



CHEFTM
CODE CAN



What's New with Chef?

Nathen Harvey

@nathenharvey



Nathen Harvey

- nharvey@chef.io
- @nathenharvey
- Co-organizer – DevOps DC
- <http://bit.ly/farmer-nathen> (explicit)
- <http://bit.ly/farmer-nathen-sfw>



Agenda

- Chef 12
- CYOA
- Q & A





Chef 12



What's NOT in Chef 12?



What's NOT in Chef 12?

- Differentiation between Open Source & Enterprise Chef Server
- WebUI – The OSS WebUI has been completely deprecated. WebUI is now a premium-only add-on.
- Support for Ruby < 2.0.
- Daemonized clients can no longer be run with --no-fork.
- CouchDB is 100% gone (was previously still part of Enterprise Chef).
- Whole Buncha daemons got the axe: opscode-account, opscode-certificate, oc_authz_migrator, opscode-org-creator, orgmapper.





What's New in Chef Server 12?



The Chef World Has Grown

New OSS since Chef 11 Launch:

- Chef Development Kit
- Test Kitchen
- Chef Provisioning (née Chef-Metal)
- Powershell DSC Support
- Chef-zero (--local-mode)
- Supermarket
- Push Jobs Went Open Source!



The Chef World Has Grown

New Premium Features since Chef 11 Launch:

- Chef Analytics
- Chef Reporting
- Chef Management Console
- Chef Server Replication
- Chef Server Pluggable HA



Changes in the Chef Community

- RFC Process – Get involved!
 - <https://github.com/opscode/chef-rfc>
- Supermarket
 - <https://supermarket.chef.io>
- LearnChef
 - <https://learn.chef.io>
- Switch to Github Issues



The Big Deal With Chef Server 12

- No more difference between OSS and Enterprise Chef Server 12
 - There is one Chef Server, and it is open source.
- Once-premium features are now free for everybody:
 - Multi-tenancy (“Organizations”)
 - Role-based access controls
 - Push Jobs Server & Client
- All Premium features are free up to 25 nodes
- Toggling Premium Features is one command.
 - `chef-server-ctl install opscode-manage`
 - `chef-server-ctl remove opscode-manage`



There is One Chef Server, And it is Open Source

- People had to choose up front whether to use OSS Chef Server, Chef-solo, or Chef Enterprise
- If you chose Chef-solo, odds are good you wound up re-implementing at least half the server anyway, and this weird competition developed because people thought that Chef the company wanted you to buy servers
- Changing your mind later was hard!
- Over time, Enterprise Chef got all the fancy new hotness, and OSS Chef went for months without a release.



Enterprise Chef Features Are Now Free

- Multi-tenancy (“Organizations”)
 - Like having multiple chef servers on your Chef Server
- Role-based access controls
 - ACLs on every single item in the Chef Server datastore
- Push Jobs
 - Once the client is in place, hit the /jobs API and kick off on-demand scripts



The Big Deal With Chef Server 12

- chef-server-ctl now manages users, permissions, & organizations
- SOLR4 under the hood for much faster search indexing (typically < 1s)
- Chef Identity for authenticating external apps such as Supermarket & Analytics
- 25% reduction in running services, 20% reduction in package size
- Full Release Notes:
 - https://docs.chef.io/server/release_notes.html



The Big Deal With Chef Client 12

- SSL Key Verification by default
- AIX Support
- Homebrew is default package manager for OSX
- Easier encrypted data bags in recipes with `data_bag_item()`
- Client-side key generation
- Windows DSC Support
- “Partial Search” integrated into main client, no need for cookbook
- Full Release Notes:
 - https://docs.chef.io/server/release_notes.html

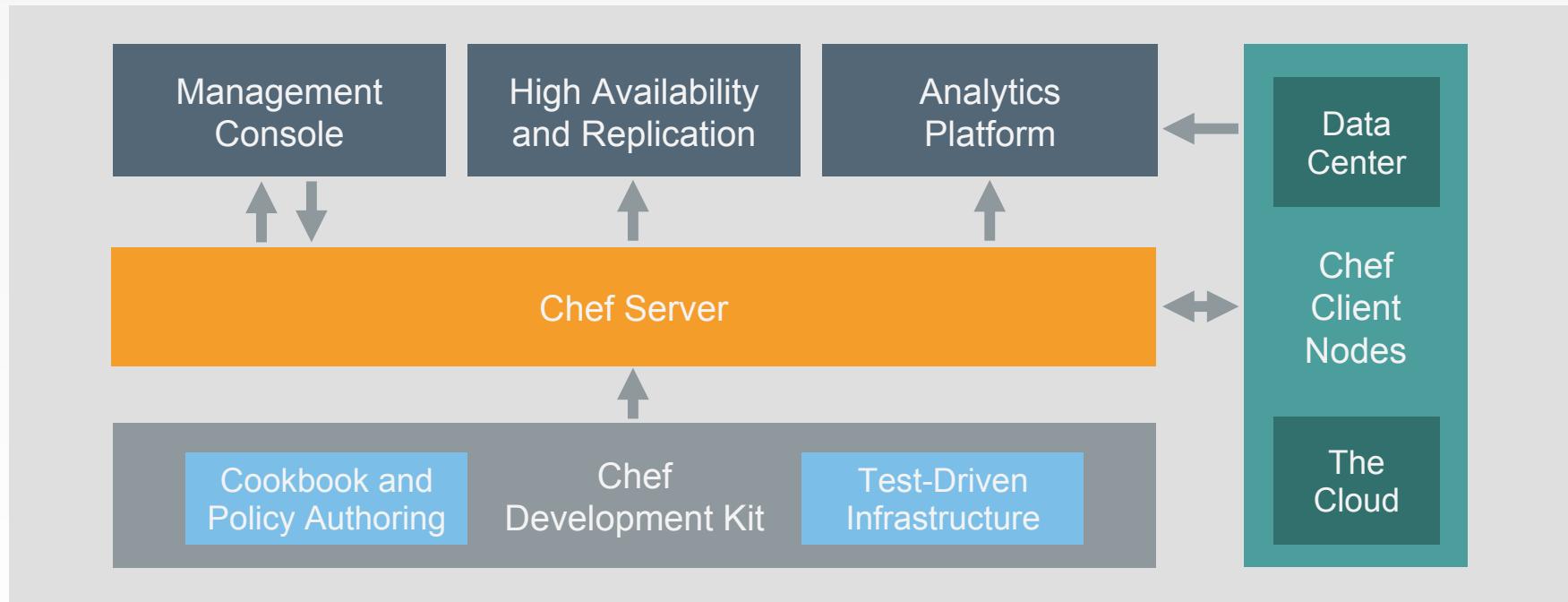


Chef 12 Delivers

	Single Code Base	<ul style="list-style-type: none">Converges open-source and premium features using a unified code baseSupports open-source innovation and delivers that value to commercial users.	<ul style="list-style-type: none">✓ Open source community benefits from timely bug fixes and improvements.✓ Commercial customers get enterprise features and commercial support for open-source tools.
	Enterprise Capabilities	<ul style="list-style-type: none">AnalyticsHigh availability and replicationManagement consoleChef development kitCommercial support for test-driven infrastructure	<ul style="list-style-type: none">✓ Visibility✓ Uptime✓ Consistency✓ Ease of use✓ Reliability
  	Ecosystem Integration	<ul style="list-style-type: none">Windows PowerShell DSCVMware vCloud AirAWS Cloud	<ul style="list-style-type: none">✓ Native integration with DSC and Azure.✓ Easily automate vSphere and vCloud workloads.✓ Automate AWS and use AWS Storage for replication.



The Chef Software Platform



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

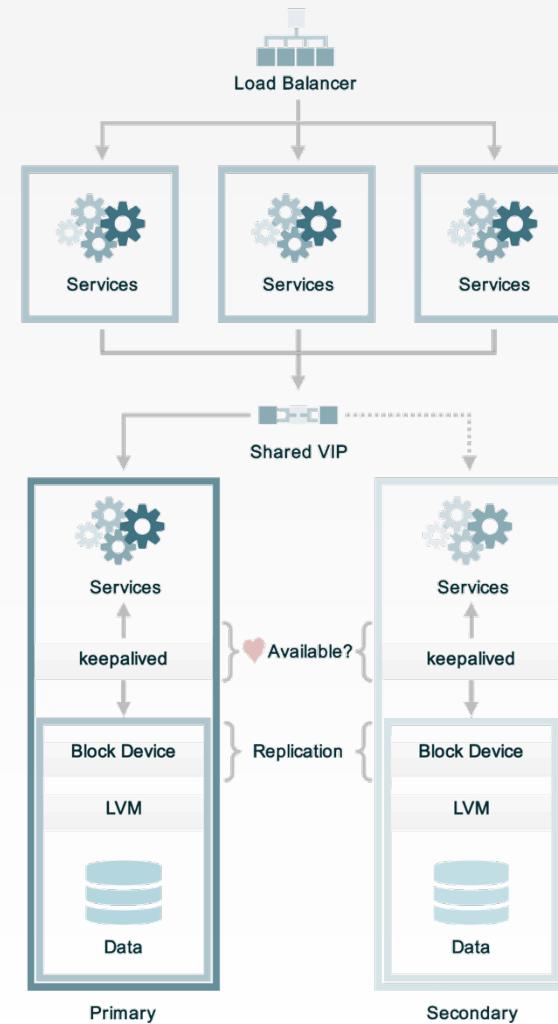
- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



High Availability

Maximum uptime

- Maximum system availability
- Works in data center and cloud environments
- Chef continues to run without interruption even if primary node fails
- Minimizes the chance of build-related delays in your delivery pipeline



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

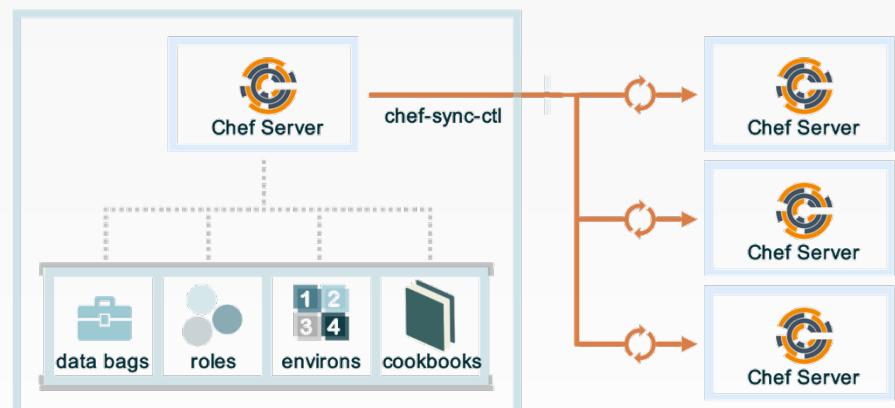
- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Replication

Consistency Across Global Systems

- Easily manage globally distributed infrastructure from single view
- Chef servers in different clusters / data centers / regions asynchronously replicate from a Master Chef server
- Infrastructure is consistent across all locations



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Chef Client and Chef Server Compatibility Matrix

Client Versions	Server Versions		
	Enterprise Chef > 11.1.8	Open Source > 11.1.0	Chef Server 12
11.x	✓	✓	✓
12.x	✓	✓	✓

- No longer supported as of Chef Client 12: Chef Client 0.10.x, 10.x
- No longer supported as of Chef Server 12: Private Chef 1.4.x



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



New Platform: IBM AIX

- Chef Client 12 supports IBM AIX 6.1 and 7.1
- Freely-available BFF installer
- Feature highlights:
 - Manage both SRC and init-system-based services
 - Install both BFF and RPM packages
 - Mount JFS2 disks and NFS shares and manage fstab
 - Native user and group management
- Compiled with IBM XL C/C++ for binary compatibility with future AIX versions



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Native Partial Search

```
search(:node,
  "role:web",
  :filter_result => {
    'name' => [ 'name' ],
    'ip' => [ 'ipaddress' ]
  }
).each do |result|
  Chef::Log.debug("I found a #{result['ip']} & #{result['name']}")
end
```



