

# Chef Integration (KITE) Techno Functional Spec

JANUARY 9, 2017

Kapil Thakkar

Email: [kapil@click2cloud.net](mailto:kapil@click2cloud.net)

Document Version – V1.1

## Version History

Version	Date	Name	Description of Change
1.0	2017-01-09	Kapil Thakkar	Draft for Chef Integration (KITE)
1.1	2017-01-09	Kapil Thakkar	Answered Shuwei questions

## Reviewers

Name	Version Approved	Role	Date
Manish Mishra	1.0	Project Manager	2017-01-09

## 1 Table of Contents

1	Introduction .....	4
1.1	Objectives .....	4
2	Prerequisites .....	4
2.1	System Description .....	4
2.2	Software .....	4
3	Knife (Drop-1) .....	4
3.1	Intro .....	4
3.2	Scope .....	4
3.3	Out of Scope .....	5
3.4	Assumptions and Constraints .....	5
3.4.1	Assumptions .....	5
3.4.2	Constraints .....	6
3.5	References .....	6
3.6	Architecture .....	6
3.6.1	Architecture Overview .....	6
3.7	CODE .....	7
3.7.1	File Names and Structure .....	7
3.8	Install Knife ECS Plugin gem .....	7
3.9	Knife ECS Commands .....	8
3.9.1	Server Create .....	8
3.9.2	Server List .....	11
3.9.3	Server Delete .....	12
3.9.4	Server Show .....	13
3.9.5	Flavor List .....	13
3.9.6	Image List .....	14
3.9.7	Region List .....	14
3.9.8	Security Group List .....	14
4	Kitchen (Drop-2) .....	15
4.1	Intro .....	15
4.2	Scope .....	15
4.3	Out of Scope .....	15
4.4	Assumptions and Constraints .....	15
4.4.1	Assumptions .....	15
4.4.2	Constraints .....	16

4.5	References .....	16
4.6	Architecture .....	16
4.6.1	Architecture Overview .....	16
4.7	Code .....	16
4.7.1	Major Tasks .....	16
4.7.2	Filenames and Structure .....	17
5	Ohai (Drop-3) .....	18
5.1	Intro .....	18
5.2	Scope .....	18
5.3	Out of Scope .....	18
5.4	Assumptions and Constraints .....	18
5.4.1	Assumptions .....	18
5.4.2	Constraints .....	19
5.5	References .....	19
5.6	Architecture .....	19
5.6.1	Architecture Overview .....	19
5.7	Explanation of each element .....	20
6	Deliverables .....	21
7	Disclaimer .....	22
8	Appendix A: Glossary .....	23
9	Appendix B: Document Change History and Sign-off .....	23
10	Point of Contact .....	23

# 1 Introduction

## 1.1 Objectives

The objective of this document is to describe technical and functional specification on KITE deliverable plugins and gem commands. This document covers scope of KITE and detailed on different parameters supported by Knife, Kitchen and Ohai plugins. The KITE will covers the implementation plugins of Knife, Kitchen and Ohai for Elastic Compute Service (ECS).

Elastic Compute Service (ECS) is a type of computing service that features elastic processing capabilities. Its management mode is simpler and more efficient than that of physical servers. Users can create instances, change the operating system, or release any number of ECS instances at any time according to business needs without upfront investment. An ECS computing services such as create instance, delete instance, list instance etc. are the manual processes. In this project our main goal is automate all these services using Knife, Kitchen, Ohai cloud plugins and chef server. Chef supports implementation of custom cloud plugin on ruby language to accomplish our task.

Chef is a powerful automation platform that transforms infrastructure into code. Whether users are operating in the cloud, on-premises, or in a hybrid environment, Chef automates how infrastructure is configured, deployed, and managed across your network, no matter its size.

# 2 Prerequisites

## 2.1 System Description

- Supporting Unix, Linux and Windows platforms with 64-bit architecture

## 2.2 Software

- The latest stable release of [Chef DK](#)
- Ruby 2.2.x or higher
- Gem packages of aliyun-ruby-api and knife-windows

# 3 Knife (Drop-1)

## 3.1 Intro

Knife is a command-line tool that provides an interface between a local chef-repo and the Chef server. Knife helps users to manage nodes, cookbooks, recipes, cloud resources, etc. Click2Cloud is developing a knife plugin which is a set of one (or more) subcommands that can be added to knife to support additional functionality that is not built-in to the base set of knife subcommands.

## 3.2 Scope

The scope of this project is to automate the instance related task through knife commands and bootstrap the node to chef server. The table below defines the list of task that are covered in this project.

Knife ECS Plugin - Commands	
1	ECS Options help content
2	Server Create
3	Server Start (Internally calling)
4	Server Stop (Internally calling)
5	Server Restart (Internally calling)
6	Server List
7	Server Delete
8	Server Show
9	Flavor List
10	Image List
11	Region List
12	Bootstrapping ECS Node (Internally calling)
13	Security Group list

### 3.3 Out of Scope

1. Other than the above tasks mentioned in [#3.2](#) will be consider as out of scope task for this project.
2. In KITE, Implementation are targeted for Linux and Windows systems only. The efforts are not considered for Mac Operating System.
3. Aliyun Marketplace images will not be part of this scope.

