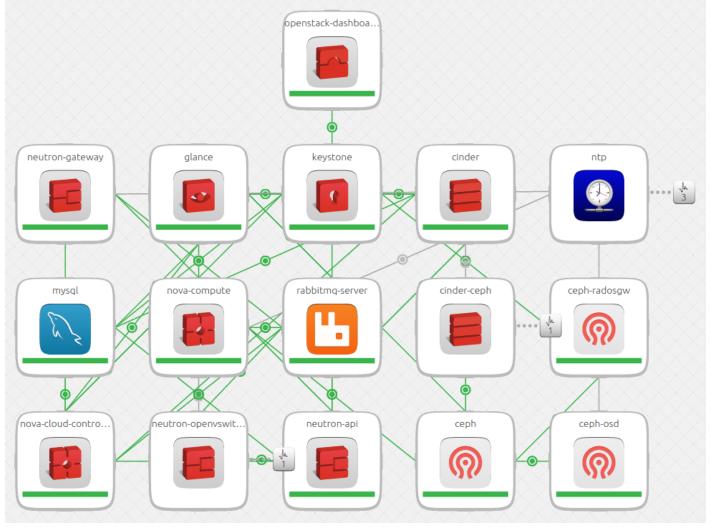


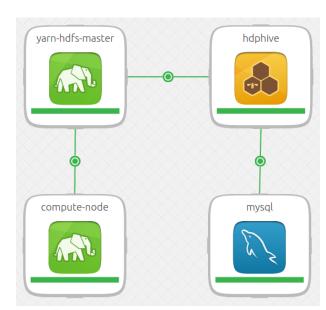
Big Data



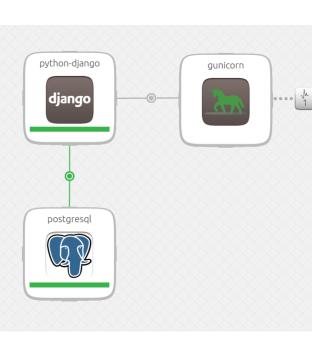




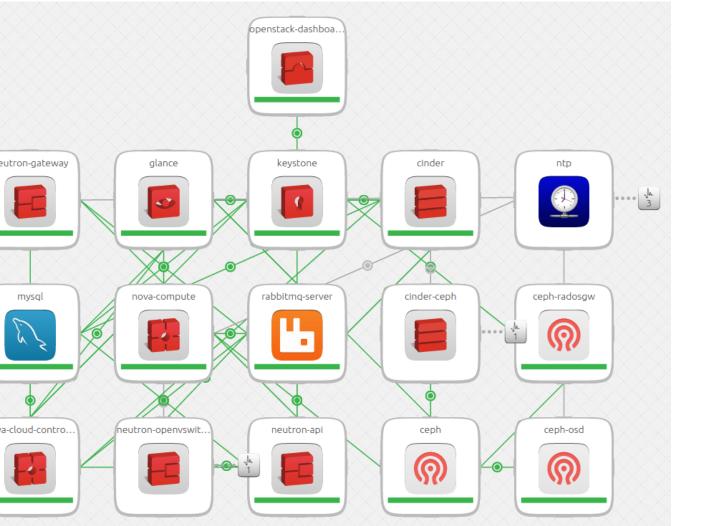


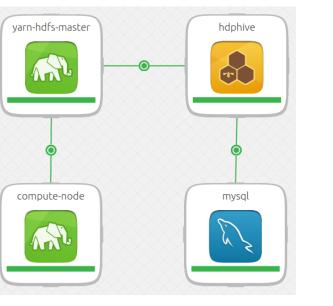


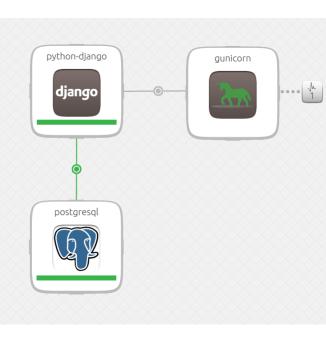
Django



"tmate-slave"



