SC12

Salt Lake City, Utah

Conference Dates:
November 10-16, 2012

Exhibition Dates:
November 12-15, 2012

IEEE computer society

Association for Computing Machinery

Infrastructure Clouds for Science and Education: Platform Tools

Kate Keahey, John Bresnahan, Patrick Armstrong, Pierre Riteau

Argonne National Laboratory

University of Chicago

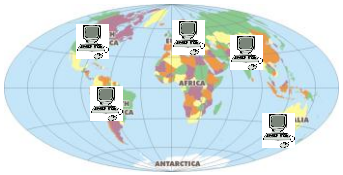
The Power of Infrastructure Clouds

Virtualization opens the flood gates

- Outsourcing

- Virtual appliances
  - Freeze your stack in time
  - Run it anywhere

- Multi-cloud applications
  - Run many copies all over the world!



- Elasticity

Harnessing The Power

- Organization tools and techniques

What Needs To Be Harnessed

- VM (appliance) creation and development
  - configuration management tools (chef, puppet)
- VM hypervisors
  - Infrastructure-as-a-Service (IaaS)
- Cloud applications
  - virtual clusters, cloudinit.d, CloudFormation
- Elasticity
  - Auto-scaling tools, phantom

What Needs To Be Organized?

- VM (appliance) creation and development
  - configuration management tools (chef, puppet)
- VM hypervisors
  - Infrastructure-as-a-Service (IaaS)
- Cloud applications
  - virtual clusters, cloudinit.d, CloudFormation,
- Elasticity
  - Auto-scaling tools, phantom

VM Applications

- An entire system frozen in time
  - Full software stacks (versions)
  - Configuration files
  - Important for science!
- A dedicated modular service
  - Web service, database, AMQP node, etc
- Demos
- A binary single file (or set of files)
  - Easy to *freeze*

Developing Appliances

- A single binary image?
  - Many developers?
  - Version control?
  - Merging conflicts?

- Base image with a description
  - Ex: Ubuntu 11.04 base images plus a set of scripts
- Configuration Management Software
  - Chef, Puppet, FG Rain, etc

Chef

- Software stack description
  - ruby and json
- A library of cookbooks
- Cookbooks contain recipes
  - Ex: apache2 server with php4
- Attributes to customize each recipe
  - Ex: on what port will apache listen
- Templates for configuration files
- Appliance developers make recipes
  - Version control can be done with git/svn/cvs…

Example Recipe

```
app_dir = node[:appdir]
ve_dir = node[:virtualenv][:path]

git app_dir do
  repository node[:autoscale][:git_repo]
  reference node[:autoscale][:git_branch]
  action :sync
  user node[:username]
  group node[:groupname]
end

execute "run install" do
    cwd app_dir
    user node[:username]
    group node[:groupname]
    command "python setup.py install"
end
```

Example Template

```
phantom:
  system:
    type: epu
    rabbit: <%= node[:autoscale][:rabbit_host] %>
    rabbit_port: <%= node[:autoscale][:rabbit_port] %>
    rabbit_ssl: False
    rabbit_user: <%= node[:autoscale][:rabbit_username] %>
    rabbit_pw: <%= node[:autoscale][:rabbit_password] %>
    rabbit_exchange: <%= node[:autoscale][:rabbit_exchange] %>
  authz:
    type: sqldb
    dburl: <%= node[:autoscale][:dburl] %>
```

```
phantom:
  system:
    type: epu
    rabbit: vm-102.uc.futuregrid.org
    rabbit_port: 5672
    rabbit_ssl: False
    rabbit_user: XXX
    rabbit_pw: PPPPPP
    rabbit_exchange: default_dashi_exchange
  authz:
    type: sqldb
    dburl: mysql://nimbus:XXXX@futuregrid.org/testphantom
```

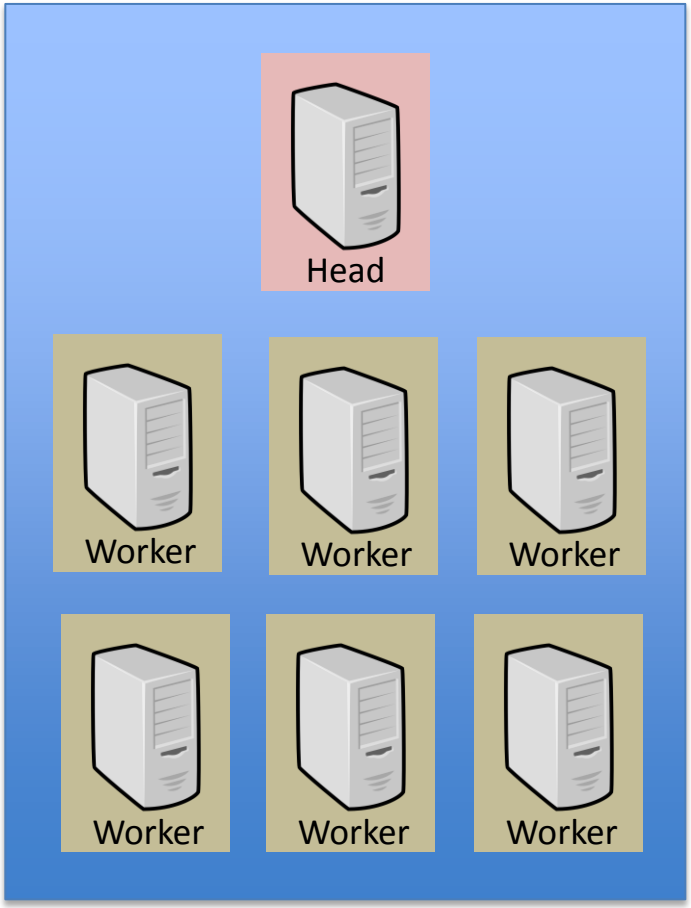
What Needs To Be Organized?