### 3.4 Assumptions and Constraints

#### 3.4.1 Assumptions

1. The technical platform such as Aliyun AK, Chef Server and Ruby SDK is available.
2. Change in any functional requirement documented below shall be treated as CR (Change Request).
3. Windows images should support WINRM services for bootstrapping.
4. This document to be freeze and sign-off before implementation start.
5. Addition of new parameter or commands specified in [#3.9](#) will be treated as CR after FSD sign-off.
6. The merge request shall be shared on GitHub for dropping code by Click2Cloud Team. The review and merging source code should be done by Aliyun Team.
7. All commands which need to integrate, are finalized by Aliyun Team under SOW.
8. In this phase Knife commands are targeted for Linux and Windows systems only.
9. All the supporting document will be provided in English language only.
10. Deploying and uploading Packages developed as a part of this project will be done by Aliyun Team. Such as hosting knife-ecs gems to gem repository.
11. Any update within REST API or Aliyun SDK for Ruby should be informed to Click2Cloud Team. Click2Cloud will do the required analysis and will identify the impact. If the impact is significant, then it will be communicated to stakeholders for further decision.
12. KITE, targets running commands for English language only. Other languages such as Chinese is not considered.

### 3.4.2 Constraints

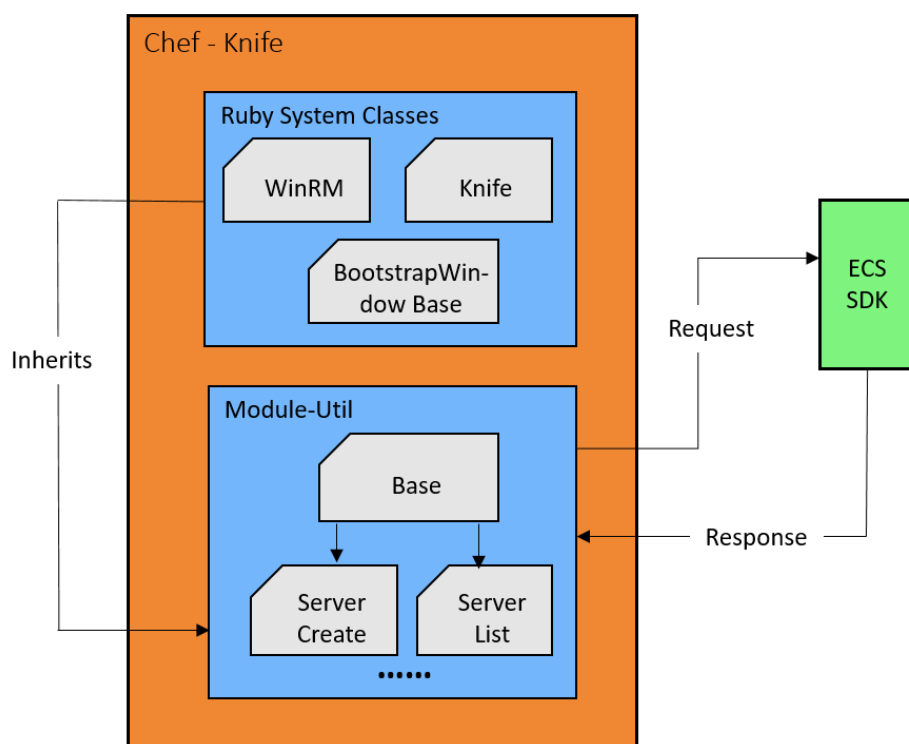
1. As currently Aliyun Web API is not working with Click2Cloud AK, team is going to use Aliyun Team AK for development and testing.

### 3.5 References

- <https://intl.aliyun.com/help/doc-detail/25485.htm?spm=a3c0i.o25484en.b99.122.pbJoYZ>
- [https://github.com/cheyang.aliyun\\_ruby\\_api](https://github.com/cheyang.aliyun_ruby_api)

### 3.6 Architecture

#### 3.6.1 Architecture Overview



As the image above:

1. The Ruby SDK is Aliyun Ruby SDK, it's available on [https://github.com/cheyang.aliyun\\_ruby\\_api](https://github.com/cheyang.aliyun_ruby_api)
2. Ruby System Classes contains the classes coming from Chef and Knife-Windows plugin.
3. Module-Util is the base area where all the command related files and the common files will be present. All the business logic and conditions implementation will be done in this module.

### 3.7 CODE

#### 3.7.1 File Names and Structure

##### 1. Knife ECS Plugin File Structure



**Note:** The above file structure shall be updated at the time of implementation (If required).

### 3.8 Install Knife ECS Plugin gem

Knife ECS plugin is a custom cloud plugin which will implement for Aliyun Cloud, it will provide several subcommands which interacts with Aliyun ECS services. Using the knife ECS plugin with knife users/organisations can manage their Aliyun ECS instances. Knife ECS makes it possible to create and bootstrap Aliyun instances in just single command.

To install knife ECS plugin run the following command on your Chef Workstation –

```
# gem install knife-ecs
```



In above example **gem install** command is used to install knife ECS plugin from public gem repository (once the gem is uploaded by Aliyun team on public gem repository).

### 3.9 Knife ECS Commands

Following are the list of knife ECS commands and its parameters.

Sr.no.	**ECS Commands**	Links
1	knife ecs server create (options)	<a href="#">Server Create</a>
2	knife ecs server list (options)	<a href="#">Server List</a>
3	knife ecs server delete SERVER [SERVER] (options)	<a href="#">Server Delete</a>
4	knife ecs server show SERVER (options)	<a href="#">Server Show</a>
5	knife ecs flavor list (options)	<a href="#">Flavor List</a>
6	knife ecs image list (options)	<a href="#">Image List</a>
7	knife ecs region list (options)	<a href="#">Region List</a>
8	knife ecs sg list (options)	<a href="#">Security Group List</a>

#### 3.9.1 Server Create

Create an Aliyun ECS instance using “**knife ecs server create**” command along with the below input parameters.