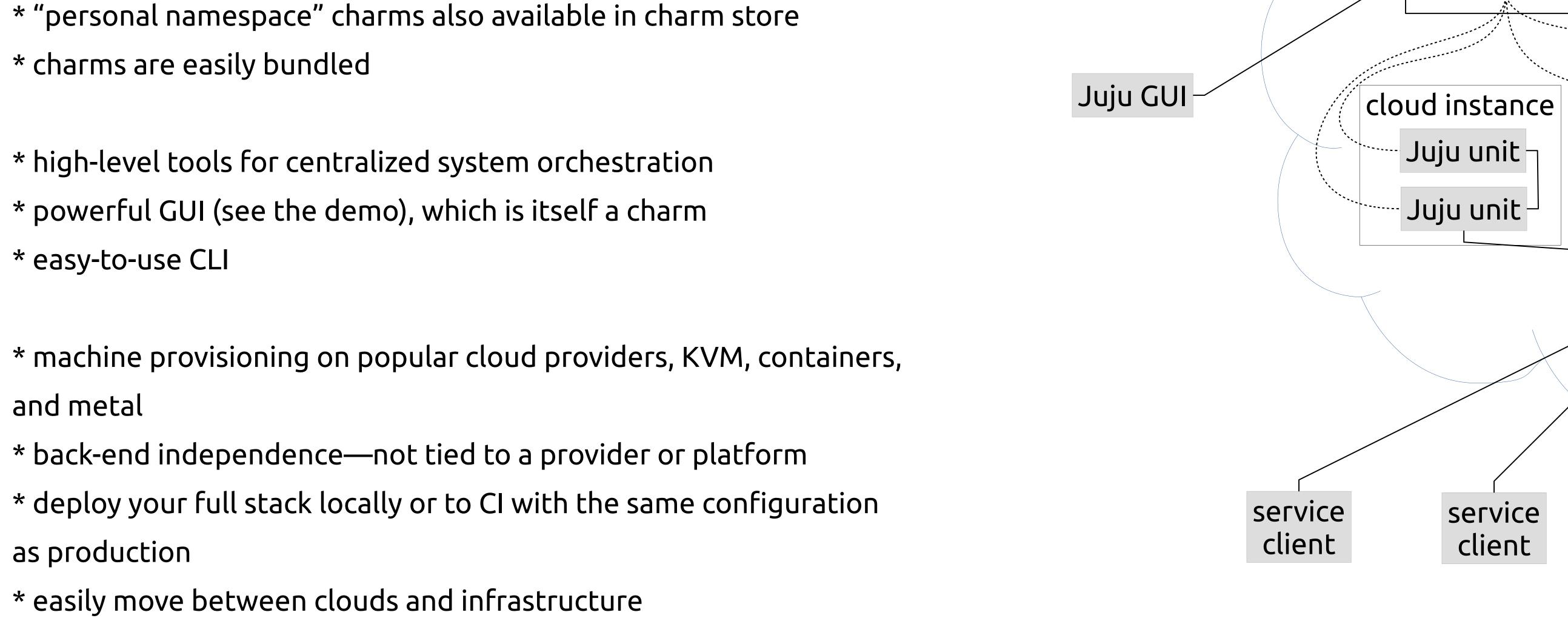




	salt	ansible	chef
Juju deploys full solution	X	X	X
existing "recipes" charm-able	X	X	X
support in charm-helpers	X	X	

Related Solutions and Juju

	docker	kubernetes	rails
deploy with Juju	X	X	X
Juju adds coordination	X	X	X
deploy into with Juju	X	X	X



Juju CLI -

JSON RPC

OpenStack Azure Joyent **HPCloud** Digital Ocean VMware (upcoming) MAAS (bare metal)

* easily move between clouds and infrastructure

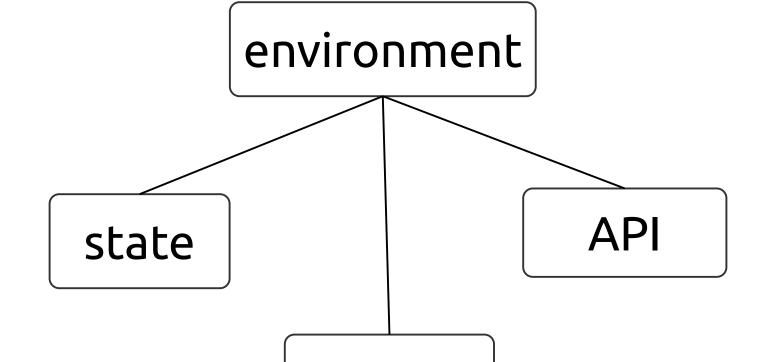
AWS (EC2, etc.)

Supported Providers

Provider Features

anything else ("manual")

Instances Storage Networking **Availability Zones** Containers/Virtualization



agents

Juju State

service client

provider

instance

cloud API

cloud instance

Juju unit -

cloud instance

Juju unit

service client

- * Captures intent for the environment
- * Drives events which realize that intent
- * Managed by the Juju server (behind API)
- * High-availability through replication

