

Beaker 101

Acceptance Test Everything



Introductions

I'm Alice Nodelman, I create automation tools.

alice@puppetlabs.com

anode on freenode irc (#puppet-dev)

What is Beaker?

- A black box acceptance testing tool.
- Provisions, configures and manages hosts under test.
- Executes test files written in Ruby with Beaker DSL functions and procedures.

DSL?

Domain-specific Language

The Beaker DSL is an extension to Ruby for PuppetLabs specific testing steps. Test files have access to all of Ruby plus these additional functions & procedures.

Get Beaker

Requires Ruby 1.8+

```
$ gem install beaker
```

```
$ beaker --help
```

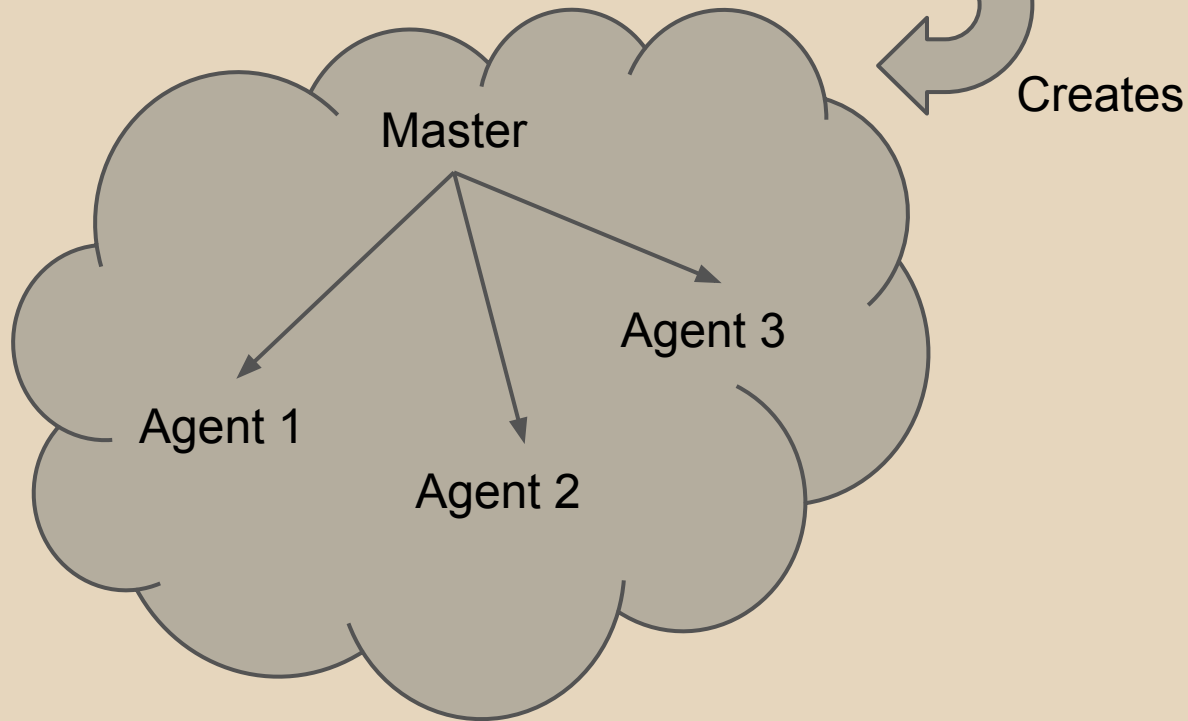
The Workflow

1. Provision hosts for testing
2. Validate & configure test hosts
3. Run provided test files on hosts
4. Report test results
5. Teardown test hosts

Command Line Basics

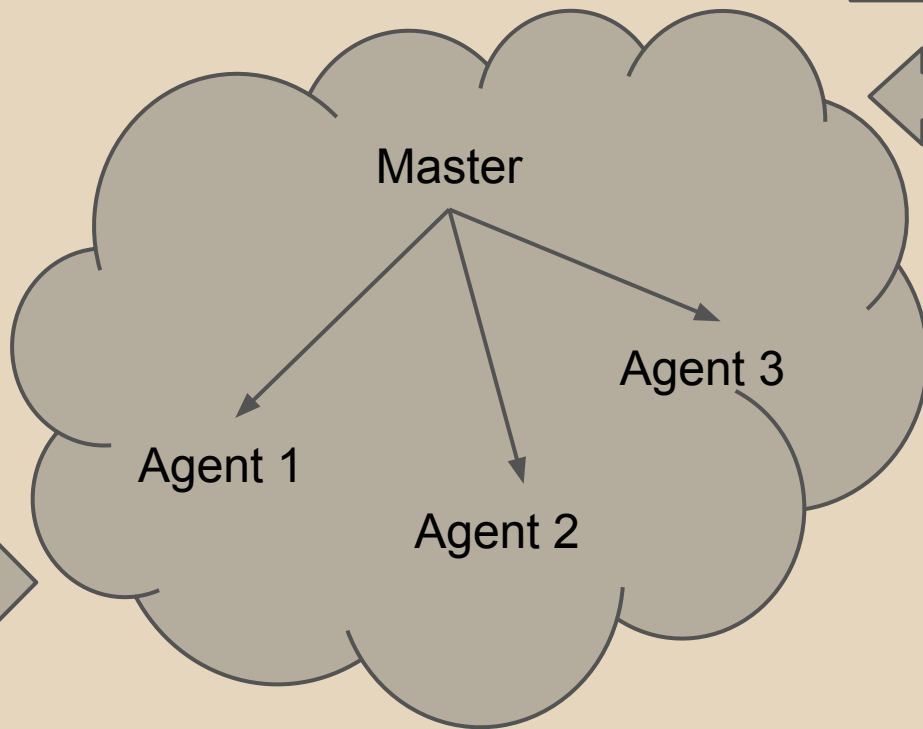
- `--hosts file`
 - to describe your test hosts
- `--log-level debug`
 - for more messaging
- `--preserve-hosts [mode]`
 - policy for cleaning up hosts post-testing
 - one of *always*, *onfail*, *never*
- `--no-provision`
 - use hosts that are already up and ready
- `--parse-only`
 - read all parameters, environment variables and config options and report the final Beaker configuration
- `--help`
 - is there to help

```
$ beaker --hosts nodes.cfg <...>
```




```
$ beaker --hosts nodes.cfg --preserve-hosts always <...>
```

