



OPSCODE



Automated Deployment of OpenStack with Chef

SCALE 9x

February 25-27, 2011



Introductions

Matt Ray

Senior Technical Evangelist

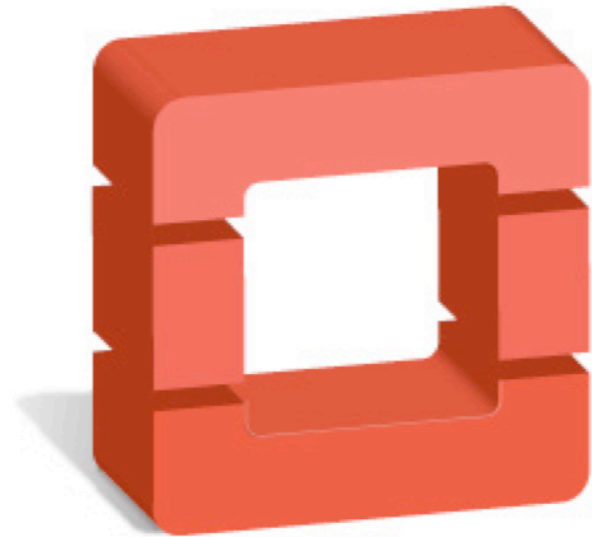
matt@opscode.com

@mattray

GitHub:mattray



OPSCODE

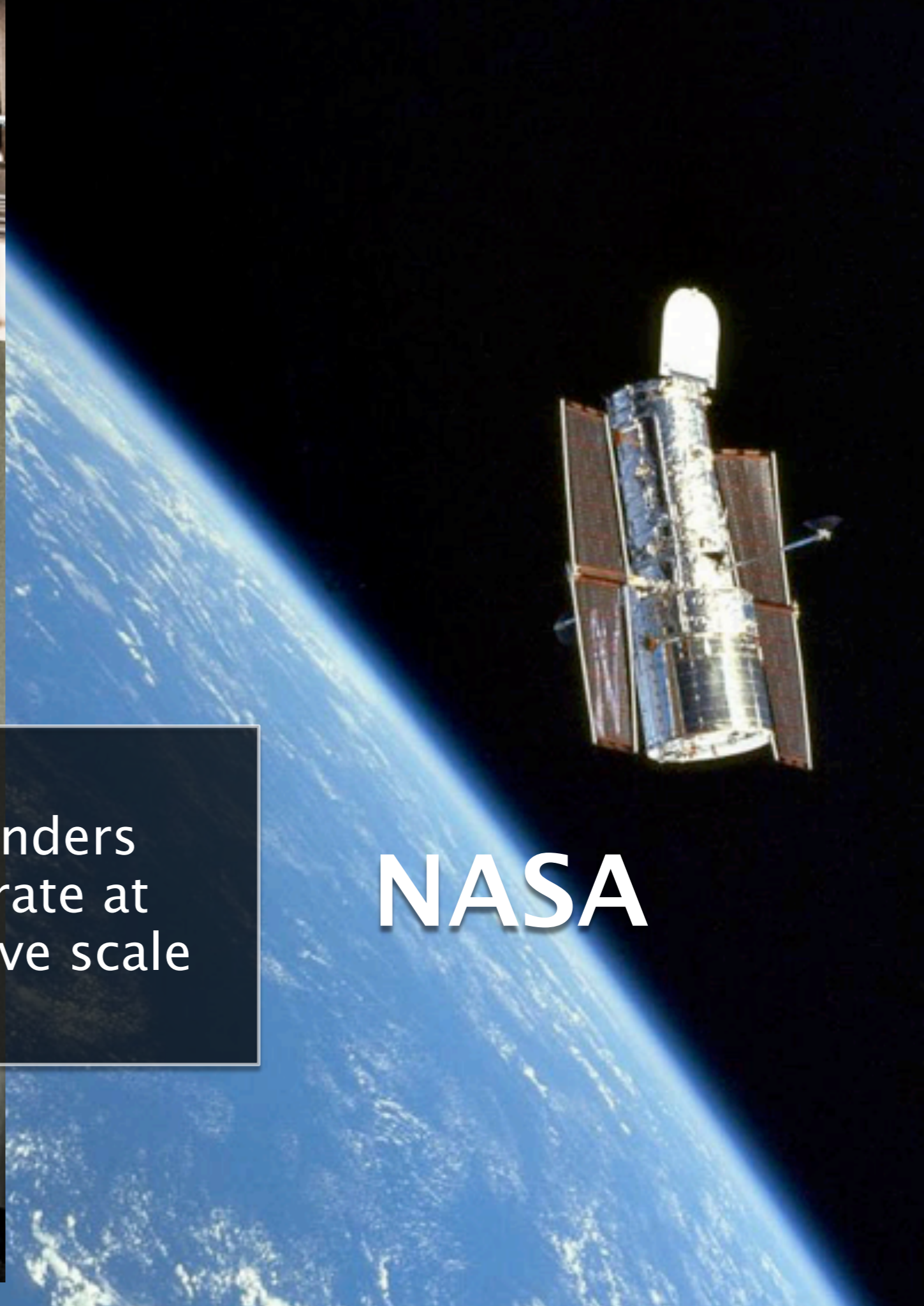


openstack™

What is OpenStack?



Founders
operate at
massive scale



NASA



OPSCODE

OpenStack: The Mission

"To produce the ubiquitous **Open Source cloud** computing platform that will meet the needs of **public and private cloud** providers regardless of size, by being **simple to implement** and **massively scalable.**"