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Handling Platform-specific values

```
apache_package = "httpd"

case node["platform"]
when "ubuntu"
  apache_package = "apache2"
when "centos"
  apache_package = "httpd"
end

package apache_package do
  action :install
end
```



value_for_platform

```
apache_package = value_for_platform(
  "ubuntu" => "apache2",
  "centos" => "httpd"
)
```

```
package apache_package do
  action :install
end
```



value_for_platform accepts `>=`, `<=`, etc.

```
mysql_version = value_for_platform(
  ["centos", "redhat", "oracle"] => {
    '<= 5.0' => '4.3.21',
    '6.0'      => '5.5.23',
    '>= 7.0'    => '6.3.20'
  }
)
```



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Encrypted Databags pre 12.x

```
# load encrypted data bag item
Chef::EncryptedDataBagItem.load("passwords", "mysql")  
  
# load normal data bag item
data_bag_item("users", "mysql")
```



Automatic Encrypted Data Bag Support

```
# Provide a secret
data_bag_item('passwords', 'private', '/etc/
some/secret.key')

# Auto-detect encrypted item and use default
secret
data_bag_item('passwords', 'private')
```



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



shell_out and shell_out! part of DSL

```
execute 'make Dan Walsh happy' do
  command 'setenforce Enforcing'
  only_if { shell_out('getenforce').stdout =~ "Disabled" }
end
```

- Use `shell_out!` to raise exceptions on fatal errors
- Use `shell_out_with_systems_locale` to escape UTF-8 normalization
- **Never use `Process.spawn`, `backticks`, `popen4`, etc. to run shell commands!**



CYOA

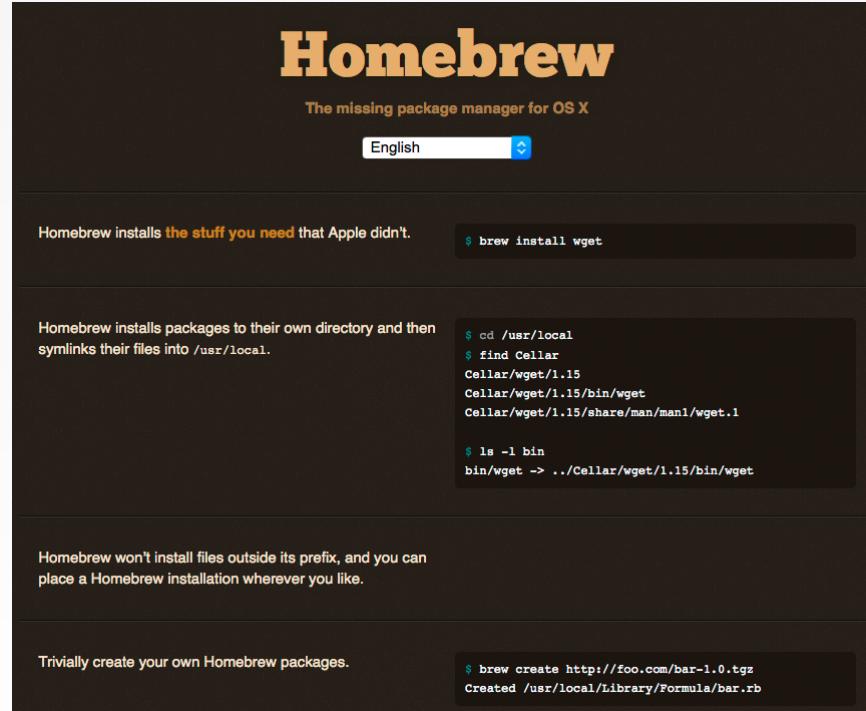
- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Homebrew Provider is Default

- Chef Client 11 and below use MacPorts as default package provider
- Chef Client 12 uses Homebrew



User Password Management Improvements

- dscl user provider supports OS X 10.7, 10.8+
 - 10.7: Password scheme is SALTED-SHA512
 - 10.8+: Password scheme is SALTED-SHA512-PBKDF2
- Pass 'salt' and 'iterations' attributes to 10.8+
- Examples and more info:
 - http://docs.chef.io/release/12-0/release_notes.html#mac-os-x-passwords



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Client-Side Key Generation

- Chef Client <= 11: Generate all keys server-side
 - Client-side key generation is possible with config option, but not default
- Chef Client 12: Clients generate their own public/private keys and send them to the server



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Minimum Ruby Versions

- Minimum Ruby is **2.0.0**
- Chef's full-stack installers bundle:
 - Ruby 2.1.5+ on UNIX/Linux
 - Ruby 2.0.0+ on Windows
- More information: <https://github.com/opscode/chef-rfc/blob/master/rfc034-ruby-193-eol.md>



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)

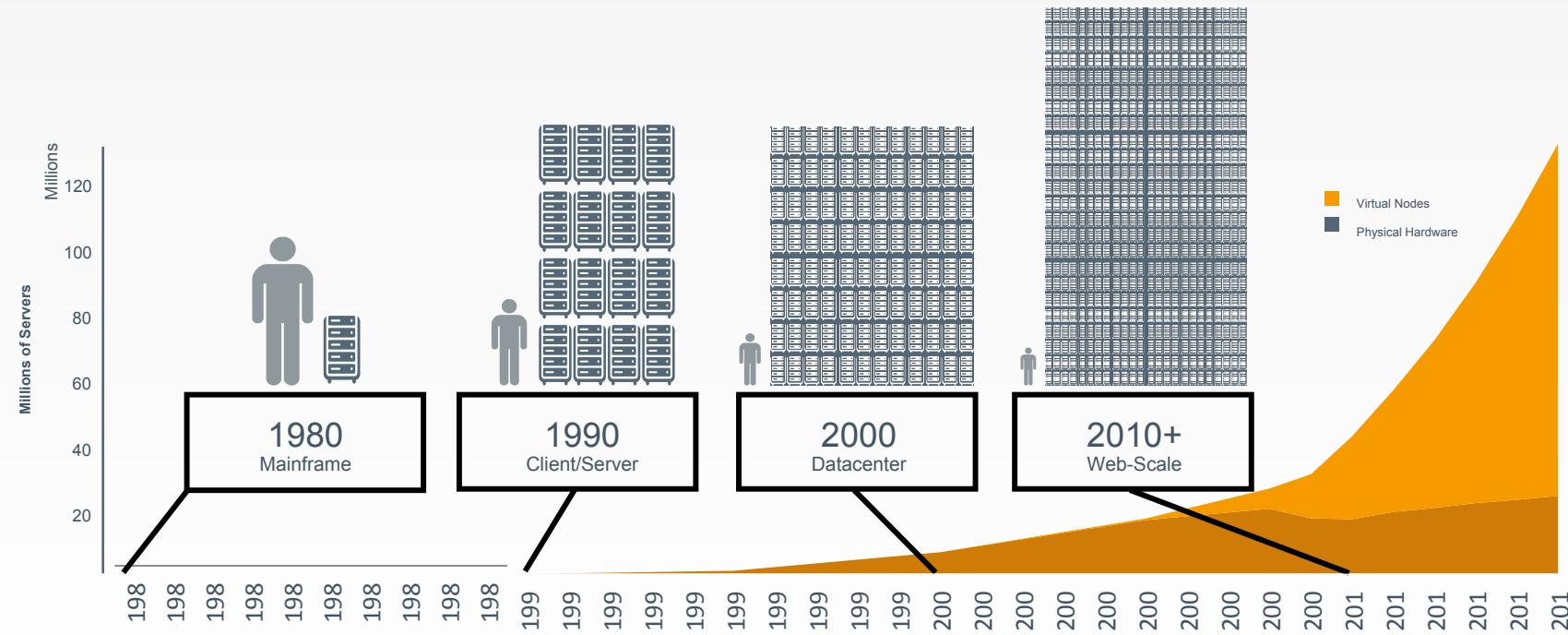




Chef Analytics



Computing Resources Have Been Increasing Exponentially



Increase Size Leads to Operational Complexity



To Operate at Velocity, Teams Need Visibility





Chef Analytics keeps
your finger on the pulse
of your infrastructure.



Chef Analytics Provides Visibility in Three Ways

TRACK CHANGES FROM ALL SOURCES

Chef Server Policy (Actions)

Chef Client Runs (Run History)

Node state (Audit Controls)

REAL-TIME VISIBILITY

Simple dashboard with search, filters and sorting options

Integrate with existing tools via API



NOTIFICATIONS

Built-in messaging and email integration

Extend notifications to your existing systems



Chef Analytics History



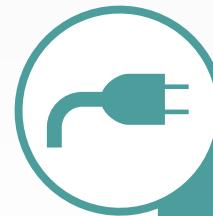
Launched in May 2014

Builds on reporting (run history) feature shipped in 2013



Adds new fundamental data collection components

Actions – track policy modifications on the Chef Server
Controls – assert the state of your infrastructure



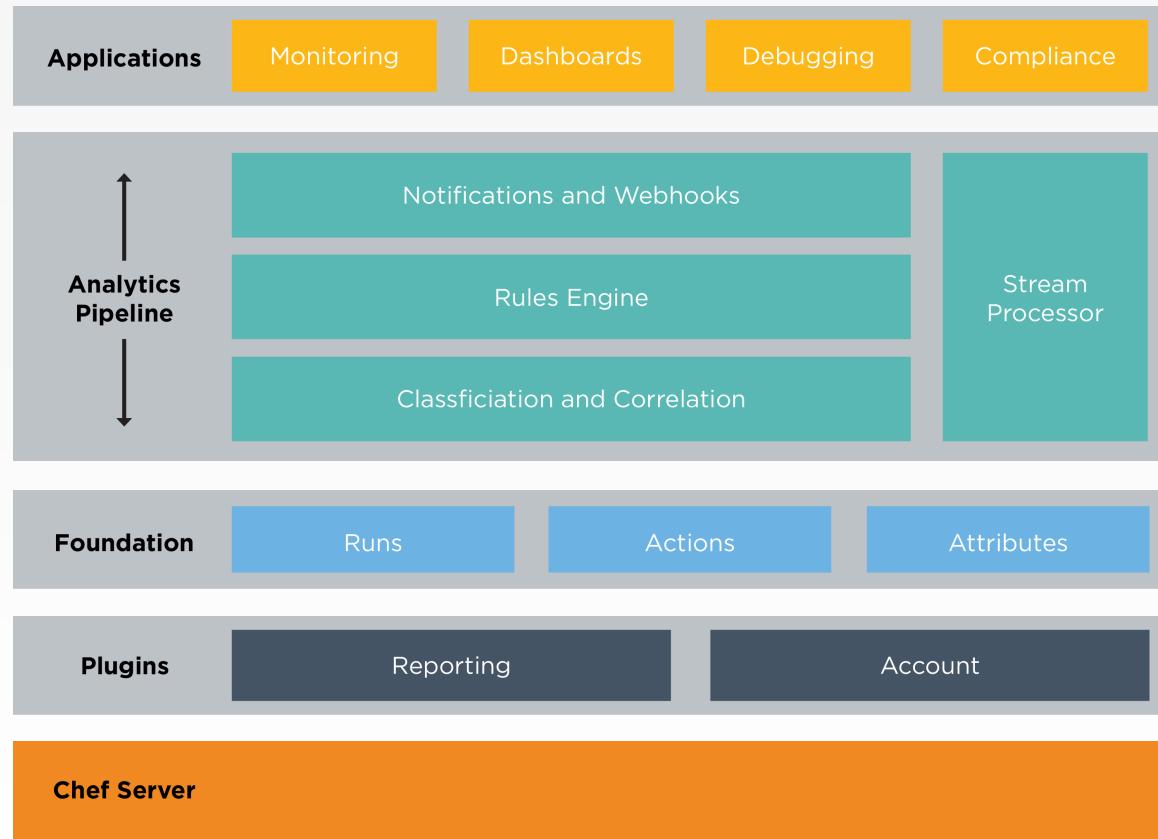
Pluggable analytics pipeline

Rules and Notifications



Delivered as premium feature of Chef 12

Chef Analytics Architecture



Chef Analytics Data Flow



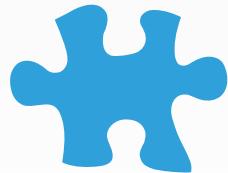
Actions – Who did what and when?



Provide a read-only view of what happened



Allow administrators to react to events as they happen or after the fact investigation



Can be customized to meet audit and compliance reporting requirements

Reporting – What changed during Chef runs ?

