ubuntu

Use Nagios to monitor your public cloud

monitoring complex Ubuntu deployments

Matt Bruzek - matthew.bruzek@canonical.com

CANONICAL

Computing infrastructure is only getting more complex

Life is too short to monitor your infrastructure by hand!

How many people here use a configuration management tool to deploy or monitor their environment?

Chef/Puppet/Ansible/other?

Everyone here is using Nagios differently

- Monitoring different things
- Different architectures or operating systems
- Different production environments
- Using different cloud provider
- Bare metal servers

... at least we all agree on Nagios!

If there was a change to your network/servers/cloud provider

could your automation tool handle a major change?

Can we continue to have artisanal infrastructures?

How many false positive monitoring errors do we get from infrastructure changes?

When you take a machine out of service, does that same process remove the host from monitoring?

If yes that is great, if not it should!

The services we deploy should know how to monitor themselves!

Crazy right? Well, what if they could?

Let’s say you were asked to monitor some new big data solution

demo time!

Before the demo…

```
juju quickstart realtime-syslog-analytics
juju expose zeppelin

juju deploy nrpe
juju deploy nagios
juju expose nagios
```

During the demo...

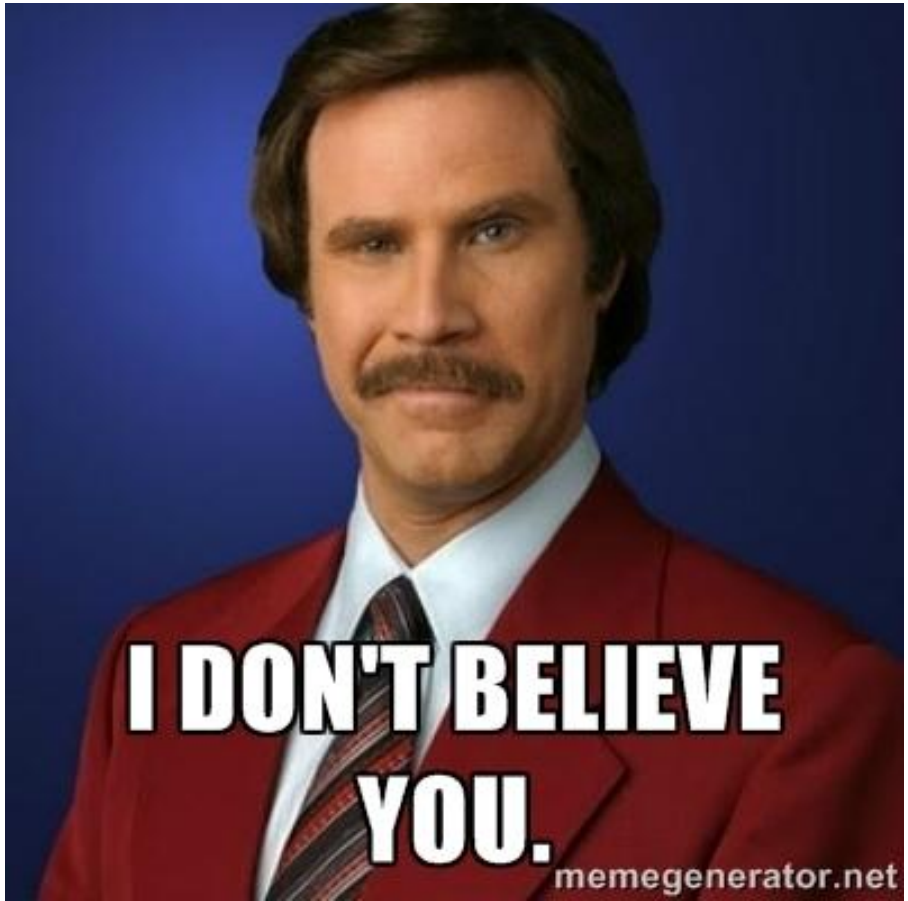
```
juju add-relation nrpe hdfs-master
juju add-relation nrpe secondary-namenode
juju add-relation nrpe compute-slave
juju add-relation nrpe flume-hdfs
juju add-relation nrpe flume-syslog
juju add-relation nrpe spark
juju add-relation nrpe yarn-master

juju add-relation nrpe nagios
```

What was that?

Juju in action with Nagios!

That looked fake



Juju

- An open source service orchestration software project
  - that provides a language to **model** cloud deployments
  - Cloud or bare metal agnostic, define the model and it is repeatable on a variety of environments.

- Not just a configuration management tool
  - Juju follows the service through the complete lifecycle
    - add or remove relations, scale up or down services

A language for the cloud?

Here are a few examples:

juju **deploy** nagios

juju **add-relation** nagios nrpe

juju **set** nagios extraconfig=`cat custom.cfg`

juju **add-unit** mysql

juju **expose** nagios

Define a model and share it with others to deploy the same model on different clouds

And **monitor** every charm (service)

How does Juju work?