Creates



Re-uses

```
$ beaker --hosts nodes.cfg --preserve-hosts always --no-provision <...>  
$ < continued development >  
$ < done with test hosts, throw away and rebuild from snapshot >
```

Start Over

Supported Test Host Providers

- Vagrant
- Google Compute Engine
- Amazon EC2
- VMware Fusion
- vSphere
- vCloud
- Solaris Zones
- Docker (coming soon!)
- OpenStack (coming soon!)

HOSTS:

ubuntu-10-04-4-x64-master:

roles:

- master
- agent
- dashboard
- database

platform: ubuntu-10.04-amd64

hypervisor : vagrant

box: ubuntu-server-10044-x64-vbox4210

box_url : <http://boxes/ubuntu-server-10044-x64-vbox4210.box>

ubuntu-10-04-4-x64-agent:

roles:

- agent

platform: ubuntu-10.04-amd64

hypervisor : vagrant

box: ubuntu-server-10044-x64-vbox4210

box_url : <http://boxes/ubuntu-server-10044-x64-vbox4210.box>

CONFIG:

pe_ver : 3.1.1

pe_dir : <https://buildsarehere/pe-builds/released/3.1.1>

HOSTS:

centos-6-master:

roles:

- master
- agent
- database
- dashboard

platform: el-6-x86_64

image: centos-6-x86_64

hypervisor: google

debian-7-agent:

roles:

- agent

platform: debian-7-amd64

image: debian-7-wheezy

hypervisor: google

CONFIG:

gce_project : beaker-compute-api-integration

gce_keyfile : /Users/user/privatekey.p12

gce_password: notasecret

gce_email : 73f51fpni@developer.gserviceaccount.com

pe_dir : https://buildsarehere/pe-builds/released/3.1.1

pe_ver : 3.1.1

Roles, what are they?

- master, agent, dashboard, database, default
- user defined!

Provide shortcuts to access machines that have particular responsibilities.

Beaker DSL - your helpful friend

- Installation
 - *install_pe, upgrade_pe, install_puppet*
- File
 - *scp_to, scp_from*
- Command execution
 - *on, shell, puppet, facter, hiera, fact_on*
- Lots more!

Let's look at tests!

```
test_name "my test"  
step "first, say hello everywhere"  
hosts.each do | host |  
  on host, "echo hello"  
end  
step "second, say hello on the master"  
result = on master, "echo hello"  
step "third, check our command output"  
assert_equal(result.stdout, "hello")
```

```
test_name "/etc/init.d/pe-mcollective restart check"
```

```
confine :except, :platform => 'solaris'
```

```
confine :except, :platform => 'windows'
```

```
confine :except, :platform => 'aix'
```

```
step "Make sure the service restarts properly"
```

```
hosts.each do |host|
```

```
  # Commands to execute on the target system.
```

```
  restart_command = "bash -c [[ -x /etc/init.d/pe-mcollective ]] && /etc/init.d/pe-mcollective restart"
```

```
  process_count_check = "bash -c '[[ $(ps auxww | grep [m]collectived | wc -l) -eq 1 ]]'"
```

```
  # Restart once
```

```
  on(host, restart_command) { assert_equal(0, exit_code) }
```

```
  # Restart again
```

```
  on(host, restart_command) { assert_equal(0, exit_code) }
```

```
  # Check to make sure only one process is running
```

```
  on(host, process_count_check) { assert_equal(0, exit_code) }
```

```
end
```


Coming up Next

- more hypervisors!
- a 'pooling api' to sit between Beaker and the cloud
- creation of a 'best practices' guide
- more and better documentation

What we haven't talked about

beaker-RSpec

- A shim between RSpec and Beaker that allows for Rspec-style test construction with full access to the Beaker DSL
- Used for Puppet module testing
- Listen to Hunter, he knows more

Documentation

Beaker code

- <https://github.com/puppetlabs/beaker>

Beaker Documentation

- <https://github.com/puppetlabs/beaker/wiki>

File a Bug

- <https://github.com/puppetlabs/beaker/issues>

Beaker DSL

- <http://rdoc.info/github/puppetlabs/beaker/master/Beaker/DSL>