Single view of what changed



- Success/failure status of individual Chef Client runs
- Rollups of success and failure counts
- Rollups of run durations over time



Drill-down detail to individual resource convergence

- State before and after convergence
- Resource Diffs (e.g. for templates, files)
- Error messages and stack traces

Audit Controls



Continually assess the baseline of your system

Create IT Controls on the state of a node

Package installations and version information, file contents, running services, open ports, registry keys, users and groups



Multiple uses

Audit and Compliance

Unmanaged fleet migration



Separation of Controls

Allows for separation of controls via integration with Chef role based access controls

Notifications



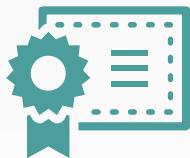
Adds a language which allows you to express rules on all Chef events

Runs

Resource convergence

Actions

Controls



Extensible for your business requirements

Build custom rules that match your infrastructure and team

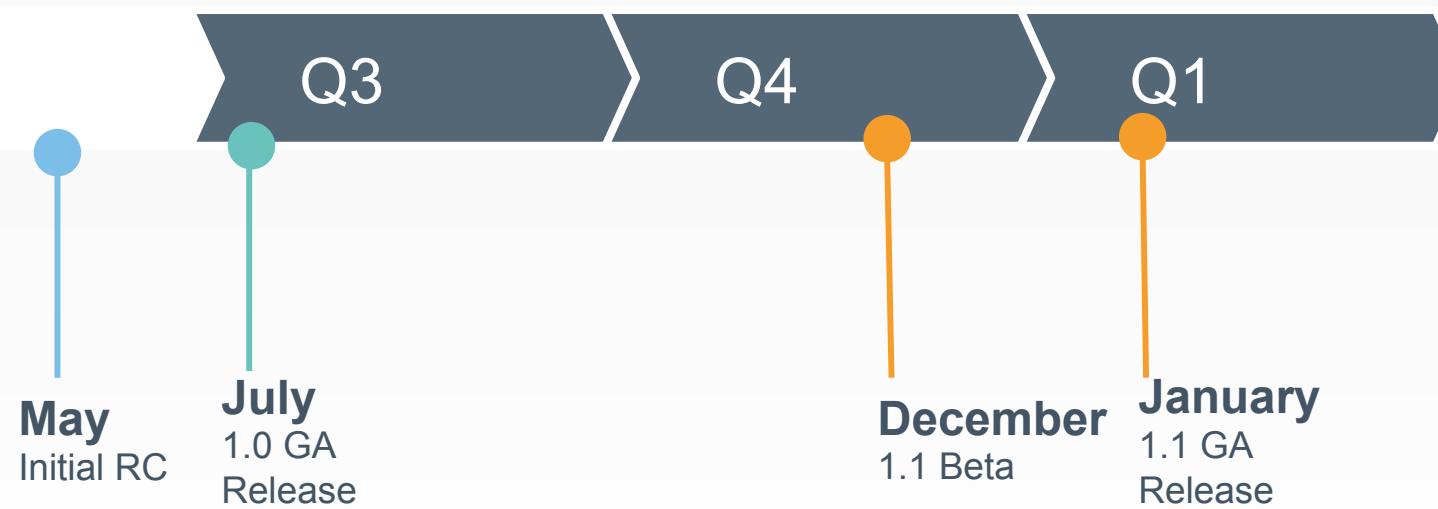


Send to different external data sinks

Messaging systems, e.g. HipChat, SMTP

Extensible to anything that supports webhook notifications

2014 Analytics Roadmap



What's next?

Integrate with more Chef data sources



Chef Solo

Push Jobs

Chef Provisioning

Add more data integrations



Splunk, AWS analytics stack, PagerDuty, ...

Enhanced Compliance tools



Enhanced audit and compliance reporting

Content for auditing against specific compliance regimes (PCI, SOX, ...)

How You Can Get Analytics

Available as a Premium Feature of Chef Server

Supported on Enterprise Chef 11.2 and Chef Server 12

Included in Chef Subscription

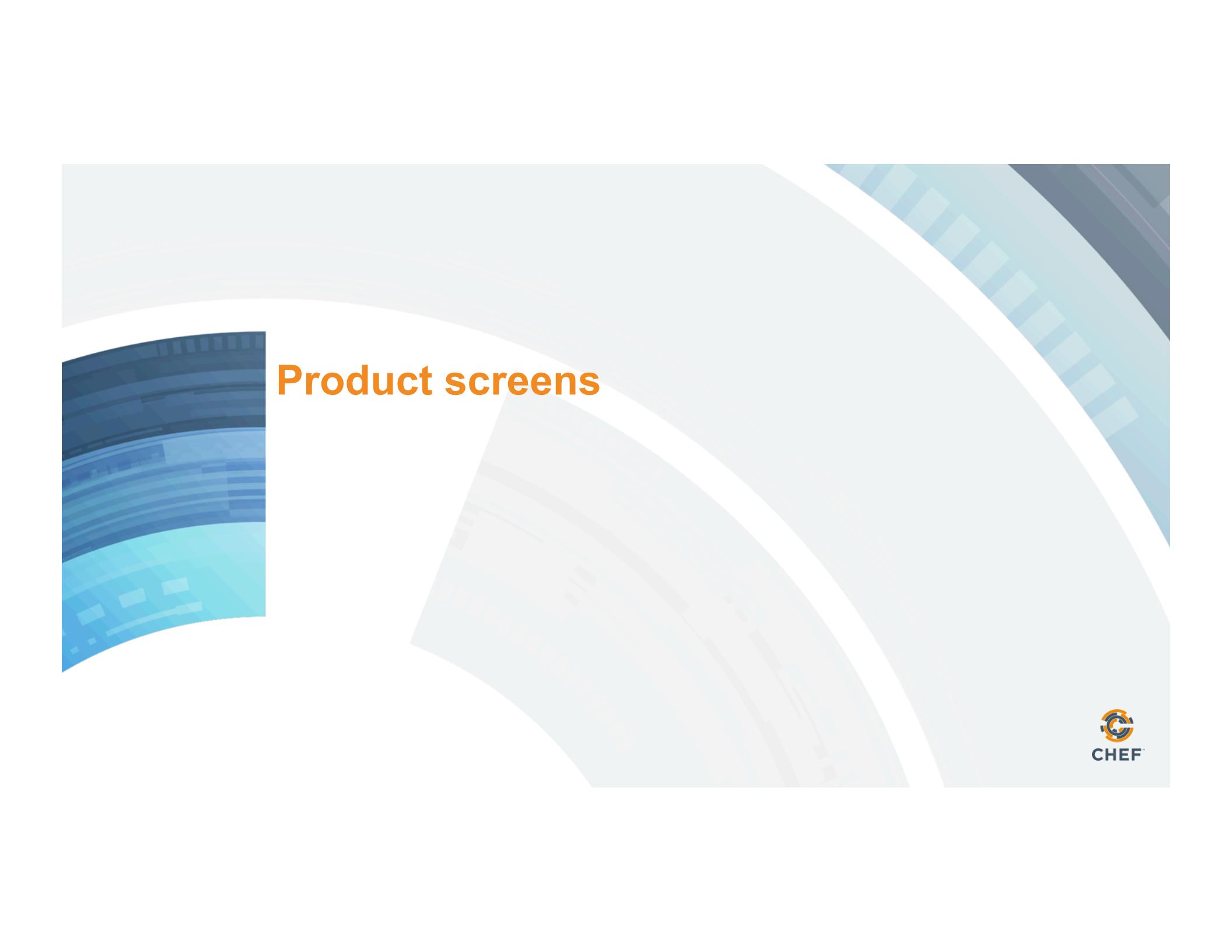
Also available free for installations less than 25 nodes

Installation instructions

https://docs.getchef.com/install_analytics.html

Download from <https://downloads.getchef.com/>





Product screens



Dashboard

Run History

Showing chef-client runs for 07/24/2014 - 07/25/2014 ▾

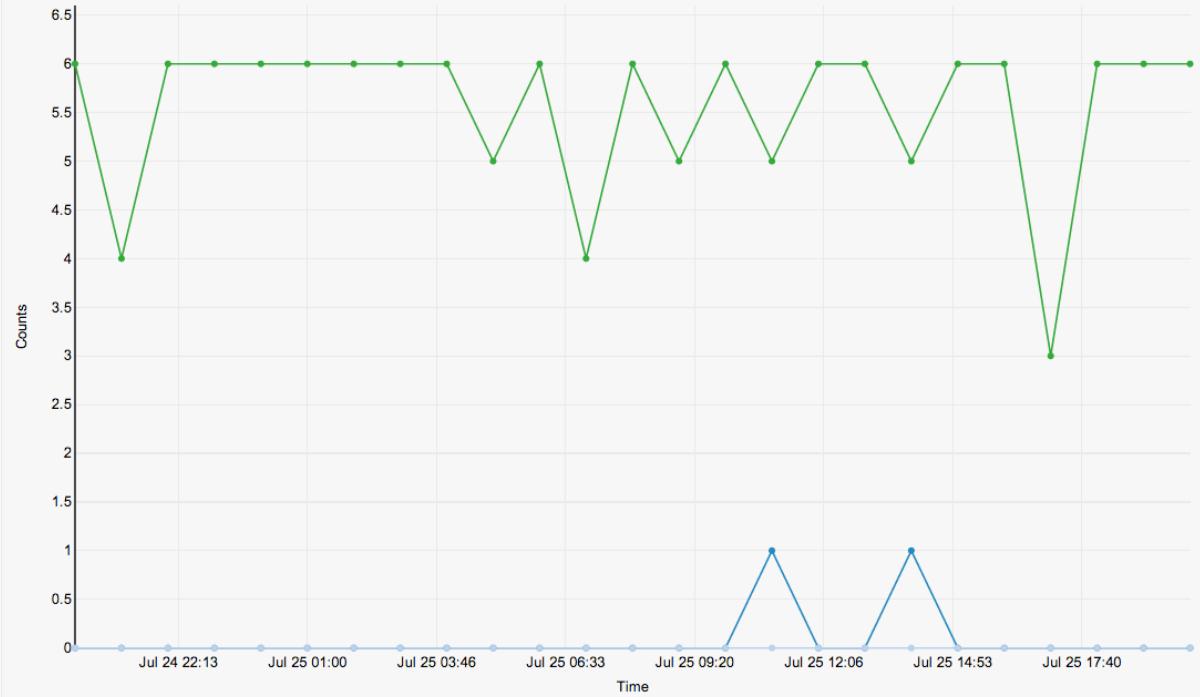
Runs Summary



Run Durations



Run Counts

■ success ■ failure ■ aborted

CHEF MANAGE

Nodes Reports Policy Administration

Feedback | Organization Signed in as James Casey 0 0

Dashboard Showing chef-client runs for 07/24/2014 - 07/25/2014 ▾

Run History Filter by environment All Environments Filter by status All

Start Time	Finish Time	Node	Duration	Resources Executed / Total	Status
2014-07-25 12:14:19	2014-07-25 12:15:07		a minute	11 / 74	success
2014-07-25 12:13:05	2014-07-25 12:13:54		a minute	11 / 74	success
2014-07-25 11:48:29	2014-07-25 11:48:35		a few seconds	0 / 2	failure
2014-07-25 11:42:47	2014-07-25 11:43:36		a minute	11 / 74	success

Run ID: 6aa8bbf0-8032-4c34-b949-f6a838057890

Details Run List Error Log

#<Net::HTTPFatalError: 503 "Service Temporarily Unavailable"> had an error:
503 "Service Temporarily Unavailable"

Stacktrace

```
C:/opscode/chef/embedded/lib/ruby/1.9.1/net/http.rb:2633:in `error!'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/http.rb:143:in `request'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/rest.rb:115:in `get'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/data_bag_item.rb:149:in `load'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/encrypted_data_bag_item.rb:122:in `load'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-sugar-2.0.0/lib/chef/sugar/data_bag.rb:57:in `encrypted_data_bag_item'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-sugar-2.0.0/lib/chef/sugar/data_bag.rb:84:in `encrypted_data_bag_item_for_environment'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-sugar-2.0.0/lib/chef/sugar/data_bag.rb:104:in `encrypted_data_bag_item_for_environment'
C:/chef/cache/cookbooks/opscode-ci/recipes/_github.rb:28:in `from_file'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/mixin/from_file.rb:30:in `instance_eval'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/mixin/from_file.rb:30:in `from_file'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/cookbook_version.rb:237:in `load_recipe'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/run_context.rb:165:in `load_recipe'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/run_context.rb:134:in `block in include_recipe'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/run_context.rb:133:in `each'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/run_context.rb:133:in `include_recipe'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/dsl/include_recipe.rb:26:in `include_recipe'
C:/chef/cache/cookbooks/opscode-ci/recipes/_build_support.rb:35:in `from_file'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/mixin/from_file.rb:30:in `instance_eval'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/mixin/from_file.rb:30:in `from_file'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/cookbook_version.rb:237:in `load_recipe'
C:/opscode/chef/embedded/lib/ruby/gems/1.9.1/gems/chef-11.12.8-x86-mingw32/lib/chef/run_context.rb:165:in `load_recipe'
```



Carworld Overview

Actions

Actions are modifications to the policy (roles, cookbooks, environments, etc...) on your Chef server. Browse a timeline view of historical actions for your organization and filter and search for specific actions.

