

CHEF Automation for DSW Tools Installation

Internal Use only

Table of Contents

[1. Introduction: 3](#_Toc498393079)

[2. Overview: 3](#_Toc498393080)

[3. About Chef: 3](#_Toc498393081)

[4. Chef At Optum: 4](#_Toc498393082)

[4.1 Chef Architecture at Optum: 4](#_Toc498393083)

[4.2 Key Components of Appstack Cookbook: 5](#_Toc498393084)

[4.3 Dependencies for Appstack Cookbook: 6](#_Toc498393085)

[4.3.1 To setup a Git/Jenkins hook: 6](#_Toc498393086)

[5. Requests to Chefaas Team: 6](#_Toc498393087)

[5.1 Sample Pipeline request: 7](#_Toc498393088)

[5.2 Sample Manage Node Request 7](#_Toc498393089)

[6. Server dependencies before chef can be run on node: 8](#_Toc498393090)

[7. Managing Chef Appstack Cook Books: 8](#_Toc498393091)

[7.1 Tools Required for Develop Appstack cookbook: 8](#_Toc498393092)

[7.2 Tools to test cookbooks: 8](#_Toc498393093)

[7.3 Adding a new server as Chef Client: 9](#_Toc498393094)

[7.4 Current Appstack Cookbooks: 11](#_Toc498393095)

[7.5 Code Hub Repos: 11](#_Toc498393096)

[7.6 Jenkins : 11](#_Toc498393097)

[7.6.1 Jenkins URL’s for Appstack Cookbooks: 11](#_Toc498393098)

[7.7 Current Recipies: 12](#_Toc498393099)

[7.8 Current Hosts: 12](#_Toc498393100)

[7.8.1 Zone Abbreviations: 12](#_Toc498393101)

[8. Step By Step process of updating cookbooks: 12](#_Toc498393102)

[8.1 Updating Appstack Optum\_MapR Cook Book 12](#_Toc498393103)

[8.2 Updating Appstack Optum\_dsi\_asc Cook Book 16](#_Toc498393104)

[8.3 Repo Server: 21](#_Toc498393105)

[8.4 Point of Contacts: 21](#_Toc498393106)

[8.5 Chef at Optum Training Links: 21](#_Toc498393107)

[9. Knowledgebase: 21](#_Toc498393108)

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Purpose/Change | Author | Date |
| 1.0 | Initial Draft | Goutham Guduguntla | 10/24/2017 |
| 2.0 | Added Deployment Flow, and updated more details Managing Cookbook section | Goutham | 11/01/2017 |
| 3.0 | Updated the doc based on Comments from Dave(step by step process to update chef cookbbok) | Goutham | 11/06/2017 |

# Introduction:

The purpose of this document is to introduce Chef Automation for installing DSW tools by using Chef Appstack Cookbooks

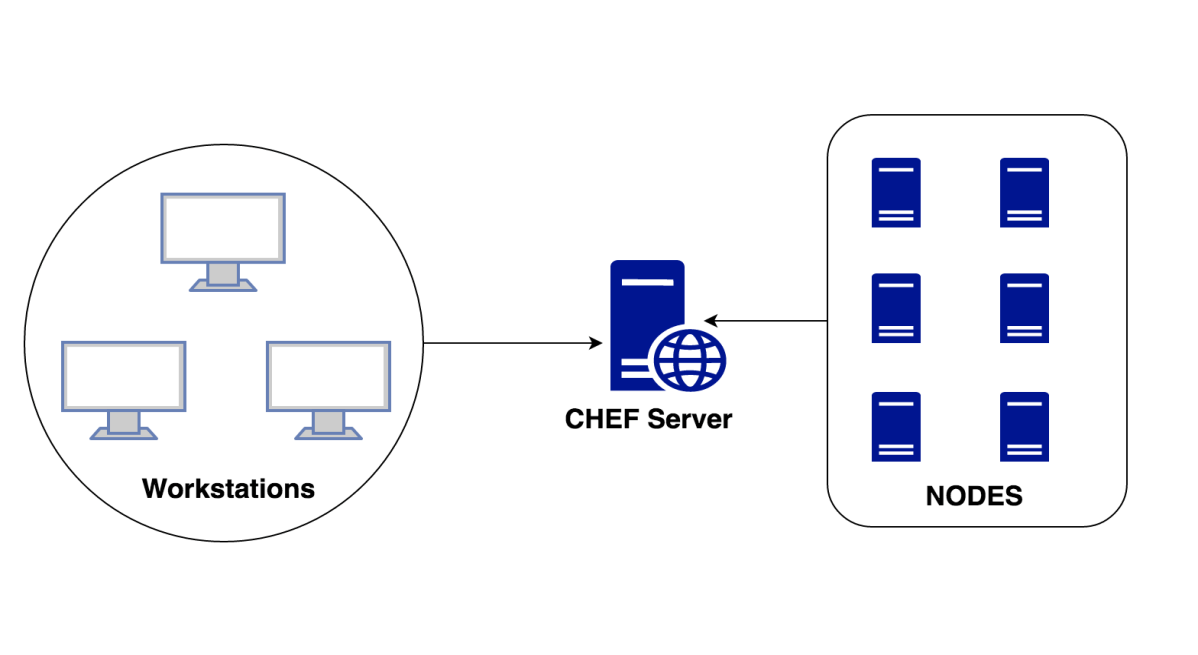
# Overview:

This Document covers the below topics

* About Chef
* Chef at Optum
* Key Components of Appstack Cookbooks
* Dependencies for Appstack Cookbooks
* Request to Chefaas team
* Managing Current Appstack Cookbooks
* Tools required to manage
* Tools required to test
* Adding a new server/pipeline
* Current Appstack Cookbooks
* Code Hub Repos
* Current Recipes
* Current list of Hosts
* Step by Step process to update Chef Cookbooks
* Repo Server Details /Chef Point of contacts

# About Chef:

Chef is an automation platform that “turns infrastructure into code,” allowing organizations or persons with large frameworks to generate a process that will save time and effort when making changes to part or all of their server. Chef works with three core components: The Chef server, workstations, and nodes



* **Chef Server:** This central server holds all your configuration data that the nodes will use for configuration.  In form of Cookbooks, Recipes, Roles, Environments and Attributes.
* **Workstation:** This machine holds all the configuration data that can later be pushed to the central chef server. Several chef command line utilities will be available in this system, which can be used to interact with nodes, update configurations etc. This is the place from which most of the work happens on a day to day basis.
* **Node**:This is nothing but a client server/system that will be registered to the central chef server, from where it can pull configuration data that needs to be applied. A node is any machine (physical, virtual, cloud, network device, etc) that is under management by Chef.

