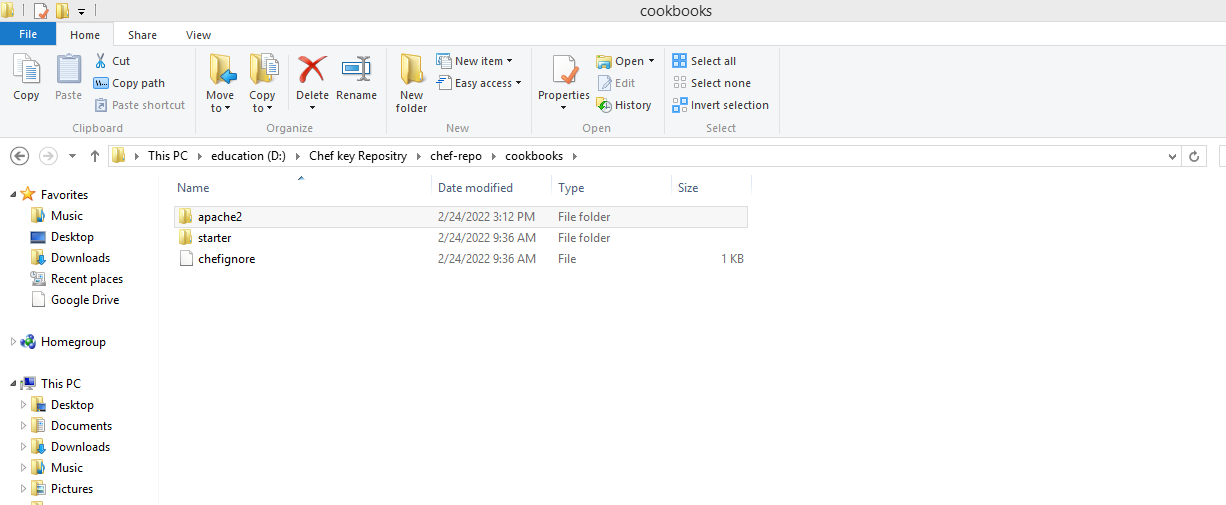
**CHEF CLASS NOTE BOOK** by Khaja

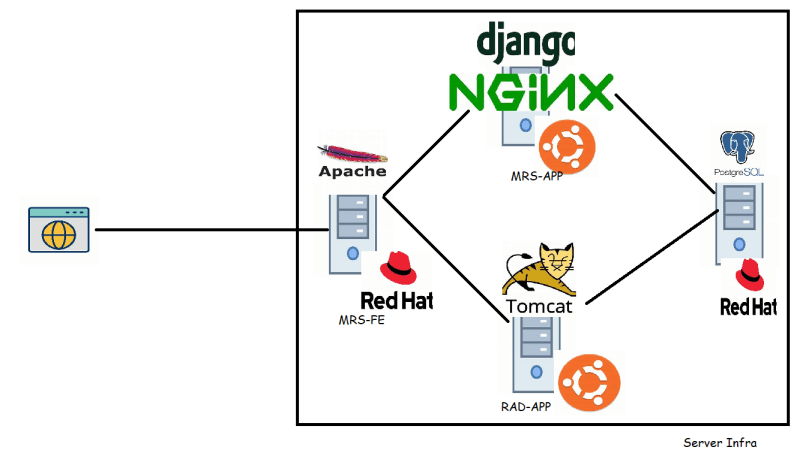
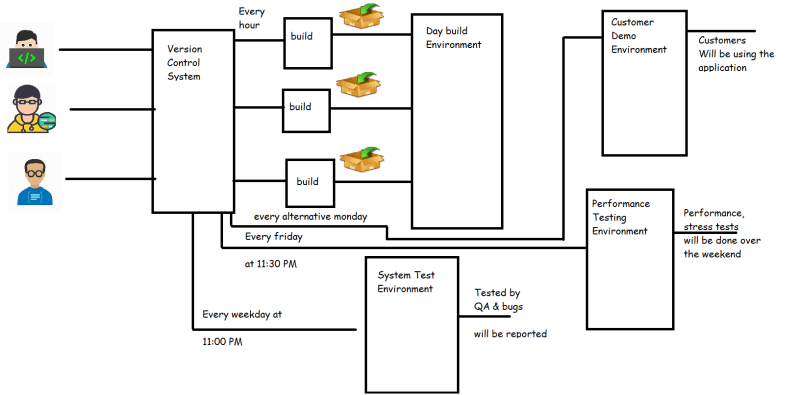
New chef repo



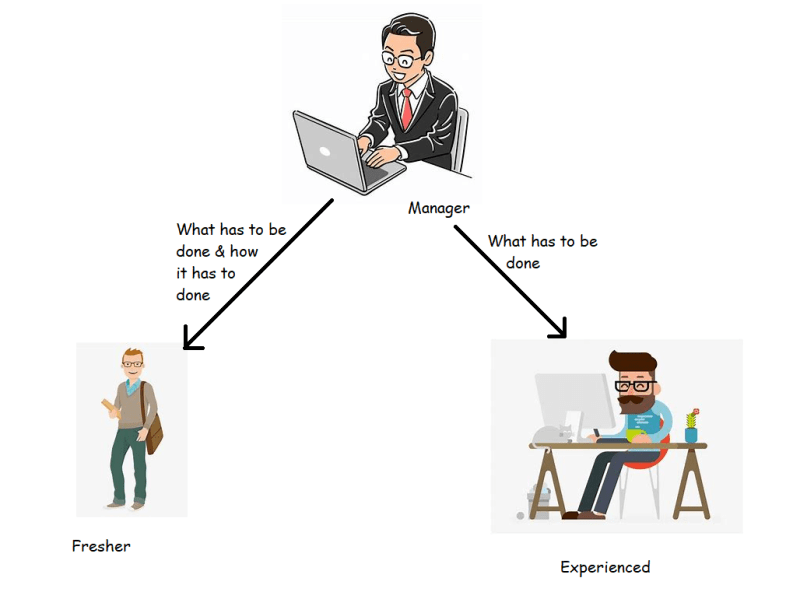
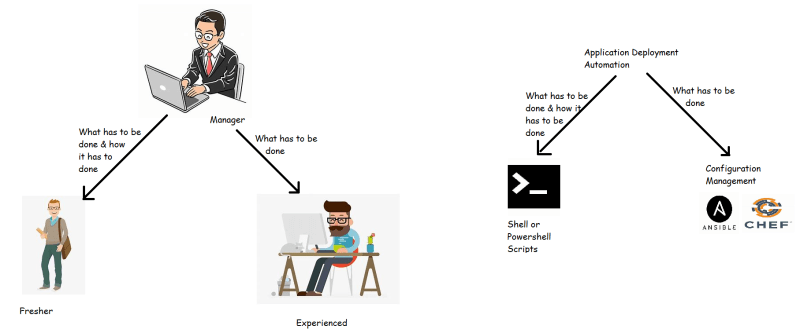
OCTOBER 26, 2021

DevOps Classroom Series – 26/Oct/2021

Scenario of an Organization (LT)

* LT is working on a project towards Health Care Software MRS
* The architecture of the project 
* To work on this project we have 3 developer teams
  + Front end
  + django/mrs developers
  + java/radiology developers
* We also have a
  + QA Team
* LT according to their QA policy wants to build
  + Day build environment (Component test)
  + System Test (Nightly build)
  + Performance test (Weekly build)
  + Customer Demo (Release build) 
* So doing the deployment of the applications manually is no longer the option.
* LT already has servers available for all the environments, We need a solution to deploy the applications and/or configure the applications so that stake holders can use it.

Possible Solutions:

* Lets take the following analogy 
* Now lets apply the same to scripting vs configuration management 
* Example:
  + Install apache server on ubuntu
    - shell/bash: sudo apt update && sudo apt install apache2 -y
    - configuration management: we will be writing in some syntax which implies Ensure apache2 is installed
  + Create a file at /usr/share/ltmrs/info.config
    - shell/bash: touch /usr/share/ltmrs/info.config or echo "" > /usr/share/ltmrs/info.config
    - configuration management: we will be writing in some syntax which implies Ensure file is present at /usr/share/ltmrs/info.config
* Configuration management are designed to be idempotent. Idempotance is a property which ensures you will get the same result irrespective of how many time you executed the script.
* If we use shell/Powershell scripts the developer of the script needs to write conditional statements almost for every step to ensure idempotance which becomes difficult to read & maintain.
* With Configuration management tools we use declarative approach to automate where we will mention our desired state (Ensure apache2 is installed), whereas in shell scripts we follow procedural approach.

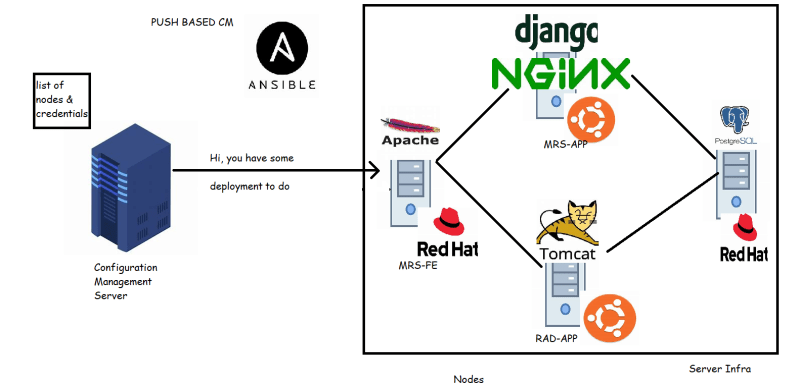
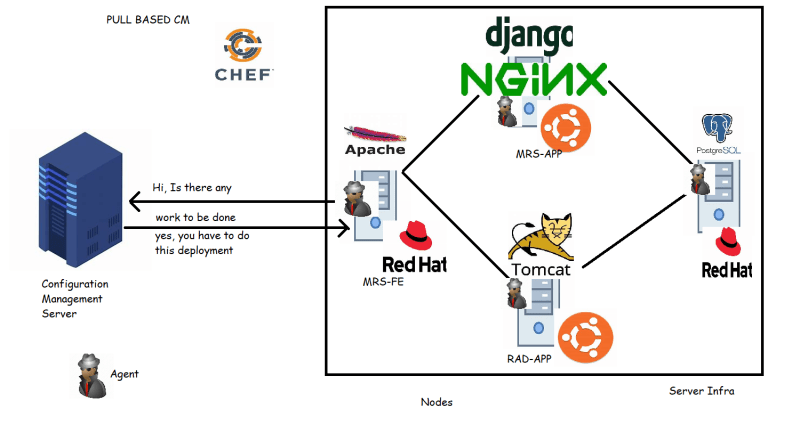
Exercise:

1. Azure Free tier Account Creation [Refer Here](https://www.youtube.com/watch?v=MdDOc9OPVDA&list=PLuVH8Jaq3mLuqXuGs6aeqxhuvCYSzB1kT)
2. AWS Free tier Account Creation [Refer Here](https://www.youtube.com/watch?v=z95MhW1gAcA&list=PLuVH8Jaq3mLszrC7lv68a0VcrDripW-HK)
3. Necessary Software Installation [Refer Here](https://www.youtube.com/watch?v=mRILfUNbsIo&list=PLuVH8Jaq3mLud3sVDvJ-gJ__0zd15wGDd&index=14)
4. For Windows 10 and above configure Windows Terminal [Refer Here](https://www.youtube.com/watch?v=qLVn2EvPsYc&list=PLuVH8Jaq3mLud3sVDvJ-gJ__0zd15wGDd&index=11)
5. Create an account in manage.chef.io [Refer Here](https://manage.chef.io/signup)

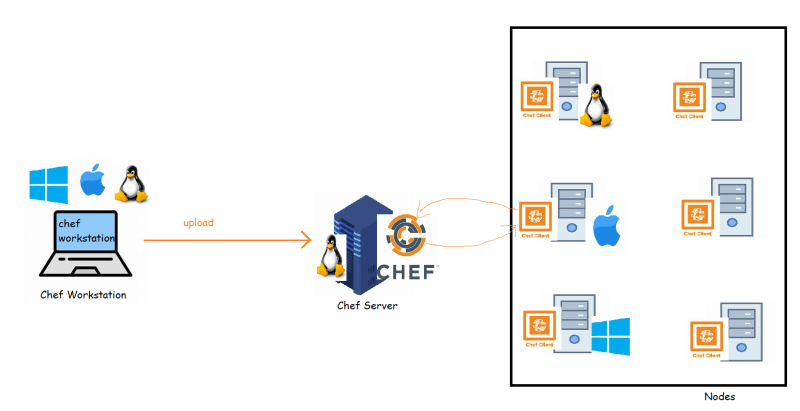
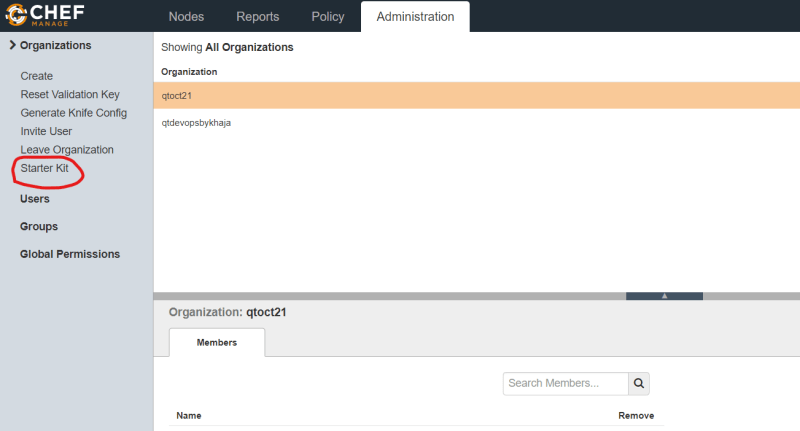
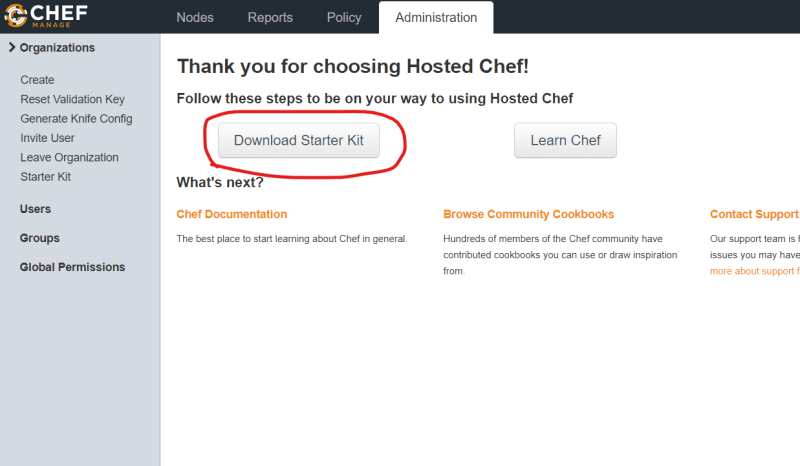
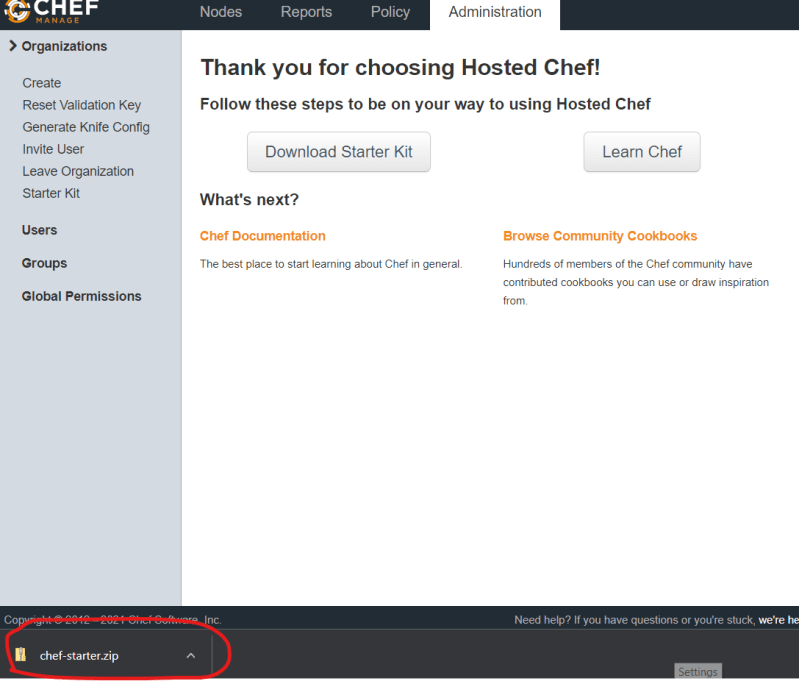
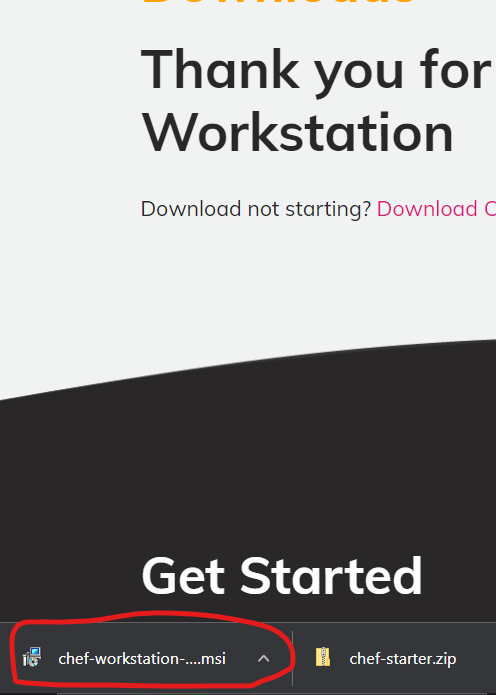
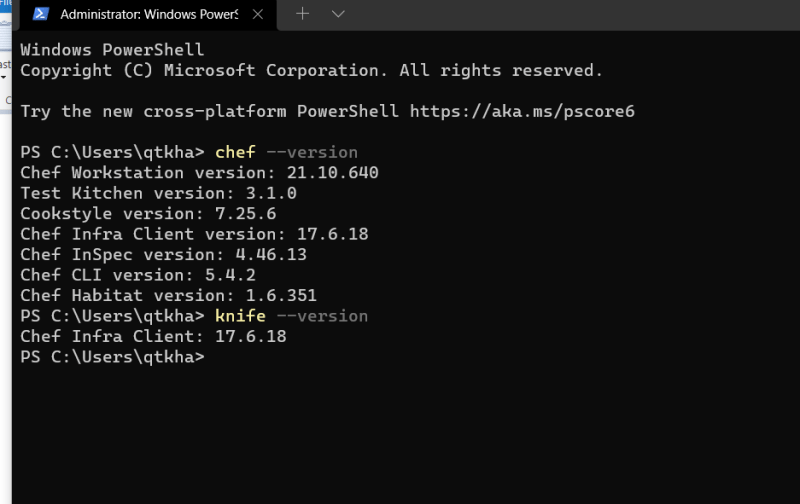
OCTOBER 27, 2021

# DevOps Classroom Series – 27/Oct/2021

## Configuration Management Architectural Models

* CM majorly has two architectural models
  + PUSH BASED CM:
    - CM server initiates communication with the nodes
    - For this CM server needs to maintain list of nodes & Credentials
    - Examples:
      * Ansible
      * Salt Stack
  + PULL BASED CM:
    - Nodes initiate communication with the CM Server
    - For this Nodes need to install CM agent
    - Examples:
      * Chef
      * Puppet  

## Chef

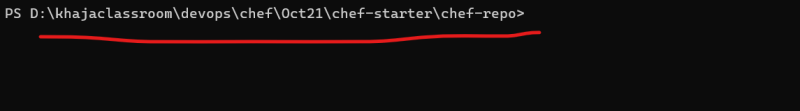
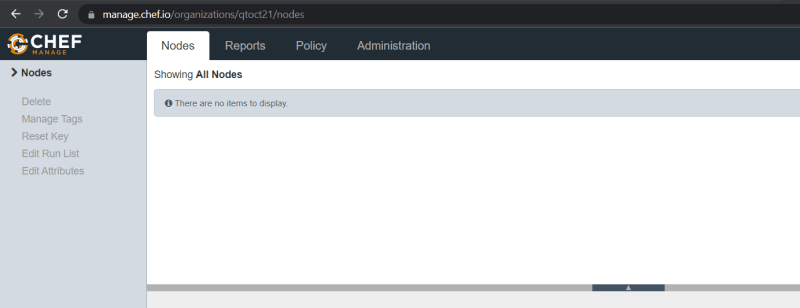
* Chef uses PULL BASED Configuration Management and can be used whether your nodes are on-premise or in cloud.
* Let’s dive into Chef Architecture 
* Bringing up the Chef Server:
  + Chef server can be installed on the Linux machine with 4 Cores and 8 GB of RAM. Here chef is free till 10 nodes
  + For learning purposes chef offers a hosted chef server which we can use, we will start learning from here [Refer Here](https://manage.chef.io/signup)
    - Create an organization and download the starter kit   
    - Note: Downloading the starter kit multiple times changes the keys used for communication
* Configuring Chef Workstation:
  + [Refer Here](https://downloads.chef.io/tools/workstation) to download the chef workstation
  + Let’s install the chef workstation 
  + Proceed with installation defaults
  + To verify if the installation is successful, after the installation is complete launch windows terminal/powershell and execute the following commands 