- VM (appliance) creation and development
  - configuration management tools (chef, puppet)
- VM hypervisors
  - Infrastructure-as-a-Service (IaaS)
- Cloud applications
  - virtual clusters, cloudinit.d, CloudFormation,
- Elasticity
  - Auto-scaling tools, phantom

Cloud Applications
Virtual Clusters

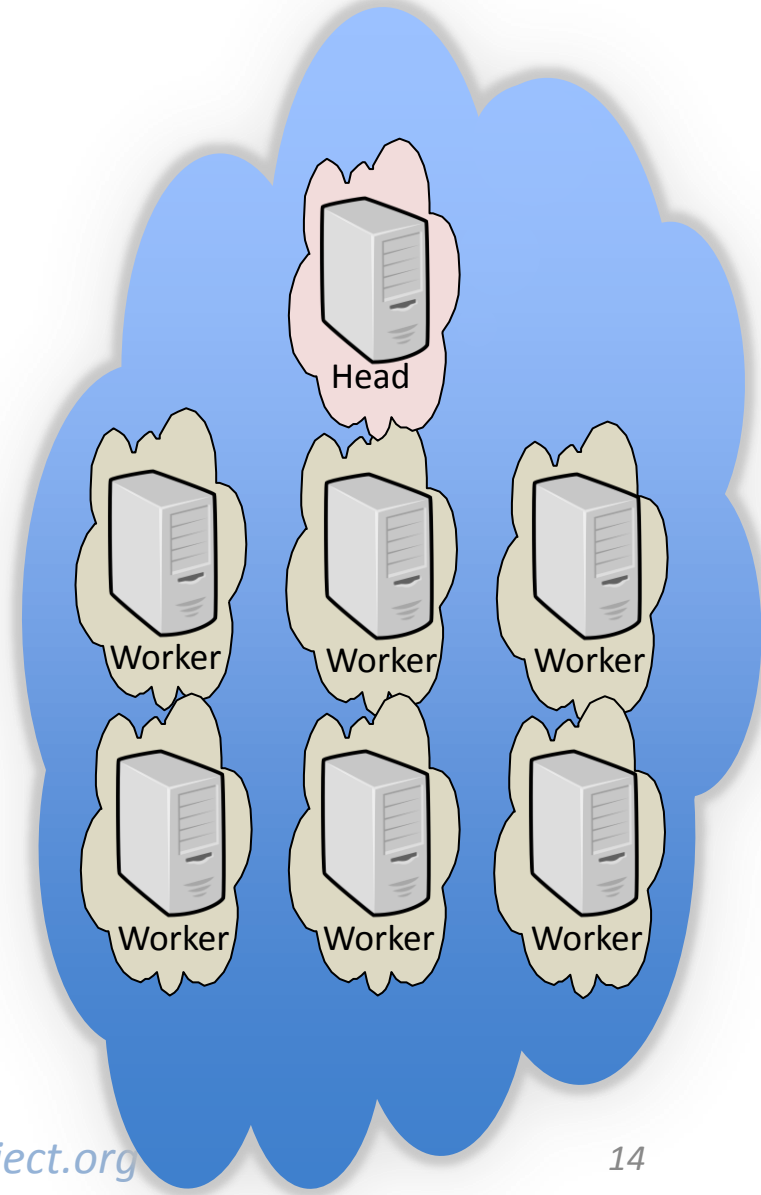
Linux Clusters

- A cheap answer to *Super Computers*
  - Many "commodity" machines interconnected to operate as single machine
- Load distributed across nodes
- Compute clusters
- A single head node with N worker nodes



Virtual Clusters

- The same thing
  - composed of VMs in a cloud
- Advantages
  - A stable, pre-fabricated workspace
  - Elastic
  - repeatable
- Disadvantages
  - Unknown hardware conditions
    - Network, noisy neighbors, etc.



Nimbus One-Click Clusters

- Virtual Clusters In Nimbus Clouds
- Turnkey, tightly-coupled cluster
  - Shared trust/security context
  - Shared configuration/context information
- Sample images
  - NFS file systems, torque queues, GridFTP servers, etc…
- Easily repeatable and distributable
  - An xml file and a VM image

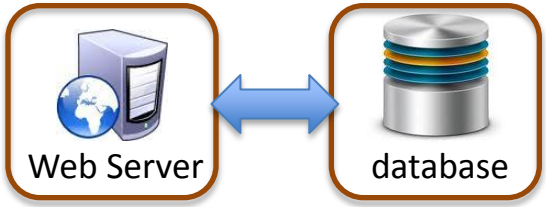
Nimbus One-Click Clusters Context

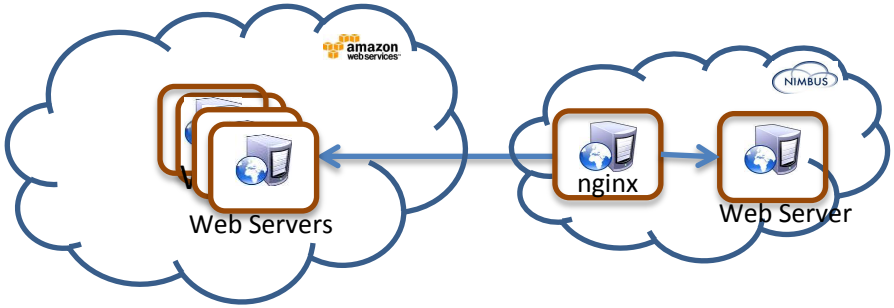
- *Personalize* a VM instance
  - seed them with secrets, access policies, and just-in-time configurations
  - Populate /etc/hosts with cluster member addresses
  - Set up SSH host-based authentication across all accounts
- Must run a light weight script (context agent) on boot.
  - Context broker/agent discussed in detail later