A **cookbook** is the fundamental unit of configuration and policy distribution. A cookbook defines a scenario and contains everything that is required to support that scenario:

* Recipes that specify the resources to use and the order in which they are to be applied
* Attribute values
* File distributions
* Templates

A **recipe** is the most fundamental configuration element within the organization. A recipe Is mostly a collection of resources, defined using patterns (resource names, attribute-value pairs, and actions); helper code is added around this using Ruby, when needed.

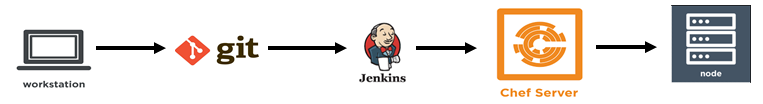
# Chef At Optum:

## Chef Architecture at Optum:

Chef at Optum is achieving the automation of infrastructure by using Appstack Cookbooks.

Appstack Cookbook is a top-level cookbooks used to bring in all the dependent cookbooks for an entire application or service. Its purpose is to consistently apply the same Infrastructure as Code across the environments it is applied to.

1. The Appstack Cookbook pattern is the pattern that the CHEFaaS team is supporting for the implementation of Chef at Optum
2. Standard Promotion Pipeline is used to get code through the CHEFaaS CICD pipeline
3. The individuals/team is responsible to create the Appstack Cookbook who are going to be managing the infrastructure for an application/server



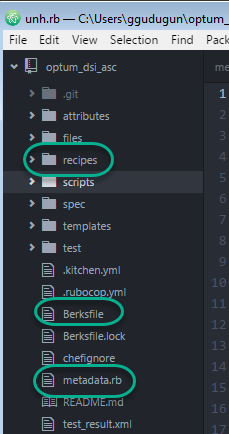
Chef Deployment Process Flow:

Architecture Overview at Optum:

1. Users are Individuals generates/updated their code using workstation(Chef DK) and commit their code using Git. (Note: Our team is responsible to edit these cookbooks)
2. When code commits through git, all my recipes get automagically tested in the Jenkins pipeline using various tools( Food Critic, Rubocop, Chef Spec, and Test Kitchen)
3. The pipeline pushes all my Appstack cookbook to the Chef server for me, and its ready to be deployed to appropriate SDLC environment when the chef-client runs on the individual nodes.

## Key Components of Appstack Cookbook:

* Berksfile
* Identifies the location of dependent cookbooks
* Metadata.rb
* Identify the dependent cookbooks to be used by the AppStack
* Specify the recipes that will be used as roles
* Recipe (Logical Role)
* Define what depended cookbooks & recipes get pulled in for a specific logical-role
* With the AppStack Pattern roles set by using an AppStack recipe as a single recipe for a nodes run list



## Dependencies for Appstack Cookbook:

1. Chef Appstack documentation to be reviewed before you start (Refer Section 8.5)
2. Chef Client needs to be installed on the requested nodes.
   1. Install the Chef Client on your node using the [ChefaaS Chef Binary Install Guide for Training](https://hubconnect.uhg.com/docs/DOC-58291)
3. Jenkins Standard Pipe line should be created ( This should be taken care by CHEFaaS team when we submit Standard Pipeline request) (Refer to section 5 and 5.1 for sample pipeline request)(Refer Section 7.6 more details about Jenkins)
4. Mange node request needs to be fulfilled by CHEFaaS team. And the Node needs to be added to the run list in chef Server. ( This should be taken care CHEFaaS team when we submit mange node request. ((Refer to section 5 and 5.2 for sample pipeline request)
5. users will need to setup a hook between Code hub and Jenkins to trigger their pipeline when updates are made to their code. (Refer section 4.3.1 for setting up Hooks)

### To setup a Git/Jenkins hook:

**Git hooks** are scripts that run automatically every time a particular event occurs in a Git repository. They let you customize Git’s internal behavior and trigger customizable actions at key points in the development life cycle

1.       Go to your repo and click settings

a.       **Link to your Git**

**Repo:** https://github.optum.com/DSI/optum\_mapr/settings/hooks

2.       On the right side of the screen select ‘settings’

3.       From the left side select hooks

4.       Then ‘Add Hooks’

5.       Select the type of hooks you are setting up (Notify Jenkins) hook

6.       Fill in the data inputs supplied below

7.       Click Create Hook

8.       Click the Test button (*If Successful it should reply back green*)

**Hook Inputs:**

* + Title = mapr\_cookbook Pipeline Hook
  + Branches = master
  + Jenkins URL = <https://jenkins.optum.com/chefaastraining/>
  + Jenkins Job Name =  mapr\_cookbook/job/mapr\_cookbook\_pipeline\_ci/
  + Jenkins token =  ggudugun