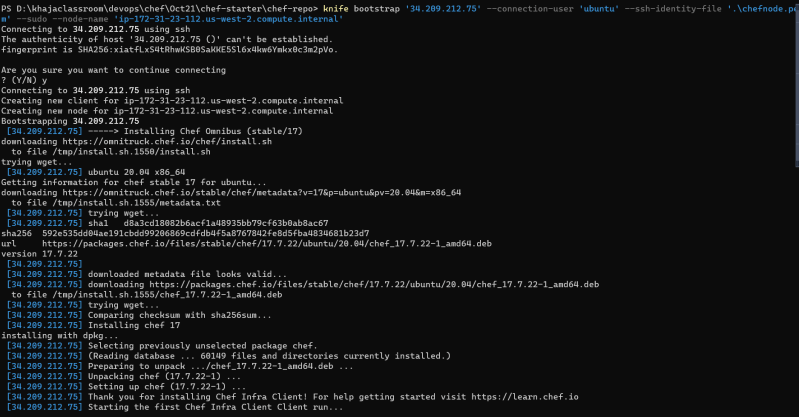
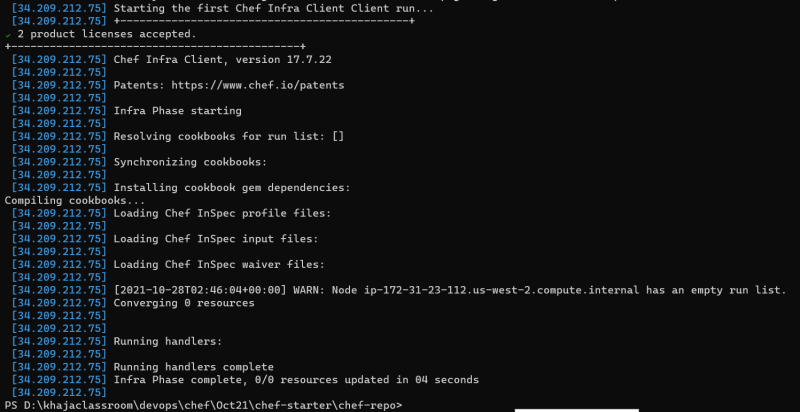
### Share this:

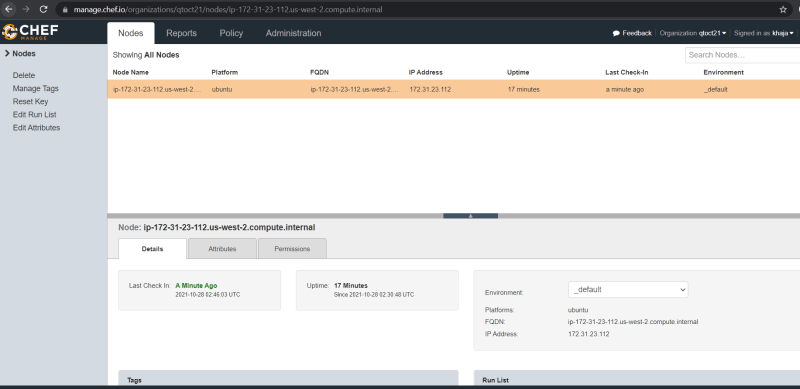
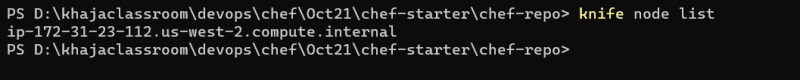
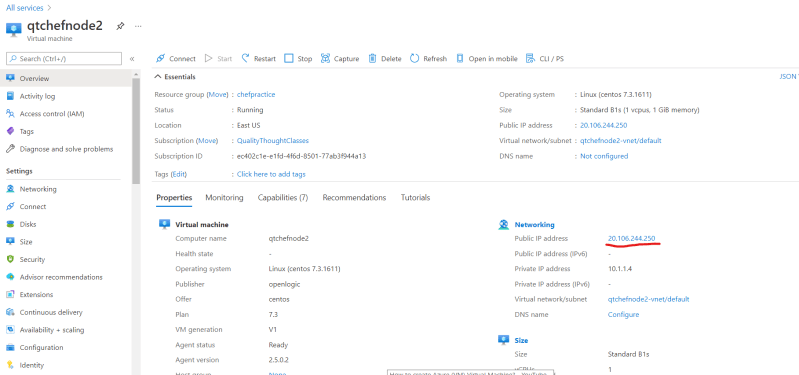
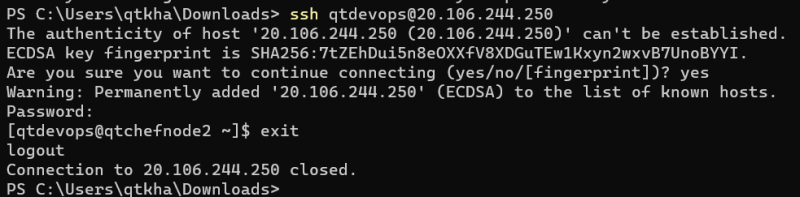
OCTOBER 28, 2021

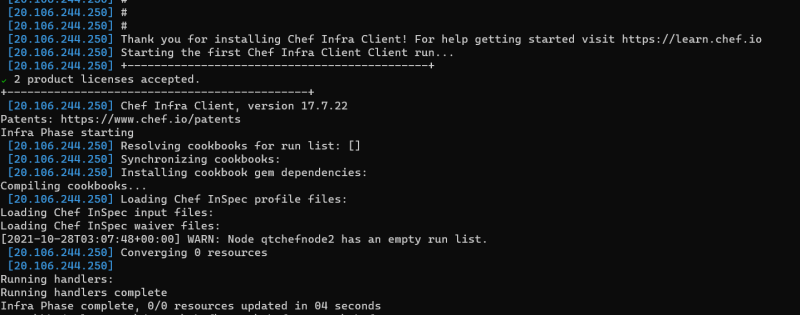
# DevOps Classroom Series – 28/Oct/2021

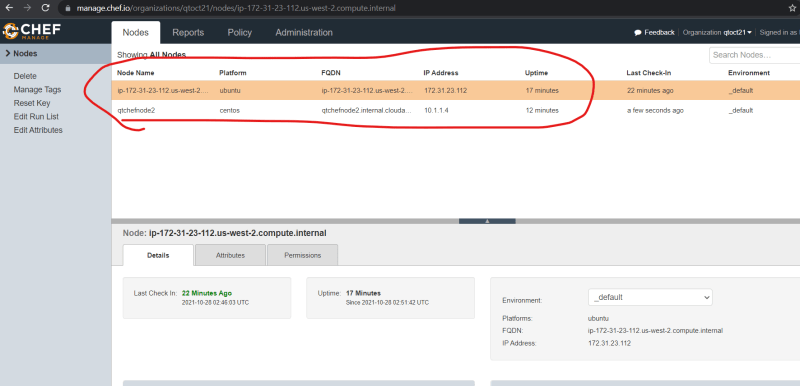
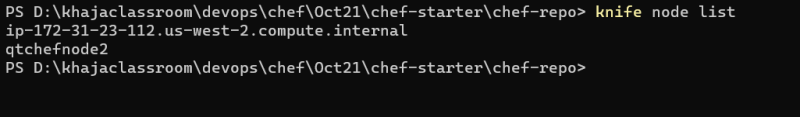
## Setting up Nodes for Chef

* In the last session, we were able to install chef workstation & create an account in hosted chef server with starter kit downloaded.
* Now let’s try to configure nodes.
* Lab Setup:
  + Creating a Ubuntu Node in AWS:
    - [Refer Here](https://www.youtube.com/watch?v=me2s3mTNwGo&list=PLuVH8Jaq3mLszrC7lv68a0VcrDripW-HK&index=2) to create an ec2 instance in AWS
    - EC2 instance created 
    - To login into this ec2 instance the command would be ssh -i <path to pem> ubuntu@<publicip>
    - On this node we need to ensure chef-client which is agent to be installed. This process is called as bootstrapping
    - To bootstrap the node, when we install chef-workstation we get a application called as knife, we can use knife & starter kit to bootstrap the node
    - cd into the chef-repo (extracted from starter kit) 
    - [Refer Here](https://docs.chef.io/workstation/knife_bootstrap/#syntax) for the knife bootstrap command
    - Login into hosted chef 
    - Now let’s try to build a command to bootstrap the node
  + 77777

0.323 

* + - Let’s observe the hosted chef management console 
    - Try form command line knife node list 
  + Create a Centos 7 vm in Azure:
    - [Refer Here](https://www.youtube.com/watch?v=P9X-4Z-NeGg&list=PLuVH8Jaq3mLuqXuGs6aeqxhuvCYSzB1kT&index=2) for steps to create Azure VM 
    - The ssh command for this node is ssh <username>@<ipaddress> 
    - Now let’s try to build the command to bootstrap the node
  + knife bootstrap '20.106.244.250' --connection-user 'qtdevops' --connection-password 'motherindia@123' --sudo --use-sudo-password --node-name 'qtchefnode2'

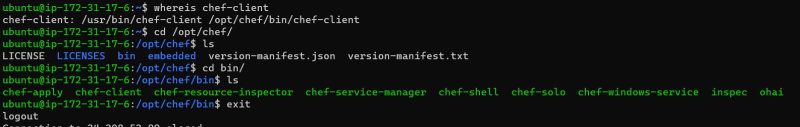


* + - Management console 
    - Execute knife node list 
* Exercise: Try to bootstrap any Linux node using knife

