Roger Ignazio – PuppetConf 2015

MANAGING MESOS, DOCKER, AND CHRONOS WITH PUPPET

MESOSPHERE

$(whoami)

ABOUT ME

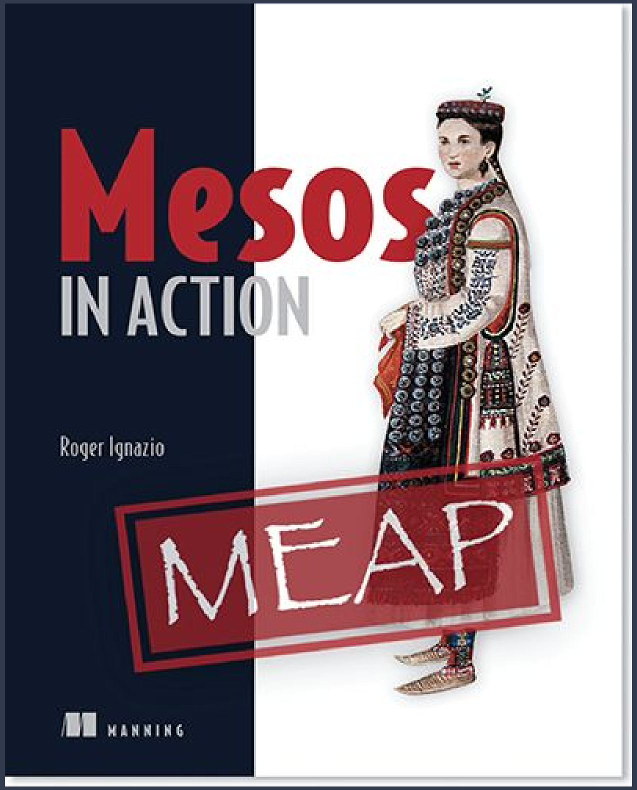
Roger Ignazio

Infrastructure Automation Engineer @ Mesosphere

@rogerignazio

$(whoami)

MESOS IN ACTION



mesosinaction.com

Code: **`ctwpuppet`**

AGENDA

- Getting started
- Deploying a Mesos cluster
- Building a Docker image
- Creating a Chronos job
- Demo
- Provisioning infrastructure – bare-metal and cloud
- Q & A

AUDIENCE POLL

GETTING STARTED

ABOUT MESOS, DOCKER, AND CHRONOS

Mesos

- Represent many machines (thousands) as a single entity
- Advertise resources directly to applications

Docker

- Easily package and deploy apps (with dependencies)
- Analogous to VMs, but minus the overhead*