Options	Required	Description
--acs-access-key	Yes	Your Aliyun Access Key ID
-K, --acs-secret-access-key	Yes	Your Aliyun API Secret Access Key
--region REGION	Yes	Your ECS region id
-Z, --zone ZONE	No	The available zone id
-f, --flavor FLAVOR	Yes	The instance type for server (m1.small, m1.medium, etc)
-I, --image IMAGE	Yes	The Image ID for Server
-g, --security-group-id ID	Yes	The security group id for this server
--ssh-password PASSWORD	Yes	The ssh password for Instance
--ssh-user USERNAME	No	The ssh username, default is root
--ssh-port PORT	No	The ssh port, default is 22
--allocate-public-ip	No	Whether allocate a public IP for the new instance.
--bind-eip ID	No	ID of Elastic IP Address bind to the new instance.
-N, --node-name NAME	No	The ECS Instance/Chef node name for your new node
--io-optimized	No	I/O optimized. Optional values are: True/False. Default value is False
--vswitch-id ID	No	The subnet ID in which to launch the instance (VPC).
--host-name	No	Host name of the ECS Instance
-T, --instance-tags [Tag=Value]	No	A tags for the Server. [Tag=Value]
--data-disk	No	Numbers of data disk add to Instance
--data-disk-size [X,Y,Z]	No	Size of the n volume, n starts from 1. In GB
--data-disk-category [X,Y,Z]	No	Category of the volume n. Values are Ultra and SSD
--disk-delete-on-term	No	Whether a volume is released with the instance. Default is True
--internet-charge-type TYPE	No	Internet charge type, which can be PayByTraffic or PayByBandwidth. Default is PayByBandwidth
--internet-max-bandwidth-out	No	Maximum outgoing bandwidth to the public network,

VALUE		measured in Mbps. Value range for PayByBandwidth: [0, 100] and PayByTraffic: [1, 100].
--instance-charge-type TYPE	No	Payment methods. Values are Postpaid or Prepaid. Default is Postpaid (Pay-as-you-go)
--auto-renew	No	Whether automatic renewal is supported. Only valid when InstanceChargeType is set PrePaid. Default is False.
-V, --verbose	No	More verbose output. Use twice for max verbosity
-v, --version	No	Show chef version
-y, --yes	No	Say yes to all prompts for confirmation
-h, --help	No	Show this message

#### *Chef and Knife Windows Specific Parameters:*

Options	Required	Description
--secret	No	The secret key to use to encrypt data bag item values
--secret-file SECRET_FILE	No	A file containing the secret key to use to encrypt data bag item values
--bootstrap-curl-options OPTIONS	No	Add options to curl when install chef-client
--bootstrap-install-command COMMANDS	No	Custom command to install chef-client
--bootstrap-no-proxy [NO_PROXY_URL NO_PROXY_IP]	No	Do not proxy locations for the node being bootstrapped; this option is used internally by Opscode
--bootstrap-protocol protocol	No	protocol to bootstrap windows servers. options: winrm/ssh
--bootstrap-proxy PROXY_URL	No	The proxy server for the node being bootstrapped
--bootstrap-template TEMPLATE	No	Bootstrap Chef using a built-in or custom template. Set to the full path of an erb template or use one of the built-in templates.
--bootstrap-url URL	No	URL to a custom installation script
--bootstrap-vault-file VAULT_FILE	No	A JSON file with a list of vault(s) and item(s) to be updated
--bootstrap-vault-item VAULT_ITEM	No	A single vault and item to update as "vault:item"
--bootstrap-vault-json VAULT_JSON	No	A JSON string with the vault(s) and item(s) to be updated
--bootstrap-version VERSION	No	The version of Chef to install
--bootstrap-wget-options OPTIONS	No	Add options to wget when installing chef-client
--ca-trust-file CA_TRUST_FILE	No	The Certificate Authority (CA) trust file used for SSL transport
--server-url URL	No	Chef Server URL
--chef-zero-host HOST	No	Host to start chef-zero on
--chef-zero-port PORT	No	Port (or port range) to start chef-zero on. Port ranges like 1000,1010 or 8889-9999 will try all given ports until one works.
-k, --key KEY	No	API Client Key
--[no-]color	No	Use colored output, defaults to enabled
-c, --config CONFIG	No	The configuration file to use
--config-option OPTION=VALUE	No	Override a single configuration option

