

CHEF Automation for DSW Tools Installation

Internal Use only

Table of Contents

[1. Intorduction: 3](#_Toc497089452)

[2. OverView: 3](#_Toc497089453)

[3. About Chef: 3](#_Toc497089454)

[4. Chef AT Optum: 4](#_Toc497089455)

[Chef Architecture at Optum: 4](#_Toc497089456)

[Key Components of Appstack Cookbook: 4](#_Toc497089457)

[Dependicies for Appstack Cookbook: 5](#_Toc497089458)

[5. Requests to Chefaas Team: 5](#_Toc497089459)

[Sample Pipeline request: 6](#_Toc497089460)

[Sample Manage Node Request 6](#_Toc497089461)

[6. Managing Chef Appastack Cook Books: 7](#_Toc497089462)

[Tools Required for Develop Appstack cookbook: 7](#_Toc497089463)

[Tools to test cookbooks: 7](#_Toc497089464)

[Current Appstack Cookbooks: 7](#_Toc497089465)

[Code Hub Repos: 7](#_Toc497089466)

[Current Recipies: 7](#_Toc497089467)

[Current Hosts: 8](#_Toc497089468)

[Repo Server: 8](#_Toc497089469)

[Point of Contacts: 8](#_Toc497089470)

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
|  |  |  |  |

# Intorduction:

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

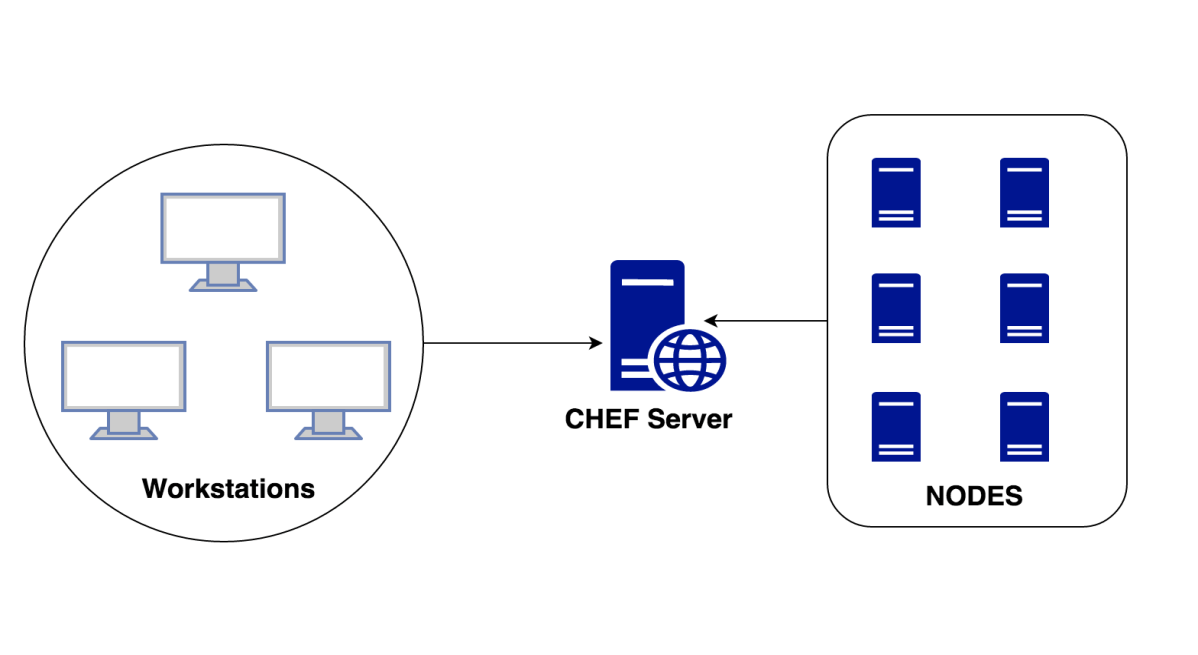
# OverView:

This Document covers the below topics

* About Chef
* Chef at Optum
* Key Components of Appstack Cookbooks
* Dependenicies for Appstack Cookbooks
* Request to Chefaas team
* Manging Current Appstack Cookbooks
* Tools required to manage
* Tools required to test
* Adding a new server
* Current Appstack Cookbooks
* Code Hub Repos
* Current Recipies
* Current list of Hosts
* 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 configuration data that the nodes will use for configuration.
* **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 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

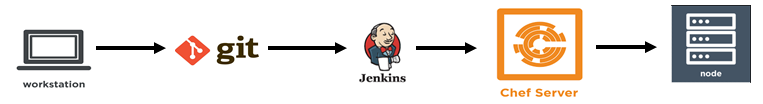
# Chef At Optum:

## Chef Architecture at Optum:

Chef at Optum is achiving 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.