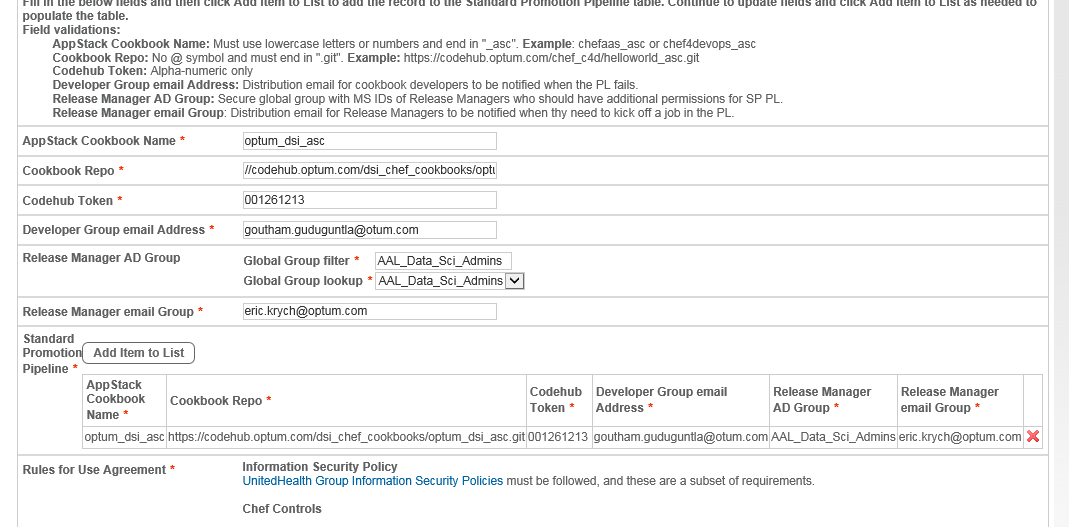
# Requests to Chefaas Team:

* The ChefaaS Enterprise Service Catalog (ESC) offering is used to collect data relating to onboarding – including the creation of the node environment. This information is important to aid ChefaaS in creating the pipeline on your behalf.

Link to the ESC: [**https://servicecatalog.uhc.com/sc/default.aspx**](https://servicecatalog.uhc.com/sc/default.aspx)

* Search for ‘Chef’ in the search bar
* Select ‘Chef as a Service’
* Select ‘Manage Nodes’ under Type of Request
* Provide all required fields
* Description of the request
* Server Name
* Environment
* AppStack cookbook name
* Recipe Name

## Sample Pipeline request:



## Sample Manage Node Request

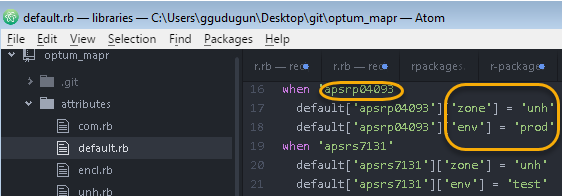
* Please refer to section 7.3 for list of cookbooks. And the recipe name is “default.rb”
* Refer section 8.2 for Chef contact details, if don’t get response for 1 day on your requests.
* After completion of request goto section 8

Once Mange node request completed follow the the below steps to deploy the code into new host:

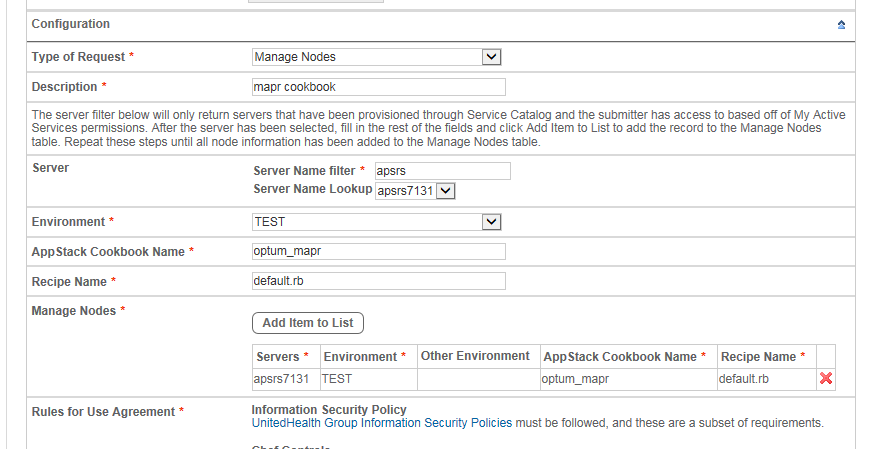
1. Get sudo access to the host to perform required validations
2. Make sure clef client installed on requested host
3. Add the new host details in requested appstack cookbook based on env(refer section 7.4 for current cookbooks).

Below are the navifations:

Appstack cookbook🡪 Attributes 🡪 default.rb 🡪add the host name, zone and env as shown below



1. Save the changes
2. Update the metadata.rb ( refer section 8.1 for how to update cookbook details which applicable for all below steps)
3. Commit the changes
4. Make sure code pushed to chef server
5. Run the chef-client on the host



# Server dependencies before chef can be run on node:

<http://ss.optum.com/sites/transactions/aal/SA/Collateral/Big_Data_Cloud_Strategy/Big_Data_Science_Spt/DSW/09_Adminstration/Support%20Docs/ServerDeploymentCheckList_template.xlsx>

# Managing Chef Appstack Cook Books:

## Tools Required for Develop Appstack cookbook:

1. **ChefDK:**

Chef DK is a package that contains all the development tools you will need when coding Chef. It combines the best of the breed tools developed by Chef community with Chef Client.

1. **GIT Bash**

Users will use GitBash to push cookbook code from their local workstations to their Codehub repositorios

1. **Atom**

Atom is a text editor, The ChefaaS team recommends [Atom](http://appstore.uhc.com/) due to its compatibility with Chef and robust plugin capabilities. Users can opt to install plugins that will check their code for Rubocop and Foodcritic errors

1. **SSH Client software**

Users can have any SSH client to run or test thier cookbook execution on the nodes

## Tools to test cookbooks:

1. **Rubocop**

**RuboCop** is a Ruby static code analyzer to ensure style conventions and best practices and evaluating the code in a cookbook against metrics like “line length” and “function size”

1. **Foodcritic**

Foodcritic is a static linting tool that analyzes all of the Ruby code that is authored in a cookbook against a number of rules, and then returns a list of violations

1. **Test Kitchen**

Test Kitchen is an integration tool for developing and testing infrastructure code and software on isolated target platforms.