Other Cloud Applications

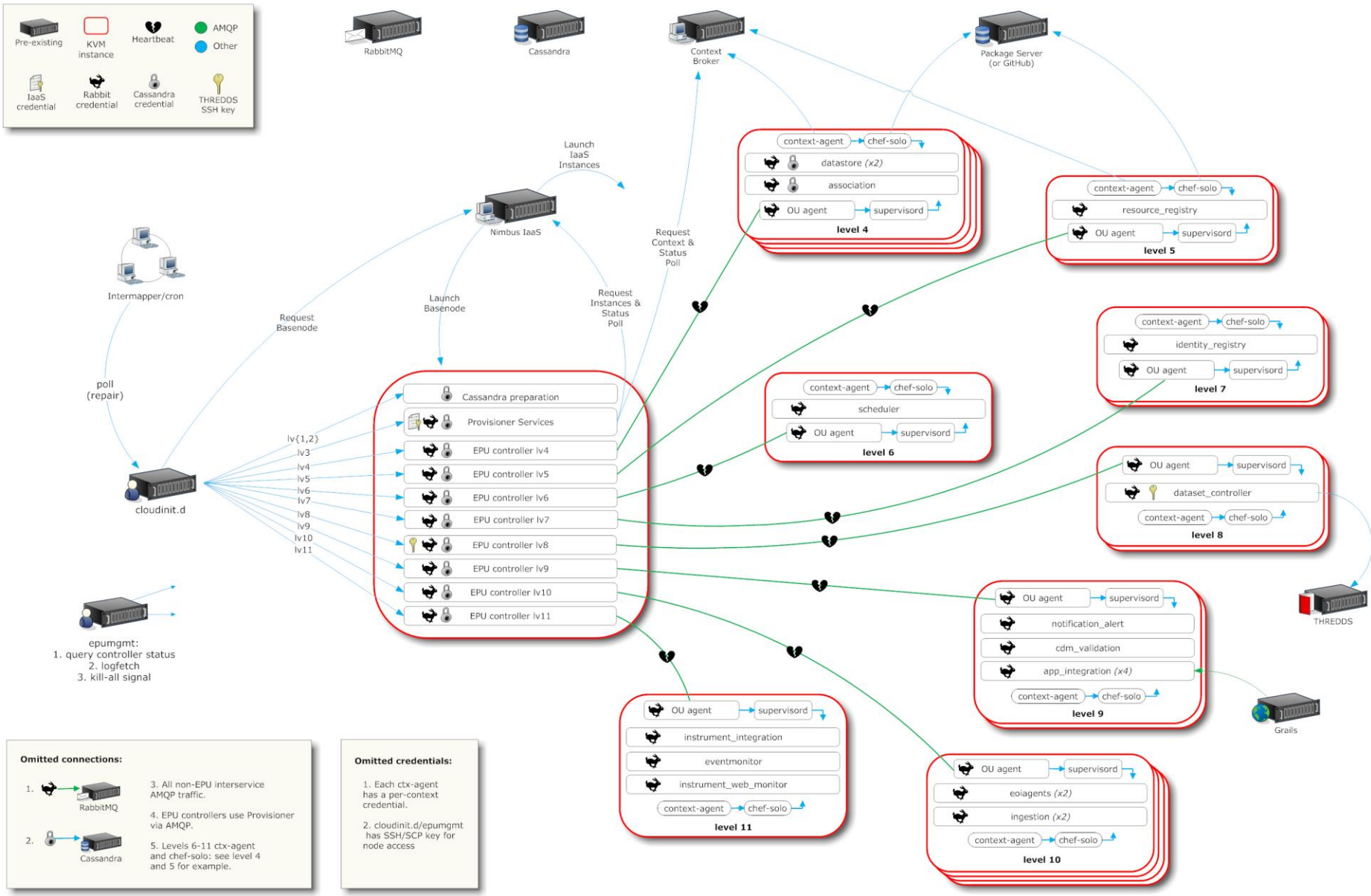
Cloud Applications

- More than 1 VM needed for the job
- Information exchange is needed
  - Manual information exchange ☹
- Multi-cloud
  - Cloud independence required