Rules

Create rules on actions that match when specific actions occur. You can send notifications, create audit events or log messages when actions occur.

Notifications

Define notification channels to use in Rules to alert your services and/or team members of any changes to infrastructure compliance. Supported notification channels are HipChat and a HTTP-based webhook mechanism for integration into arbitrary systems.

Audits (Beta)

Audits events are triggered via rules which may match any event in the Chef analytics system. You can view past audits and search by category and level.

Help



Search actions

 Search Default ▾

From the beginning of time

 up to now Export ▾

Showing 1 - 25 of 147

 Save ▾Client web updated Node - web - 4 days ago Manage run_id:9a918fa6-3be7-496d-b8b0-e2aec52e6e01User james updated Node - db - 4 days ago Manage run_id:96fe1941-58e1-4e1b-ac2e-036ea322d1dcUser james updated Node - app - 4 days ago Manage run_id:22f7a0ac-d632-4a3a-8e52-d8f2d791fc88User james updated Node - web - 4 days ago Manage run_id:09070399-3b39-4800-a9a3-324f5d14cb88User james updated Node - db - 4 days ago Manage run_id:0347e2e7-f5cc-4add-8866-2e7c898501f4Client db updated Node - db - 4 days ago Manage



Search actions

[Search](#)[\[x Entity Type\]](#) [\[x Entity Name\]](#)[+](#)[Default](#) [From](#) *the beginning of time* [up to](#) *now*[↻](#)[Export](#) Showing 1 - 3 of 3[Save](#)Client carworld-validator created [Client - db](#) - 4 days ago[Manage](#)[run_id:251a9853-2c72-4eae-a33f-3d53c5cd4238](#)

Details — 3:08 pm - 4 December 2014

Remote Hostname: 73.37.70.73

Remote Request ID: 251a9853-2c72-4eae-a33f-3d53c5cd4238

Service Hostname: manage-demo.opscode.us

Service Request ID: g3IAA2QAEGVY2h1ZkAxMjcuMC4wLjEDAAIkVwAACUAAAA

User james deleted [Client - db](#) - 4 days ago[Manage](#)Client carworld-validator created [Client - db](#) - 6 days ago[Manage](#)



Rules

Active	Priority	Name	Updated By	Updated At	
<input checked="" type="checkbox"/>	0	unknown cookbook upload	james	12/4/14 3:34 PM	
<input type="checkbox"/>	0	controls audit	james	12/4/14 2:55 PM	
<input checked="" type="checkbox"/>	0	audit runs	james	12/4/14 3:34 PM	
<input checked="" type="checkbox"/>	0	Tagging	james	12/4/14 2:56 PM	
<input checked="" type="checkbox"/>	0	Node deletion	james	12/4/14 2:55 PM	
<input checked="" type="checkbox"/>	0	Lifecycle Webhook	james	12/4/14 2:46 PM	

**Rule Group:** [unknown cookbook upload](#)*Last Updated 4 days ago by james*[Active](#) Priority: **0**

```
rules 'unknown cookbook upload'
  rule on action
  when
    entity_type = 'version' and requestor_name != 'jenkins'
  then
    audit:error("Non Jenkins user uploading cookbook")
  end
end
```

[Cancel](#) [Delete](#)[Save](#)



Notifications

Name	Type	Updated By	Updated At	X
local_webhook	Http	james	12/2/14 11:52 AM	



Http Notification: [Io](#)

Webhooks allow external services to be notified when certain events happen in Chef Analytics. When the specified events happen, we'll send a POST request to each of the URLs you provide.

URL to send message to*



Authenticator Type [None](#)

Username

Password

[Cancel](#)[Delete](#)[Save](#)

[Filter by level](#)[Filter by type](#)[Default](#)From *the beginning of time*

up to

now

Showing 1 - 25 of 200

 warn	Audit was triggered by run converge 663dc2ec-dae6-4d9c-a58b-5767a2d44e65 - 4 days ago Run list: recipe[apt], recipe[controls-test]	Audit Description: Run Converged 454929d0-4a9d-4596-bc58-e35a283bcadf
 warn	Audit was triggered by run start 454929d0-4a9d-4596-bc58-e35a283bcadf - 4 days ago db	Audit Description: Run Started 454929d0-4a9d-4596-bc58-e35a283bcadf
 warn	Audit was triggered by run converge 9e490ce3-fe55-4f6c-82cd-995551deebf8 - 4 days ago Run list: recipe[apt]	Audit Description: Run Converged 9a918fa6-3be7-496d-b8b0-e2aec52e6e01
 warn	Audit was triggered by run start 9a918fa6-3be7-496d-b8b0-e2aec52e6e01 - 4 days ago web	Audit Description: Run Started 9a918fa6-3be7-496d-b8b0-e2aec52e6e01
 info	Audit was triggered by action db - 4 days ago Client - db	Audit Description: Action delete on client db
 info	Audit was triggered by action db - 4 days ago Node - db	Audit Description: Action delete on node db
 info	Audit was triggered by action web - 4 days ago	Audit Description: Action delete on client web



		Filter by level	Filter by type
<input checked="" type="checkbox"/> Default		From <input type="text" value="the beginning of time"/>	<input type="checkbox"/> up to <input type="text" value="now"/>
Showing 1 - 25 of 200			
<input checked="" type="checkbox"/> warn	Audit was triggered by run converge 663dc2ec-dae6-4d9c-a58b-5767a2d44e65 - 4 days ago	Audit Description: Run Converged 454929d0-4a9d-4596-bc58-e35a283bcadf	
Run list: recipe[apt], recipe[controls-test]			
Run ID:	454929d0-4a9d-4596-bc58-e35a283bcadf	Total Resource Count:	0
Node Name:	db	Updated Resource Count:	0
Start Time:	3:39 pm - 4 December 2014	Run List:	recipe[apt], recipe[controls-test]
End Time:	3:40 pm - 4 December 2014	Tags:	run:454929d0-4a9d-4596-bc58-e35a283bcadf
<input checked="" type="checkbox"/> warn	Audit was triggered by run start 454929d0-4a9d-4596-bc58-e35a283bcadf - 4 days ago	Audit Description: Run Started 454929d0-4a9d-4596-bc58-e35a283bcadf	
db			
Run ID:	454929d0-4a9d-4596-bc58-e35a283bcadf	Start Time:	3:39 pm - 4 December 2014
Node Name:	db	Tags:	run:454929d0-4a9d-4596-bc58-e35a283bcadf
<input checked="" type="checkbox"/> warn	Audit was triggered by run converge 9e490ce3-fe55-4f6c-82cd-995551deebf8 - 4 days ago	Audit Description: Run Converged 9a918fa6-3be7-496d-b8b0-e2aec52e6e01	
Run list: recipe[apt]			
Run ID:	9a918fa6-3be7-496d-b8b0-e2aec52e6e01	Total Resource Count:	9
Node Name:	web	Updated Resource Count:	5
Start Time:	3:35 pm - 4 December 2014	Run List:	recipe[apt]
End Time:	3:38 pm - 4 December 2014	Tags:	run:9a918fa6-3be7-496d-b8b0-e2aec52e6e01
<input checked="" type="checkbox"/> warn	Audit was triggered by run start 9a918fa6-3be7-496d-b8b0-e2aec52e6e01 - 4 days ago	Audit Description: Run Started 9a918fa6-3be7-496d-b8b0-e2aec52e6e01	
web			

Example Rules



Controls - PCI Compliance

- PCI 2.3 - Encrypt all non-console administrative access such as browser/Web-based management tools.

Control

```
controls 'port compliance' do
  control port(23) do
    it "has nothing listening"
    expect(port(23)).to_not
    be_listening
  end
end
end
```

Rule

```
rules 'PCI 2.3 – Confirm telnet port not available'
rule on run_control
when
  name = 'should be listening' and
  resource_type = 'port' and
  resource_name = '23' and
  status != 'success'
then
  audit:error("PCI 2.3 – Encrypt all non-console
  administrative access such as browser/Web-based
  management tools.")
  notify("security-team@financialcorp.com")
end
end
```



Controls - SOX Compliance

- SOX Section 302.4.B – Establish verifiable controls to track data access.

Control

```
controls 'password authentication' do
  control file('/etc/ssh/sshd_config') do
    it "is disabled"
    expect(file('/etc/ssh/sshd_config'))
      .to match(/PasswordAuthentication\s+no/)
  end
end
end
```

Rule

```
rules 'SOX Section 302.4.B – force key based auth'
  rule on run control
  when
    name = 'is disabled' and
    resource_type = 'File' and
    resource_name = '/etc/ssh/sshd_config' and
    status = 'failed'
  then
    audit:error("SOX Section 302.4.B – Establish verifiable
  controls to track data access.")
    notify('security-team')
  end
end
```



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



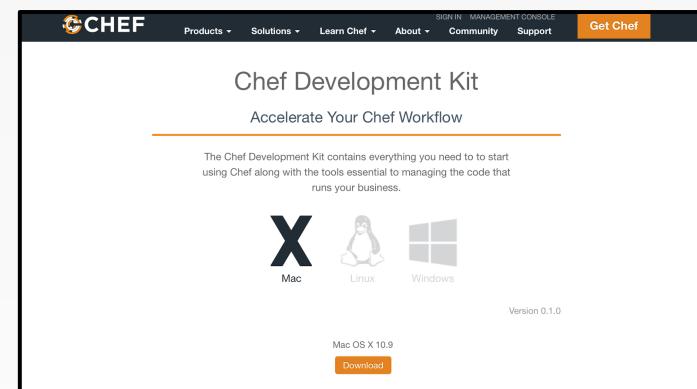


Chef Development Kit



Increase CHEF adoption through ChefDK

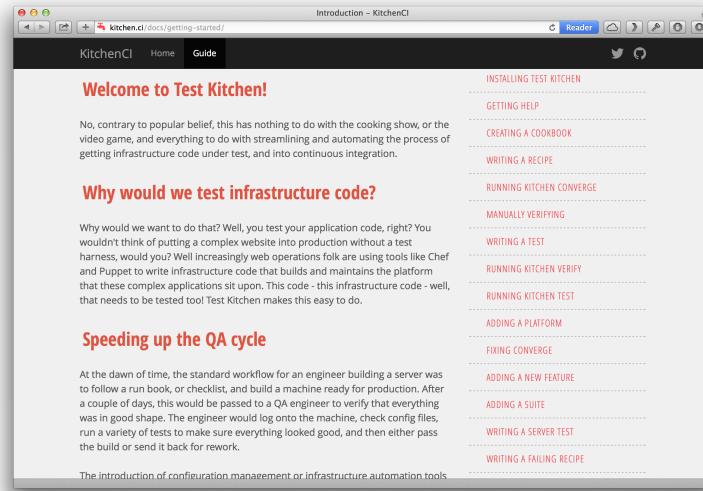
- **ChefDK:** CHEF Software Development Kit, fully supported with the Chef Premium Subscription
- **Workflow Definition:** Our recommendation on the process to test and verify your infrastructure code before committing it to source control and shipping it to production.
- **Workflow Enhancement:** Based on customer feedback and use cases



ChefDK: In the Box

First Class Support on Windows, Linux, and OSX for the entire suite of Chef development tools

- **Test Kitchen:** Virtualized testing harness
- **Berkshelf:** Dependency solver
- **Chef-Vault:** Secrets management
- **Rubocop / Foodcritic:** Code linting
- **Chefspec:** In-memory Unit Testing
- **Chef.bin:** New wrapper binary to tie it all together, with new extensible cookbook generators.