--[no-]create-ssl-listener	No	Create ssl listener, enabled by default.
--defaults	No	Accept default values for all questions
--disable-editing	No	Do not open EDITOR, just accept the data as is
-e, --editor EDITOR	No	Set the editor to use for interactive commands
-s	No	The secret key to use to decrypt data bag item values. Will be rendered on the node at c:/chef/encrypted_data_bag_secret and set in the rendered client config.
-E, --environment ENVIRONMENT	No	Set the Chef environment (except for in searches, where this will be flagrantly ignored)
--ephemeral EPHEMERAL_DEVICES	No	Comma separated list of device locations (eg - /dev/sdb) to map ephemeral devices
--[no-]fips	No	Enable fips mode
-j JSON_ATTRIBS, --json-attributes	No	A JSON string to be added to the first run of chef-client
--json-attribute-file FILE	No	A JSON file to be used to the first run of chef-client
-F, --format FORMAT	No	Which format to use for output
-A, --forward-agent	No	Enable SSH agent forwarding
--hint HINT_NAME[=HINT_FILE]	No	Specify Ohai Hint to be set on the bootstrap target. Use multiple --hint options to specify multiple hints.
--[no-]host-key-verify	No	Verify host key, enabled by default.
--install-as-service	No	Install chef-client as a Windows service
--keytab-file KEYTAB_FILE	No	The Kerberos keytab file used for authentication
-R KERBEROS_REALM, --kerberos-realm	No	The Kerberos realm used for authentication
--kerberos-service KERBEROS_SERVICE	No	The Kerberos service used for authentication
--[no-]listen	No	Whether a local mode (-z) server binds to a port
--msi-url URL	No	Location of the Chef Client MSI. The default templates will prefer to download from this location. The MSI will be downloaded from chef.io if not provided.
-u, --user USER	No	API Client Username
--node-ssl-verify-mode [peer none]	No	Whether or not to verify the SSL cert for all HTTPS requests.
--[no-]node-verify-api-cert	No	Verify the SSL cert for HTTPS requests to the Chef server API.
--prerelease	No	Install the pre-release chef gems
--print-after	No	Show the data after a destructive operation
-r, --run-list RUN_LIST	No	Comma separated list of roles/recipes to apply
--session-timeout Minutes	No	The timeout for the client for the maximum length of the WinRM session
--ssl-peer-fingerprint FINGERPRINT	No	ssl Cert Fingerprint to bypass normal cert chain checks
--template-file TEMPLATE	No	Full path to location of template to use. [DEPRECATED] Use -t / --bootstrap-template option instead.
--windows-auth-timeout MINUTES	No	The maximum time in minutes to wait to for authentication over the transport to the node to succeed. The default value is 25 minutes.
--winrm-authentication-protocol AUTHENTICATION_PROTOCOL	No	The authentication protocol used during WinRM communication. The supported protocols are

		basic,negotiate,kerberos. Default is 'negotiate'.
--winrm-codepage Codepage	No	The codepage to use for the winrm cmd shell
-P, --winrm-password PASSWORD	No	The WinRM password
-p, --winrm-port PORT	No	The WinRM port, by default this is '5985' for 'plaintext' and '5986' for 'ssl' winrm transport
--winrm-shell SHELL	No	The WinRM shell type. Valid choices are [cmd, powershell, elevated]. 'elevated' runs powershell in a scheduled task
--winrm-ssl-verify-mode SSL_VERIFY_MODE	No	The WinRM peer verification mode. Valid choices are [verify_peer, verify_none]
-t, --winrm-transport TRANSPORT	No	The WinRM transport type. Valid choices are [ssl, plaintext]
-x, --winrm-user USERNAME	No	The WinRM username
--validation-key-url URL	No	Path to the validation key
--use-sudo-password	No	Execute the bootstrap via sudo with password
-d, --distro DISTRO	No	Bootstrap a distro using a template. [DEPRECATED] Use --bootstrap-template option instead.
-z, --local-mode	No	Point knife commands at local repository instead of server

The tasks like Start, Stop, Restart and Bootstrapping discussed in [#3.2](#) are implicitly calling functions within Create and Delete commands.

### 3.9.2 Server List

This command “**knife ecs server list**” is used to see the list of instances and its status.

Options	Required	Description
--acs-access-key	Yes	Your Aliyun Access Key ID
-K, --acs-secret-access-key	Yes	Your Aliyun API Secret Access Key
--region REGION	Yes	Your ECS region id
-Z, --zone ZONE	No	The available zone id
-n, --no-name	No	Do not display name tag in output
-V, --verbose	No	More verbose output. Use twice for max verbosity
-v, --version	No	Show chef version
-y, --yes	No	Say yes to all prompts for confirmation
-h, --help	No	Show this message

#### Chef Specific Parameters:

Options	Required	Description
--server-url URL	No	Chef Server URL
--chef-zero-host HOST	No	Host to start chef-zero on
--chef-zero-port PORT	No	Port (or port range) to start chef-zero on. Port ranges like 1000,1010 or 8889-9999 will try all given ports until one works.
-k, --key KEY	No	API Client Key
--[no-]color	No	Use coloured output, defaults to enabled

-c, --config CONFIG	No	The configuration file to use
--config-option OPTION=VALUE	No	Override a single configuration option
--[no-]create-ssl-listener	No	Create ssl listener, enabled by default.
--defaults	No	Accept default values for all questions
--disable-editing	No	Do not open EDITOR, just accept the data as is
-e, --editor EDITOR	No	Set the editor to use for interactive commands
-E, --environment ENVIRONMENT	No	Set the Chef environment (except for in searches, where this will be flagrantly ignored)
--[no-]fips	No	Enable fips mode
-F, --format FORMAT	No	Which format to use for output
--[no-]listen	No	Whether a local mode (-z) server binds to a port
-u, --user USER	No	API Client Username
--print-after	No	Show the data after a destructive operation
-z, --local-mode	No	Point knife commands at local repository instead of server

### 3.9.3 Server Delete

This command “**knife ecs server delete [SERVER]**” is used to delete server instance.

Options	Required	Description
--acs-access-key-id	Yes	Your Aliyun Access Key ID
-K, --acs-secret-access-key	Yes	Your Aliyun API Secret Access Key
--region REGION	Yes	Your ECS region id
-Z, --zone ZONE	No	The available zone id
-N, --node-name NAME	No	The ECS Instance/Chef node name for your new node
-P, --purge	No	Destroy corresponding node and client on the Chef Server,
-V, --verbose	No	More verbose output. Use twice for max verbosity
-v, --version	No	Show chef version
-y, --yes	No	Say yes to all prompts for confirmation
-h, --help	No	Show this message