OpenStack Founding Principles

Apache 2.0 license (OSI), open development process

Open design process, 2x year public Design Summits

Publicly available open source code repositories

Open community processes documented and transparent

Commitment to drive and adopt open standards

Modular design for deployment flexibility via APIs



OPSCODE

Community with Broad Support





OPSCODE



OpenStack **Compute**

Software to **provision virtual machines** on standard hardware at massive scale

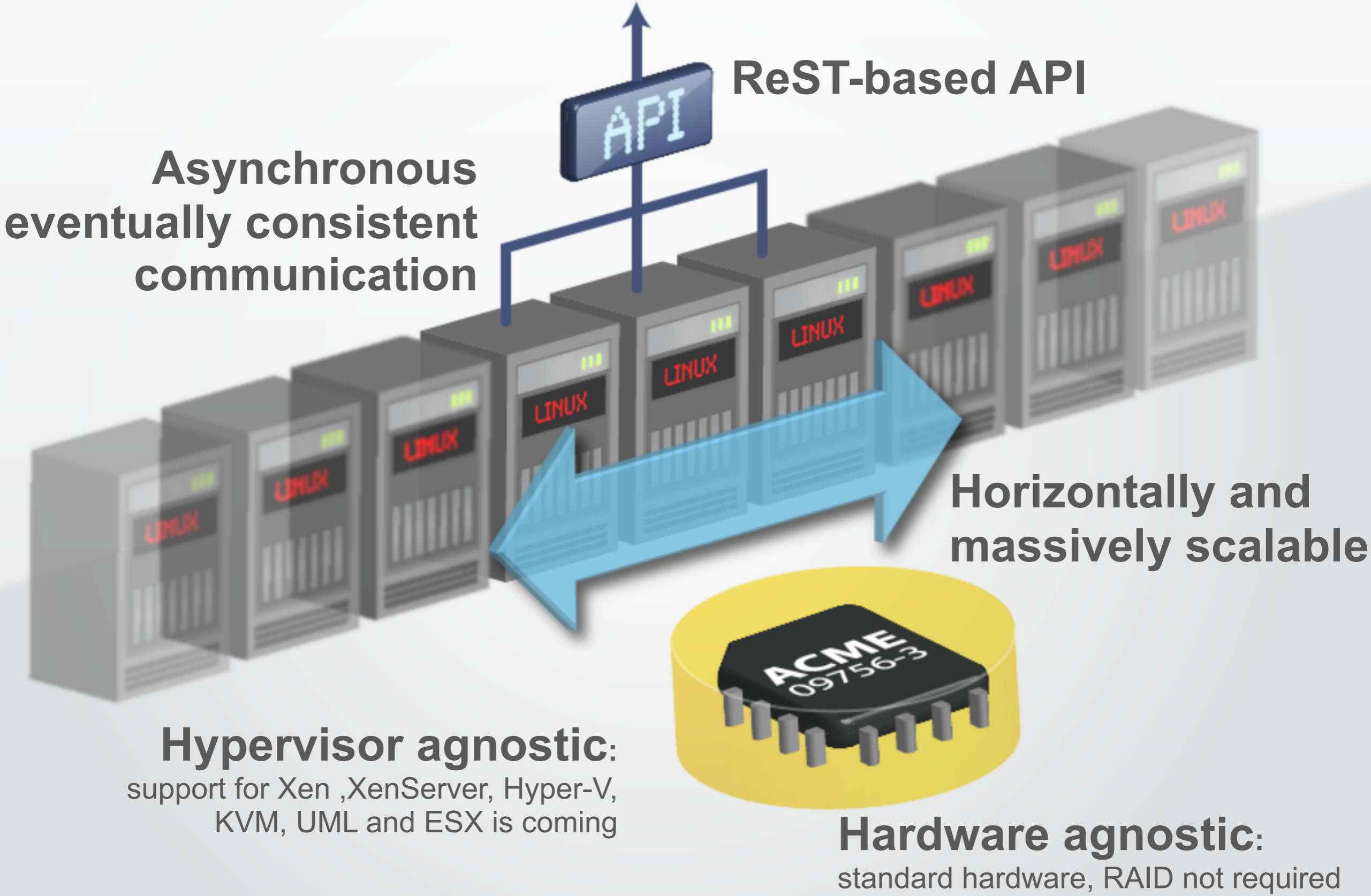
creating open source software to build public and private clouds