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)





Chef Provisioning



Chef Lifecycle

1. Provision a server, virtual machine, or cloud instance
2. Install Chef
3. Configure Chef
4. Run Chef
5. GOTO step 4



knife bootstrap

1. Provision a server, virtual machine, or cloud instance
2. Install Chef
3. Configure Chef
4. Run Chef
5. GOTO step 4



Provisioning in AWS

- Let's use AWS as our case study
- We'll look at the various ways Chef can help you provision instances there



```
$ knife ec2 server create -I ami-4ab46b3d -f t1.micro -g sg-884f2eed,sg-8b4f2eee --ssh-user ubuntu -N nathen_hw_knife -r "recipe[hello_world]"
```

```
ec2-54-77-67-44.eu-west-1.compute.amazonaws.com Chef Client finished, 85/149 resources updated in 38.998005693 seconds

Instance ID: i-80d76866
Flavor: t1.micro
Image: ami-4ab46b3d
Region: eu-west-1
Availability Zone: eu-west-1a
Security Groups: default
Security Group Ids: sg-884f2eed, sg-8b4f2eee
Tags: Name: nathen_hw_knife
SSH Key: chef-nathenharvey-eu
Root Device Type: ebs
Root Volume ID: vol-602e1f7c
Root Device Name: /dev/sda1
Root Device Delete on Terminate: true
Public DNS Name: ec2-54-77-67-44.eu-west-1.compute.amazonaws.com
Public IP Address: 54.77.67.44
Private DNS Name: ip-172-31-2-160.eu-west-1.compute.internal
Private IP Address: 172.31.2.160
Environment: _default
Run List: recipe[hello_world]
```



knife ec2

- “Infrastructure as Command Line” instead of “Infrastructure as Code”
- What if I want many instances?
- What if I want multiple tiers of instances?



© r.n.
CHEF

<https://flic.kr/p/eycPj7>

The classic operations playbook

```
sleep 90 && knife ssh "role:opscode-authz OR role:opscode-chef OR role:opscode-solr OR \
role:monitoring*" "sudo chef-client"
```

rabbitmq (omnibus)

```
## knife Chef 0.10 syntax
knife ec2 server create -G rs-preprod,platform-server --flavor m1.small -I ami-1136fb78 -x ubuntu \
-E rs-preprod \
-r 'role[rabbitmq],role[rs-preprod]' -i ~/.ssh/our_ssh_key
```

Create and attach an elastic IP, or find the old rabbitmq elastic IP and attach it.
Rerun chef-client
update environment data bag item "rabbitmq_ip" with assigned elastic IP, upload data bag with knife.
run chef-client on all the servers that are rabbitmq producers or consumers:

```
sleep 90 && knife ssh "role:opscode-account OR role:opscode-chef OR role:opscode-solr OR \
role:opscode-job-worker OR role:monitoring*" "sudo chef-client"
```

opscode-solr

```
## knife Chef 0.10 syntax
knife ec2 server create -G platform-server,rs-preprod \
--flavor cl.medium -I ami-1136fb78 -x ubuntu \
-E rs-preprod \
-r 'role[opscode-solr],role[rs-preprod]' \
-Z us-east-1c \
-i ~/.ssh/our_ssh_key
```

Attach an elastic IP.
Rerun chef-client
update environment data bag item "solr_ip" with assigned elastic IP, upload data bag with knife.
then add it into rotation after waiting for node to be indexed

```
sleep 90 && knife ssh "role:opscode-lb-int OR role:monitoring*" "sudo chef-client"
```

opscode-authz (omnibus)

```
knife ec2 server create -G rs-preprod,platform-server --flavor cl.medium -I ami-1136fb78 -x ubuntu \
-E opscode-authz \
-r 'role[opscode-authz]' -i /etc/opscode/authz.pem
```



Spiceweasel

- Command-line tool for batch loading Chef infrastructure
- Generate and executes knife commands

```
nodes:  
- serverA:  
  run_list: role[base]  
  options: -i ~/.ssh/mray.pem -x user --sudo  
- serverB serverC:  
  run_list: role[base]  
  options: -i ~/.ssh/mray.pem -x user --sudo -E production  
- rackspace 3:  
  run_list: recipe:mysql,role[monitoring]  
  options: --image 49 --flavor 2 -N db{n}  
- windows_winrm winboxA:  
  run_list: role[base],role[iisserver]  
  options: -x Administrator -P 'super_secret_password'  
- windows_ssh winboxB winboxC:  
  run_list: role[base],role[iisserver]  
  options: -x Administrator -P 'super_secret_password'
```



<https://github.com/mattray/spiceweasel>

Chef Provisioning

- Allows creation of instances in Chef Recipes
- Allows for more programmatic creation
- Allows for multiple tiers to be created in one shot
- Moves more towards “Infrastructure as Code”



<https://flic.kr/p/knDPjc>

Chef Provisioning

```
machine 'web1' do
  recipe 'apache'
end
```



Chef Provisioning - Drivers

- AWS
- Azure
- Fog
- Vagrant
- Docker
- LXC
- Hanlon
- ...and more

```
machine 'web1' do  
  recipe 'apache'  
end
```



The classic operations playbook

```
sleep 90 && knife ssh "role:opscode-account OR role:opscode-chef OR role:opscode-solr OR \
role:opscode-job-worker OR role:monitoring" "sudo chef-client"
```

rabbitmq (omnibus)

```
## knife Chef 0.10 syntax
knife ec2 server create -G rs-preprod,platform-server --flavor m1.small -I ami-1136fb78 -x ubuntu \
-E rs-preprod \
-r 'role[rabbitmq],role[rs-preprod]' -i ~/.ssh/our_ssh_key
```

Create and attach an elastic IP, or find the old rabbitmq elastic IP and attach it.

Rerun chef-client

update environment data bag item "rabbitmq_ip" with assigned elastic IP, upload data bag with knife.

run chef-client on all the servers that are rabbitmq producers or consumers:

```
sleep 90 && knife ssh "role:opscode-account OR role:opscode-chef OR role:opscode-solr OR \
role:opscode-job-worker OR role:monitoring" "sudo chef-client"
```

opscode-solr

```
## knife Chef 0.10 syntax
knife ec2 server create -G platform-server,rs-preprod \
--flavor cl.medium -I ami-1136fb78 -x ubuntu \
-E rs-preprod \
-r 'role[opscode-solr],role[rs-preprod]' \
-Z us-east-1c \
-i ~/.ssh/our_ssh_key
```

Attach an elastic IP.

Rerun chef-client

update environment data bag item "solr_ip" with assigned elastic IP, upload data bag with knife.

then add it into rotation after waiting for node to be indexed

```
sleep 90 && knife ssh "role:opscode-lb-int OR role:monitoring" "sudo chef-client"
```

opscode-authz (omnibus)

```
knife ec2 server create -G rs-preprod,platform-server --flavor cl.medium -I ami-1136fb78 -x ubuntu \
-E opscode-authz \
-r 'role[opscode-authz],role[rs-preprod]' \
-Z us-east-1c \
-i ~/.ssh/our_ssh_key
```



<https://flic.kr/p/eycPj7>

Chef Provisioning

```
require 'chef/provisioning'

machine_batch do
  machines %w(primary secondary web1 web2)
end

machine_batch do
  machine 'primary' do
    recipe 'initial_ha_setup'
  end
end

machine_batch do
  machine 'secondary' do
    recipe 'initial_ha_setup'
  end
end

machine_batch do
  %w(primary secondary).each do |name|
    machine name do
      recipe 'rest_of_my_configuration'
    end
  end
end
```



Configure Chef Provisioning

- Installed as part of Chef Development Kit
- Configure AWS
 - Create an AWS credentials file
- Gems
 - chef-provisioning
 - chef-provisioning-aws



AWS Credentials File



OPEN IN EDITOR: ~/.aws/config

```
[default]
Region=us-east-1
aws_access_key_id = <AWS_ACCESS_KEY_ID>
aws_secret_access_key = <AWS_SECRET_ACCESS_KEY>
```

SAVE FILE!



Simple Example

```
require 'chef/provisioning/  
aws_driver'  
with_driver 'aws'  
  
machine 'nathen_web1' do  
  recipe 'hello_world'  
end
```



Execute the Recipe

```
$ chef-client --local-mode simple.rb
```

```
[2015-02-04T12:53:22+00:00] INFO: Started chef-zero at http://localhost:8889 with repository at /Users/nathenharvey/chef_london_users/chef-repo
One version per cookbook

[2015-02-04T12:53:22+00:00] INFO: Forking chef instance to converge...
Starting Chef Client, version 12.0.3
[2015-02-04T12:53:22+00:00] INFO: *** Chef 12.0.3 ***
[2015-02-04T12:53:22+00:00] INFO: Chef-client pid: 23233
[2015-02-04T12:53:31+00:00] INFO: Run List is []
[2015-02-04T12:53:31+00:00] INFO: Run List expands to []
[2015-02-04T12:53:31+00:00] INFO: Starting Chef Run for nharveycul215
[2015-02-04T12:53:31+00:00] INFO: Running start handlers
[2015-02-04T12:53:31+00:00] INFO: Start handlers complete.
[2015-02-04T12:53:31+00:00] INFO: HTTP Request Returned 404 Not Found : Object not found: /reports/nodes/nharveycul215/runs
resolving cookbooks for run list: []
[2015-02-04T12:53:31+00:00] INFO: Loading cookbooks []
Synchronizing Cookbooks:
Compiling Cookbooks...
[2015-02-04T12:53:35+00:00] WARN: Node nharveycul215 has an empty run list.
Converging 7 resources
Recipe: @recipe_files:::/Users/nathenharvey/chef_london_users/chef-repo/web.rb
```



Test and Repair

Resources follow a test and repair model

```
machine 'nathen_web1'
```



Test and Repair

Resources follow a **test** and **repair** model

```
machine 'nathen_web1'
```

Test nathen_web1 exist?

Yes

Test and Repair

Resources follow a **test** and **repair** model

```
machine 'nathen_web1'
```

Test nathen_web1 exist?

Yes

Done

Test and Repair

Resources follow a **test** and **repair** model

machine 'nathen_web1'

Test nathen_web1 exist?

Yes

Done

No

Test and Repair

Resources follow a **test** and **repair** model

machine 'nathen_web1'

Test nathen_web1 exist?

Yes

Done

No

Create it



Test and Repair

Resources follow a **test** and **repair** model

machine 'nathen_web1'

Test nathen_web1 exist?

Yes

Done

Repair

No

Create it



Behold! The power of a loop

- Need multiple instances?
- That's easy!

```
num_webservers = 3
(0...num_webservers).each do |i|
  machine "nathen_web_0#{i}" do
    recipe "hello_world"
  end
end
```



More with AWS

- Security Groups
- Elastic Load Balancers
- VPCs
- Auto Scaling Groups
- SQS Queues
- ...and more
- <https://github.com/chef/chef-provisioning-aws>



Security Group

```
with_data_center 'us-east-1' do
  aws_security_group "nathen-provisioning-security-group" do
    inbound_rules [
      { :ports => 22, :protocol => :tcp, :sources => ["0.0.0.0/0"] },
      { :ports => 80, :protocol => :tcp, :sources => ["0.0.0.0/0"] }
    ]
  end
end
```



Security Groups

```
{  
  "id": "nathan-provisioning-security-group",  
  "security_group_id": "sg-1d452878"  
}
```

- Information is stored in data bag items



Load Security Group ID

```
security_group = data_bag_item('aws_security_groups',  
                           'nathan-provisioning-security-group')  
  
security_group_id = security_group['security_group_id']
```



Use the Security Group

```
with_machine_options :bootstrap_options => {
  :instance_type => 't1.micro',
  :security_group_ids => [security_group_id]
}
```



Elastic Load Balancer

```
load_balancer "nathen-elb" do
  load_balancer_options :availability_zones => ['us-east-1a', 'us-east-1c', 'us-east-1d', 'us-east-1e'],
                        :listeners => [
    {
      :port => 80,
      :protocol => :http,
      :instance_port => 80,
      :instance_protocol => :http,
    }
  ]
  machines machine_names
end
```