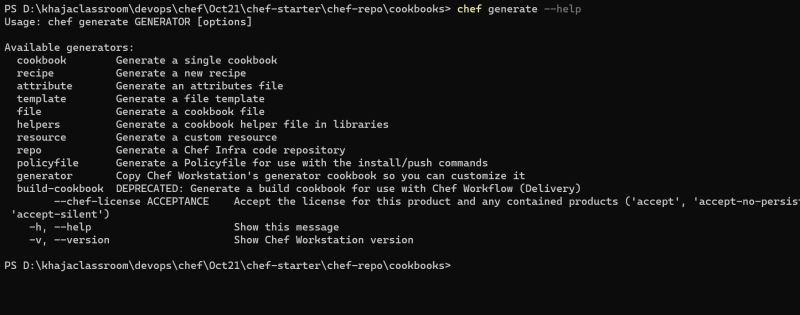
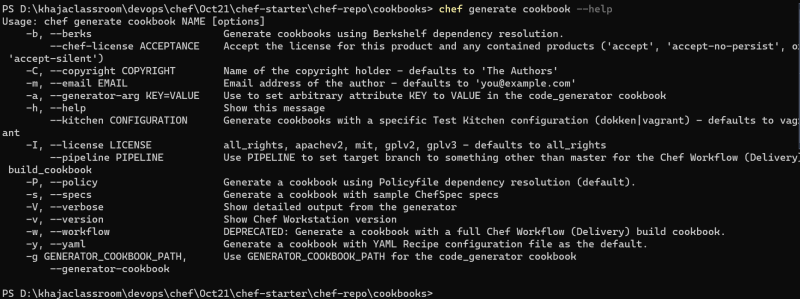
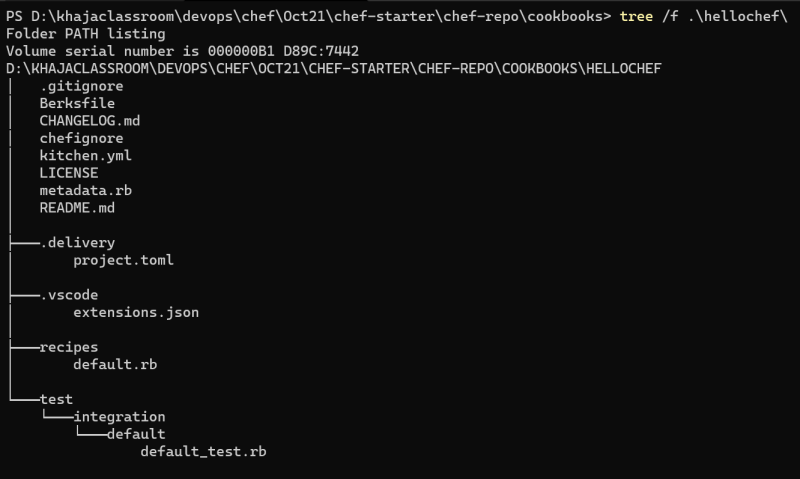
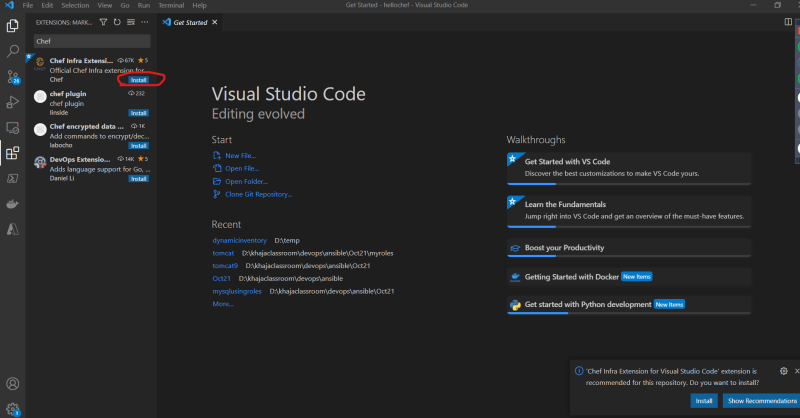
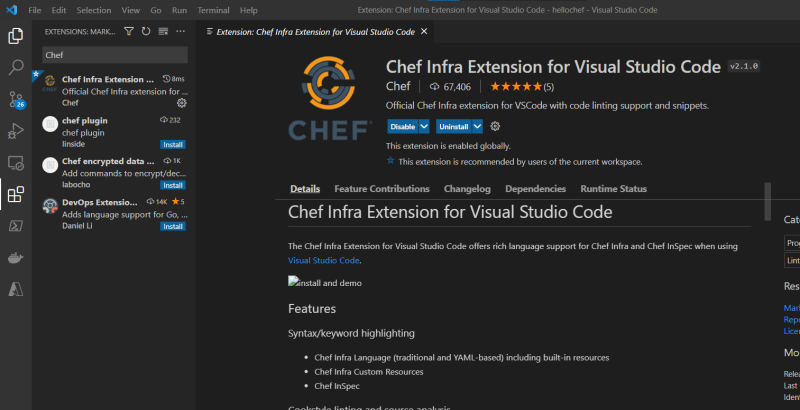
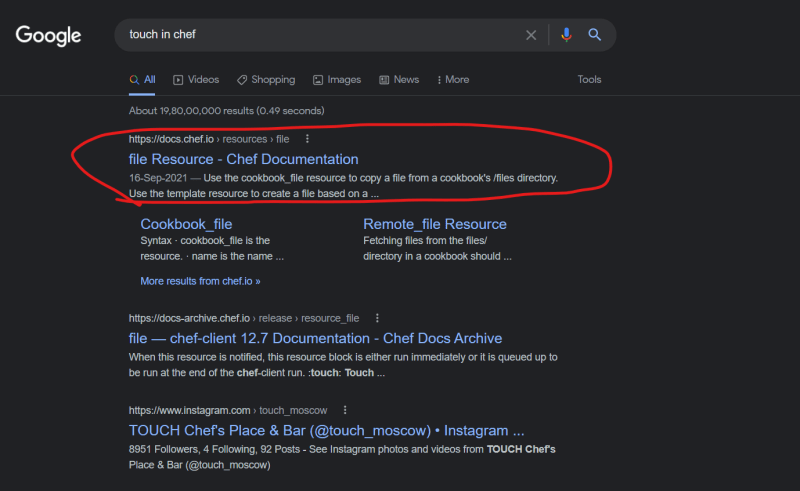
OCTOBER 29, 2021

# DevOps Classroom Series – 29/Oct/2021

## Chef-Client

* Let’s login into the chef nodes and understand what has installed 

## Chef-Workstation

* One or more workstations are configured to allows users to work with chef
* The users in chef develop
  + cookbooks:
    - Has recipes, which in turn has chef resources placed in some order.
* The Chef infra language is based on Ruby, chef has its own DSL (Domain specific language) to express our desired state.
* Step 1: Create a cookbook   
* Step 2: Examine the contents of the cookbook folder 
* Step 3: Configure Visual Studio Code with Chef Extension  
* Step 4: Find a resource to create a file on the node [Refer Here](https://docs.chef.io/resources/file/#syntax) 
* Step 5: Understand Resource Syntax

<resource type> '<name>' do

<property> <value>

...

<action>

end

* Step 6: Lets create a file in /tmp/fromchef & for this lets write the following in recipes\default.rb