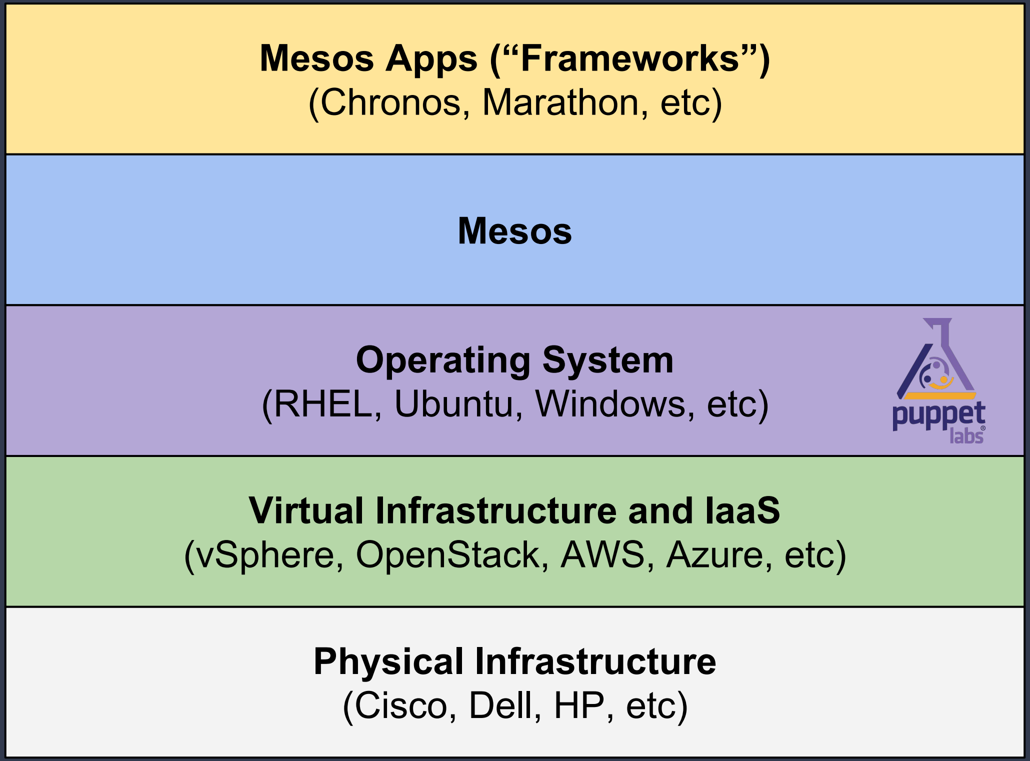
Chronos

- Distributed, highly available Cron for Mesos
- Run scheduled tasks in cgroups, Docker containers

ABOUT PUPPET

- Declare *desired* state for your infrastructure
- Wide range of OS support
- Idempotent
- Extensible (via custom facts, types, providers)
- Open source – Apache License, version 2

PUPPET'S ROLE


Mesos Apps ("Frameworks")
(Chronos, Marathon, etc)
Mesos
Operating System
(RHEL, Ubuntu, Windows, etc)