Provisioning Production Environments



Chef Provisioning – Provisioning Node



Chef Server



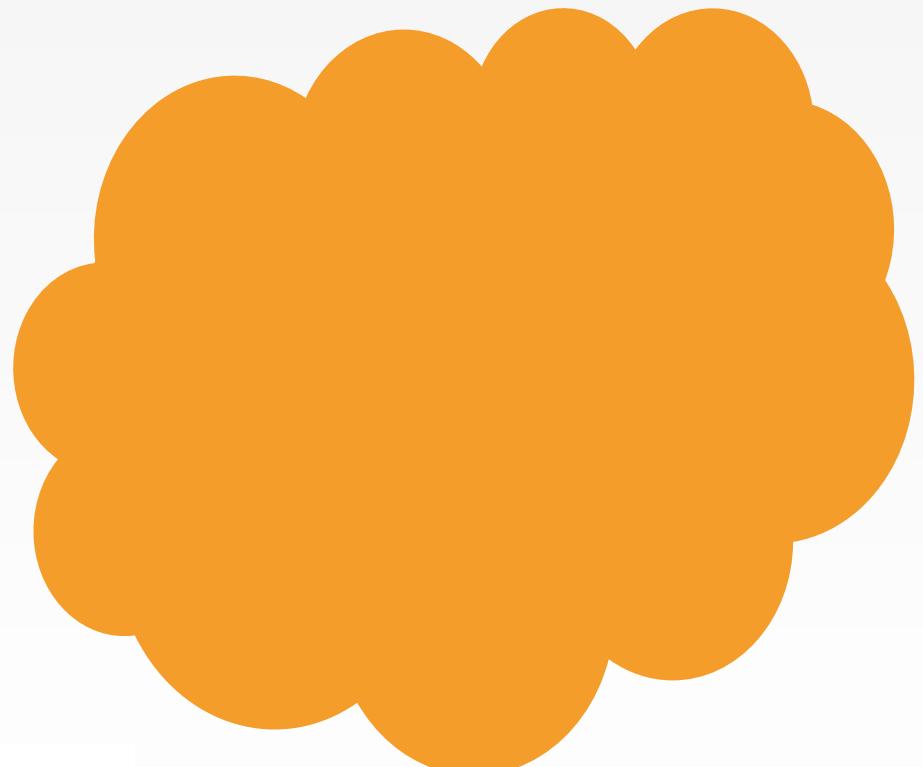
AWS



Chef Provisioning – Provisioning Node



Chef Server



AWS

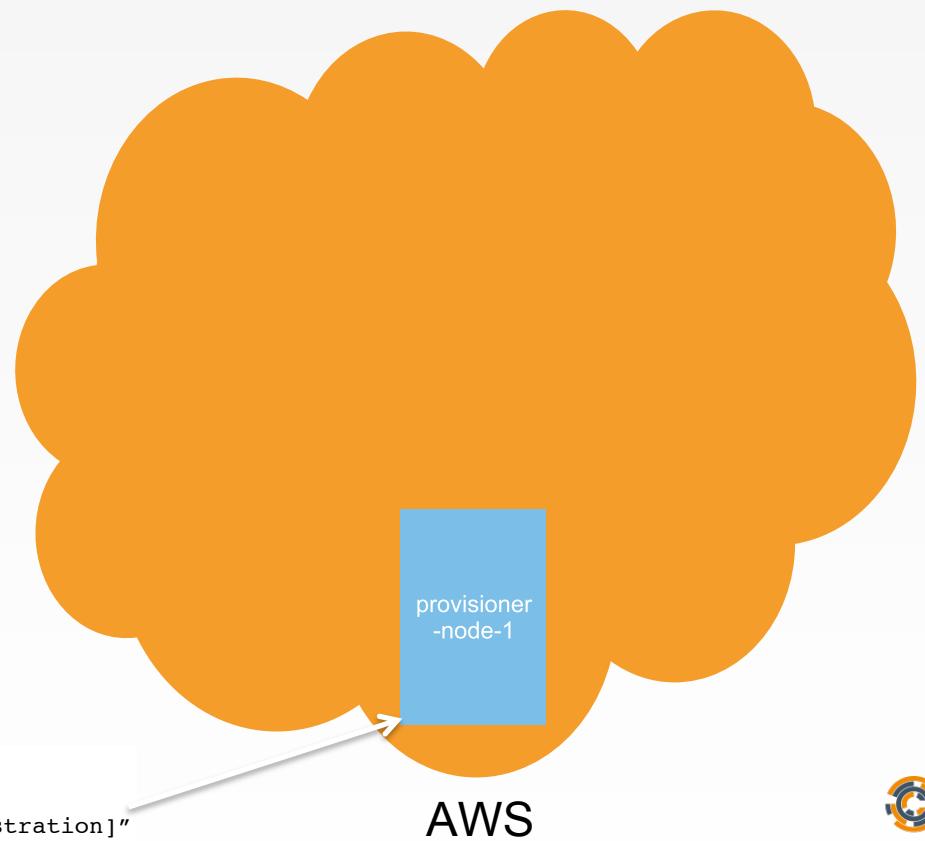


```
$ knife ec2 server create \
--node-name provisioner-node-1 \
--run-list "recipe[provision]", "recipe[provision::orchestration]"
```

Chef Provisioning – Provisioning Node



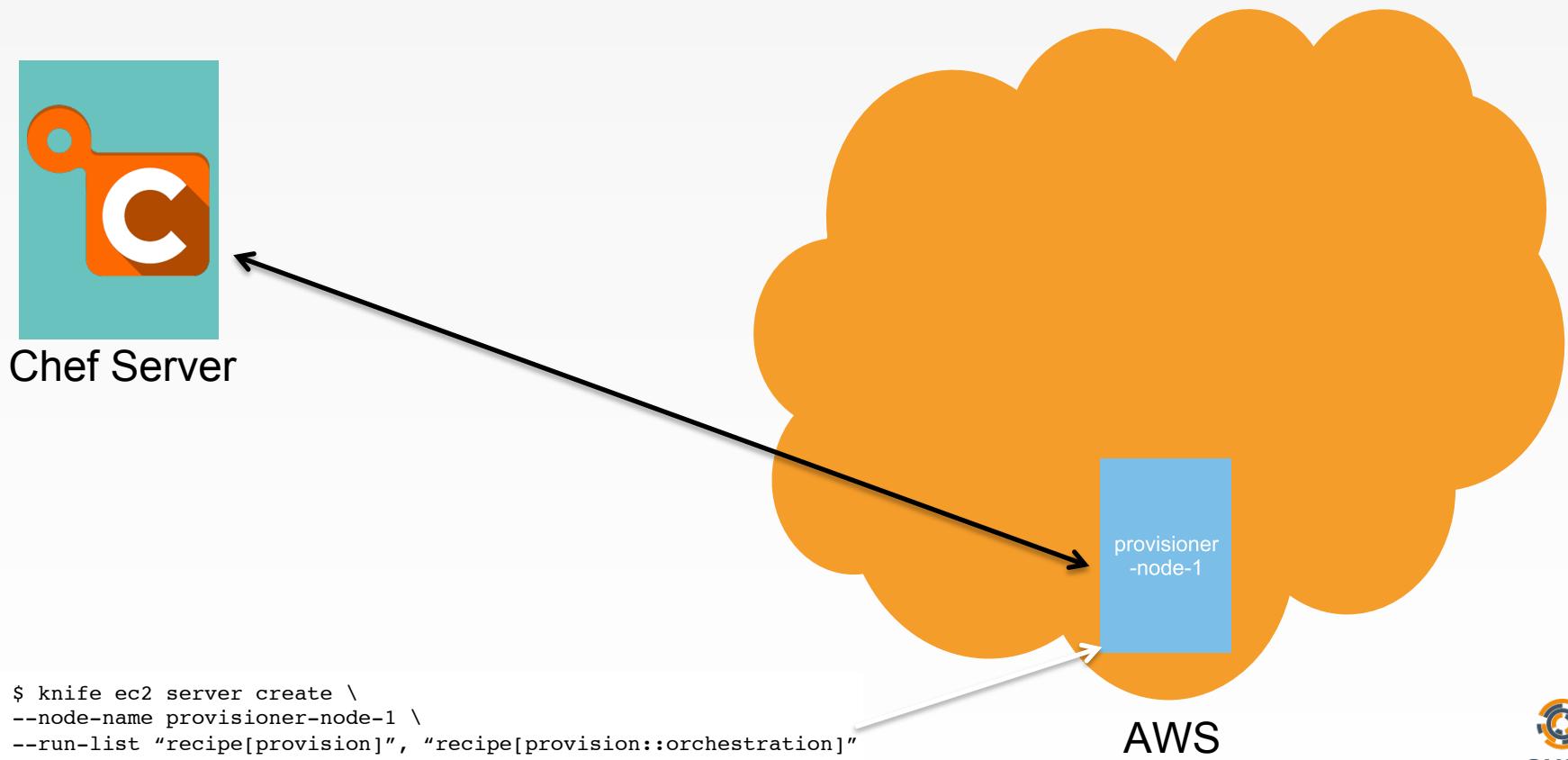
Chef Server



```
$ knife ec2 server create \
--node-name provisioner-node-1 \
--run-list "recipe[provision]", "recipe[provision::orchestration]"
```



Chef Provisioning – Provisioning Node



Chef Provisioning – Provisioning Node



Chef Server

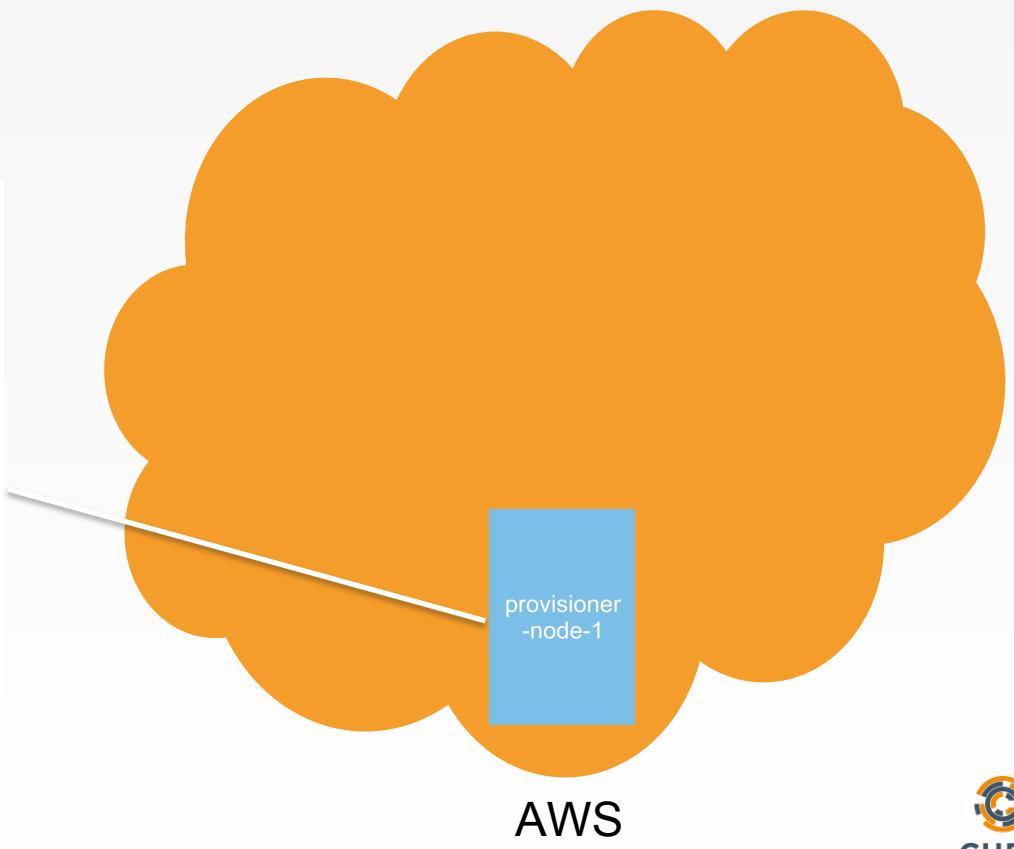
```
require 'chef/provisioning'

machine_batch do
  machines %w(primary secondary
web1 web2)
end

machine_batch do
  machine 'primary' do
    recipe 'initial_ha_setup'
  end
end

machine_batch do
  machine 'secondary' do
    recipe 'initial_ha_setup'
  end
end

machine_batch do
  %w(primary secondary).each do |name|
    machine name do
      recipe 'rest_of_my_shit'
    end
  end
end
```