file '/tmp/fromchef' do

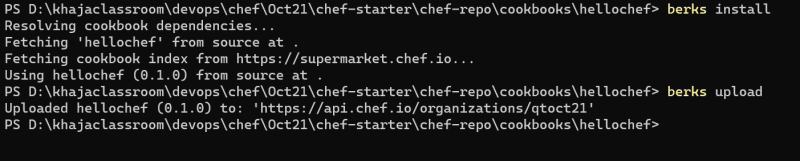
action :touch

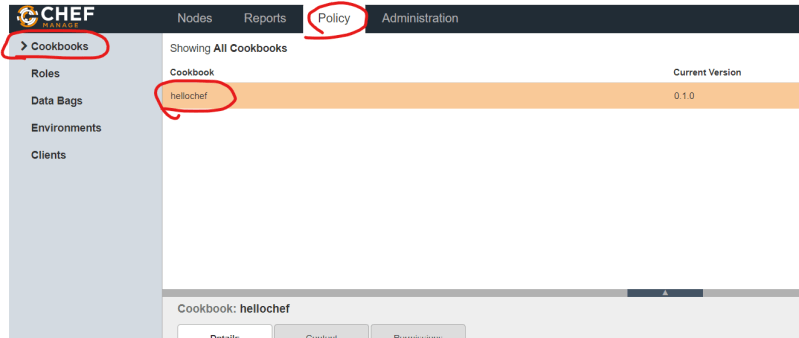
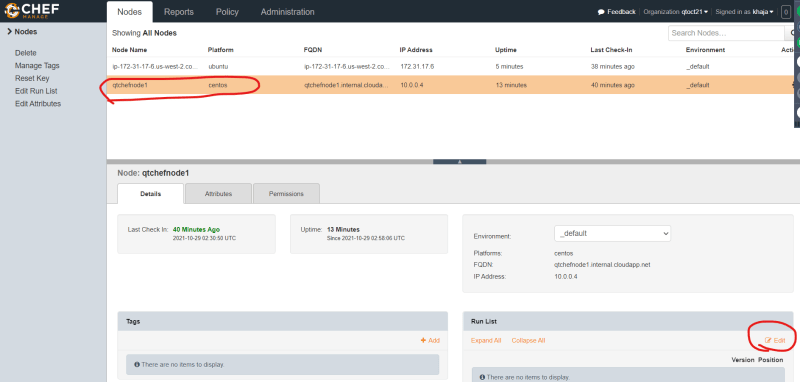
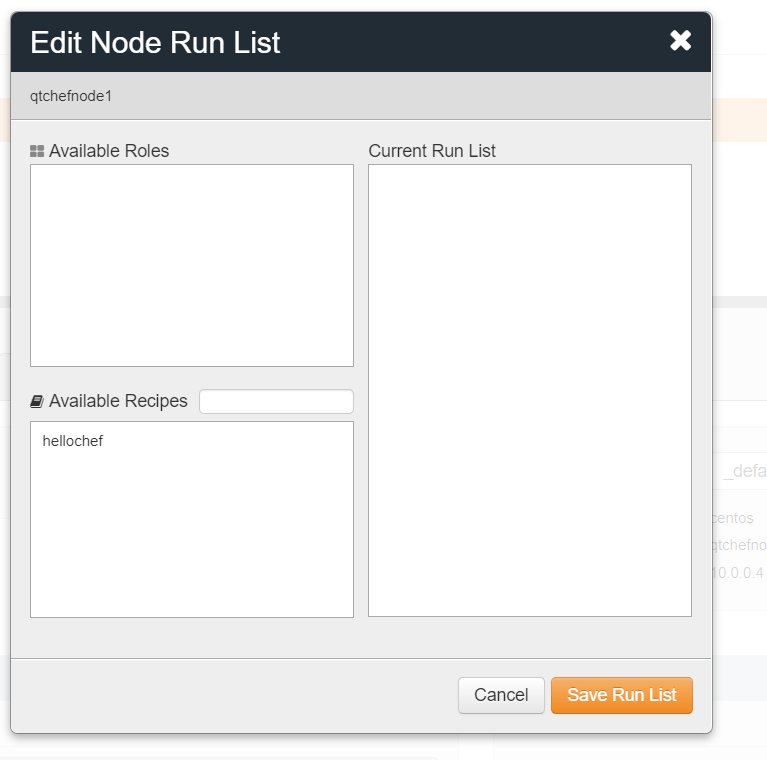
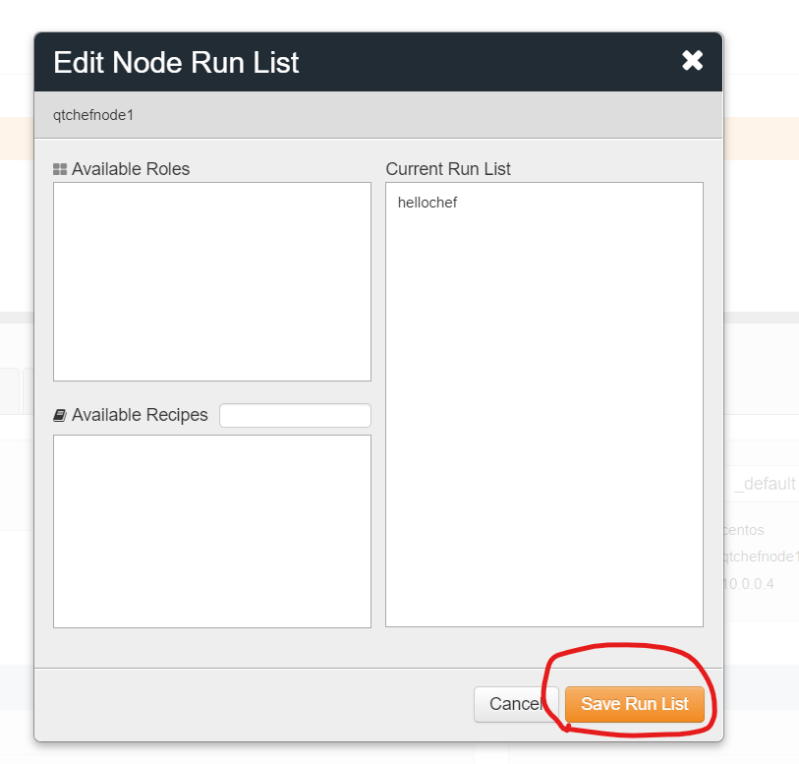
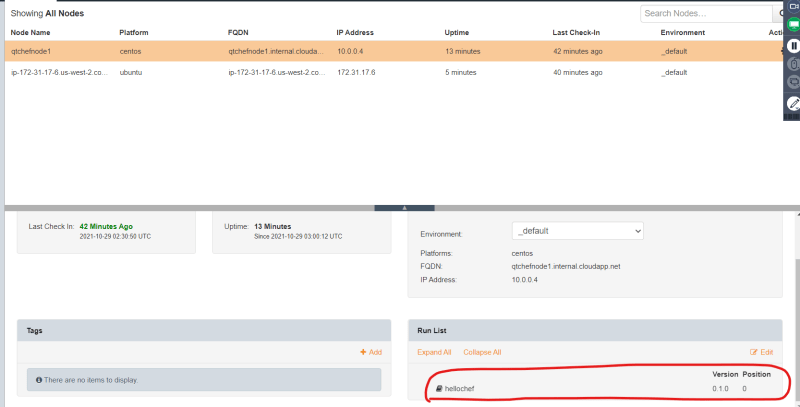
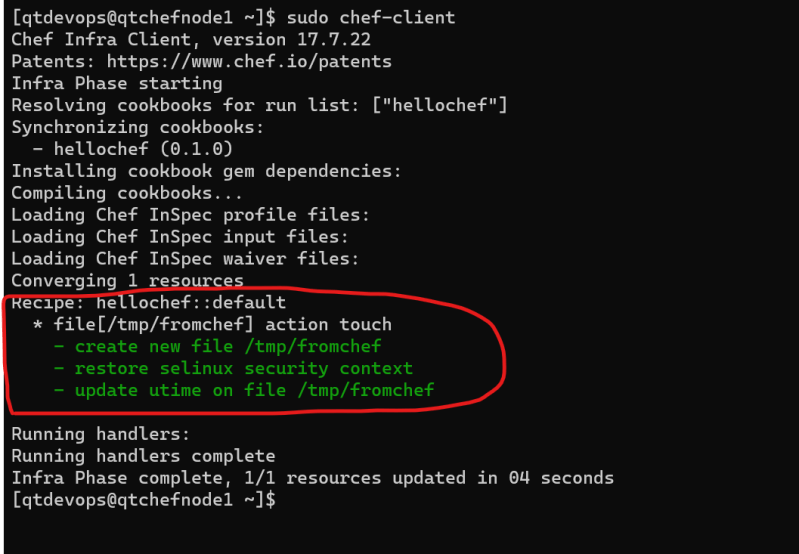
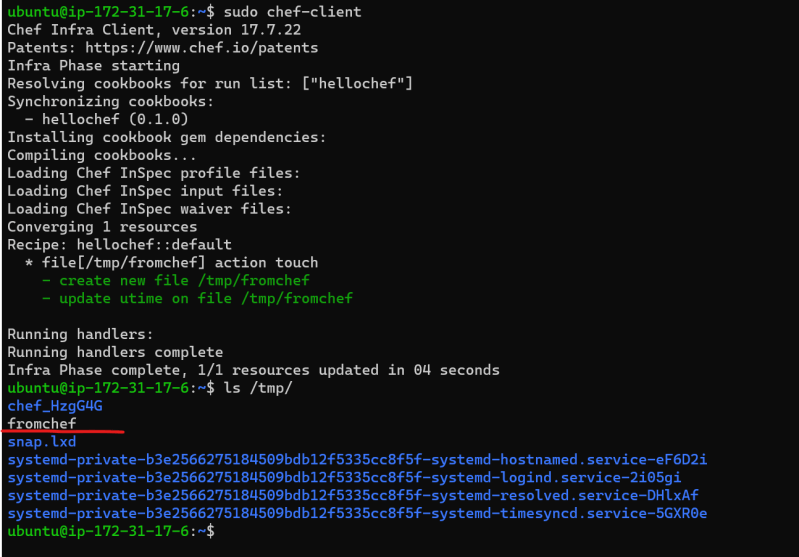
end

* Step 7: Lets upload our cookbook to chef server:

berks install

berks upload

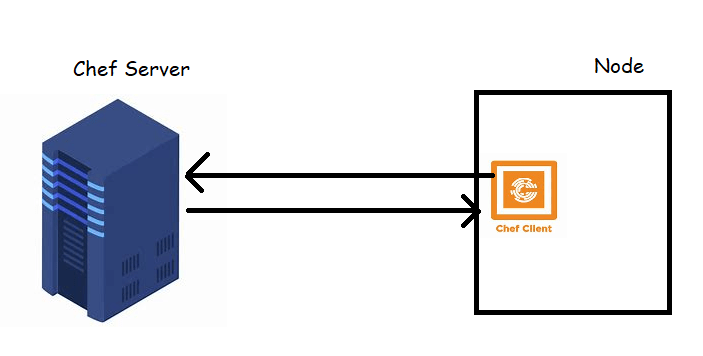
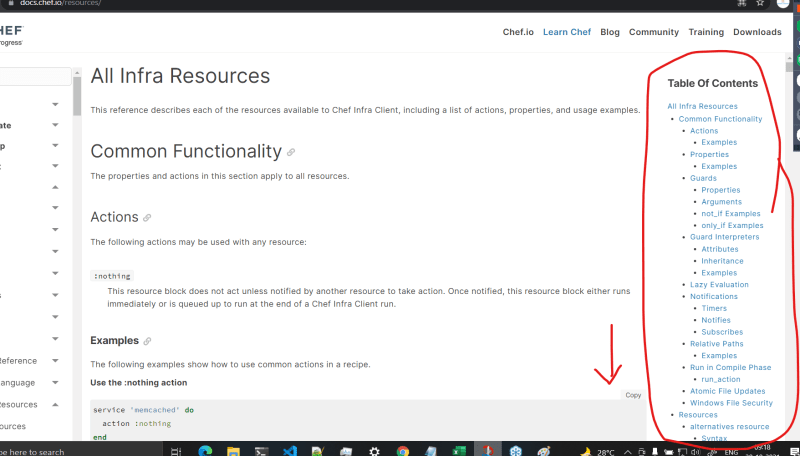


* Step 8: Verify whether cookbooks are uploaded or not 
* Step 9: Configuring this cookbook to be executed on our nodes
  + Every node has a run\_list which specifies the recipes to be executed when the node communicates (asks) the server
  + Now let’s change run\_list for the azure node    
  + Whenever azure node tries to speak with chef server, our recipe will be executed. In this cases rather than waiting for azure node to communicate with chef server, let’s manually try to do this by logging into azure node.
  + when node wants to speak with chef-server the following command is executed sudo chef-client  Preview
* Step 10: Do the same for aws node 

OCTOBER 30, 2021

# DevOps Classroom Series – 30/Oct/2021

## Basic Terminology

* Convergence:
  + It is the when node (chef-client) interacts with chef-server.
  + During the convergence, the chef server responds back with the recipes in run\_list configured for the node to be executed.
  + By default the convergence happens every 1/2 hour and this can be changed 
* Resource:
  + In chef resource is smallest unit of work that automates some activity and allows us to express the desired state
  + [Refer Here](https://docs.chef.io/resources/) for all the infra resources. 

## Approach for Automating Deployments

* To automate any deployment,
  + make a note of all the manual steps involved
  + Try to execute them on some test server to ensure manual steps are correct
* Once you have consolidated steps, for each step try to find a chef resource.