A Simplified Deployment Scenario



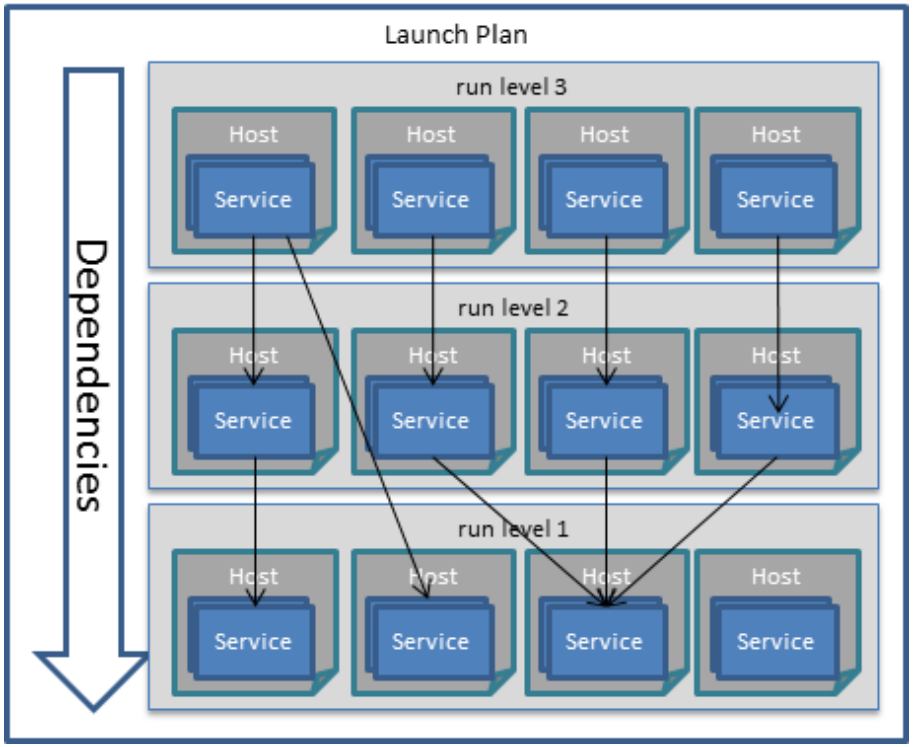
cloudinit.d

- Multicloud VM dependency management
  - Uses the libcloud abstraction library
- Integrated with chef solo
- ini file format descriptions
  - Coupled with any executable script
- Launch plan end-users/operators
  - Lightweight
  - Copy launch plan and “one click” action
  - Easily reconfigured for various clouds
- Launch plan/application developers:
  - Minimal software assumptions (ssh)
  - “Stem cell” deployment approach
  - Incremental launch plan development

```
[svc-alamoHTTP]
iaas_key: XXXXXX
iaas_secret: XXXX
iaas_host: alamo.futuregrid.org
iaas_port: 8443
iaas: Nimbus
image: ubunut10.10
ssh_username: ubuntu
localsshkeypath: ~/.ssh/fg.pem
readypgm: http-test.py
bootpgm: http-boot.sh
```

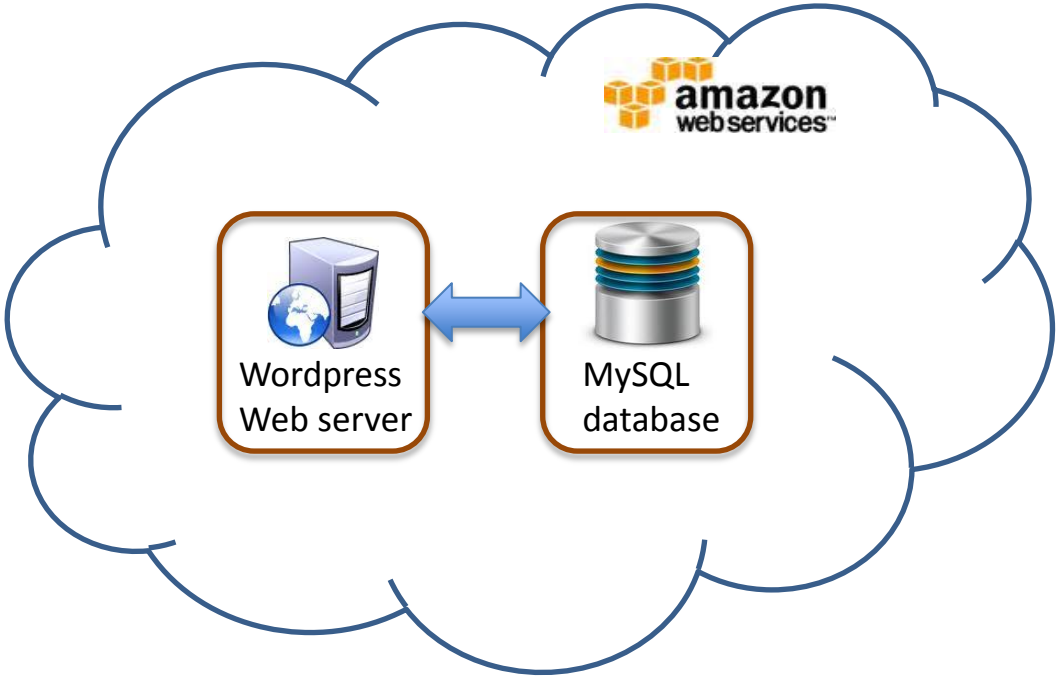
cloudinit.d Overview

- Services
- Run Levels
  - Collections of services without dependencies on each other
- Launch Plan
  - An ordered set of run levels