puppet labs
Virtual Infrastructure and IaaS
(vSphere, OpenStack, AWS, Azure, etc)
Physical Infrastructure
(Cisco, Dell, HP, etc)

GETTING STARTED

PUPPET'S ROLE

If Mesos is the abstraction layer for your applications,
Puppet is the abstraction layer for infrastructure management

GETTING STARTED

PUPPET'S ROLE

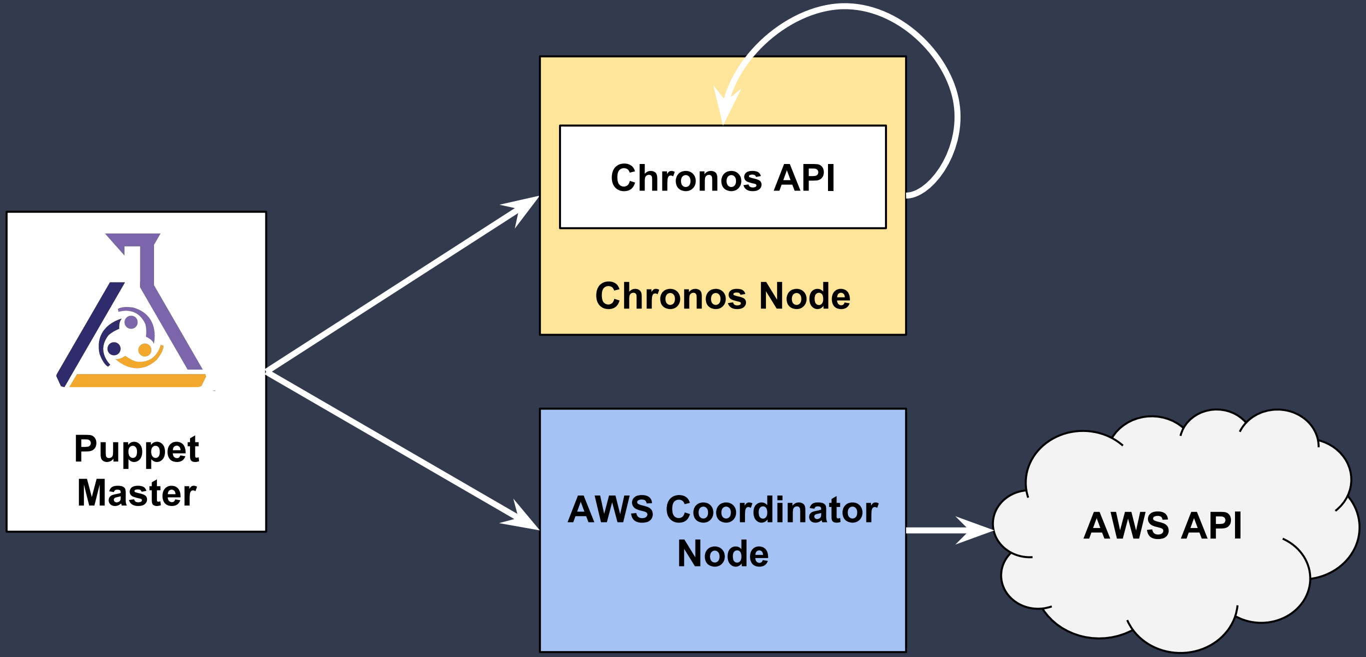
If Mesos is the abstraction layer for your applications,
Puppet is the abstraction layer for infrastructure management