## Activity-1: Installing a Lamp Server on Ubuntu 20.04

* [Refer Here](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04) for the manual steps
* Manual steps:

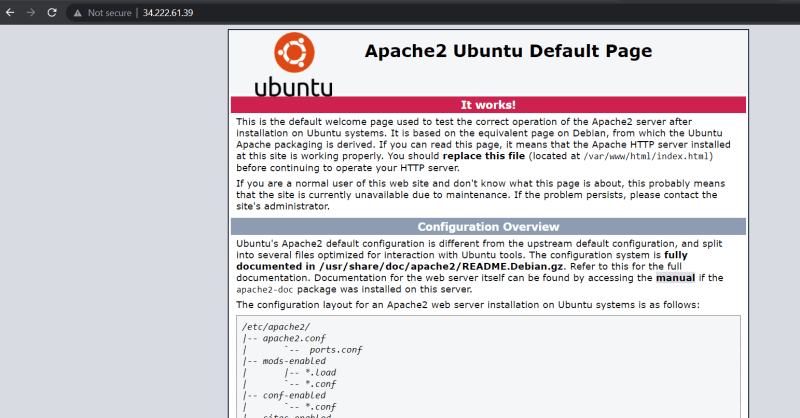
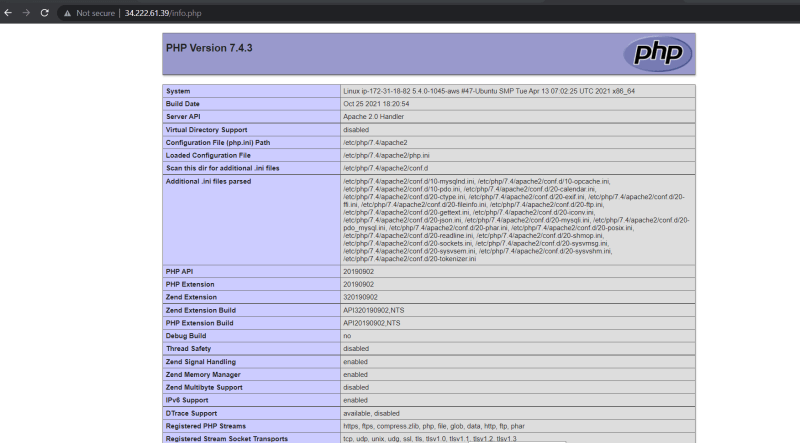
sudo apt update

sudo apt install apache2 -y

sudo apt install php libapache2-mod-php php-mysql -y

echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

sudo systemctl restart apache2

* Evaluate whether these commands work or not:
  + Let’s create an Ubuntu 20.04 VM and execute the commands
  + Navigate to http://<public-ip&gt; 
  + Navigate to http://<public-ip>/info.php 
* Goal: To automate the above mentioned manual steps in chef
* Steps to Achieve:
  + We can start by creating a cookbook
  + finding right resources
  + Expressing the desired state
  + Upload the cookbook
  + Wait for convergence or manually force the chef-client run

### Share this:

OCTOBER 31, 2021

# DevOps Classroom Series – 31/Oct/2021

## Automating deployment of PHP

* Manual Steps

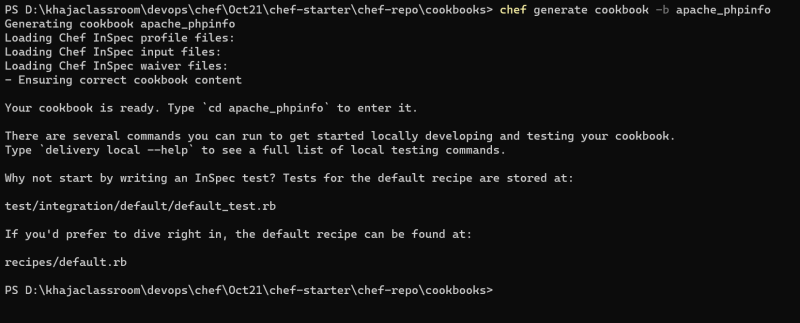
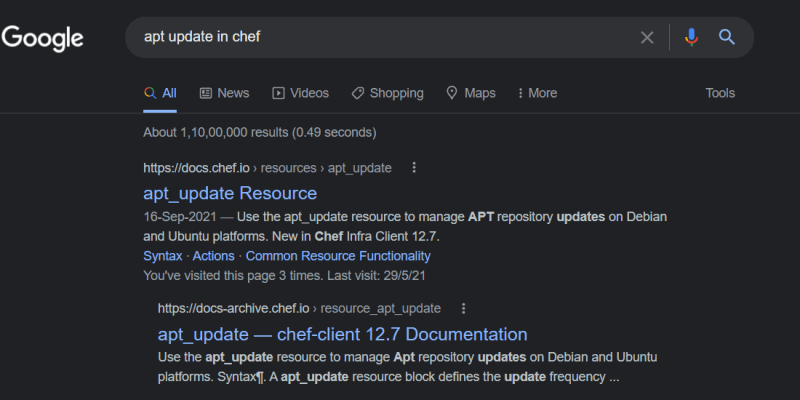
sudo apt update

sudo apt install apache2 -y

sudo apt install php libapache2-mod-php php-mysql -y

echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

sudo systemctl restart apache2

* Steps:
  + Create a cookbook 
  + Launch visual studio code in the folder apache\_phpinfo
  + Navigate to the recipes\default.rb file
  + Find a resource for the first step sudo apt update 
  + The resource found is
* apt\_update 'update ubuntu packages' do
* ignore\_failure true
* action :update
* end
  + Find a resource for apt in chef [Refer Here](https://docs.chef.io/resources/apt_package/) for the resource found
  + Let’s look at examples [Refer Here](https://docs.chef.io/resources/apt_package/#examples)

apt\_update 'update ubuntu packages' do

ignore\_failure true

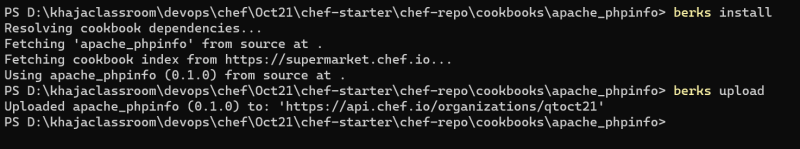
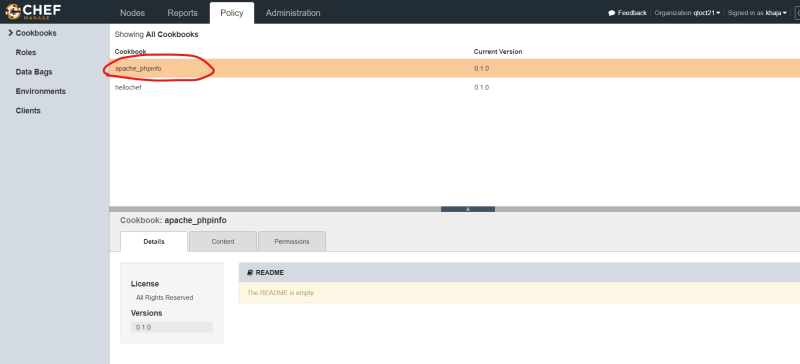
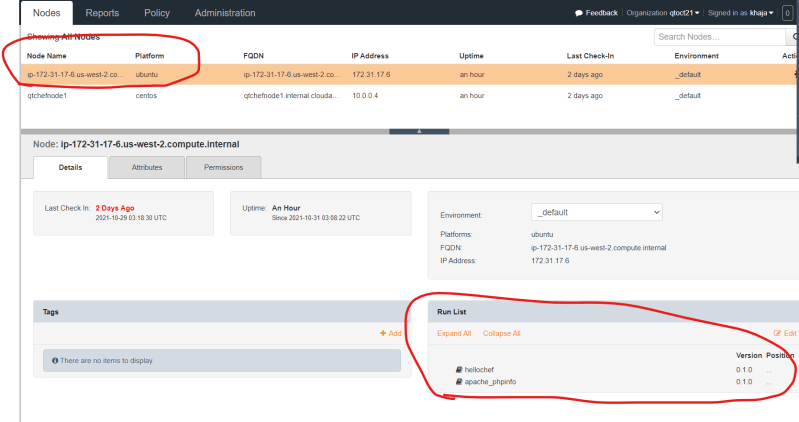
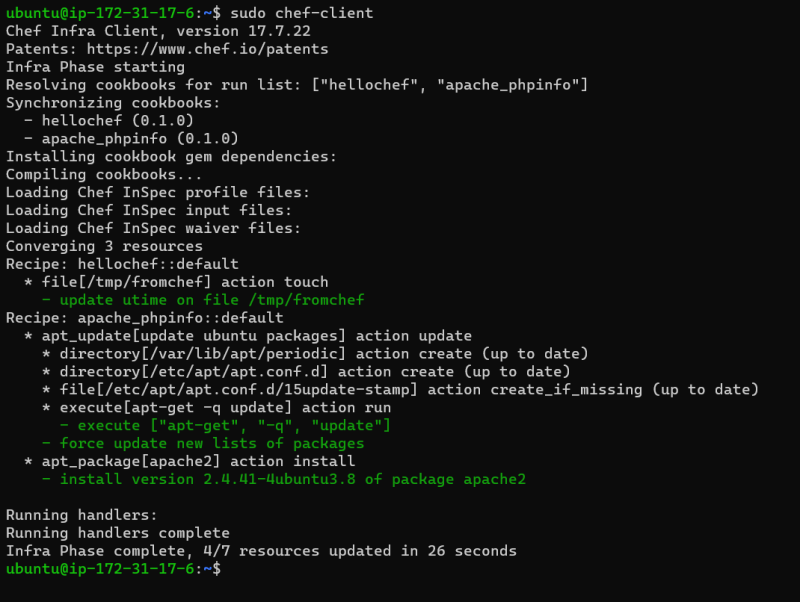
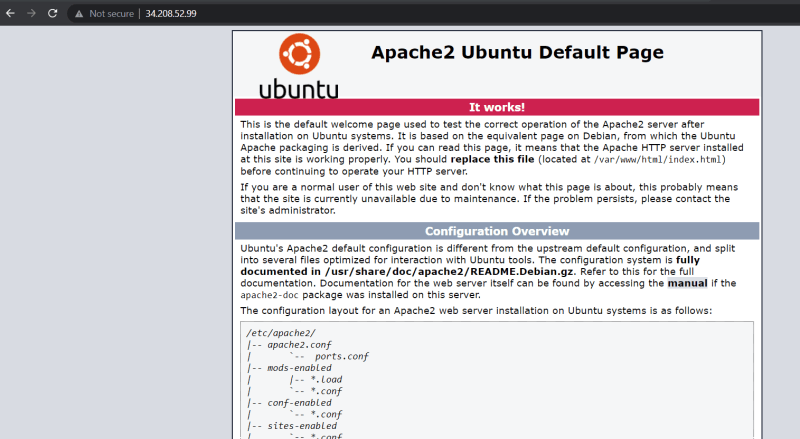
action :update

end

apt\_package 'apache2' do

action :install

end

* + Let’s try to upload our cookbook and change the run\_list of the Ubuntu node for the steps done so far   
  + Now let’s manually force the convergance.  
  + Now let’s try to automate next set of steps sudo apt install php libapache2-mod-php php-mysql -y
  + The Recipe so far looks as shown below

apt\_update 'update ubuntu packages' do

ignore\_failure true

action :update

end

apt\_package 'apache2' do

action :install

end

apt\_package 'php' do

action :install

end

apt\_package 'libapache2-mod-php' do

action :install

end

apt\_package 'php-mysql' do

action :install

end

* + The next step is to create a file with some content echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php. [Refer Here](https://docs.chef.io/resources/file/) for the chef file resource
  + The recipe so far looks as shown below