### Chef Specific Parameters:

Options	Required	Description
--server-url URL	No	Chef Server URL
--chef-zero-host HOST	No	Host to start chef-zero on
--chef-zero-port PORT	No	Port (or port range) to start chef-zero on. Port ranges like 1000,1010 or 8889-9999 will try all given ports until one works.
-k, --key KEY	No	API Client Key
--[no-]color	No	Use colored output, defaults to enabled
-c, --config CONFIG	No	The configuration file to use
--config-option OPTION=VALUE	No	Override a single configuration option
--[no-]create-ssl-listener	No	Create ssl listener, enabled by default.
--defaults	No	Accept default values for all questions

--disable-editing	No	Do not open EDITOR, just accept the data as is
-e, --editor EDITOR	No	Set the editor to use for interactive commands
-E, --environment ENVIRONMENT	No	Set the Chef environment (except for in searches, where this will be flagrantly ignored)
--[no-]fips	No	Enable fips mode
-F, --format FORMAT	No	Which format to use for output
--[no-]listen	No	Whether a local mode (-z) server binds to a port
-u, --user USER	No	API Client Username
--print-after	No	Show the data after a destructive operation
-z, --local-mode	No	Point knife commands at local repository instead of server

### 3.9.4 Server Show

This command “**knife ecs server show SERVER**” is used to describe the instance information.

Options	Required	Description
--acs-access-key-id	Yes	Your Aliyun Access Key ID
-K, --acs-secret-access-key	Yes	Your Aliyun API Secret Access Key
--region REGION	Yes	Your ECS region id
-Z, --zone ZONE	No	The available zone id
-V, --verbose	No	More verbose output. Use twice for max verbosity
-v, --version	No	Show chef version
-y, --yes	No	Say yes to all prompts for confirmation
-h, --help	No	Show this message

### 3.9.5 Flavor List

This command “**knife ecs flavor list**” is used to see the list of instance types available in selected region.

Options	Required	Description
--acs-access-key-id	Yes	Your Aliyun Access Key ID
-K, --acs-secret-access-key	Yes	Your Aliyun API Secret Access Key
--region REGION	Yes	Your ECS region id
-Z, --zone ZONE	No	The available zone id
-V, --verbose	No	More verbose output. Use twice for max verbosity
-v, --version	No	Show chef version
-y, --yes	No	Say yes to all prompts for confirmation
-h, --help	No	Show this message

### 3.9.6 Image List

This command **"knife ecs image list"** is used to see the list of images available in selected region.

Options	Required	Description
--acs-access-key-id	Yes	Your Aliyun Access Key ID
-K, --acs-secret-access-key	Yes	Your Aliyun API Secret Access Key
--region REGION	Yes	Your ECS region id
-Z, --zone ZONE	No	The available zone id
--image-owner-alias	No	The alias of the image owner. Values are: system   self   others   marketplace. Default value is "system + self + others"
-V, --verbose	No	More verbose output. Use twice for max verbosity
-v, --version	No	Show chef version
-y, --yes	No	Say yes to all prompts for confirmation
-h, --help	No	Show this message

### 3.9.7 Region List

This command **"knife ecs region list"** is used to see the list of regions available in Aliyun Cloud.

Options	Required	Description
--acs-access-key-id	Yes	Your Aliyun Access Key ID
-K, --acs-secret-access-key	Yes	Your Aliyun API Secret Access Key
-V, --verbose	No	More verbose output. Use twice for max verbosity
-v, --version	No	Show chef version
-y, --yes	No	Say yes to all prompts for confirmation
-h, --help	No	Show this message

### 3.9.8 Security Group List

This command **"knife ecs sg list"** is used to see the list of security groups available in selected region.

Options	Required	Description
--acs-access-key-id	Yes	Your Aliyun Access Key ID
-K, --acs-secret-access-key	Yes	Your Aliyun API Secret Access Key
--region REGION	Yes	Your ECS region id
-Z, --zone ZONE	No	The available zone id
-V, --verbose	No	More verbose output. Use twice for max verbosity
-v, --version	No	Show chef version
-y, --yes	No	Say yes to all prompts for confirmation
-h, --help	No	Show this message

## 4 Kitchen (Drop-2)

### 4.1 Intro

Kitchen driver allows to automate testing process to ensure the code has done the right thing, it is designed specifically to make it easy to plug that testing process into your continuous integration workflow to improve infrastructure code cookbook data across any combination of platforms and test suites. Kitchen driver creates ECS instance of given platform, tests cookbooks and destroys the instance when it's done.

### 4.2 Scope

In this scope Click2Cloud is going to develop Kitchen ECS driver which requires kitchen.yml file to support cookbook testing across cloud providers and can also supports all testing frameworks used by the Ruby. The kitchen.yml includes driver, provisioner, platforms, verifiers and test suites which helps in running the kitchen driver.

Following platforms are in development scope of project for kitchen ECS driver:

#	Kitchen Driver Standard Platforms
1.	CentOS
2.	Debian
3.	FreeBSD
4.	Ubuntu
5.	Windows
6.	Aliyun Linux

### 4.3 Out of Scope

1. Other than the above tasks mentioned in [#4.2](#) will be consider as out of scope task for this project.

### 4.4 Assumptions and Constraints

#### 4.4.1 Assumptions

1. The technical platform such as Aliyun AK, Chef Server and Ruby SDK is available.
2. Change in any functional requirement documented below shall be treated as CR (Change Request).
3. Windows images should support WINRM services for bootstrapping.
4. This document to be freeze and sign-off before implementation start.
5. The merge request shall be shared on GitHub for dropping code by Click2Cloud Team. The review and merging source code should be done by Aliyun Team.
6. All the supporting document will be provided in English language only.
7. Deploying and uploading Packages developed as a part of this project will be done by Aliyun Team. Such as hosting kitchen-ecs gem to gems repository.
8. Any update within REST API or Aliyun SDK for Ruby should be informed to Click2Cloud Team. Click2Cloud will do required analysis and will identify the impact. If the impact is significant, then it will be communicated to stakeholders for further decision.
9. KITE, targets running test commands for English language only. Other languages such as Chinese is not considered



