Deploying with Ubuntu Cloud
Cloud computing stack

Salesforce.com, GoogleDocs, Office, etc...

GoogleApps, Java, Azure, etc...

Amazon, GoGrid, 3Tera, OpenStack, etc...

Storage

Network

Xen
KVM
VMWare
HyperV
e tc...

HP
IBM
Dell
Lenovo
e tc...
Ubuntu Cloud

- (Ubuntu One)
- Juju (CloudFoundry)
- Ubuntu Cloud Infrastructure and Guest
  - KVM, Xen
  - LXC
  - Orchestra (Bare metal provisioning)
- Storage
- Network
- x86
- ARM
What is Ubuntu Cloud?

- Ubuntu is the reference OS for IaaS (Cloud Infrastructure)
- Ubuntu is the leading Guest OS (Cloud Guest)
- Juju is the base for the next generation Ubuntu
Why Ubuntu Cloud Infrastructure?

• Best in class deployment tools
  - Orchestra
  - Juju
  - Can adapt to any customer preferred tool

• Best in class management tools
  - Landscape
  - Juju
  - Nagios, Collectd and other OSS

• Ubuntu provides the base components for most OpenStack distributors
  - HP Cloud
  - Dell OpenStack
  - Rackspace Cloud Builder

• Ubuntu and OpenStack synchronized development releases

• Ubuntu is the reference OS for OpenStack

Why OpenStack?

• Fastest ever growing OpenSource project
  - in terms of partners (HP, Citrix, Dell, Cisco, RackSpace, etc...)
  - in terms of contributions (43 companies involved in the Diablod release)
  - in terms of time to Market (5 major public clouds expected by EO 2011)

• Built to support the needs of public clouds (which supersedes private)

• Built to provide no SPOF

• Massively extensible in terms of functionalities
  - SAN
  - Network
  - Hypervisors
  - etc..
Ubuntu Cloud Infrastructure
Ubuntu Cloud Infrastructure

An IaaS stack based on 3 pillars

Openstack Compute (aka Nova)
• Delivers instances and disk space on demand to run Ubuntu Cloud Guest or most other Operating Systems.
• Logical equivalent to AWS EC2

Openstack Image Service (aka Glance)
• Delivers machines images locally and remotely
• Ubuntu Cloud Guest enabled
• Logical equivalent of AWS AMI Server

Openstack Object Storage (aka Swift)
• Delivers Peta-scale object storage capabilities with redundancy built-in.
• Logical equivalent to AWS S3
Build your infrastructure as a service public or private cloud

- Based on OpenStack
- Supporting KVM, Xen & LXC hypervisors
- All components can be made highly available
- EC2, EBS and S3 compatible
- One of the workload made easy to deploy with Orchestra
Orchestra
Ubuntu Orchestra

Install Orcherstra on your first server
Associate profiles with MAC addresses via the web or API interface
Turn on your hardware, the deployment happens, your infrastructure is ready!
Ubuntu Orchestra

Bare metal deployment from the Ubuntu Installer

- Zero touch deployment of complex workloads
- Configuration management
- Customizable

Components

- Cobbler
- Fact database
- Cloud-Init
- Pluggable management tools (Juju as default)
Installing Orcherstra

```
> sudo apt-get install ubuntu-orchestra-server
```
Installing Orcherstra

Ubuntu Orchestra Provisioning Server can manage address and name allocation for provisioned systems. If you manage your DNS and DHCP elsewhere, you should disable this option.

Enable Orchestra managed DNS/DHCP:

<Yes>  <No>
Installing Orchestra

Package configuration

**Configuring ubuntu-orchestra-provisioning-server**

Ubuntu Orchestra Provisioning Server manages DHCP for address allocation for the provisioned systems. If the network range for the DHCP clients is different from the default (192.168.1.5, 192.168.1.200), you should set it here.

An example of how a network range should be specified is as follows:

```
10.10.10.2,10.10.10.254
```

Set the network range for DHCP Clients:

```
10.55.55.2,10.55.55.7
```

<ok>
Installing Orchestra

Package configuration

Configuring ubuntu-orchestra-provisioning-server

Ubuntu Orchestra Provisioning Server manages DHCP for address allocation for the provisioned systems. If the Provisioning Server is NOT the default Gateway for the provisioned systems, you should set the default Gateway here, otherwise leave this blank.

Set default Gateway for DHCP Clients:

10.55.55.1

<Ok>
Package configuration

Configuring ubuntu-orchestra-provisioning-server

Ubuntu Orchestra Provisioning Server manages DHCP for address allocation for the provisioned systems. If these systems are required to be under a domain, you should enter it here.