But it can also be more ...

GETTING STARTED

PUPPET'S ROLE

Custom types and providers can interact with external services (AWS, Chronos, ...)



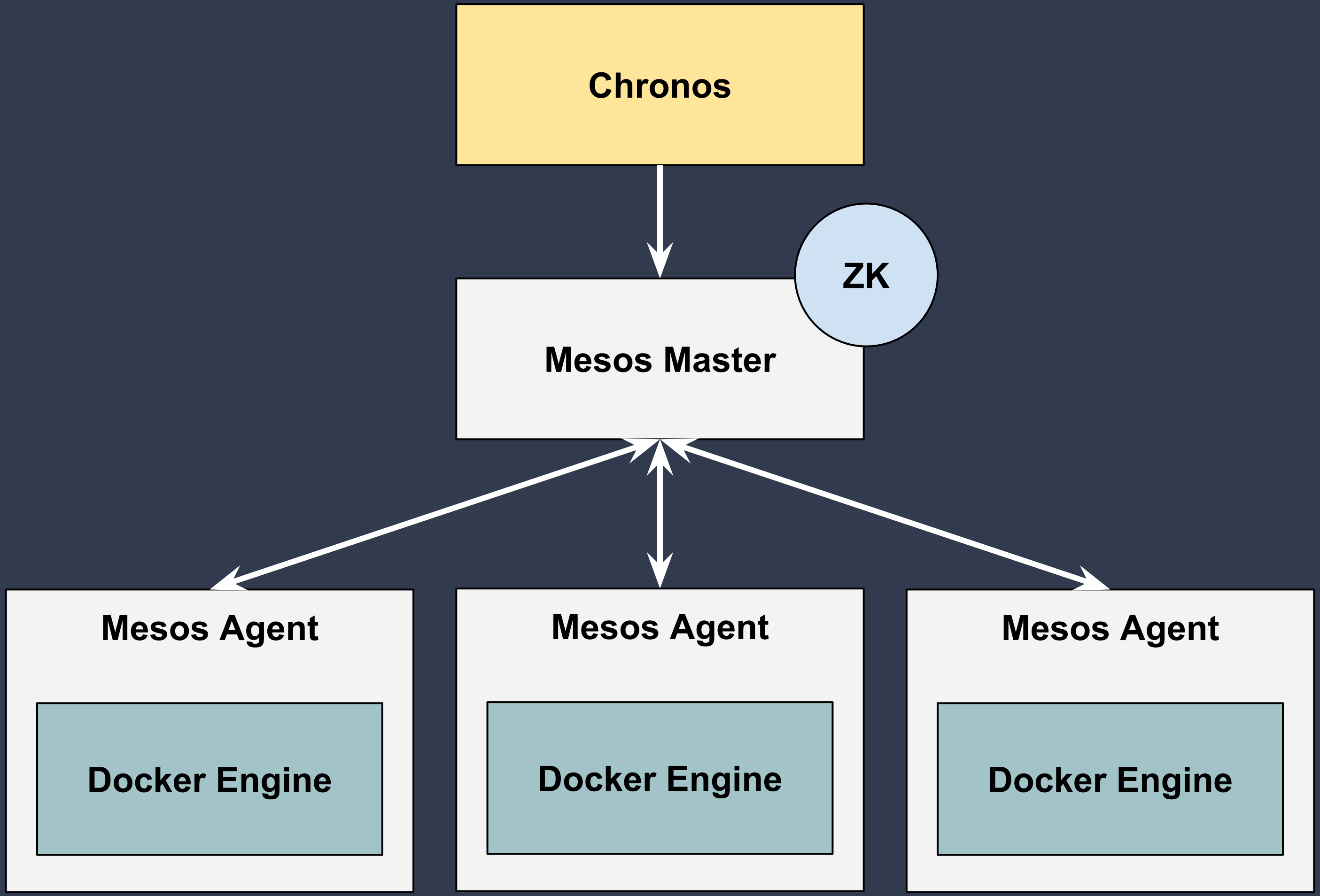
DEPLOYING MESOS, DOCKER, AND CHRONOS

DEPLOYMENT OVERVIEW

- Install/configure Mesos, ZooKeeper, Docker
- Stage a Docker image on the Mesos agents
- Install and configure Chronos
- Create a Chronos job (that runs in a Docker container)

DEPLOYING MESOS, DOCKER, AND CHRONOS

DEPLOYMENT OVERVIEW



DEPLOYMENT OVERVIEW

- Puppet’s roles/profiles pattern
- Using the following Puppet modules
    - deric-zookeeper
    - deric-mesos
    - garethr-docker
    - puppetlabs-chronos

All of these modules are open source and available via the Puppet Forge:
https://forge.puppetlabs.com

DEPLOYING MESOS (MASTER)

```
class role::mesos::master {
  include profile::base
  include profile::chronos
  include profile::mesos::master
  include profile::zookeeper
}
```

DEPLOYING MESOS (MASTER)

```
class profile::mesos::master {
  include profile::mesos::common
  class { '::mesos::master':  # From deric-mesos
    listen_address => $::ipaddress_eth0,
    work_dir       => '/var/lib/mesos',
    options        => {
      log_dir   => '/var/log/mesos',
      quorum    => '1',
    },
  }
}
```

DEPLOYING ZOOKEEPER

```
class profile::zookeeper {
  include java              # Include defaults from puppetlabs-java

  class { '::zookeeper':  # From deric-zookeeper
    client_ip => $::ipaddress_eth0,
    id        => '1',
    repo      => 'cloudera',
    require   => Class['java'],
  }
}
```

DEPLOYING MESOS (AGENT)

```
class role::mesos::agent {
  include profile::base
  include profile::docker
  include profile::mesos::agent
}
```

DEPLOYING MESOS (AGENT)

```
class profile::mesos::agent {
  include profile::mesos::common

  class { '::mesos::slave':  # From deric-mesos
    listen_address => $::ipaddress_eth0,
    work_dir       => '/var/lib/mesos',
    options        => {
      log_dir => '/var/log/mesos',
    },
  }
}
```

DEPLOYING MESOS (COMMON)

```
class profile::mesos::common {
  class { '::mesos':  # From deric-mesos
    repo      => 'mesosphere',
    zookeeper => 'zk://192.168.248.10:2181/mesos',
  }
}
```