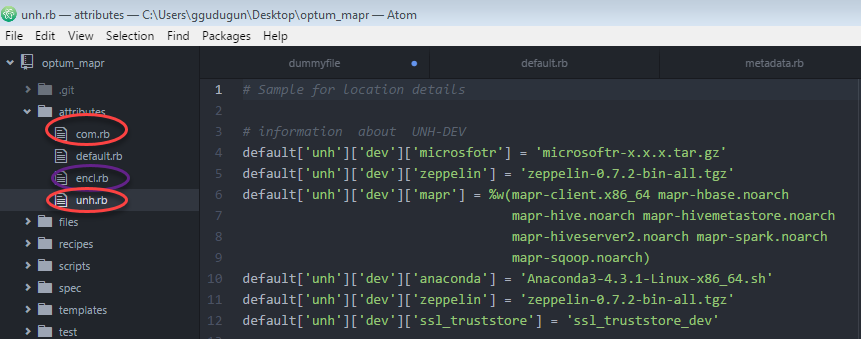
1. **Chef Spec**

ChefSpec is a framework that tests resources and recipes as part of a simulated chef-client run.

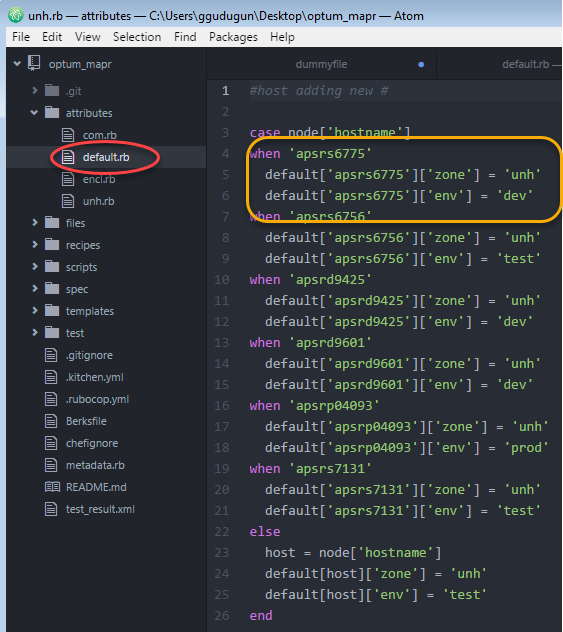
## Adding a new server as Chef Client:

When a new Host required to be deployed using Chef automation Below details needs to be added to Appstack Cook Books.

Note: Decide the zone before you add the host to Cookbook.



* Have the list of versions and make sure the software/packages are available at remote server. ( Refer section 8.3 for Repo server details)
* Navigate to desired cookbook (optum\_mapr or optum\_dsi\_asc)
* Under the cookbook, navigate to attributes folder default.rb and list the new host in case statement.



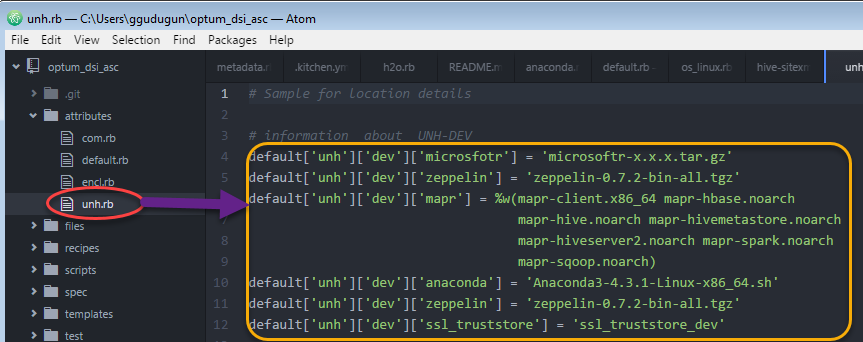
Ex: when 'apsrs6775'

default['apsrs6775']['zone'] = 'unh'

note: zone abbreviations

default['apsrs6775']['env'] = 'dev'

* Under the cookbook, navigate to Attributes Folder
* Based on the zone, provide the list of packages/versions in respected file.



Ex: UNH Dev Host file needs to be updataed as below in unh.rb file.

default['unh']['dev']['microsfotr'] = 'microsoftr-x.x.x.tar.gz'

note:screenshot

## Current Appstack Cookbooks:

Optum\_mapr (Testing Queue)

Optum\_dsi\_asc (Production Queue)

optum\_dsi\_dsw\_dev\_stg\_asc (Dev/Stage)

## Git Hub Repos:

Production

**https://github.optum.com/DSI/optum\_dsi\_asc**

Dev/Stage

**https://github.optum.com/DSI/optum\_dsi\_dsw\_dev\_stg\_asc**

Test

[**https://github.optum.com/DSI/optum\_mapr**](https://github.optum.com/DSI/optum_mapr)

## Jenkins :

Jenkins is a self-contained, open source automation server which can be used to automate all sorts of tasks related to building, testing, and deploying software.