Set the domain name for DHCP Clients:

canonical.com

<Ok>
Orchestra ready

• Latest Ubuntu ISO is downloaded
• Package repositories are setup
• PXE, TFTP, DHCP and Cobbler are ready

→ Let's define some servers
Declaring servers for Orchestra

```bash
sudo cobbler system add\n    --name="cempedak.canonical.com"\n    --mac-address="00:24:81:e4:59:9c"\n    --ip-address="10.55.55.2"\n    --dns-name="cempedak.canonical.com"\n    --hostname="cempedak.canonical.com"\n    --profile="oneiric-x86_64-ensemble"\n    --mgmt-classes="orchestra-juju-available"\n    --kopts="console=ttyS0,9600n8 DEBCONF_DEBUG=developer netcfg/dhcp_timeout=120 netcfg/choose_interface=eth0"
```
Juju
Devops Distilled
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
What drives DevOps?

- Speed of the deployment
- Continuous Integration, Automated Testing, etc.
- Fast change vs. Stability
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.
Manages Services, not Machines
Elevate to Juju

- Service Orchestration
- Configuration Management
- Virtualization
- Operating System
- Hardware
Juju, DevOps Distilled

**Dev**
- Reuse existing deployment charms in an openly-accessible repository of shared expertise
- Reproduce deployments for test and staging purposes
- Rapid deployment of your dependencies for development purposes
- Compose whole systems from individual application components and describe the entire deployment

**Ops**
- Explicit control over deployment, configuration and upgrade options
- See what’s deployed and track usage in the cloud
- Create and share charms for new applications
- Monitor, scale, shrink and adjust deployment parameters in real time
- Explicitly connect different components and maintain those relationships over time
- Collaborate with developers on the exact deployment and upgrade processes
- Get more done: implement decisions immediately regardless of infrastructure scale

http://juju.ubuntu.com
Juju's Charms

• Charms are a shareable, re-usable, and repeatable expressions of DevOps best practices.

• You can use them unmodified, or easily change and connect them to fit your needs.

• Deploying a charm is similar to installing a package on Ubuntu: ask for it and it’s there, remove it and it’s completely gone.
Juju is a community of DevOps expertise.

- Most of the application you want will be available in Juju.

- Juju provides direct and free access to a DevOps community-contributed collection of charms
Juju provides service orchestration

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

• Charms developed, tested, and deployed on your own hardware will operate the same in an EC2 API compatible cloud, including OpenStack.
Juju is intelligent

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

• There’s no need to learn a domain specific language (DSL) to use Juju or create charms. You can be up and running with your own charm in minutes
Juju's internals
Juju treats individual services as atoms that are described as charms and can be instantiated one or many times.
Each charm (or atom) define dependencies and/or provides.
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.

Relations are to charms what bounds are to atoms.

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
- 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
Deploying OpenStack with Juju
Back to the Orchestra server

- > sudo apt-get install juju
- > sudo mkdir -p ~/.juju
- > sudo vi ~/.ensemble/environments.yaml

juju: environments
  environments:
    orchestra:
      type: orchestra
      orchestra-server: 10.55.55.7
      storage-url: http://10.55.55.7/webdav
      orchestra-user: cobbler
      orchestra-pass: cobbler
      admin-secret: fooooo
      acquired-mgmt-class: orchestra-juju-acquired
      available-mgmt-class: orchestra-juju-available
Deploying OpenStak

- `juju bootstrap`
- `juju deploy mysql --placement=local`
- `juju deploy rabbit-mq --placement=local`
- `juju deploy --config=~/charms/openstack.yaml nova-cloud-controller`
- `juju add-relation nova-cloud-controller rabbitmq`
- `juju add-relation nova-cloud-controller mysql`
- `juju deploy --config=~/charms/openstack.yaml glance`
- `juju add-relation glance mysql`
- `juju add-relation glance:image-service nova-cloud-controller:image-service`
- `juju deploy --config=~/charms/openstack.yaml --repository=`pwd` nova-compute`
- `juju add-relation nova-compute rabbitmq`
- `juju add-relation nova-compute mysql`
- `juju add-relation nova-cloud-controller:nova-network nova-compute:nova-network`
- `juju add-relation glance:image-service nova-compute:image-service`
OpenStack is now ready!
Scaling OpenStack

- > juju add-unit nova-compute

or

- > juju add-unit nova-cloud-controller
Juju key take-outs

- Juju is Easy
- Juju can deploy on bare metal with Orchestra
- Juju can deploy on Openstack and AWS
- Juju is available today
- Juju is supported today
- Juju is innovative technology that bring value by it's simplicity
- Juju provides a growing collections of charms for leading workloads
Thank you

Nick Barcet
<nick.barcet@canonical.com>