DEPLOYING DOCKER

```
include ::docker           # Include defaults from garethr-docker

class { '::mesos::slave': # Let’s reconfigure the Mesos agent
  ...
  options => {
    containerizers                => 'docker,mesos',
    isolation                     => 'cgroups/cpu,cgroups/mem',
    executor_registration_timeout => '5mins',
  },
}
```

DEPLOYING CHRONOS

```
class profile::chronos {
  include ::chronos   # Include defaults from puppetlabs-chronos
}
```

BUILDING DOCKER IMAGES WITH PUPPET

GETTING STARTED WITH PUPPET AND DOCKER

Synopsis:

- Build a Docker image declaratively

Two approaches:

- `puppet agent` – pre-shared key to use existing Puppet infra
- `puppet apply` – directly apply manifests during build

GETTING STARTED WITH PUPPET AND DOCKER

Synopsis:

- Build a Docker image declaratively

Two approaches:

- `puppet agent` – pre-shared key to use existing Puppet infra
- **`puppet apply` – directly apply manifests during build**

GETTING STARTED WITH PUPPET AND DOCKER

```
FROM debian:wheezy
MAINTAINER Roger Ignazio <roger@mesosphere.com>
WORKDIR /tmp

RUN curl -sOL https://apt.puppetlabs.com/puppetlabs-release-wheezy.deb
RUN dpkg -i puppetlabs-release-wheezy.deb
RUN apt-get update
RUN apt-get -y install puppet
COPY * ./
RUN puppet apply example.pp
```

GETTING STARTED WITH PUPPET AND DOCKER

```
package { ['ruby', 'ruby-dev', 'build-essential']: ensure => installed, }

package { 'httparty': ensure => installed, provider => gem, }

file { '/usr/bin/query_mesos':
  ensure => file,
  mode   => '0755',
  source => '/tmp/query_mesos.rb',
}
```

GETTING STARTED WITH PUPPET AND DOCKER

```
Step 10 : RUN puppet apply example.pp
 ---> Running in 12eda5e24ff8
Notice: Compiled catalog for 90c88c41cdaa.bad in environment production in 0.16 seconds
Notice: Package[build-essential]/ensure: ensure changed 'purged' to 'present'
Notice: File[/usr/bin/query_mesos]/ensure: defined content as '{md5}
e44268ac8e31f75f1aeee961d0ebe36b'
Notice: Package[ruby-dev]/ensure: ensure changed 'purged' to 'present'
Notice: Package[httparty]/ensure: created
Notice: Finished catalog run in 33.22 seconds
 ---> 1a8fefd724ee
Removing intermediate container 12eda5e24ff8
Successfully built 1a8fefd724ee
```

STAGING DOCKER IMAGES ON NODES

Using the garethr-docker Puppet module

```
docker::image { 'rogerignazio/basic-puppet-example':
  image_tag => 'latest',
}
```

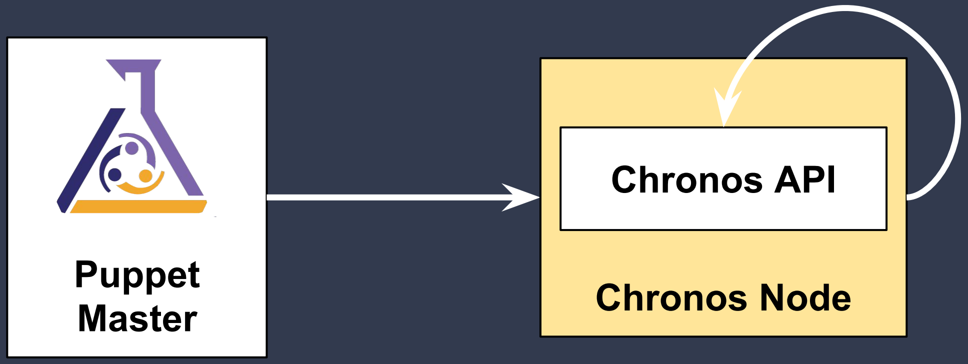
Equivalent to

```
$ docker pull rogerignazio/basic-puppet-example:latest
```