AWS



Chef Provisioning – Provisioning Node



Chef Server

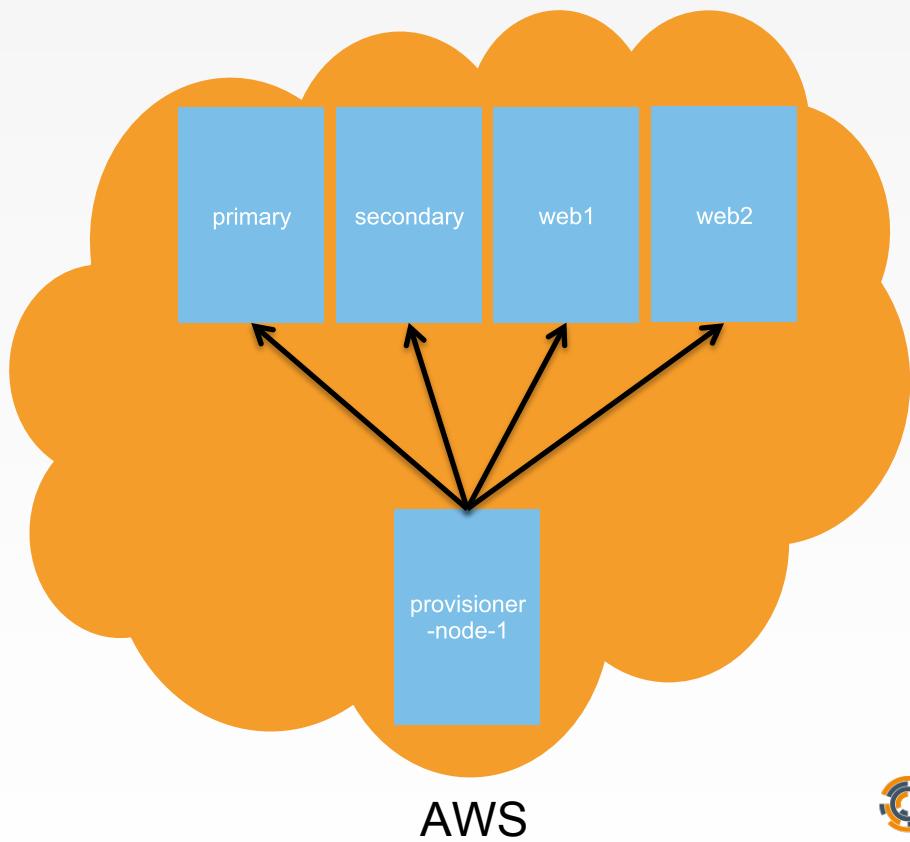
```
require 'chef/provisioning'

machine_batch do
  machines %w(primary secondary
web1 web2)
end

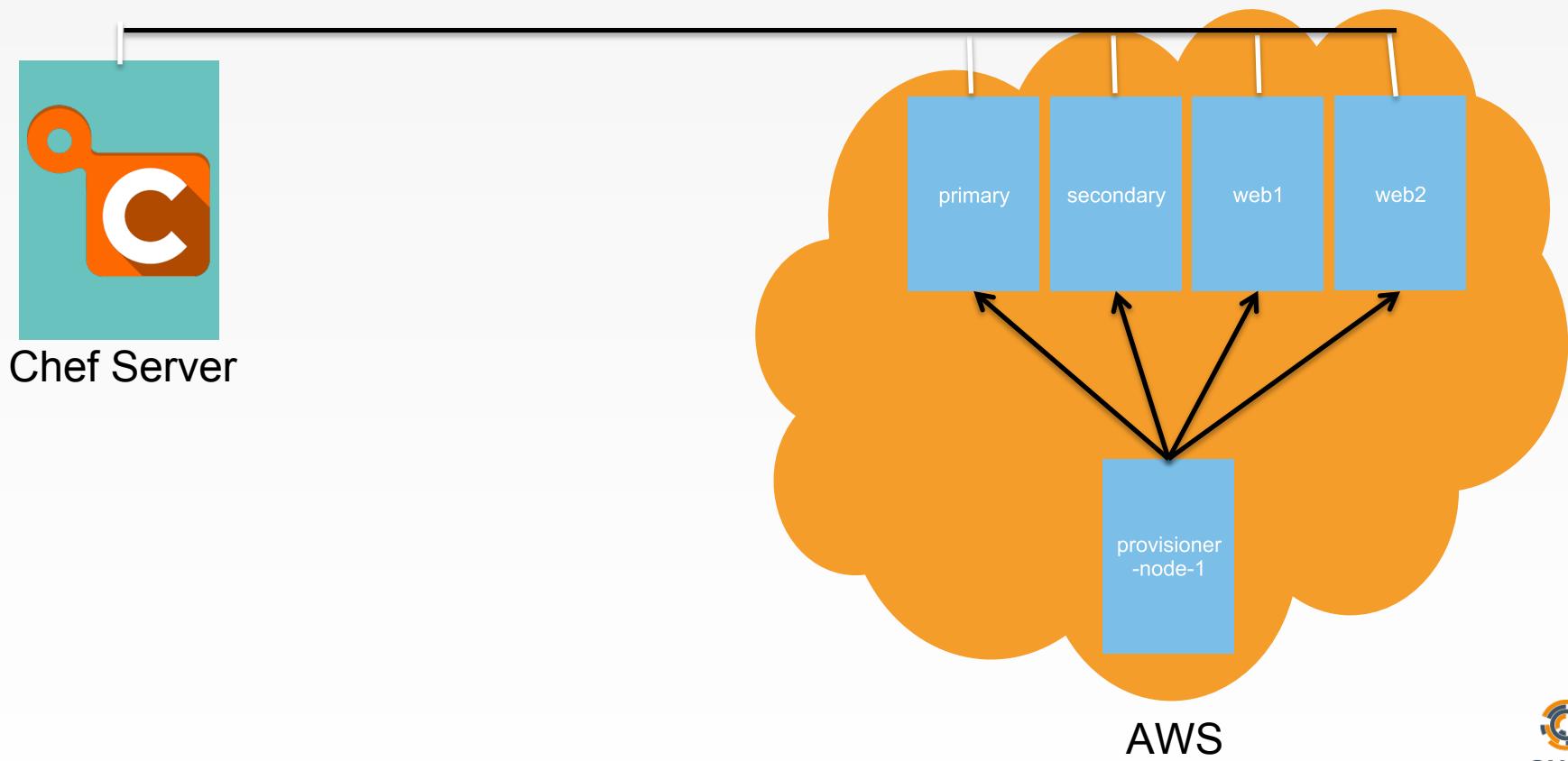
machine_batch do
  machine 'primary' do
    recipe 'initial_ha_setup'
  end
end

machine_batch do
  machine 'secondary' do
    recipe 'initial_ha_setup'
  end
end

machine_batch do
  %w(primary secondary).each do |name|
    machine name do
      recipe 'rest_of_my_shit'
    end
  end
end
```



Chef Provisioning – Provisioning Node



chef-server-cluster

- chef-server-cluster Cookbook
- Built On chef-provisioning
- Stand up Chef Server 12 AWS in a tiered configuration
- Stand up Analytics

Yo dawg, I heard you like Chef — Edit

42 commits 1 branch 0 releases 2 contributors

branch: master / +

add api_fqdn to chef-server.rb.erb

patrick-wright authored on Dec 2, 2014 latest commit d01bcfdacd

File	Description	Time Ago
attributes	updated to support chef-provisioning	3 months ago
libraries	Metal Refactor: Launch an Chef Server 12 Cluster	4 months ago
recipes	stash pivotal.pem and upload to frontends	2 months ago
spec	Metal Refactor: Launch an Chef Server 12 Cluster	4 months ago
templates/default	add api_fqdn to chef-server.rb.erb	2 months ago
test	Metal Refactor: Launch an Chef Server 12 Cluster	4 months ago
.gitignore	Initial attempt	11 months ago
.kitchen.yml	fix Rakefile syntax. update .kitchen.yml recipe	3 months ago
.rubocop.yml	Initial attempt	11 months ago
Berksfile	Metal Refactor: Launch an Chef Server 12 Cluster	4 months ago
CHANGELOG.md	Initial attempt	11 months ago
Gemfile	updated to support chef-provisioning	3 months ago
README.md	rename chef-provisioning to setup-provisioner	3 months ago
Rakefile	fix Rakefile syntax. update .kitchen.yml recipe	3 months ago
chefignore	Initial attempt	11 months ago
metadata.rb	increment patch version	2 months ago

<https://github.com/opscode-cookbooks/chef-server-cluster>



Chef Provisioning Recap

- Machine resource for creating instances
- Drivers for many providers
- Programmatically declare your infrastructure as code
- Testable
- Repeatable



What's on the horizon?

- Version 1.0 coming soon!
- IAM Roles support
- More resources...



How you can contribute

- Use it
- Fork the repositories
- Write new drivers
- Add resources
- Open issues



Where to go from here

- Chef-provisioning

<https://github.com/chef/chef-provisioning>

- Chef-provisioning-aws

<https://github.com/chef/chef-provisioning-aws>

- Gitter IM Channel

<https://gitter.im/chef/chef-provisioning>



CYOA

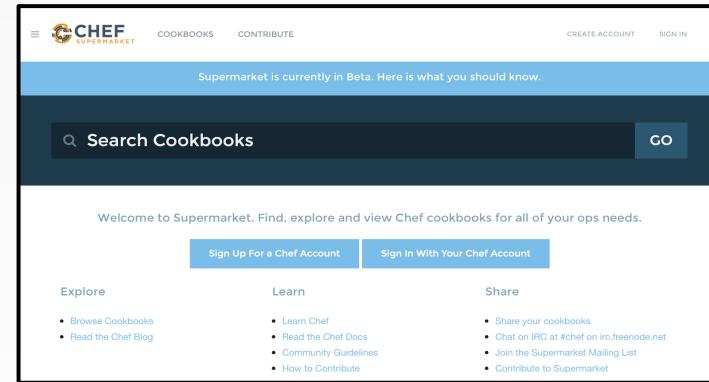
- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



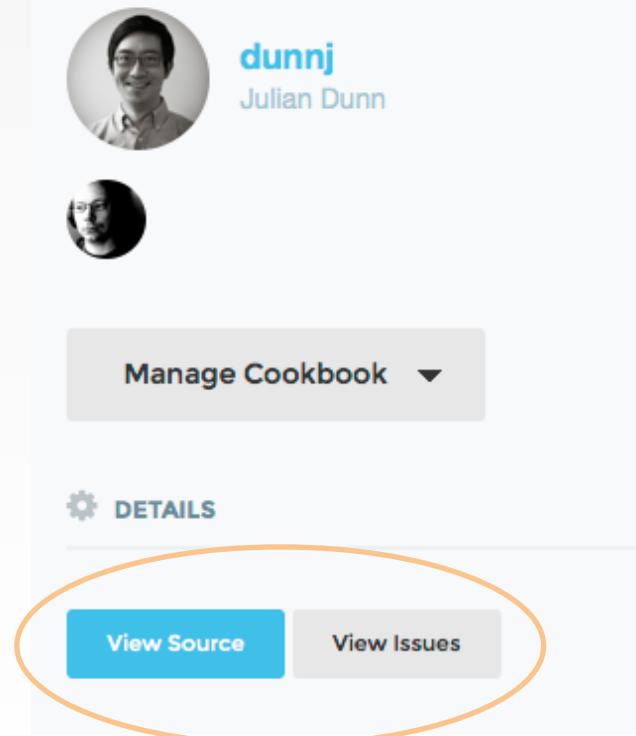
Community Cookbook Repo - Supermarket

- It's open source!
- Anyone can leverage this collection of code and best practices
- Organizations can host Supermarket internally/on-premise to empower their own technology community



New Cookbook Metadata Fields

- name is now mandatory
- issues_url and source_url for integration with Supermarket



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)





Chef on Windows



Speed Improvements

- Ruby 2.x improves garbage collection speed on Windows
- Appbundler (Chef-developed product) resolves & locks dependencies at package build time
 - Like "bundler", but bundler resolves dependencies at runtime
- More performance work planned in 2015
 - Ruby 2.1+ for Windows
 - Lazy loading of knife plugins
 - Speed up Ohai/minimize data collection



Reboot resource

```
# Reboot at end of Chef Client run
reboot "app_requires_it" do
  action :request_reboot
  reason "Need to reboot when the run completes successfully."
  delay_mins 5
end

# Reboot now
reboot "app_requires_it_now" do
  action :reboot_now
  reason "Need to reboot right now."
end
```