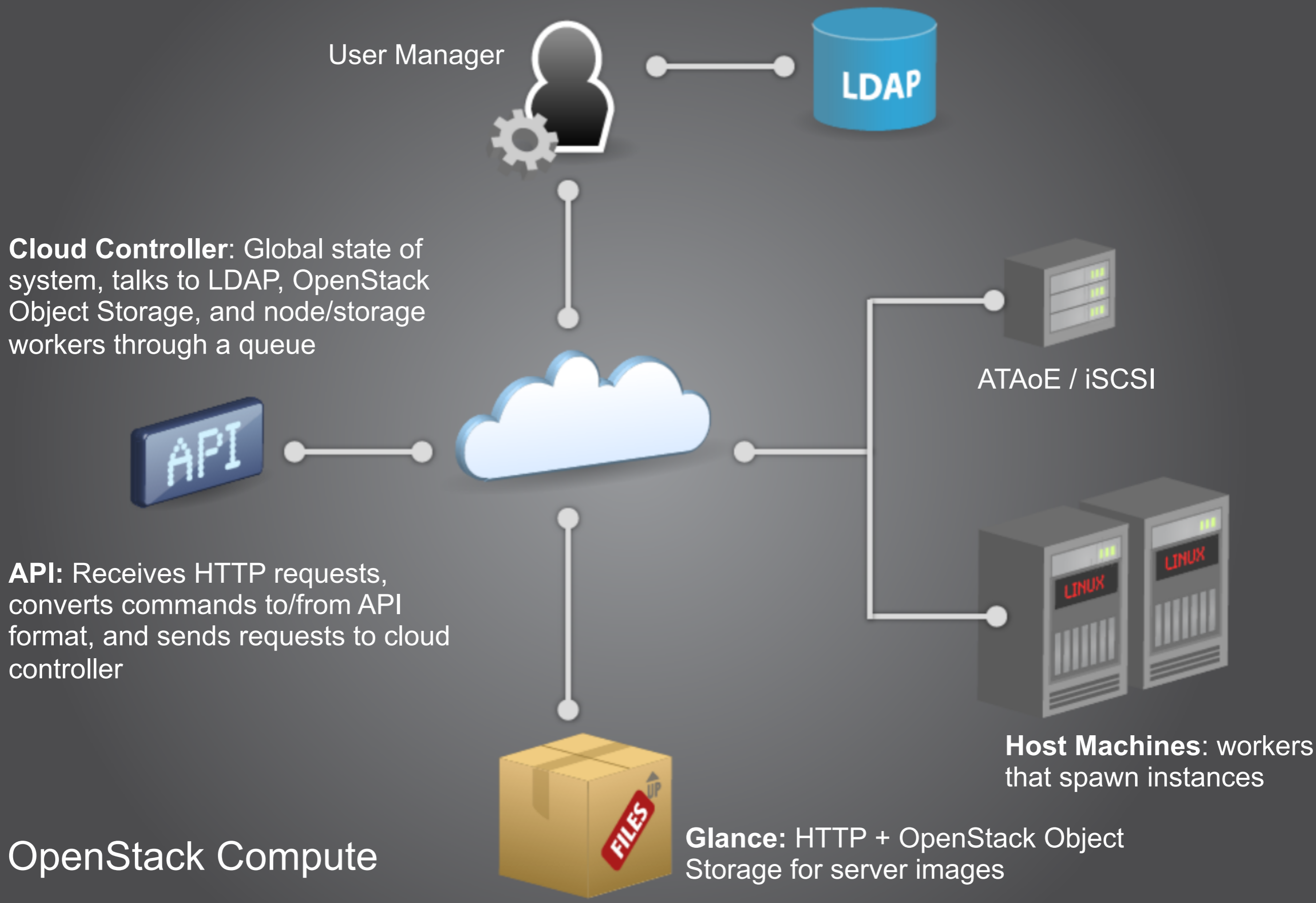


OpenStack
Object Storage

Software to reliably **store billions of objects** distributed across standard hardware

OpenStack Compute Key Features





User Manager

LDAP

Cloud Controller: Global state of system, talks to LDAP, OpenStack Object Storage, and node/storage workers through a queue

API

ATAoE / iSCSI

API: Receives HTTP requests, converts commands to/from API format, and sends requests to cloud controller

Host Machines: workers that spawn instances

OpenStack Compute

FILES

Glance: HTTP + OpenStack Object Storage for server images



OPSCODE

Hardware Requirements

OpenStack is designed to run on industry standard hardware, with flexible configurations

Compute

x86 Server (Hardware Virt. recommended)
Storage flexible (Local, SAN, NAS)

Object Storage

x86 Server (other architectures possible)
Do not deploy with RAID (can use controller for cache)



OPSCODE

Why is OpenStack important?

Open eliminates vendor lock-in

Working **together**, we all go faster

Freedom to federate, or move
between clouds



OPSCODE



Chef

What is Chef?

Chef enables Infrastructure as Code

Manage configuration as **idempotent**
Resources.

Put them together in **Recipes**.

Track it like **Source Code**.

Configure your servers.



OPSCODE

At a High Level

A **library** for configuration management

A **configuration management** system

A **systems integration** platform

An **API** for your entire Infrastructure



OPSCODE

Fully **automated** Infrastructure



OPSCODE