#### 4.4.2 Constraints

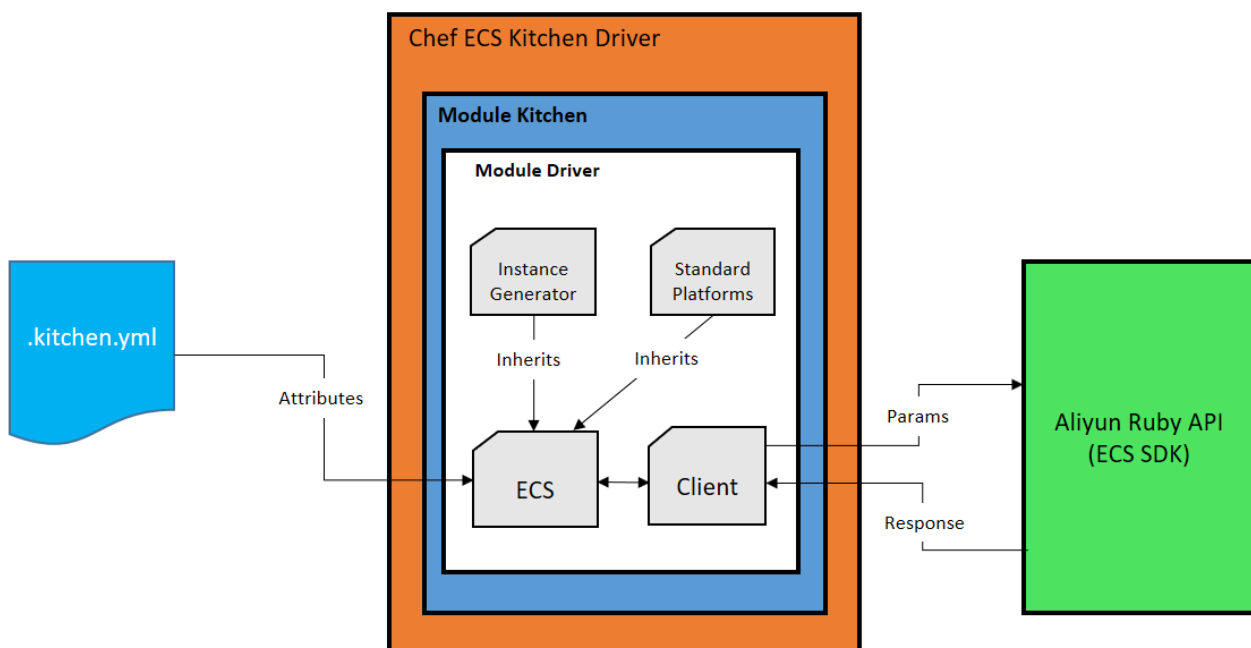
1. As currently Aliyun Web API is not working with Click2Cloud AK, team is going to use Aliyun Team AK for development and testing.

#### 4.5 References

- <https://intl.aliyun.com/help/doc-detail/25485.htm?spm=a3c0i.o25484en.b99.122.pbJoYZ>
- [https://github.com/cheyang.aliyun\\_ruby\\_api](https://github.com/cheyang.aliyun_ruby_api)

#### 4.6 Architecture

##### 4.6.1 Architecture Overview



Above architecture shows flow of ECS kitchen driver, `kitchen.yml` contains attributes such as driver, provisioner, platforms, verifier, and test suites these attributes will be given to ECS class, then it will identify the configurations according to attributes. Along with some inherited features from instance generator and standard platforms ECS calls services of SDK with the help of client class to create, test and destroy instances, here **Aliyun Ruby API** SDK is used to communicate with Aliyun ECS for the machine instance create and destroy.