MANAGING CHRONOS JOBS WITH PUPPET

A CUSTOM TYPE AND PROVIDER

- Bundled with a module
- Found at `lib/puppet/type` and `lib/puppet/provider`
- Model the API of an external service – as Puppet code



A CUSTOM TYPE AND PROVIDER

```
chronos_job { 'fetch_mesos_master_metrics':
  command      => 'query_mesos 192.168.248.10',
  job_schedule => 'R/2015-10-09T00:00:00.000Z/PT1M',
  container    => {
    type  => 'DOCKER',
    image => 'rogerignazio/basic-puppet-example',
  },
  cpus         => 0.5,
  mem          => 256,
  owner        => 'roger@mesosphere.com',
}
```

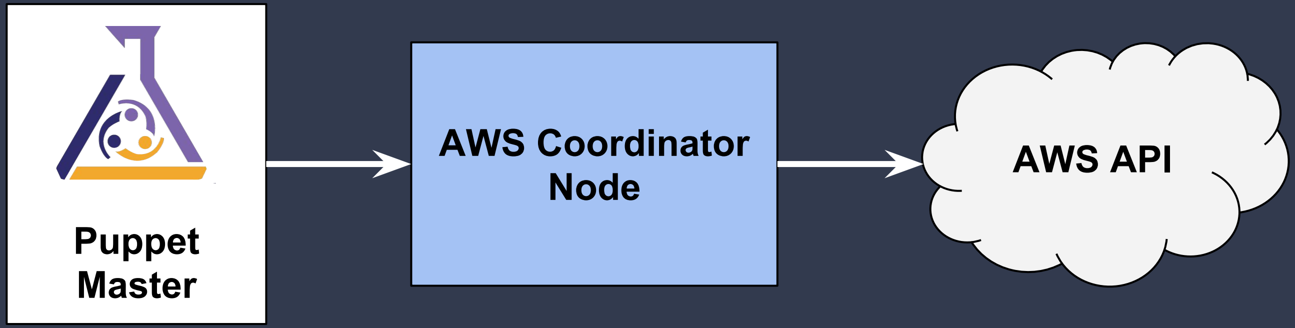
DEMO

PROVISIONING INFRASTRUCTURE

PROVISIONING INFRASTRUCTURE

CLOUD PROVISIONING WITH AWS

- Declare AWS infrastructure as Puppet resources
- Custom types and providers hit the AWS API
  - Ensures resources are in desired state



PROVISIONING INFRASTRUCTURE

CLOUD PROVISIONING WITH AWS

```
ec2_instance { 'mesos-slave-NN':
  ensure          => present,
  region          => 'us-west-2',      # US West (Oregon)
  image_id        => 'ami-4dbf9e7d',   # AWS RHEL 7.1 image
  instance_type   => 'c4.xlarge',      # 4 CPUs, 7.5 GB mem
  security_groups => ['mesos-aws-secgrp'],
}
```

CLOUD PROVISIONING WITH AWS

Some of the available resource types:

- `ec2_instance`
- `ec2_securitygroup`
- `ec2_vpc`
- `elb_loadbalancer`
- `route53_a_record`