Principles

Idempotent
Data-driven
Sane defaults
Hackability
TMTOWTDI

Open Source and Community

Apache licensed

Large and active community

Over 280 individual contributors (60+ corporate)

Community is Important!



OPSCODE



Etsy



maxmedia





OPSCODE



Chef

How does it Work?



OPSCODE



Chef

How does it Work?

Magic!



OPSCODE



Chef

How does it Work?

Magic!

(no really)



OPSCODE

Chef Client runs on **your System**



OPSCODE

Chef Client runs on **your System**

ohai!



OPSCODE

Clients talk to the Chef Server



OPSCODE

The Opscode Platform is a hosted Chef Server



OPSCODE

**We call each system
you configure a Node**



OPSCODE

Nodes have Attributes

```
{
  "kernel": {
    "machine": "x86_64",
    "name": "Darwin",
    "os": "Darwin",
    "version": "Darwin Kernel Version 10.4.0: Fri Apr 23 18:28:53 PDT 2010;
root:xnu-1504.7.4~1/RELEASE_I386",
    "release": "10.4.0"
  },
  "platform_version": "10.6.4",
  "platform": "mac_os_x",
  "platform_build": "10F569",
  "domain": "local",
  "os": "darwin",
  "current_user": "mray",
  "ohai_time": 1278602661.60043,
  "os_version": "10.4.0",
  "uptime": "18 days 17 hours 49 minutes 18 seconds",
  "ipaddress": "10.13.37.116",
  "hostname": "morbo",
  "fqdn": "morbomorbo.local",
  "uptime_seconds": 1619358
}
```

Kernel info!

Platform info!

Hostname and IP!



OPSCODE

Nodes have a **Run List**

What **Roles** and **Recipes**
to Apply in **Order**



OPSCODE

Nodes have Roles



OPSCODE

Nodes have **Roles**

webserver, database, monitoring, etc.