### Jenkins URL’s for Appstack Cookbooks:

**Test/Dev Pipeline URL:**

[**https://jenkins.optum.com/chefaastraining/job/mapr\_cookbook/**](https://jenkins.optum.com/chefaastraining/job/mapr_cookbook/)

**Prod Pipeline URL:**

[**https://jenkins.optum.com/chefaas/job/optum\_dsi\_asc/**](https://jenkins.optum.com/chefaas/job/optum_dsi_asc/)

**Dev/Stage Pipeline URL:**

## Current Recipies:

|  |  |
| --- | --- |
| **Recipies** | **Code Components** |
| mapr | Env File, Mapr Packages, Mapr symlinks, Java, Ssl files, site.xmls |
| r | Microsoft R, configuration files |
| rstudio | r studio, rstudio server, configuration files and service enable and strat |
| anaconda | Anconda, node js,jupyterhub,pycharm, pyspark, configuration files and  Service enable and start |
| drivers | Nz, oracle, ojdbc, teradata, nifi, sqlsvr,skel, process moniter and other  configuration files |
| zeppelin | Zeppelin Instation and service enable and start |
| Weka | Weka |
| h2o | h2o and sparking water |
| Openrefine | Open refine, service start and enable |
| os\_linux | Linux Ov, Bare OS, Docker, cluster shell, Julia |

## Current Hosts:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Hosts** | **Zone** | **Env** | **OS Version** |
| 1 | apsrd9425 | unh | Dev | release 7.3 |
| 2 | apsrd9601 | unh | Dev | release 7.3 |
| 3 | apsrp04093 | unh | Prod | release 6.9 |
| 4 | apsrp04067 | unh | STG | release 6.9 |
| 5 | apsrp04068 | unh | STG | release 6.9 |
| 6 | apsrs7131 | unh | Test | release 6.9 |
| 7 | apsrs6775 | unh | Dev | release 7.3 |
| 8 | apsrs6756 | unh | Test | release 7.3 |

### Zone Abbreviations:

UNH : United

COM : Commercial

ENCL : Commercial Enclave

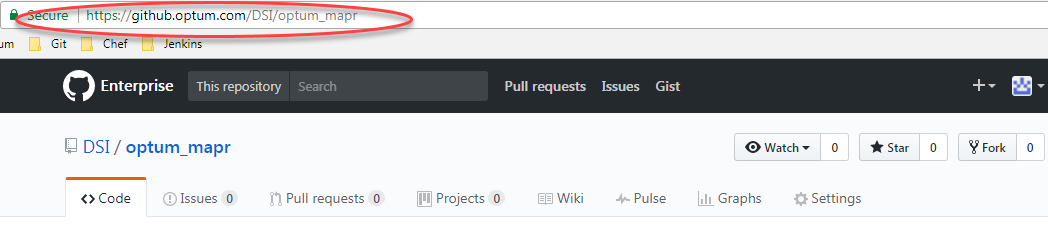
<http://ss.optum.com/sites/transactions/aal/SA/Collateral/Big_Data_Cloud_Strategy/Big_Data_Science_Spt/Support/Big%20Data%20Science%20Platform%20Architecture.vsd>

# Step By Step process of updating cookbooks:

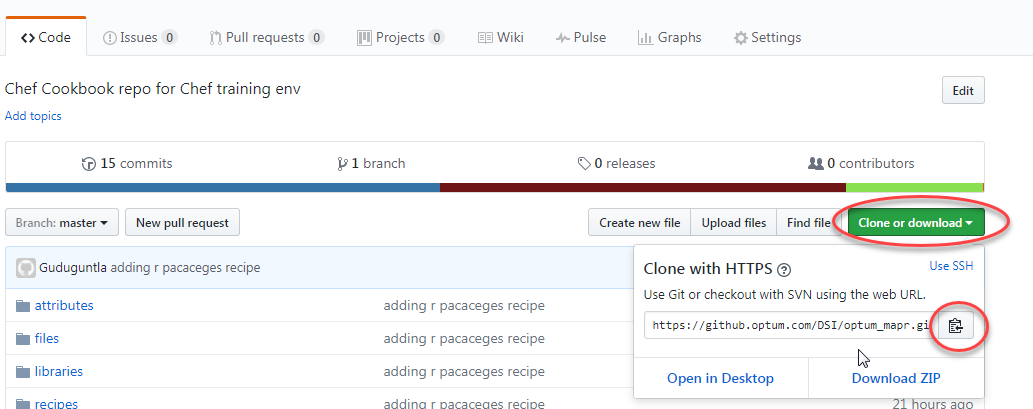
## Updating Appstack Optum\_MapR Cook Book

* Clone the Git Repos to your local desktop (ex: GitBash)

[**https://github.optum.com/DSI/optum\_mapr**](https://github.optum.com/DSI/optum_mapr)



* Click on Clone and copy to click board the url location

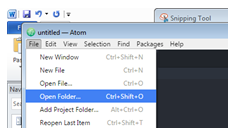


* Clone your Cookbook to local Desktop using Git Bash Tool

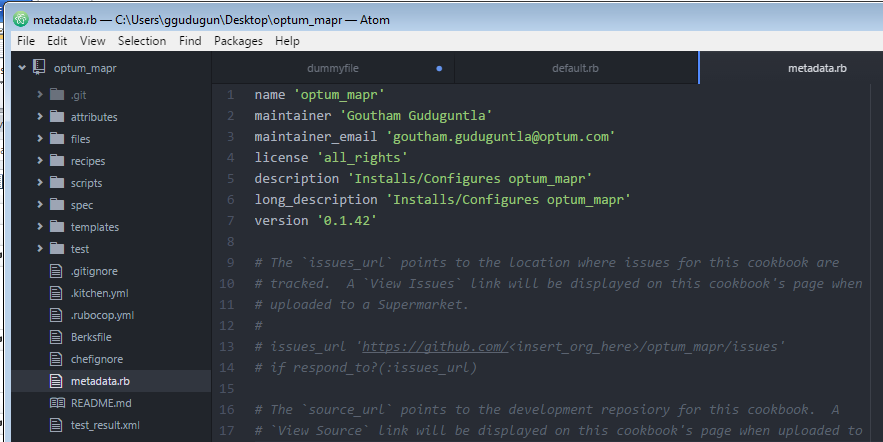


* Open the cookbooks using any note book (ex: Atom)

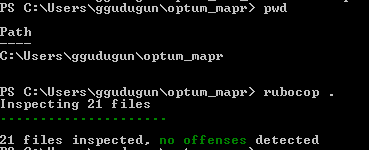
Navigation 🡪 Launch your Atom Tool 🡪 File 🡪 Open Folder 🡪 Choose the cookbook folder where you have cloned in your previous location.