* The AppStack Cookbook pattern is the pattern that the CHEFaaS team is supporting for the implementation of Chef at Optum
* Standard Promotion Pipeline is used to get code through the CHEFaaS CICD pipeline
* 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:



* 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)
* 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)
* 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

## Dependicies for Appstack Cookbook:

* Chef Client needs to be installed on the requested nodes.
* Mange node request needs to be fulfilled by CHEFaaS team. And Node is needs to update with the runlist in chef Server. ( this should be taken care CHEFaaS team when we submit mangenode request )
* Jenkins Standard Pipe line should be created ( This should be taken care by CHEFaaS team when we submit Satandard Pipeline request)
* users will need to setup a hook between Codehub and Jenkins to trigger their pipeline when updates are made to their code.

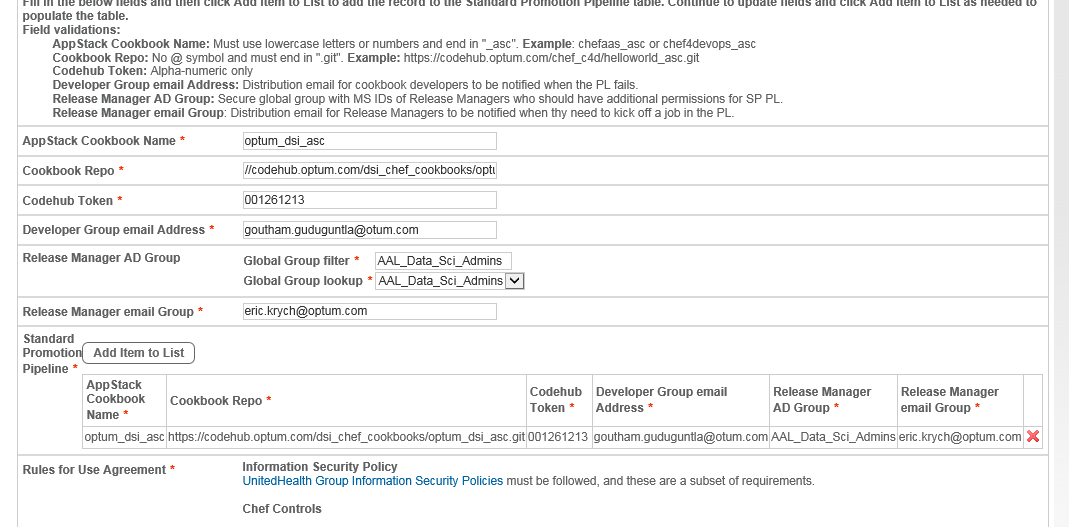
# Requests to Chefaas Team:

* 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.

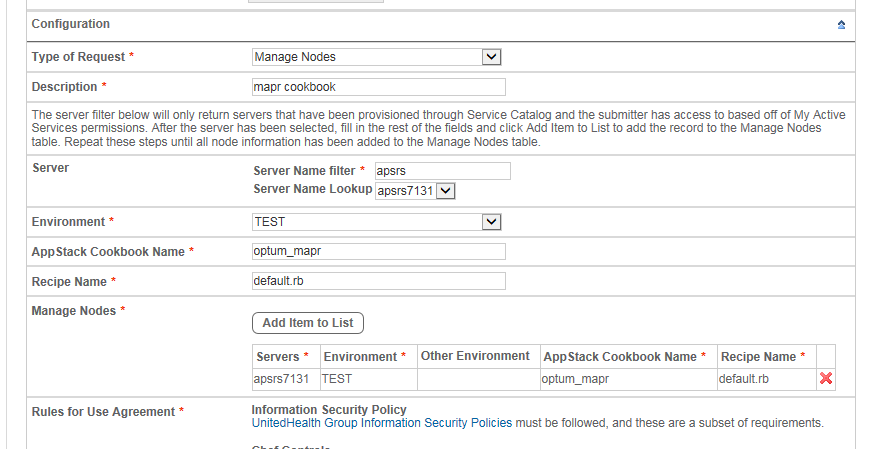
Link to the ESC: [**https://servicecatalog.uhc.com/sc/default.aspx**](https://servicecatalog.uhc.com/sc/default.aspx)

* 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



# Managing Chef Appastack 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 Git to push cookbook code from their local workstations to their Codehub repositories

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: Deside 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.

* Navigate to desired cookbook (optum\_mapr or optum\_dsi\_asc)
* Under the cookbook, navigate to default.rb and list the new host in case statement.

Ex: when 'apsrs6775'

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

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'

## Current Appstack Cookbooks:

Optum\_mapr

Optum\_dsi\_asc

## Code Hub Repos:

[**https://codehub.optum.com/dsi\_chef\_cookbooks/optum\_dsi\_asc.git**](https://codehub.optum.com/dsi_chef_cookbooks/optum_dsi_asc.git)

[**https://codehub.optum.com/dsi\_chef\_cookbooks/optum\_mapr.git**](https://codehub.optum.com/dsi_chef_cookbooks/optum_mapr.git)

## Jenkins URL’s:

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

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

## 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 | apsrs7131 | unh | Test | release 6.9 |
| 5 | apsrs6775 | unh | Dev | release 7.3 |
| 6 | apsrs6756 | unh | Test | release 7.3 |

## Repo Server:

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

## Point of Contacts:

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

## Chef at optum Training Links:

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

<https://hubconnect.uhg.com/docs/DOC-68735>

<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)