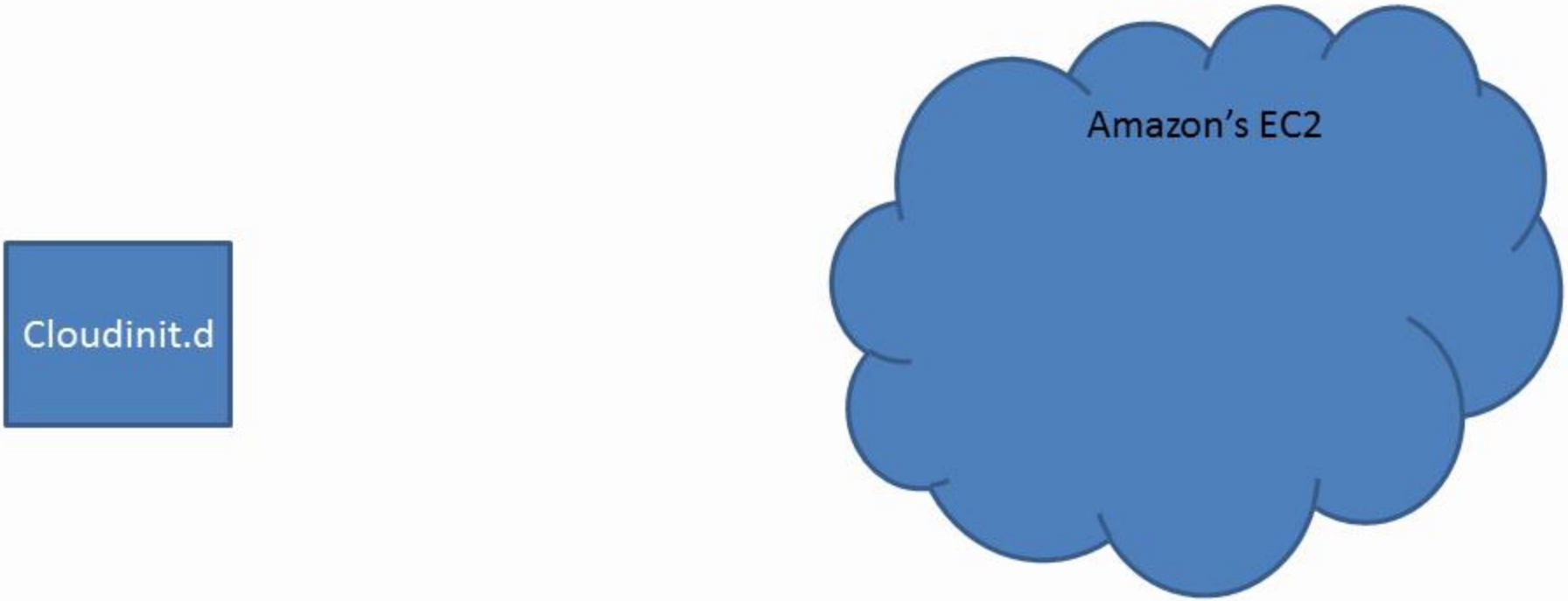


cloudinit.d example

- Wordpress example
- 2 virtual machine running in EC2
  - MySQL
  - Wordpress
- MySQL contact information injected into wordpress



Creating a Wordpress Service with cloudinit.d



What Needs To Be Organized?

- VM (appliance) creation and development
  - configuration management tools (chef, puppet)
- VM hypervisors
  - Infrastructure-as-a-Service (IaaS)
- Cloud applications
  - virtual clusters, cloudinit.d, CloudFormation,
- Elasticity
  - Auto-scaling tools, phantom