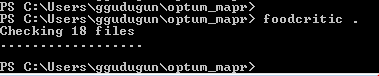
* Update recipes/files or Attribute bases on your requirements and save



* Update the metdata file version number for each commit.(Imp step) Please validate above screenshot.
* Test your cookbook for any errors using chef dk using below commands
* Rubocop .



* Foodcritic .



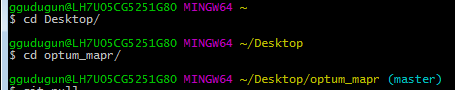
* Chef exec rspec





* Commit the changes using Git Bash, please follow below steps for best practices

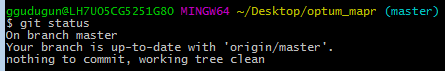
Navigate to cookbook location in your GitBash



1. Git Pull



1. Git Status



1. Git Commit



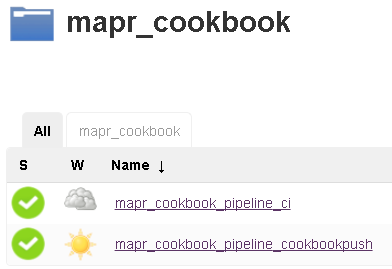
1. Git Push



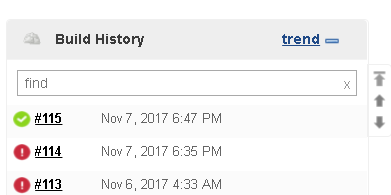
* Once after Push Navigate to your Jenkins URL’s based on Env and validate your build

( below Ex: to validate in Test cookbook)

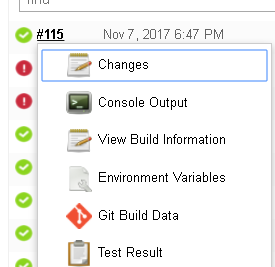
* Click on the link: <https://jenkins.optum.com/chefaastraining/job/mapr_cookbook/>
* User will see below links with Status(S) as Success 



* Click on mapr\_cookbook\_pipeline\_ci
* User will see a Build History on left hand side of screen as below. (115 is succeeded build and 114 and 113 are failed builds in below image)

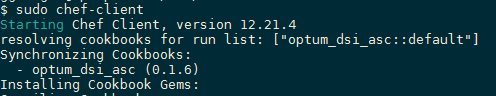


* Highlight your cursor on build and select console output for detailed execution.



* After successful build your Cookbook will be uploaded to Chef Server.
* Run your cookbook on the server to test you changes using below command. For immediate changes or wait for next scheduled chef-client job( Its scheduled for every 30 mins)

Sudo chef-client



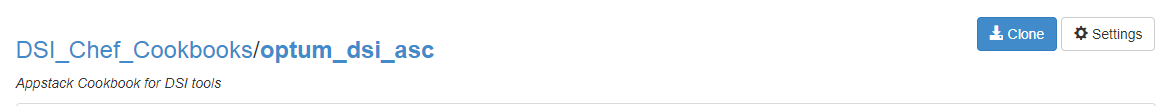
* After successful execution your chef-client will update all your resources in you node.



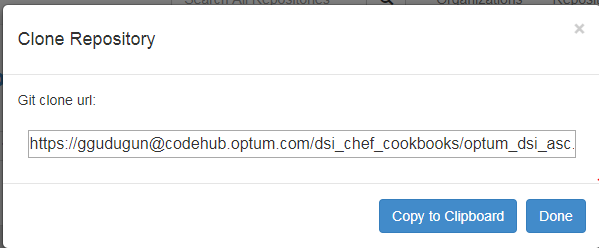
## Updating Appstack Optum\_dsi\_asc Cook Book

* Clone the Git Repos to your local desktop (ex: GitBash)

**https://github.optum.com/DSI/optum\_dsi\_asc**



* Click on Clone and copy to click board the url location

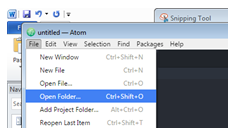


* Clone your Cookbook to local Desktop using Git Bash Tool

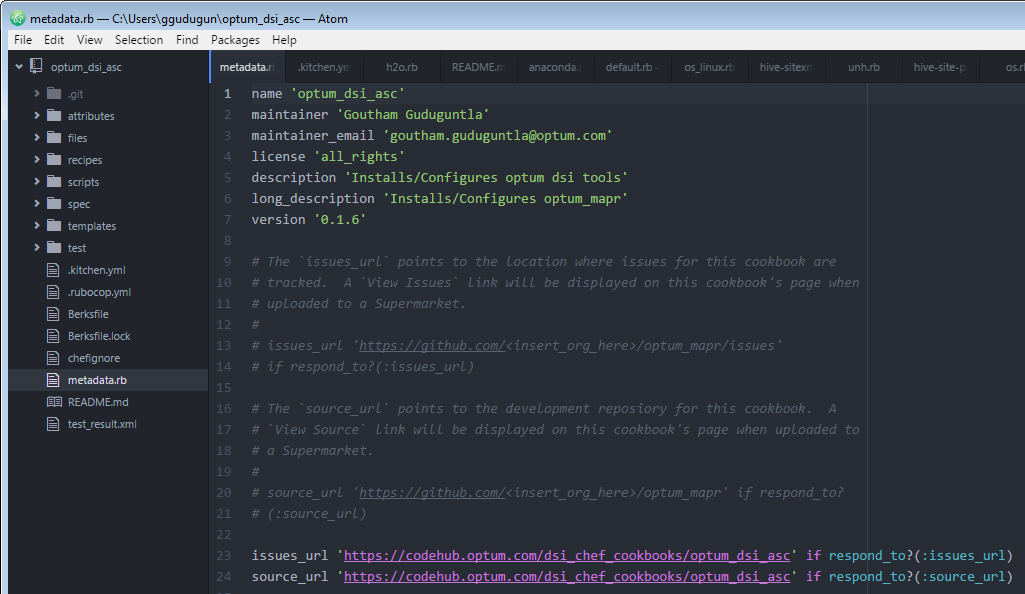


* Open the cookbooks using any note book (ex: Atom)