#

# Cookbook:: apache\_phpinfo

# Recipe:: default

#

# Copyright:: 2021, The Authors, All Rights Reserved.

apt\_update 'update ubuntu packages' do

ignore\_failure true

action :update

end

apt\_package 'apache2' do

action :install

end

apt\_package 'php' do

action :install

end

apt\_package 'libapache2-mod-php' do

action :install

end

apt\_package 'php-mysql' do

action :install

end

file '/var/www/html/info.php' do

content '<?php phpinfo(); ?>'

action :create

end

* + Now there is one last step to restart the service apache2 [Refer Here](https://docs.chef.io/resources/service/)
  + The recipe so far looks as shown below
  + #

# Cookbook:: apache\_phpinfo

# Recipe:: default

#

# Copyright:: 2021, The Authors, All Rights Reserved.

apt\_update 'update ubuntu packages' do

ignore\_failure true

action :update

end

apt\_package 'apache2' do

action :install

end

apt\_package 'php' do

action :install

end

apt\_package 'libapache2-mod-php' do

action :install

end

apt\_package 'php-mysql' do

action :install

end

file '/var/www/html/info.php' do

content '<?php phpinfo(); ?>'

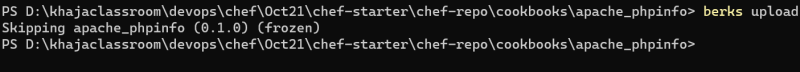
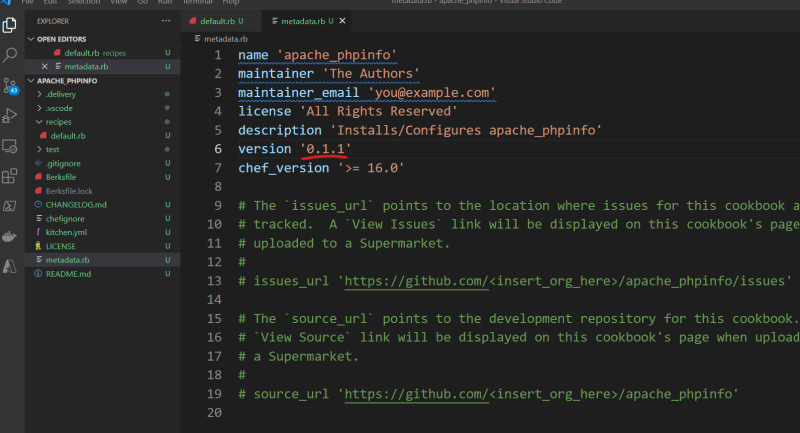
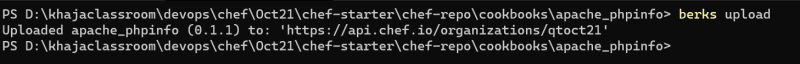
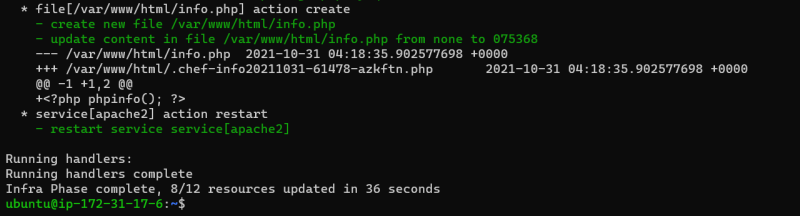
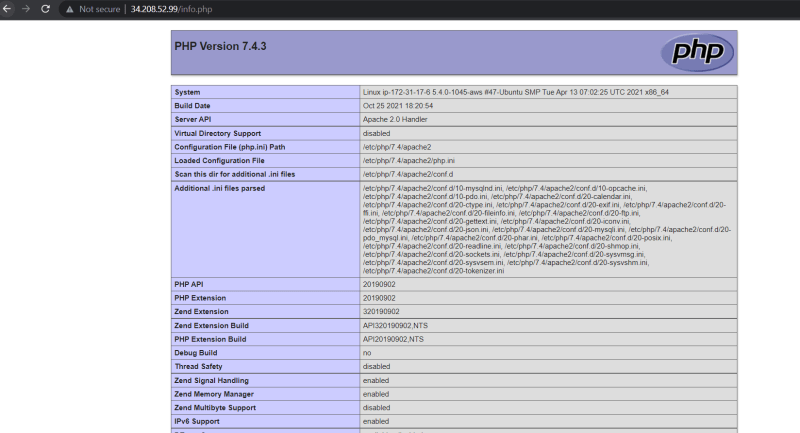
action :create

end

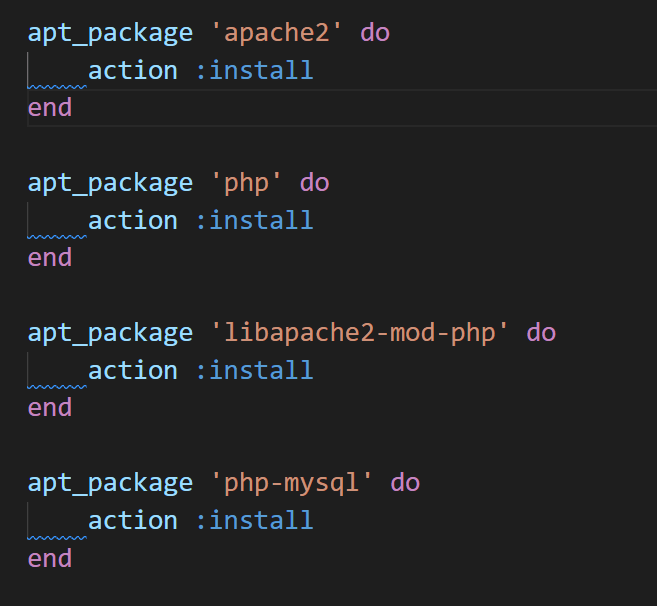
service 'apache2' do

action :restart

end

* + We have already execute berks install we need to execute berks upload 
  + Berks has not uploaded the new changes, as berks will allow the upload only if the version does not exist in the chef server. In our case, 0.1.0 version is already uploaded to chef server, So to upload the cookbook we need to change the version number.
  + Changing version number:
    - Navigate to the file apache\_phpinfo\metadata.rb and change the version  
  + Now let’s upload the new version of the cookbook 
* Since the run\_list is same, lets manually force convergence and check whether the deployment works   
* [Refer Here](https://github.com/asquarezone/ChefZone/tree/master/Oct21/chef-starter/chef-repo/cookbooks/apache_phpinfo) for the cookbook developed in the class

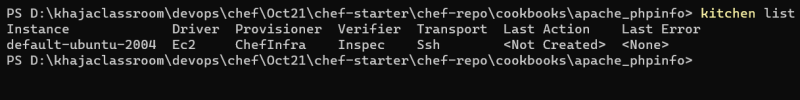
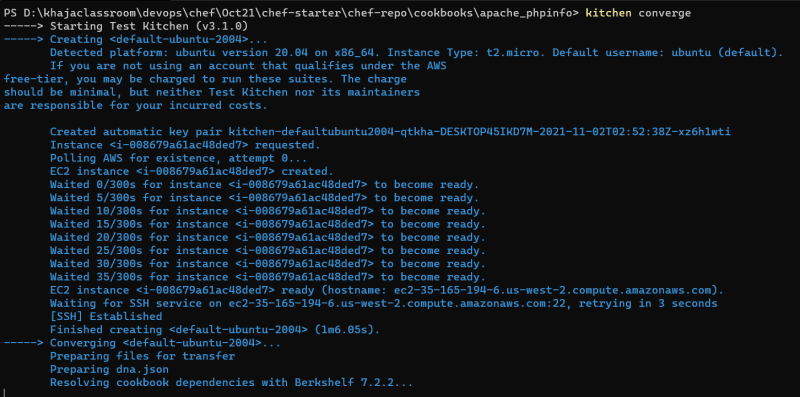
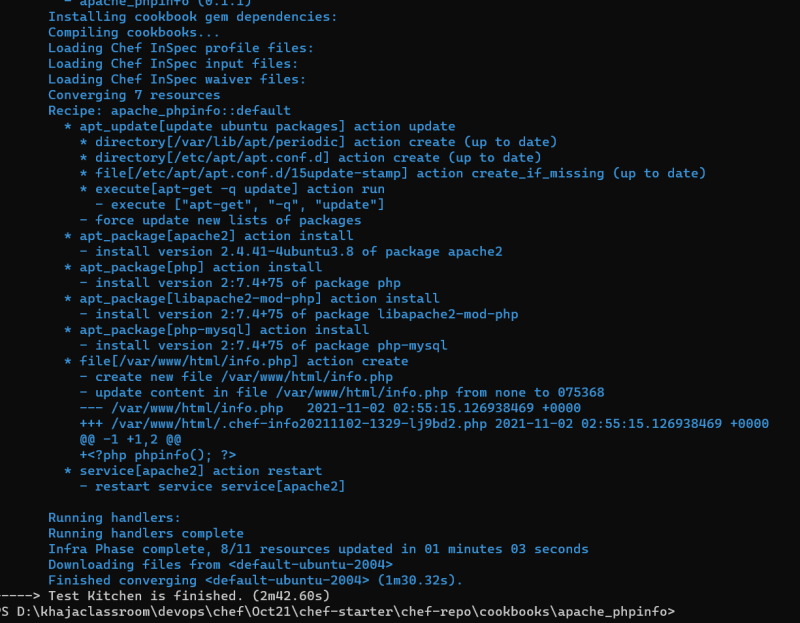
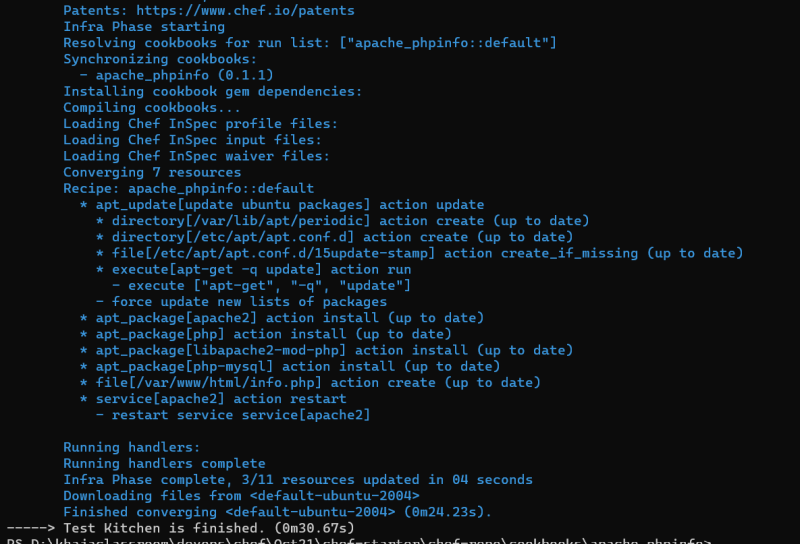
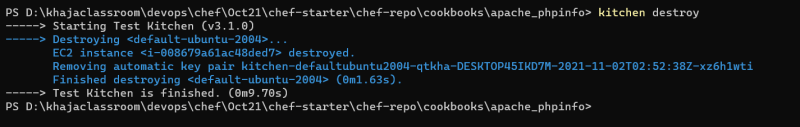
## Improvements to be done