Windows Event Logging

- Chef Client 12 on Windows logs to Application Event Log by default

Application Number of events: 451 (0) New events available

Level	Date and Time	Source	Event ID	Task Category
Error	11/8/2014 4:16:24 AM	Chef	10003	None
Information	11/8/2014 4:16:01 AM	Chef	10001	None
Information	11/8/2014 4:15:54 AM	Chef	10000	None
Error	11/8/2014 4:12:44 AM	Chef	10003	None
Information	11/8/2014 4:12:24 AM	Chef	10001	None
Information	11/8/2014 4:12:01 AM	Chef	10000	None
Error	11/8/2014 4:11:52 AM	Chef	10003	None
Information	11/8/2014 4:11:20 AM	Chef	10001	None
Information	11/8/2014 4:11:14 AM	Chef	10000	None
Information	11/8/2014 2:49:19 AM	Windows Error Repor...	1001	None
Error	11/7/2014 11:46:43 PM	Chef	10003	None
Information	11/7/2014 11:46:38 PM	Chef	10001	None
Information	11/7/2014 11:46:27 PM	Chef	10000	None
Information	11/7/2014 11:45:09 PM	MsiInstaller	1033	None

Event 10003, Chef

General Details

```
Failed Chef Client run 1d419e04-0c3f-4d76-b05e-261ce126b774 in 4.874996 seconds.
Exception type: TypeError
Exception message: Registry is not a module
Exception backtrace: c:/chef/cache/cookbooks/windows/libraries/registry_helper.rb:361:in `<top (required)>'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context/cookbook_compiler.rb:191:in `load'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context/cookbook_compiler.rb:191:in `block in load_libraries_from_cookbook'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context/cookbook_compiler.rb:188:in `each'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context/cookbook_compiler.rb:188:in `load_libraries_from_cookbook'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context/cookbook_compiler.rb:99:in `block in compile_libraries'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context/cookbook_compiler.rb:98:in `each'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context/cookbook_compiler.rb:98:in `compile_libraries'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context/cookbook_compiler.rb:71:in `compile'
C:/opscode/chef/embedded/apps/chef/lib/chef/run_context.rb:97:in `load'
C:/opscode/chef/embedded/apps/chef/lib/chef/policy_builder/expand_node_object.rb:73:in `setup_run_context'
C:/opscode/chef/embedded/apps/chef/lib/chef/client.rb:235:in `setup_run_context'
C:/opscode/chef/embedded/apps/chef/lib/chef/client.rb:397:in `run'
C:/opscode/chef/embedded/apps/chef/lib/chef/application.rb:243:in `run_with_graceful_exit_option'
C:/opscode/chef/embedded/apps/chef/lib/chef/application.rb:220:in `block in run_chef_client'
C:/opscode/chef/embedded/apps/chef/lib/chef/local_mode.rb:38:in `with_server_connectivity'
```

Log Name: Application
Source: Chef
Event ID: 10003
Level: Error
User: N/A
OpCode:

Logged: 11/7/2014 11:46:43 PM
Task Category: None
Keywords: Classic
Computer: WIN-CGBR7K9L4MS.canhazchef.com

More Information: [Event Log Online Help](#)



Guard Interpreter Defaults

- Guards: `not_if/only_if` expressions in resources
- Guard interpreter: what the guard expression is executed with

Resource		
Client Version	powershell_script	batch
11.x	32-bit cmd.exe	32-bit cmd.exe
12.x	64-bit PowerShell	64-bit cmd.exe

More information: <http://tinyurl.com/guard-interpreter>



Windows Service Startup Type

```
windows_service "BITS" do
  action :configure_startup
  startup_type :manual
end
```

- Must use `windows_service` to do this; `:configure_startup` action and `startup_type` are not supported by 'service'



Windows PowerShell DSC

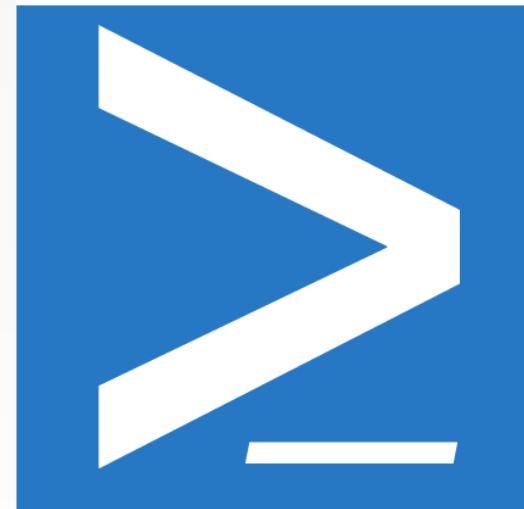
Comprehensive Integration

Multi-Platform Automation

- Easily automate Windows and Linux workloads in data center and Azure Cloud.
- Treat any DSC resource as a Chef resource.
- Improve collaboration between development and operations teams in heterogeneous environments.

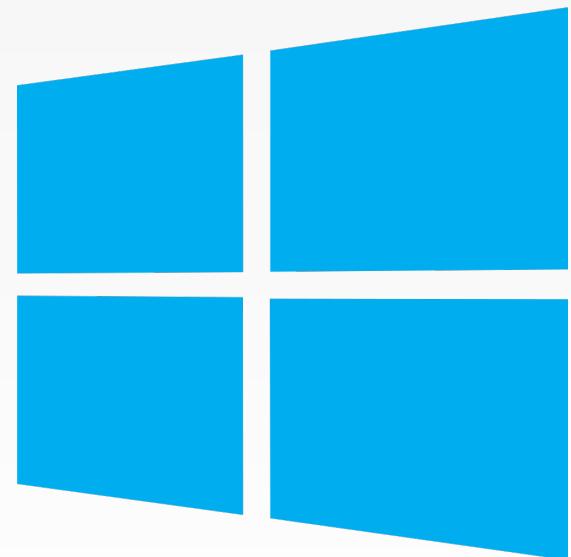
Azure Integration

- Ease of configuration, automation and management.
- Microsoft Open Technologies has [published a collection of three Chef cookbooks](#) for configuring Azure Virtual Machines.
- Reduce risk, management, and development cycle times.



Chef on Windows

- **Cross-Platform resources:**
 - file, remote_file, cookbook_file, template
 - directory, remote_directory
 - user, group
 - mount (can take CIFS paths)
 - env
 - service
 - execute
 - ruby_block
 - many others...



Chef on Windows

- **Unique to Windows:**
 - registry_key (new in Chef 11.0.0)
 - powershell_script (new in Chef 11.6.0)
 - batch (new in Chef 11.6.0)
 - Automatic architecture handling (:i386 vs. :x86_64)
 - Automatic Windows filesystem redirector handling (Wow64)
- **Long-term roadmap:**
 - Move all Windows-specific resources to core and out of 'Windows' cookbook



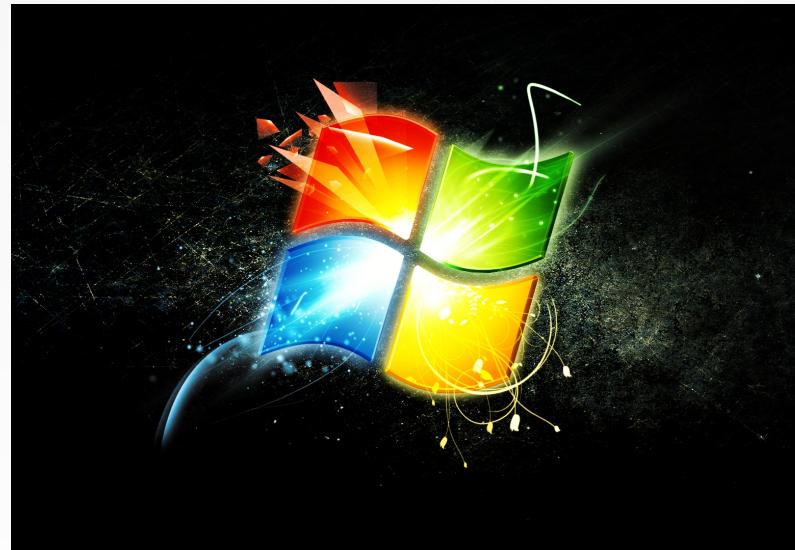
Chef on Windows

- **A few Windows-only cookbooks:**
 - azure (Community)
 - chocolatey (Community)
 - eventlog (Community)
 - iis (Chef)
 - msoffice (Community)
 - powershell (Chef)
 - sql_server (Chef)
 - webpi (Chef)
 - windows (Chef)
 - wix (Chef)
- **A few of the Windows-specific helper functions in the language:**
 - :registry_has_subkeys?
 - :registry_key_exists?
 - :registry_value_exists?
 - :windows_8_1?
 - :windows_server_2012_r2?
 - :marketing_name
 - :core? :cluster? :datacenter?
- **Many more!**



Chef on Windows

- **The Windows Cookbook** includes a number of additional resources and helper libraries.
- See <https://github.com/opscode-cookbooks/windows> for a full list
- **Highlights:**
 - windows_auto_run
 - windows_feature
 - windows_package
 - windows_path
 - windows_reboot
 - windows_zipfile
- **More:** windows_printer, windows_printer_port, windows_task



Chef on Windows: DSC Integration (Released)

```
dsc_script 'IIS' do
  code <<-EOH
    windowsfeature 'iis'
  {
    name = 'web-server'
  }
EOH
end
```

- Released – Embedding DSC configurations into Chef recipes with `dsc_script`
- Upcoming – 1:1 Mapping DSC resources to Chef resources
 - Try it out with the DSC cookbook! <https://supermarket.chef.io/cookbooks/dsc>
- See it live: Chef and Microsoft's Desired State Configuration
 - <https://www.youtube.com/watch?v=mXaAlawzNic>



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)





Chef in 2015



Themes

- **Improving velocity at CHEF Software, Inc**
- **Compliance at Velocity**
- **Moar Integrations!**
- **Management Console Delight**
- **Broader Enterprise Offerings**
- **TDI improvements and provisioning**
- **Ecosystem made more awesome**



Other Awesome Stuff

- More than twirling dials and knobs
- Configuration, Visibility, and Delivery at scale
- Community Engineering group



ChefConf



<http://chefconf.com>



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

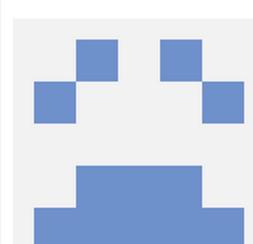
- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



DevOps DC Meetup

DevOpsDC

Home Members Sponsors Photos Pages Discussions More Group tools My profile



Change photo

Washington, DC
Founded Feb 9, 2011

About us... devops 1,154 Group reviews 25

DevOps in the Nation's Capital

+ SCHEDULE A NEW MEETUP

Upcoming 8 Suggested 0 Past Calendar

DevOpsDC Meetup - April 2015

Excella Consulting
2300 Wilson Blvd, Suite 630, Arlington, VA ([map](#))



Join us for the DevOpsDC Meetup Group!
Agenda: • 6:30 - 7:00 - Meet, greet, and eat
• 7:00 - 7:15 - Introductions and Announcements
• 7:15 - 8:45 - Presentations,... [LEARN MORE](#)

Hosted by: [Peter Burkholder](#) (Co-Organizer)

Tue Apr 14 6:30 PM

x NOT GOING 80 going 1 comment

[MORE](#)





NEW MEMBER

<http://www.meetup.com/DevOpsDC/>



ChefConf



<http://chefconf.com>



Upcoming Chef Training

- Chef Intermediate Topics – April 28-29
- Chef Fundamentals – April 30-May 1
- Discount code: MEETUP



DevOpsDays DC

- June 11-12
- USPTO, Alexandria
- Now Open
 - Registration
 - CFP
 - Sponsorships



<http://www.devopsdays.org/events/2015-washington-dc/>



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)



Thank you!

- What questions can I answer for you?
- nharvey@chef.io
- [@nathenharvey](https://twitter.com/nathenharvey)
- <http://bit.ly/farmer-nathen> (explicit)
- <http://bit.ly/farmer-nathen-sfw>



CYOA

- [High Availability](#)
- [Replication](#)
- [Client / Server Compatibility](#)
- [AIX](#)
- [Partial Search](#)
- [Platform-specific values](#)
- [Encrypted Databags](#)
- [shell out](#)
- [OS X](#)
- [Windows](#)

- [Client-side key generation](#)
- [Supermarket](#)
- [Ruby versions](#)
- [Chef Analytics](#)
- [Chef Development Kit](#)
- [Chef Provisioning](#)
- [Roadmap](#)
- [Chef in DC](#)
- [Ask me Anything](#)





CHEFTM
CODE CAN