OPSCODE

Roles have a **Run List**

What **Roles** and **Recipes**
to **Apply** in **Order**



```
name "webserver"  
description "Systems that serve HTTP traffic"
```

```
run_list(  
  "role[base]",  
  "recipe[apache2]",  
  "recipe[apache2::mod_ssl]"  
)
```



**Can include
other roles!**

```
default_attributes(  
  "apache" => {  
    "listen_ports" => [ "80", "443" ]  
  }  
)
```

```
override_attributes(  
  "apache" => {  
    "max_children" => "50"  
  }  
)
```



OPSCODE

Chef manages **Resources on Nodes**



Resources

- ▶ Have a **type**
- ▶ Have a **name**
- ▶ Have **parameters**
- ▶ Take **action** to put the resource in the declared state

```
package "apache2" do
  version "2.2.11-2ubuntu2.6"
  action :install
end
```

```
template "/etc/apache2/apache2.conf" do
  source "apache2.conf.erb"
  owner "root"
  group "root"
  mode 0644
  action :create
end
```

Declare a description of the state a part of the node should be in



OPSCODE

Resources take action through Providers



OPSCODE

Recipes are lists of **Resources**



OPSCODE

Recipes

Evaluate and apply
Resources in the **order**
they appear

```
package "apache2" do
  version "2.2.11-1ubuntu2.6"
  action :install
end

template "/etc/apache2/apache2.conf" do
  source "apache2.conf.erb"
  owner "root"
  group "root"
  mode 0644
  action :create
end
```



OPSCODE

Order Matters



OPSCODE

Recipes are just **Ruby!**

```
extra_packages = case node[:platform]
  when "ubuntu", "debian"
    %w{
      ruby1.8
      ruby1.8-dev
      rdoc1.8
      ri1.8
      libopenssl-ruby
    }
  end
```

```
extra_packages.each do |pkg|
  package pkg do
    action :install
  end
end
```



OPSCODE

Cookbooks are
packages for Recipes



OPSCODE

Cookbooks

Distributable

`cookbooks.opscode.com`

Infrastructure as Code

Versioned



OPSCODE

Cookbooks

Recipes

Files

Templates

Attributes

Metadata



OPSCODE

**Data bags store
arbitrary data**



OPSCODE

A user data bag item...

```
% knife data bag show users mray
{
  "comment": "Matt Ray",
  "groups": "sysadmin",
  "ssh_keys": "ssh-rsa SUPERSEKRATS mray@morbo",
  "files": {
    ".bashrc": {
      "mode": "0644",
      "source": "dot-bashrc"
    },
    ".emacs": {
      "mode": "0644",
      "source": "dot-emacs"
    }
  },
  "id": "mray",
  "uid": 7004,
  "shell": "/usr/bin/bash"
}
```




OPSCODE

Environments manage
versioned infrastructure

A close-up photograph of a red-handled utility knife with a silver blade, set against a dark background. The knife is the central focus, with its handle and blade clearly visible. The text "Command-line API utility, Knife" is overlaid on the image in a white and green font.

Command-line API utility, **Knife**



Search

- ▶ CLI or in **Ruby**
- ▶ Nodes are searchable
- ▶ Roles are searchable
- ▶ Recipes are searchable
- ▶ Data bags are searchable


```
$ knife search node 'platform:ubuntu'  
search(:node, 'platform:centos')  
$ knife search role 'max_children:50'  
search(:role, 'max_children:50')  
$ knife search node 'role:webserver'  
search(:node, 'role:webserver')  
$ knife users 'shell:/bin/bash'  
search (:users, 'group:sysadmins')
```



OPSCODE



Chef

A photograph of a server room with rows of server racks on both sides of a central aisle. A person is standing in the aisle, looking at a device. The racks are filled with cables and hardware. The ceiling has a grid of lights.

HOW TO: Turn Racks of Standard Hardware Into a Cloud with OpenStack



OPSCODE

What Works Today?

Compute (Nova)

Single machine installation

- ▶ MySQL, RabbitMQ, OpenLDAP
- ▶ Nova-(api|scheduler|network|objectstore|compute)
- ▶ Role: nova-single-machine-install

Multi-machine

- ▶ Role: nova-multi-controller (1)
- ▶ Role: nova-multi-compute (N)



OPSCODE

Role: nova-single-machine-install

```
name "nova-single-machine-install"
description "Installs everything required to run Nova on a single
machine"
run_list(
  "recipe[apt]",
  "recipe[nova::mysql]",
  "recipe[nova::openldap]",
  "recipe[nova::rabbit]",
  "recipe[nova::common]",
  "recipe[nova::api]",
  "recipe[nova::scheduler]",
  "recipe[nova::network]",
  "recipe[nova::objectstore]",
  "recipe[nova::compute]",
  "recipe[nova::setup]",
  "recipe[nova::creds]",
  "recipe[nova::finalize]"
)
```