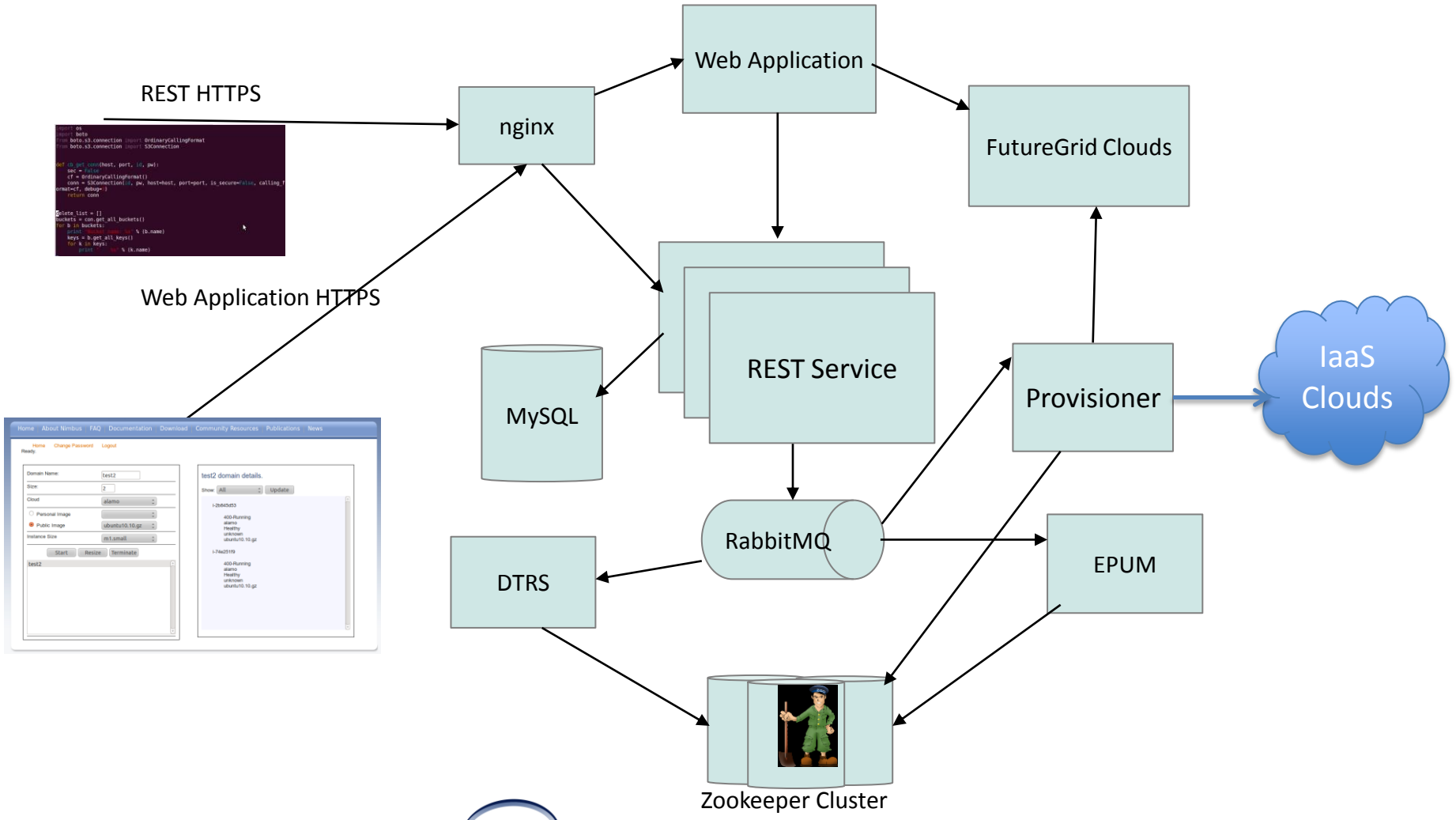
Scaling Considerations

- Reasons to scale
  - Business vs science
    - Cost vs quota
- Lossy environment
  - VMs fail more often than bare metal
  - N preserving
- Spot instances
  - If the price is right
- Backfill
  - If resources are idle

Infrastructure Platform Goals

- Multi-cloud
  - Work across private, community and commercial clouds
- Any Scale
  - Scale in response to a diverse set of sensors/triggers
  - Both system and application sensors
- High Availability
  - “Any VM can die”: system or user VMs
  - Minimizing time to recovery (TTR)
- Your Polices, Our Enactment
  - User-defined sensors/triggers and policies
- Engineered from the ground up to work with infrastructure clouds
- Easy on the user