#### 4.7 Code

##### 4.7.1 Major Tasks

###### ▪ Create Instance:

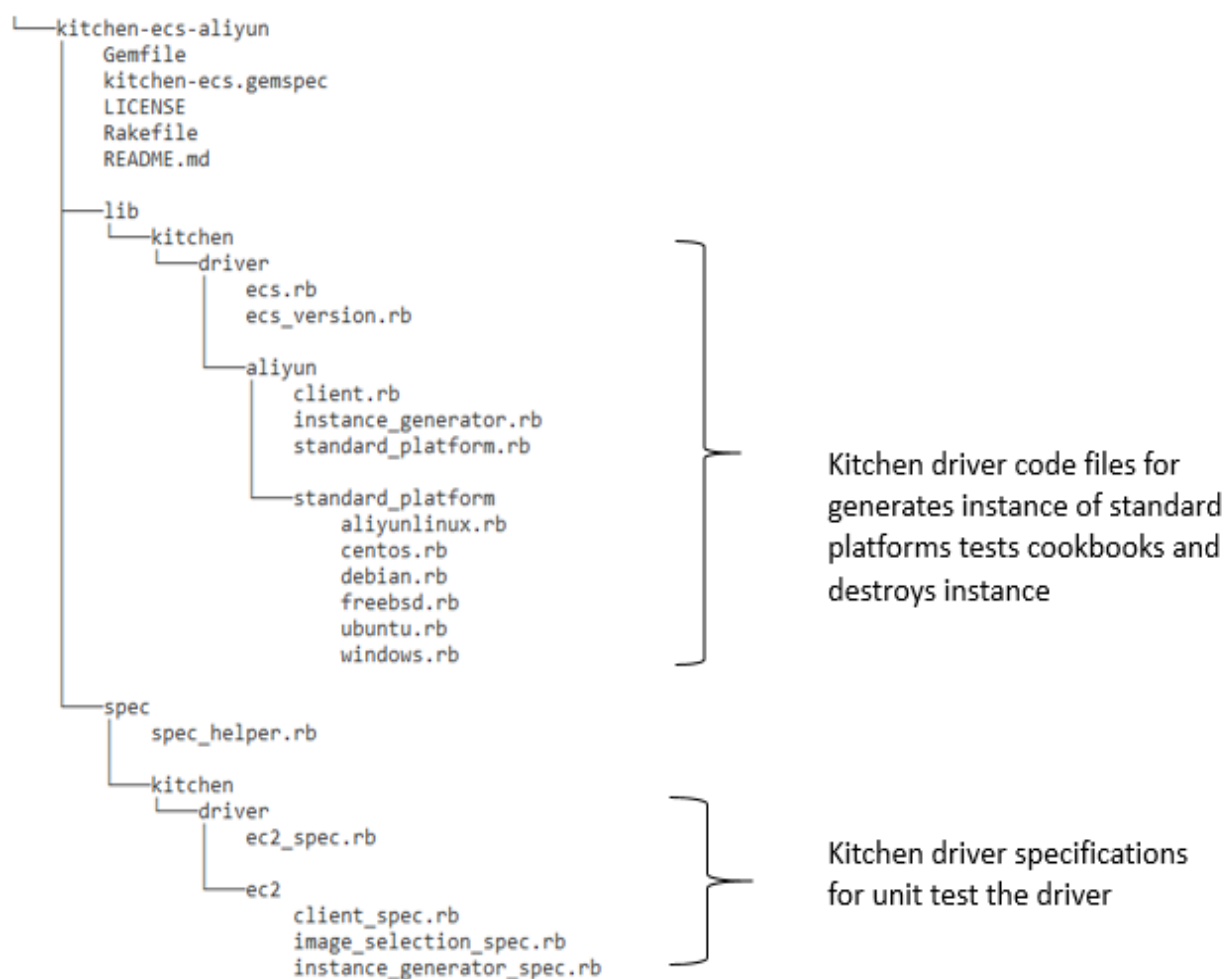
Kitchen create will create a virtual machine instance for every suite and platform combination defined in your `kitchen.yml`, This is so that each test suite can be run in its own isolated environment.

Create the instance, do what is needed to converge on that instance (such as installing the chef-client, uploading cookbooks, starting the chef-client run, and so on), setup anything else needed for testing

- **Destroy Instance:**  
When kitchen virtual machine instance gets created and test suite run operation completed after this the instance data will be fetched and it will be destroyed automatically using this task internally.
- **Image Search:**  
Image search is done on the basis of the specification in kitchen.yml file for creation the instance of specified configuration
- **Platforms**  
Platforms are detected from searched images and will be taken from the standard platforms for the creation of instance.

The above given tasks are some major tasks which Click2Cloud identified and its need to be implemented for Kitchen driver to work with Aliyun cloud, some internal tasks also be there to handle this operations.

#### 4.7.2 Filenames and Structure



## 5 Ohai (Drop-3)

### 5.1 Intro

Ohai is a tool that is used to detect attributes on a node, and then provide these attributes to the chef-client at the start of every chef-client runs. Ohai is required by the chef-client and must be present on a node.

### 5.2 Scope

The scope of this project is to get the ECS node information. The table below list the information which Click2Cloud is going to include in the Ohai plugin.

#	List of elements to include under ECS
1	Public IP
2	Private IP
3	Instance ID
4	Instance Name
5	CPU
6	Memory
7	Bandwidth
8	IO Optimize
9	Region
10	Tag

### 5.3 Out of Scope

1. Other than the above tasks mentioned in [#5.2](#) will be consider as out of scope task for this project.

### 5.4 Assumptions and Constraints

#### 5.4.1 Assumptions

1. The technical platform such as Aliyun AK, Chef Server and Ruby SDK is available.
2. Aliyun Ruby SDK is available and it contains all the methods related to our scope.
3. Change in any functional requirement documented below shall be treated as CR (Change Request).
4. This document to be freeze and sign-off before implementation start.
5. Addition of element in ECS Ohai mentioned in [#5.7](#) will be treated as CR after FSD sign-off.
6. The merge request shall be shared on GitHub for dropping code by Click2Cloud Team. The review and merging source code should be done by Aliyun Team.

7. Click2Cloud are assuming that all the element which is listed in the scope [#5.2](#) are able to retrieve using API.
8. All the supporting document will be provided in English language only.
9. Deploying and uploading Packages developed as a part of this project will be done by Aliyun Team.
10. Any update within REST API or Aliyun SDK for Ruby should be informed to Click2Cloud Team. Click2Cloud will do required analysis and will identify the impact. If the impact is significant, then it will be communicated to stakeholders for further decision.
11. KITE, targets running commands for English language only. Other languages such as Chinese is not considered.