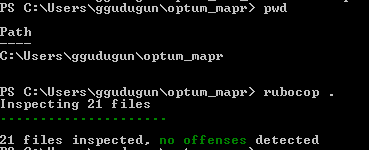
Navigation 🡪 Launch your Atom Tool 🡪 File 🡪 Open Folder 🡪 Choose the cookbook folder where you have cloned in your previous location.



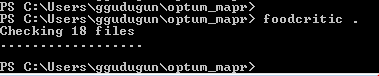
* Update recipes/files or Attribute bases on your requirements and save



* Update the metdata file version number for each commit.(Imp step) Please validate above screenshot.
* Test your cookbook for any errors using chef dk using below commands
* Rubocop .



* Foodcritic .



* Chef exec rspec



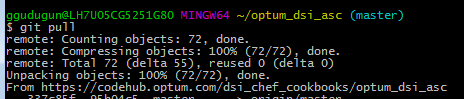


* Commit the changes using Git Bash, please follow below steps for best prctices

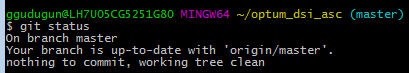
Navigate to cookbook location in your Gitbash



* 1. Git Pull



* 1. Git Status



* 1. Git Commit



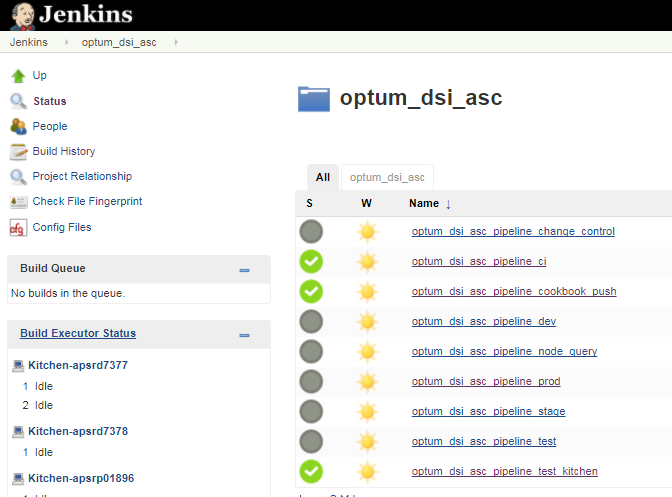
* 1. Git Push



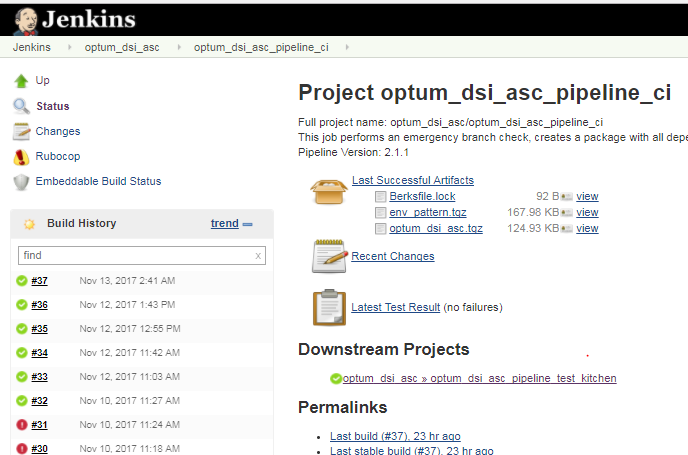
* Once after Push Navigate to your Jenkins URL’s based on Env and validate your build

( below Ex: to validate in Test cookbook)

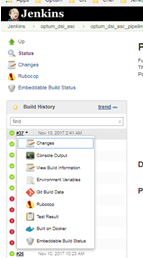
* Click on the link: <https://jenkins.optum.com/chefaas/job/optum_dsi_asc/>
* User will see below links with Status(S) as Success 



* Click on optum\_dsc\_asi\_ci
* User will see a Build History on left hand side of screen as below. (37 is succeeded build and 31 and 30 are failed builds in below image)



* Highlight your cursor on build and select console output for detailed execution.



* After successful build your Cookbook will be uploaded to Chef Server using optum\_dsi\_asc\_pipeline\_cookbook\_push

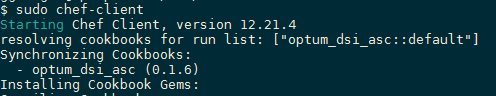


* For Production Cookbooks Its mandatory to succeed optum\_dsi\_asc\_pipeline\_test\_kitchen (during this step cookbook will be tested using vagrant instance)



* Run your cookbook on the server to test you changes using below command. For immediate changes or wait for next scheduled chef-client job( Its scheduled for every 30 mins)

Sudo chef-client



* After successful execution your chef-client will update all your resources in you node.



## Repo Server:

[**http://apsrs6756.uhc.com/softwares/**](http://apsrs6756.uhc.com/softwares/)

## Point of Contacts:

[**chef\_coe@optum.com**](mailto:chef_coe@optum.com)

**Kuthyar, Mihir** [**mihir.kuthyar@optum.com**](mailto:mihir.kuthyar@optum.com)

**Scalora, Salvatore A** [**salvatore\_scalora@optum.com**](mailto:salvatore_scalora@optum.com)

**Mandal, Suvadip <mandal\_suvadip@optum.com>**

## Chef at Optum Training Links:

Here are a couple of links for resources regarding the AppStack cookbook /Jenkin Pipelines and Git.