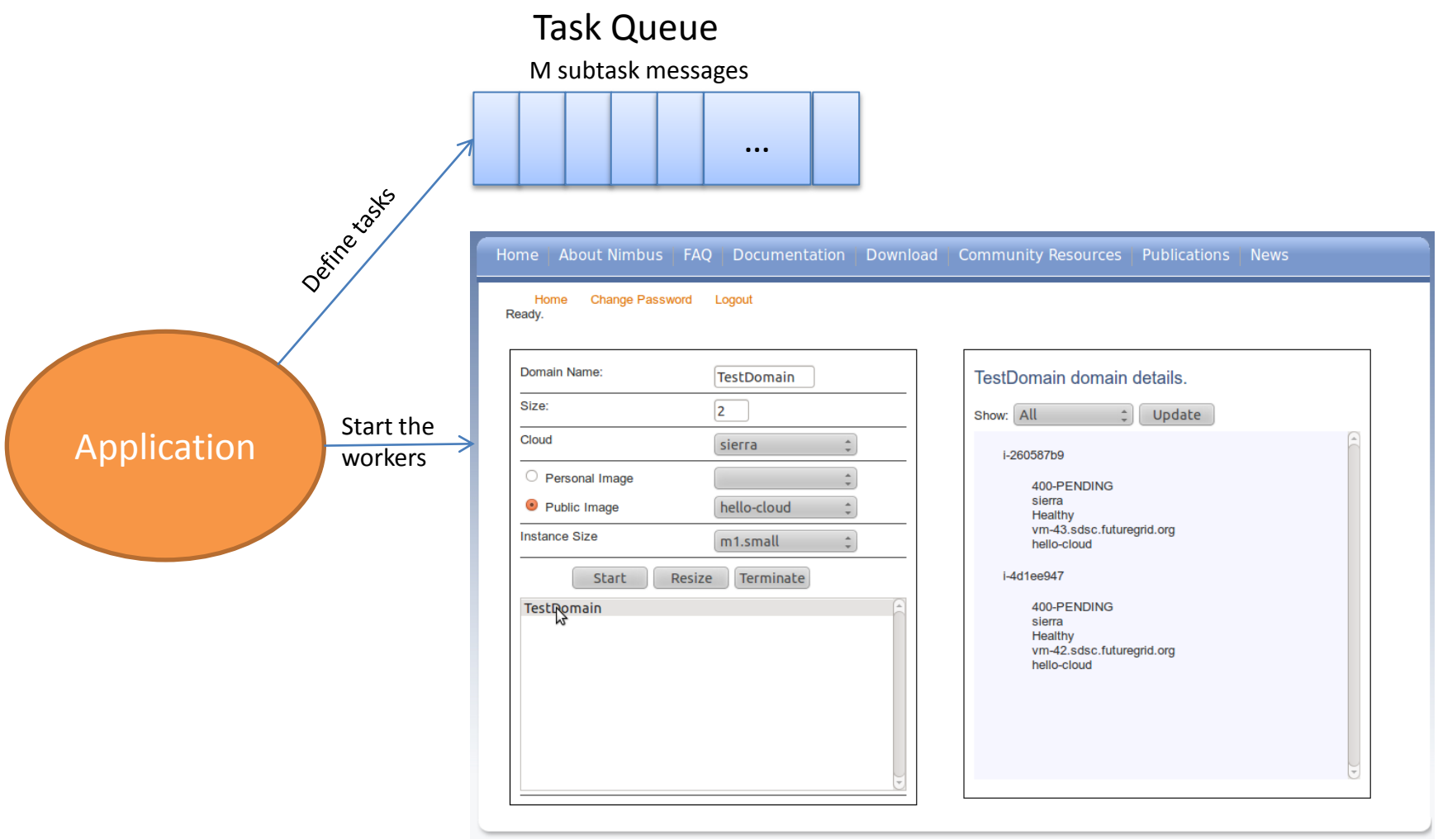
Phantom Architecture



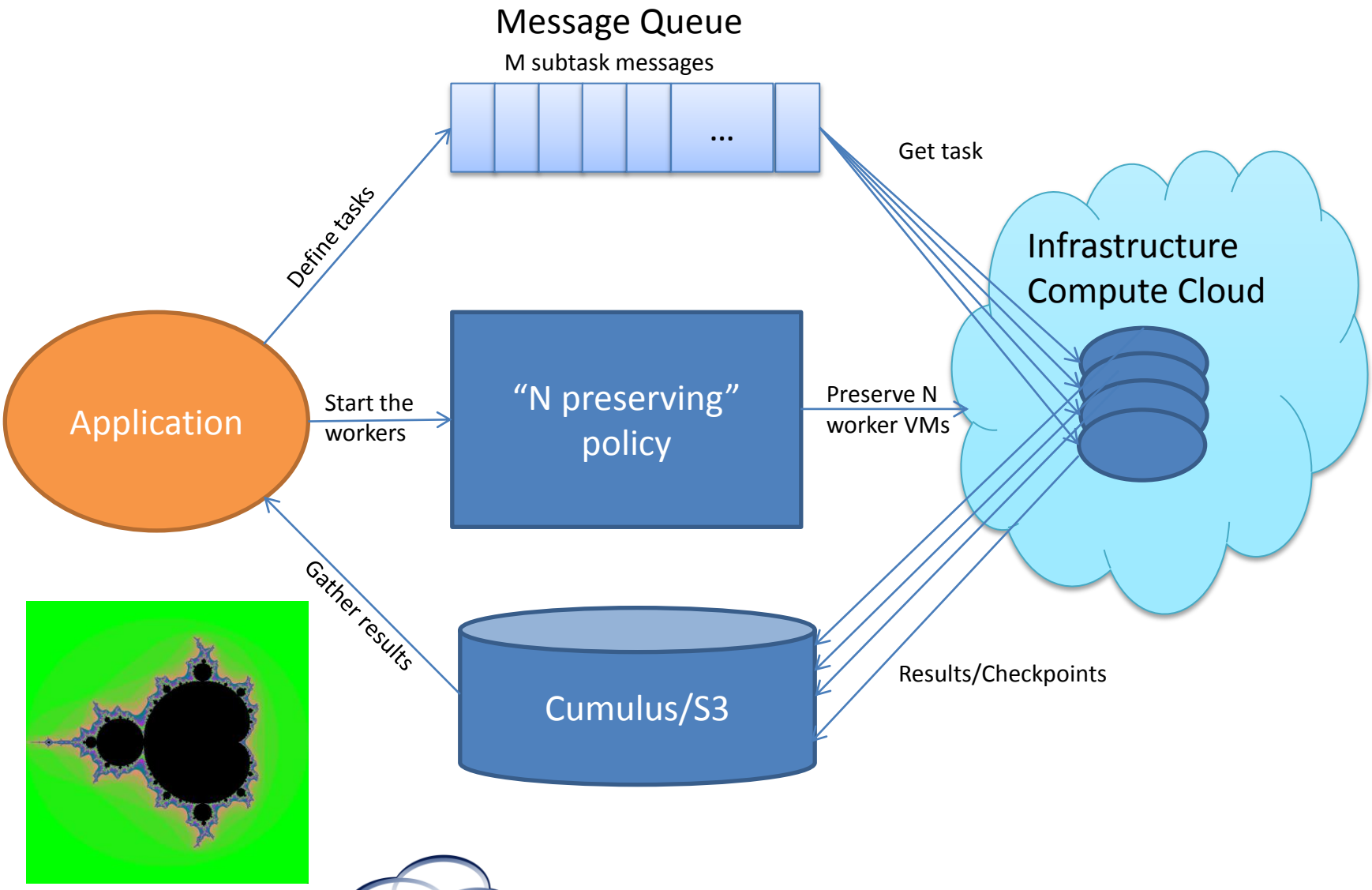
How Can Science Plug Into This Power

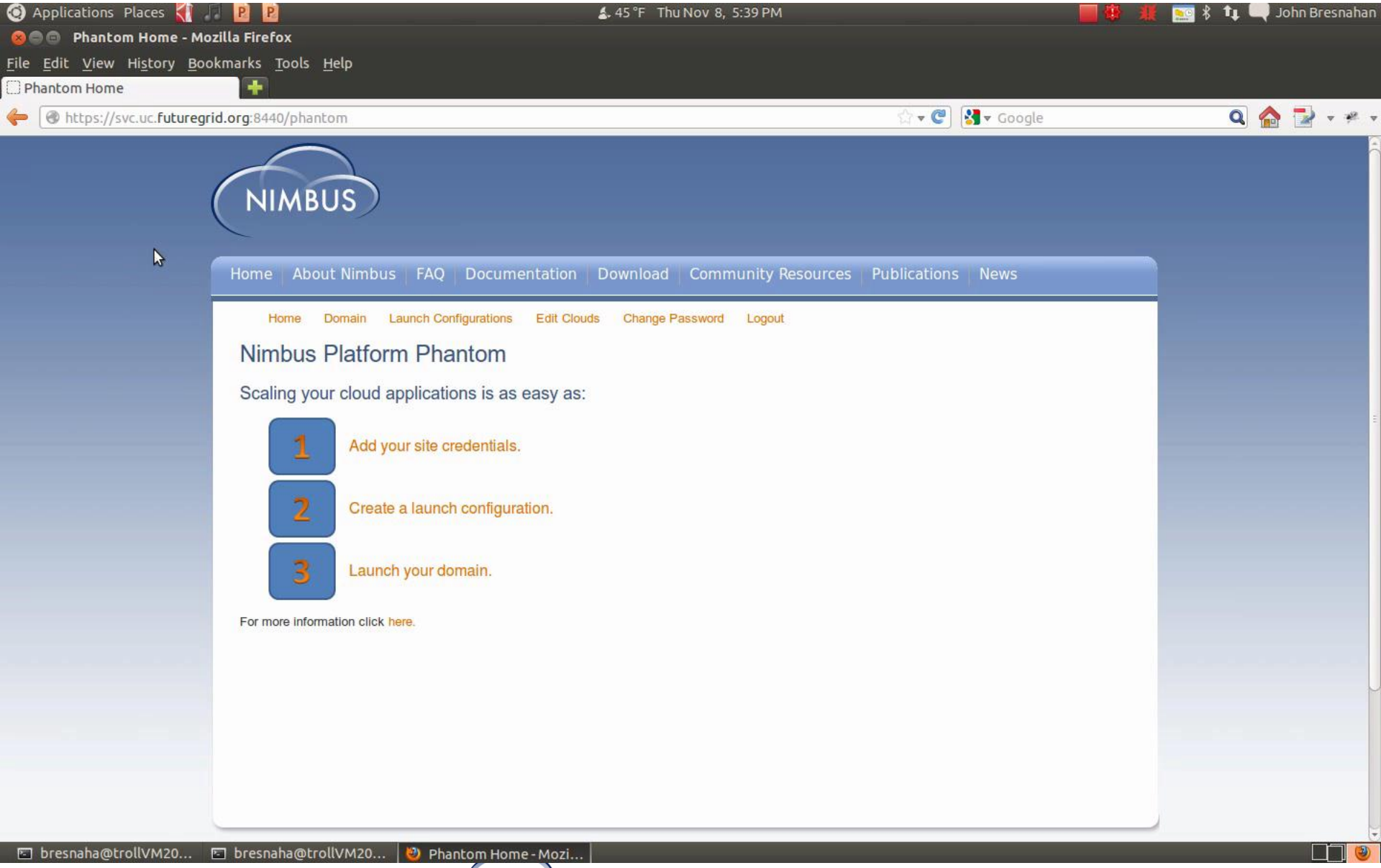
Example Embarrassingly Parallel Scientific Application Demonstration

Using Nimbus Domains



Using Nimbus Domains



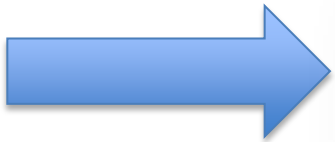

Applications Places
45 °F Thu Nov 8, 5:39 PM
John Bresnahan
Phantom Home - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Phantom Home
https://svc.uc.futuregrid.org:8440/phantom
Google
NIMBUS
Home About Nimbus FAQ Documentation Download Community Resources Publications News
Home Domain Launch Configurations Edit Clouds Change Password Logout