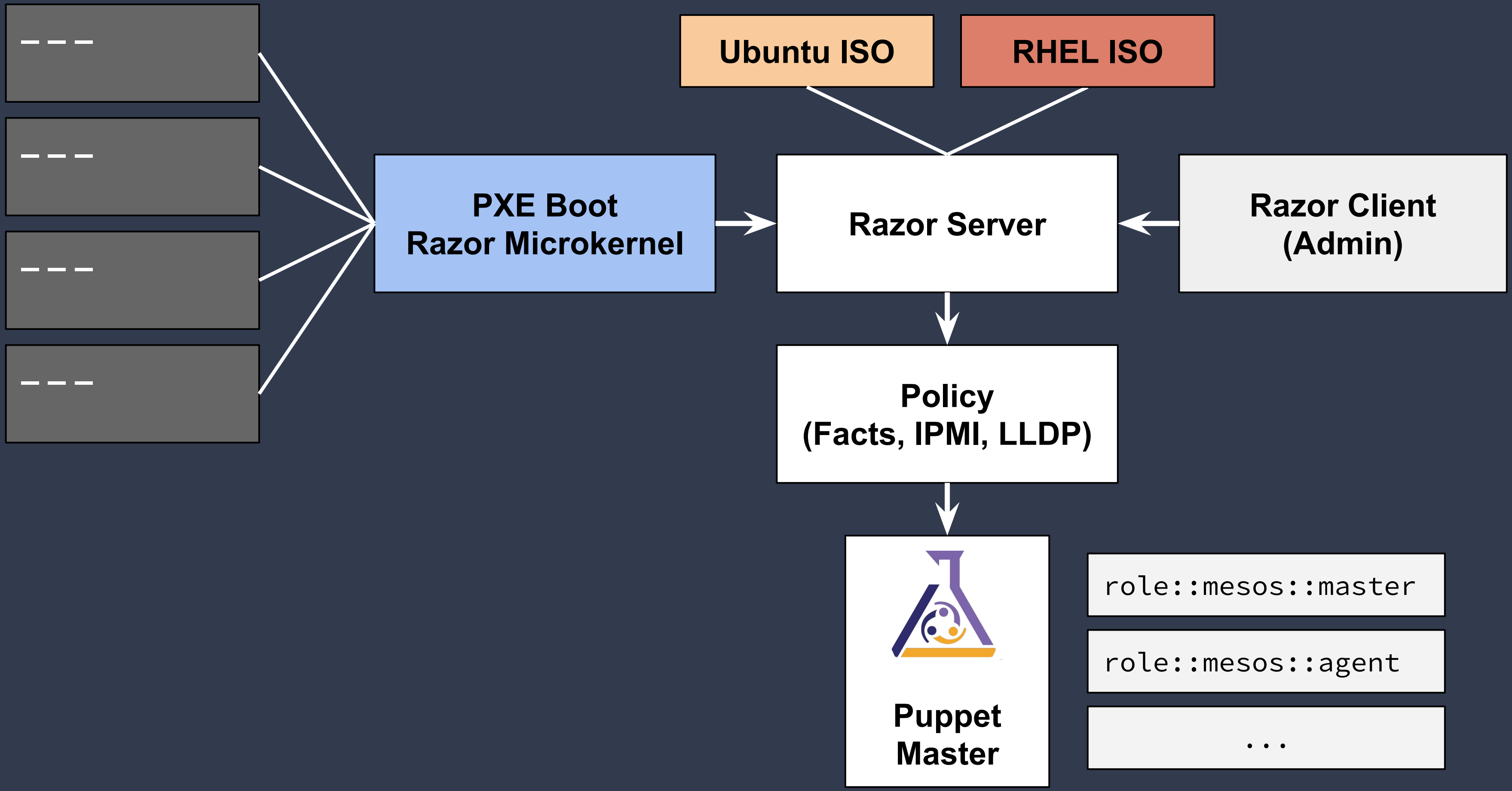
A more complete example: http://bit.ly/puppet-aws-example

BARE-METAL PROVISIONING WITH RAZOR

- Auto-discover inventory
- Policy-based provisioning
- Pluggable “brokers”
- Razor is open source – Apache License, v2

PROVISIONING INFRASTRUCTURE

BARE-METAL PROVISIONING WITH RAZOR



BARE-METAL PROVISIONING WITH RAZOR

For more information, check out
http://bit.ly/razor-intro

Q & A

```
puppetconf_talk { 'managing_mesos':
  ensure  => presented,
  speaker => 'Roger Ignazio',
  email   => 'roger@mesosphere.com',
  twitter => '@rogerignazio',
}
```