#### 5.4.2 Constraints

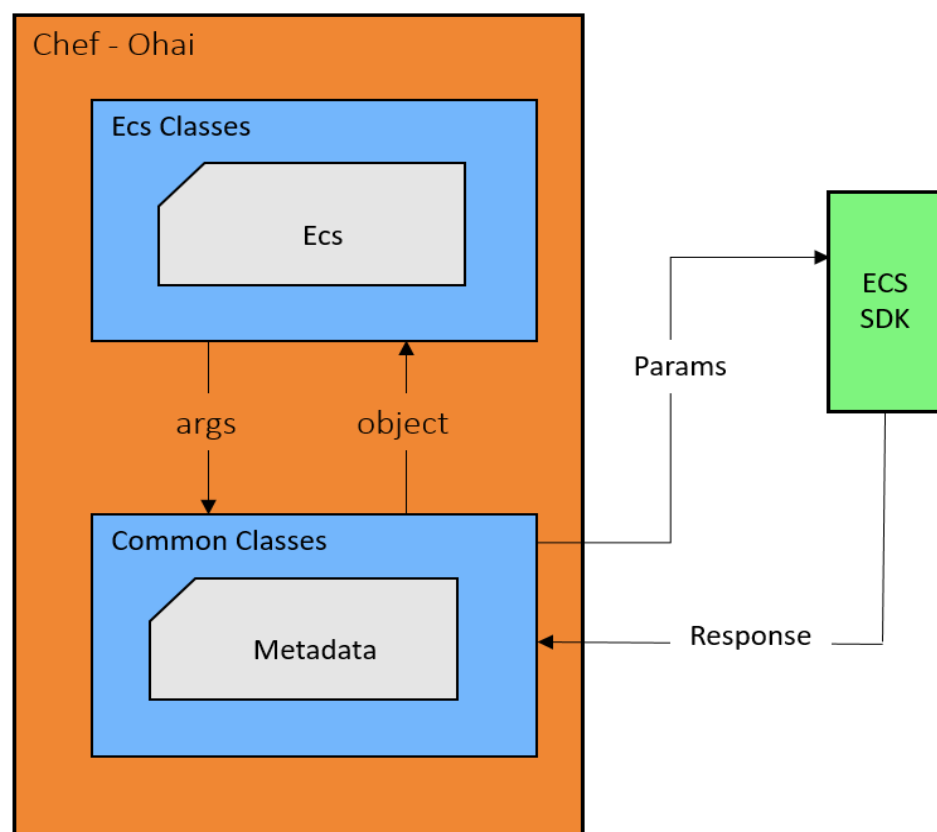
1. As currently Aliyun Web API is not working with Click2Cloud AK, team is going to use Aliyun Team AK for development and testing.

### 5.5 References

- [https://github.com/cheyang.aliyun\\_ruby\\_api](https://github.com/cheyang.aliyun_ruby_api)

### 5.6 Architecture

#### 5.6.1 Architecture Overview



As the image above:

1. The **Ruby SDK** is Aliyun Ruby SDK, it's available on [https://github.com/cheyang/aliyun\\_ruby\\_api](https://github.com/cheyang/aliyun_ruby_api)
2. The **Metadata** module is the module which actually send the request and get the response from API.
3. The **ECS** module is used to define the parameters if any, process the response and give the final output.

## 5.7 Explanation of each element

The Output of Ohai will be in the JSON Format contains the below element.

Name	Type	Description
<b>Public IP</b>	<i>String</i>	Public IP that associated with the Instance.
<b>Private IP</b>	<i>String</i>	Private IP that associated with the Instance.
<b>Instance ID</b>	<i>String</i>	Unique Id of an Instance.
<b>Instance Name</b>	<i>String</i>	Display name of the instance, which is a string of 2 to 128 Chinese or English characters. It must begin with an uppercase/lowercase letter or a Chinese character and can contain numerals, ".", "_", or "-". The instance name is displayed on the Alibaba Cloud console.
<b>CPU</b>	<i>string</i>	Displays the number of Core's.
<b>Memory</b>	<i>string</i>	Memory allocated to an instance.
<b>Bandwidth</b>	<i>String</i>	Bandwidth allocated to an instance
<b>IO Optimize</b>	<i>Boolean</i>	I/O optimized. Optional values are: <ul style="list-style-type: none"> <li>• False: no I/O Optimized</li> <li>• True: I/O Optimized</li> </ul> Default value: False
<b>Region</b>	<i>String</i>	ID of the region to which an instance belongs.
<b>Tag</b>	List	A list of hash/dictionaries of instance tags, '[{"tag_key": "value", "tag_value": "value"}]'

## 6 Deliverables

Deliverables Items	Knife	Kitchen	Ohai
KITE Gems	Yes	Yes	-NA-
User Guide	Yes	Yes	Yes
Github Readme	Yes	Yes	Yes
Test Cases	Yes	Yes	Yes
Rspec Unit Tests	Yes	Yes	Yes

## 7 Disclaimer

This **DRAFT** Specification is being forwarded to you strictly for informational purposes and sign-off requirement before development starts. This document covers functional and technical requirement of KITE and its plugin implementation with Chef for Aliyun (Alibaba) Cloud. The specification is "AS IS," "WITH ALL FAULTS" and Click2Cloud makes no warranties, and disclaims all warranties, express, implied, or statutory related to the specifications. THE CORPORATIONS ARE NOT LIABLE FOR ANY INCOMPLETENESS OR INACCURACIES. THE CORPORATIONS ARE NOT LIABLE FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES RELATING TO THE SPECIFICATIONS OR THEIR USE.

## 8 Appendix A: Glossary

No.	Initial Name	Description
1	CR	Change Request
2	FSD	Functional Specification Document
3	ECS	Elastic Compute Service
4	SG	Security Group

## 9 Appendix B: Document Change History and Sign-off

Version No.	Date	Name	Sign-Off by
V1.0	1/9/2017	Shuwei	Please answer some question
V1.1	1/10/2017	Shuwei	Shuwei sign-off

## 10 Point of Contact

Name	Designation	Email
Manish Mishra	Project Manager	<a href="mailto:manish@click2cloud.net">manish@click2cloud.net</a>
Kapil Thakkar	Sr. Lead Software Design Engineer	<a href="mailto:kapil@click2cloud.net">kapil@click2cloud.net</a>