Nimbus Platform Phantom
Scaling your cloud applications is as easy as:
1 Add your site credentials.
2 Create a launch configuration.
3 Launch your domain.
For more information click here.
bresnaha@trollVM20...
bresnaha@trollVM20...
Phantom Home - Mozi...

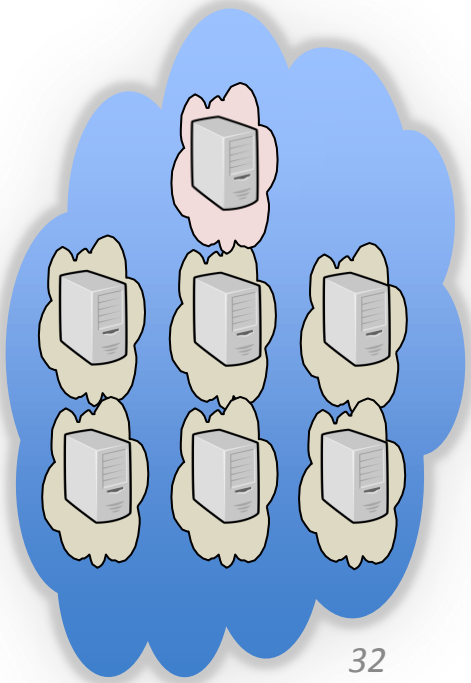
One Click Cluster Exercise

- Examine the one click cluster definition
- Launch a virtual cluster
- Inspect the virtual cluster

```
<?xml version="1.0" encoding="UTF-8"?>

<cluster>
 <workspace>
  <name>head-node</name>
  <image>base-cluster-cc14.gz</image>
  <quantity>1</quantity>
  <nic wantlogin="true">public</nic>
  <ctx>
   <provides>
     <identity />
     <role>nfsserver</role>
   </provides>
```





32