OPSCODE

Role: nova-multi-controller

```
name "nova-multi-controller"
```

```
description "Installs requirements to run the Controller node in a  
Nova cluster"
```

```
run_list(  
  "recipe[apt]",  
  "recipe[nova::mysql]",  
  "recipe[nova::openldap]",  
  "recipe[nova::rabbit]",  
  "recipe[nova::common]",  
  "recipe[nova::api]",  
  "recipe[nova::objectstore]",  
  "recipe[nova::compute]",  
  "recipe[nova::setup]",  
  "recipe[nova::creds]",  
  "recipe[nova::finalize]"  
)
```



OPSCODE

Role: nova-multi-compute

```
name "nova-multi-compute"
```

```
description "Installs requirements to run a Compute node in a Nova cluster"
```

```
run_list(  
    "recipe[apt]",  
    "recipe[nova::network]",  
    "recipe[nova::compute]",  
)
```

Starting with a provisioned server

- ▶ Ubuntu 10.10
(preseed)

- ▶ openssh-server

- ▶ virtual-machine-host

```
knife bootstrap crushinator.localdomain ~/.ssh/id_rsa -x mray \  
--sudo -d ubuntu10.04-gems
```





OPSCODE

Installation

▶ **Cookbooks uploaded**

```
$ knife cookbook upload -a
```

```
$ knife cookbook list
```

▶ **Roles uploaded**

```
$ rake roles
```

```
$ knife role list
```

▶ **Nodes ready**

```
$ knife node list
```



OPSCODE

AMIs

```
name "nova-ami-urls"  
description "Feed in a list URLs for AMIs to download"  
default_attributes(  
  "nova" => {  
    "images" =>  
["http://192.168.11.7/ubuntu1010-UEC-localuser-image.tar.gz"]  
    }  
  )
```

```
$ knife role from file roles/nova-ami-urls.rb
```

- ▶ Use an existing AMI
- ▶ Update URL to your own



OPSCODE

Assign the Roles

```
$ knife node run_list add crushinator.localdomain "role[nova-ami-urls]"
{
  "run_list": [
    "role[nova-ami-urls]"
  ]
}
```

```
$ knife node run_list add crushinator.localdomain "role[nova-single-machine-install]"
{
  "run_list": [
    "role[nova-ami-urls]",
    "role[nova-single-machine-install]",
  ]
}
```



OPSCODE

chef-client

```
mray@ubuntu1010:~$ sudo chef-client
[Fri, 25 Feb 2011 11:52:59 -0800] INFO: Starting Chef Run (Version
0.9.12)
...
[Fri, 25 Feb 2011 11:56:05 -0800] INFO: Chef Run complete in
5.911955 seconds
[Fri, 25 Feb 2011 11:56:05 -0800] INFO: cleaning the checksum cache
[Fri, 25 Feb 2011 11:56:05 -0800] INFO: Running report handlers
[Fri, 25 Feb 2011 11:56:05 -0800] INFO: Report handlers complete
```



OPSCODE

sudo su - nova

```
nova@$ nova-manage service list
h00-26-6c-f4-1e-a0 nova-scheduler enabled  :-) 2011-02-25 18:30:45
h00-26-6c-f4-1e-a0 nova-network enabled   :-) 2011-02-25 18:30:48
h00-26-6c-f4-1e-a0 nova-compute enabled   :-) 2011-02-25 18:30:50
```

```
nova@$ euca-describe-images
IMAGE ami-90hgmwai nova_amis/maverick-server-uec-amd64-vmlinuz-
virtual.manifest.xml admin available private i386 kernel true
IMAGE ami-h8wh0j17 nova_amis/maverick-server-uec-
amd64.img.manifest.xml admin untarring private i386 machine
ami-90hgmwai
```

```
nova@$ euca-run-instances ami-h8wh0j17 -k mykey -t m1.tiny
RESERVATION r-uur39109 admin default
INSTANCE i-00000001 ami-h8wh0j17 scheduling mykey (admin,
None) 0 m1.tiny 2011-02-25 18:34:01 unknown zone
```




sudo su - nova (page 2)

```
nova@$ euca-describe-instances
RESERVATION  r-uur39109      admin default
INSTANCE     i-00000001    ami-h8wh0j17 10.0.0.2  10.0.0.2  running
             mykey (admin, h00-26-6c-f4-1e-a0)  0        m1.tiny  2011-02-25
18:34:01 nova

nova@$ ssh -i mykey.priv ubuntu@10.0.0.2
The authenticity of host '10.0.0.2 (10.0.0.2)' can't be established.
RSA key fingerprint is 91:21:ef:5d:33:17:24:cb:f6:65:dd:27:1d:1c:
50:ad.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.2' (RSA) to the list of known
hosts.
```