[Chefaas AppStack Cookbook Presentation](https://hubconnect.uhg.com/docs/DOC-68735)

[CHEFaaS AppStack Cookbook Workshop](https://hubconnect.uhg.com/docs/DOC-68084)

[ChefaaS Training Environment Onboarding](https://oneconnect.uhg.com/docs/DOC-52154)

[ChefaaS Training (Simplified) Pipeline View](https://hubconnect.uhg.com/docs/DOC-90702)

[ChefaaS FAQs](https://oneconnect.uhg.com/docs/DOC-49772)

[ChefaaS Rules for Use](https://oneconnect.uhg.com/docs/DOC-68774)

[Chefaas Pipeline User Guide](https://hubconnect.uhg.com/docs/DOC-68773)

[ChefaaS Production Pipeline View](http://it-sdd.uhc.com/sites/IES/Projects/IES15_Flexible_Self_Service_Provisioning_CDA_(FSSP_D)/Document%20Library/C4D%20Public%20Documents/CHEFaaS%20Production%20Pipelines.pptx)

[ChefaaS @ Optum Onboarding Guide for Prod](https://hubconnect.uhg.com/docs/DOC-68772)

[Git Quick Start Guide](https://hubconnect.uhg.com/docs/DOC-57013)

# Installing and Updating R packages

## R Packages Installation

Appstack cookbooks should have below Recipes, Resources and libraries to Install the R packages on required hosts by using input json file.

**Rpackages.rb**: This recipe is used to install required dependencies for installing r packages.

And also it schedules the crontab to run the chef-client local by using the input json file.

And also this recipe check the status of packages by calling the default.rb libraries

**r-packages-install.rb:** This recipe is only upload to chef server and which calls the package\_install using package\_install method to install R packages.

**Package\_install.rb**: This resource is used to perform certain actions( install/upgrade and remove) by using the R commands provided in it.

## Installing the New R package

1. Select your app stack cook book where the host Present based on your env.
2. Navigate to Files and update the file r-packages\_300.json file with new package name and the version as below format

"abind": "1.4-5" (“package name”: “version”)

Note: Do not ignore the “:” and space between the package name and version.

1. Make sure these packages are available in repo server.\*refer 8.3 for repo server details)

<http://apsrs6756.uhc.com/R/src/contrib/>

1. Update the metadata file and commit the changes.
2. After the scheduled cron runs the scheduled chef-client(chef-client -o recipe[optum\_mapr::r-packages-install] -j /opt/r\_packages\_300.json) package will be installed during scheduled window.
3. In case of failure, you will get notified through the email that package is not installed.

# Knowledgebase:

* How to write recipes (best practices, etc)

<https://blog.chef.io/2017/09/28/chef-101-the-road-to-best-practices/>

* High level overview

<https://docs.chef.io/chef_overview.html>

* Getting user setup (installing, configuring the applications and where to install them?)

All Below tools are available in Appastore <http://appstore.uhc.com/Home/Index>

Users required have below tools on their local desktop

* GitBash
* Atom
* ChefDK
* Linux Shell
* Server dependencies before chef can be run on node
* For all deployments:
* 64-bit architecture
* For a standalone deployment:
* 4 total cores (physical or virtual)
* 8 GB of RAM or more
* 5 GB of free disk space in /opt
* 5 GB of free disk space in /var
* Chef commands.

Chef generator commands

Usage: chef generate GENERATOR [options]

Available generators:

app Generate an application repo

cookbook Generate a single cookbook

recipe Generate a new recipe

attribute Generate an attributes file

template Generate a file template

file Generate a cookbook file

helpers Generate a cookbook helper file in libraries

lwrp Generate a custom resource

resource Generate a custom resource

repo Generate a Chef code repository

policyfile Generate a Policyfile for use with the install/push commands

generator Copy ChefDK's generator cookbook so you can customize it

build-cookbook Generate a build cookbook for use with Delivery

-h, --help Show this message

-v, --version Show chef version

Chef execution commands:

chef-client Run chef client mode

chef-client -z -o recipe[cookbook] Run chef in local mode

chef-solo -c solo.rb -j node.json Run chef in chef solo mode

For more chef and knife commands:

<https://docs.chef.io/ctl_chef.html>

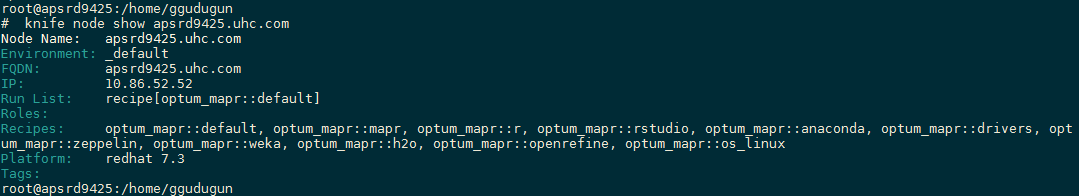
<https://docs.chef.io/knife.html>

* Chef client is always running on the nodes, can you highlight and example is more

Unfortunately we don’t have control this. We need to work with CHEFaaS team to remove the recipes from run list

Where as in our test instance we have access to Knife commands, we can remove run list from node.

|  |
| --- |
| knife node apsrd9425.uhc.com run\_list add recipe[optum\_mapr::default] |
| knife node apsrd9425.uhc.com run\_list remove recipe[optum\_mapr::default] |



* Is there any Reporting?

The Chef management console provides ways for users to configure the timeframe around which a report is built, and then to review the reports that are available for that timeframe.

Use Reporting to keep track of what happens during the execution of chef-client runs across all of the machines that are under management by Chef