cloud images, cloud-init, and charm code

Charms

Charms

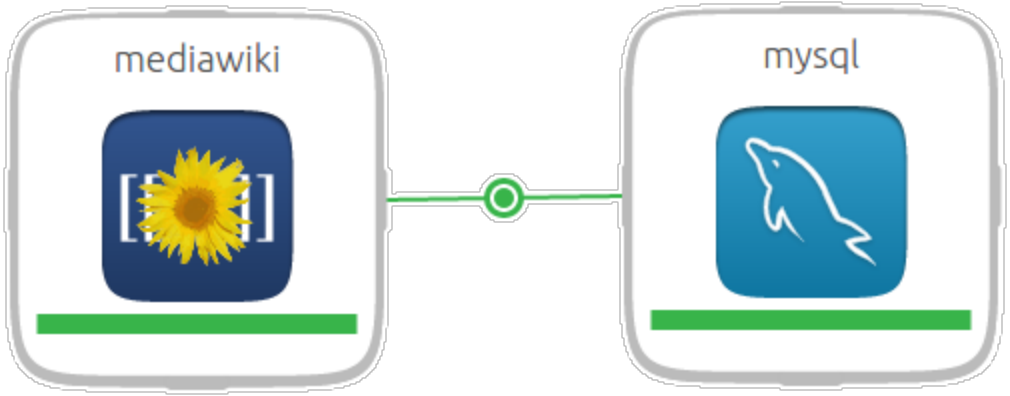
- A set of instructions or code that follow a **lifecycle** of a service
  - From install, configure, start, to relate, scale, stop and destroy
  - Contain the **operational intelligence** to respond to lifecycle events
    - For example how to monitor this service with Nagios
- An executable white paper
  - Take the knowledge of industry experts and distill that into a charm
- Each charm is its own software project

Charms

- A collection of files with specific directory structure
- Charms can be written in any language
  - From scripting languages to compiled binary files, really anything
- The code must take advantage of the event hook architecture
  - Juju fires events, each event has a hook, charm code responds to events

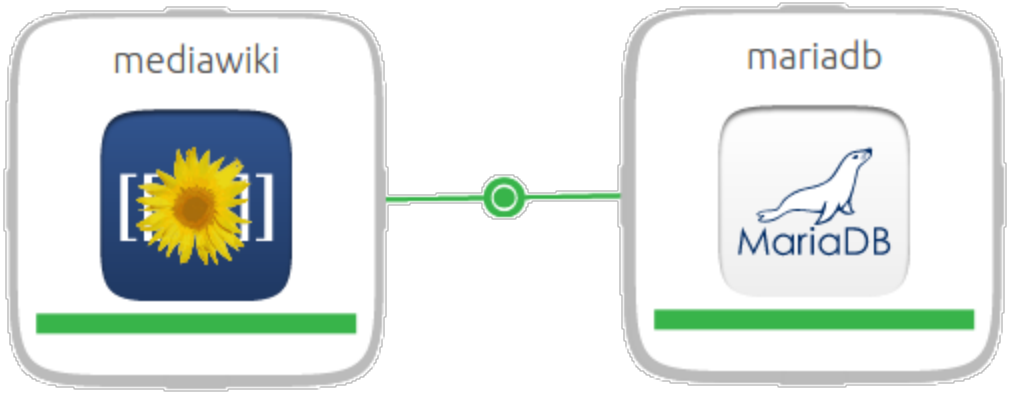
Charms are connectable

mediawiki requires a database, mysql provides a database



Charms are interchangeable

mariaDB also provides a database



Charms are not only Ubuntu!

Ubuntu workloads

CentOS workloads

Windows workloads

That is right, charms can deploy Microsoft services!

http://www.cloudbase.it/juju/

Charm are building blocks

Over 260 charms in the “charm store”

mostly free and open source software

Curated for quality and reliability

The nagios and nrpe charms can monitor every charm in Juju

There is room for improvement

Bundles are the model

Bundles define deployment with configuration and constraints

Bundles are:

- Collection of charms (services)
- Relations
- Configuration

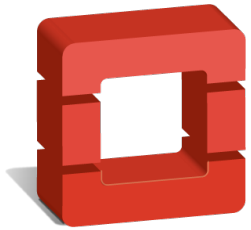
Bundles can be deployed in a single step.

Bundles allow easy repeatability for sharing complex deployments with other people.

In Juju the model is portable

Move your application to another cloud

Juju works on almost all the clouds



openstack™

Windows Azure™

Google Compute Engine

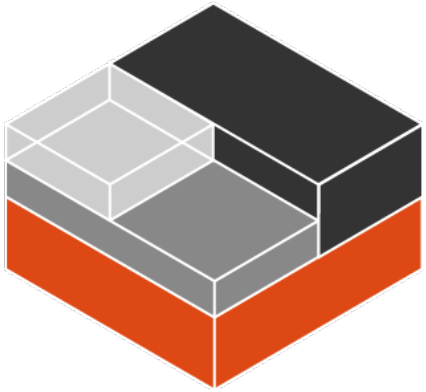
Cloud

Joyent

No clouds? Juju works with bare metal too

vmware vSphere

Metal As A Service (MAAS)



Linux Containers (LXC)

Supported cloud providers

Manage deployments on these environments

- Amazon Web Services (AWS)
- DigitalOcean (DO)
- Google Compute Engine (GCE)
- Joyent
- HP Public Cloud (OpenStack)
- Any OpenStack cloud
- Windows Azure
- Bare Metal (using MAAS)
- Local (LXC, or KVM)
- Vagrant
- VMWare vSphere
- Manual Provisioning (any other cloud using ssh)

What does that portability give you?

The ability to run on the cloud that runs your application the best

Summary

create **model** solutions, deploy them in a consistent and **reproducible** way

and you can monitor every service in the deployment with Nagios!

Where to go for more information

https://jujucharms.com

Github: https://github.com/juju/juju

Mailing list: https://lists.ubuntu.com/mailman/listinfo/juju

Nagios charm: https://jujucharms.com/nagios/

NRPE charm: https://jujucharms.com/nrpe/

Nagios & Ubuntu

Where are we?

Debian package archive (apt)

Nagios core 3

Nagios plugins 1.5

Installing Nagios core 4 on Ubuntu is still a manual process

Manual can’t be done at scale

We (I) am working on it!