OPSCODE

The Moment of Truth

```
Linux i-00000001 2.6.35-24-virtual #42-Ubuntu SMP Thu Dec 2 05:15:26
UTC 2010 x86_64 GNU/Linux
Ubuntu 10.10
```

```
Welcome to Ubuntu!
```

```
<SNIP>
```

```
See "man sudo_root" for details.
```

```
ubuntu@i-00000001:~$
```



OPSCODE

How Did We Get Here?



OPSCODE

Forked from Anso Labs' Cookbooks

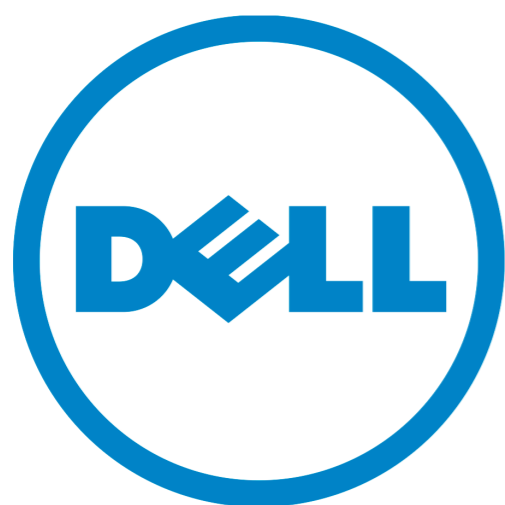
Bootstrapped by Opscode

Chef Solo/Vagrant installs for Developers

<http://github.com/ansolabs/openstack-cookbooks>



Who's involved so far?



OPSCODE





OPSCODE

What's Next?



OPSCODE

Nova needed enhancements

Happy Path-only!

KVM-only

MySQL-only

Flat DHCP network-only

Swift and Glance integration

More Roles



OPSCODE

Dashboard

**Graphical interface for managing
instantiation of AMIs**

Django application

**dashboard.rb recipe already exists in
nova cookbook**



Knife

- ▶ **Nova has same API as Amazon**

- ▶ `knife ec2 server create 'role[base]' -I ~/.ssh/my.pem -x ubuntu -G default -i ami-a403f6xd -f m1.micro`

- ▶ **Fog reportedly supports OpenStack already**

- ▶ **Simply need to pass URL of nova-api server and credentials**

- ▶ **<http://tickets.opscode.com/browse/CHEF-1757>**

- ▶ `knife openstack server create 'role[base]' -I ~/.ssh/my.pem -x ubuntu -G default -i ami-a403f6xd -f m1.micro`



OPSCODE

Object Storage (Swift)

- ▶ Recipes originated from Anso Labs' repository
 - ▶ <https://github.com/ansolabs/openstack-cookbooks>
- ▶ Included in the 'bexar' branch
- ▶ Untested so far



OPSCODE

Image Registry (Glance)

- ▶ Recipes originated from Anso Labs' repository
 - ▶ <https://github.com/ansolabs/openstack-cookbooks>
- ▶ Included in the 'bexar' branch
- ▶ Untested so far



OPSCODE

Scaling changes how we deploy OpenStack!



OPSCODE

Deployment Scenarios

- ▶ Single machine is relatively simple
- ▶ Controller + Compute nodes is a known quantity for small installations
- ▶ Nova + Swift + Glance in large installations
- ▶ Services separated and HA configurations supported
- ▶ Documentation and Chef Roles will be the solution



OPSCODE

Cactus, Diablo, ...

Development continues...

Branches for each stable release

Design Summit in April

Design Summit in the Fall



OPSCODE

Get Involved!

<https://github.com/mattray/openstack-cookbooks/tree/bexar>

<http://lists.openstack.org>

<http://lists.opscode.com>

#chef on irc.freenode.net

#openstack on irc.freenode.net

matt@opscode.com

jordan@openstack.com