* We are using resources so that they work only for Ubuntu, we should be more generic
* We are using same resource with multiple values it would be better to use some kind of looping mechanism 
* Every time when convergence is happening the service is getting restarted which would lead to down times, we need to fix this.
* We are uploading multiple versions of the cookbook in development phase, if we have some mechanism to test what we are developing and only upload working versions to server it would be good

NOVEMBER 2, 2021

# DevOps Classroom Series – 02/Nov/2021

## Test Kitchen

* We can use test kitchen to test cookbook across any combination of platforms
* When we create a cookbook, we have a file with in cookbook kitchen.yaml
  + Drivers:
    - Drivers help test kitchen to create test instances.
    - Chef supports Creating test instances on cloud providers and also on virtual box
      * AWS
      * Azure
      * GCP
      * Virtual Box (Vagrant)
    - [Refer Here](https://docs.chef.io/workstation/kitchen/#drivers) for the documentation
  + Key Concepts
    - A platform represents the operating system where the cookbook will be tested
    - A suite which will have a run\_list
    - A provisioner chef\_zero or chef\_solo
* Let’s start working on a cookbook apache\_phpinfo and set up the ec2 driver [Refer Here](https://directdevops.blog/2019/04/02/test-kitchen-setup-with-aws/)
* Also [Refer Here](https://github.com/test-kitchen/kitchen-ec2) for the kitchen ec2 driver documentation
* Install aws cli [Refer Here](https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html)
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/f140408cf788ed67e85361cc0ca9a5c450d2e8c6) for the changes done in kitchen.yaml
* Kitchen has commands, execute them as mentioned below    \*
* To check syntax install ruby [Refer Here](https://rubyinstaller.org/) and then execute the command ruby -c <path to recipe> Preview
* In chef, we have a generic package resource [Refer Here](https://docs.chef.io/resources/#package-resource). [Refer Here](https://github.com/asquarezone/ChefZone/commit/fc126327743ddb15aa28cd72726be033ba476109) for the changes done to use a generic package.
* Let’s converge using test kitchen kitchen converge 
* Let’s use array for installing packages [Refer Here](https://github.com/asquarezone/ChefZone/commit/76a8bda7623d282aa99527a5c3ce2e87b616bc1c)
* Let’s try to check if our changes works on a fresh ubuntu instance.
* Let’s delete the ec2 created by kitchen 
* Now let’s do kitchen converge to simulate fresh installation
  + converge

NOVEMBER 4, 2021

# DevOps Classroom Series – 03/Nov/2021

## Just enough Ruby for Chef

* [Refer Here](https://docs.chef.io/ruby/) for the official docs from chef
* Local variables

x = 1 # number

y = 'hello' # string

z = "hai" # string

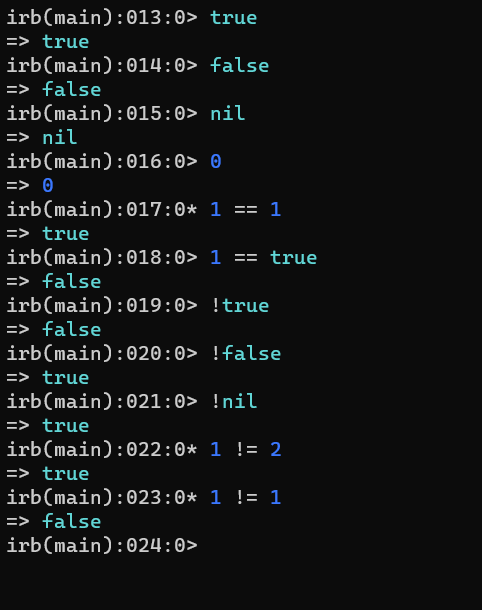
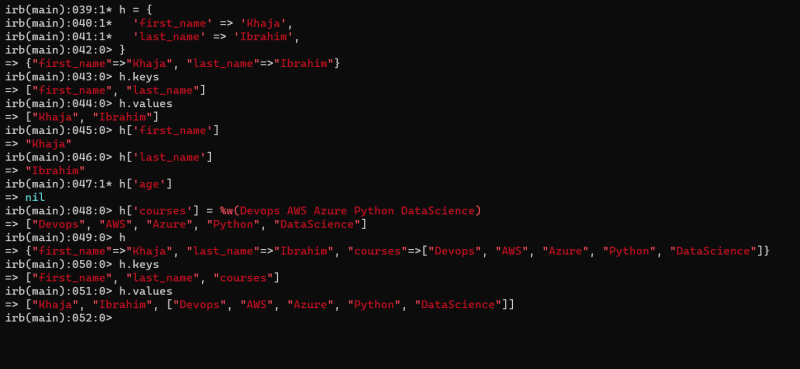
* Embed a ruby variable in a string

topic = 'chef'

'Hello all, Welcome to #{topic}'

"Hello all, Welcome to #{topic}"



* True and false (Boolean) 
* Arrays
* White space Arrays: This is shortcut for creating arrays without quotes and commas
* Hash: 
* Conditional Statement using if else:

if <condition>

# if block

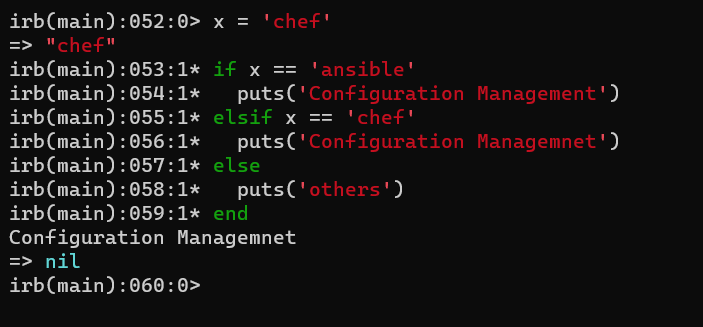
elsif <condition>

# else if block

else

# else block

end

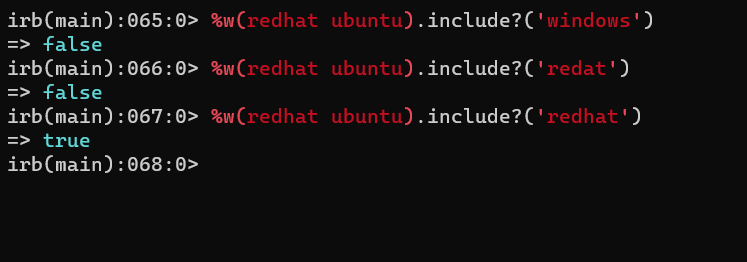


* Looping using each

%w(redhat ubuntu).each do |osname|

puts(osname)

end

* Other useful statements 

## Chef contd..

* Now let’s try to apply whitespace array to array for package names in the apache\_phpinfo cookbook [Refer Here](https://github.com/asquarezone/ChefZone/commit/2db6785c2ea917ee3fc2992bbcf89c4232b8bcf2) for the change set
* Now apply the kitchen convergence and verify if everything works or not
* Now our aim is to ensure this cookbook works for red hat instances also. The manual steps are

sudo yum install httpd -y

sudo systemctl enable httpd

sudo systemctl start httpd

sudo yum install php php-mysql php-fpm -y

echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

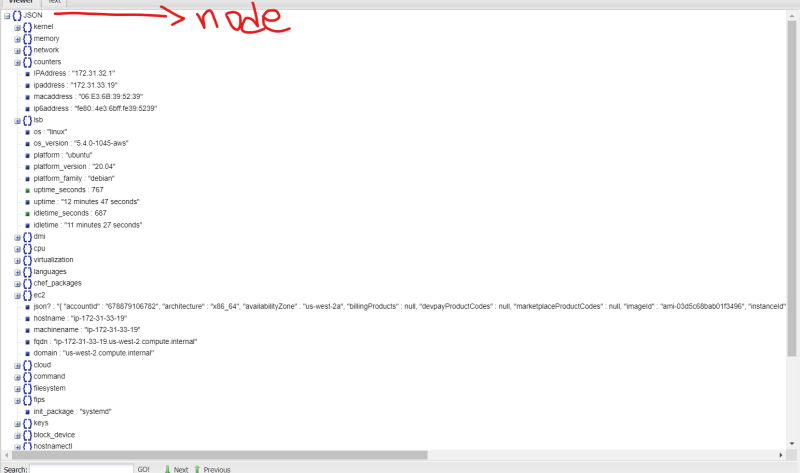
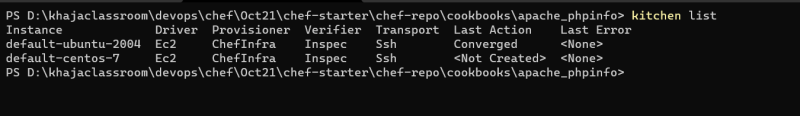
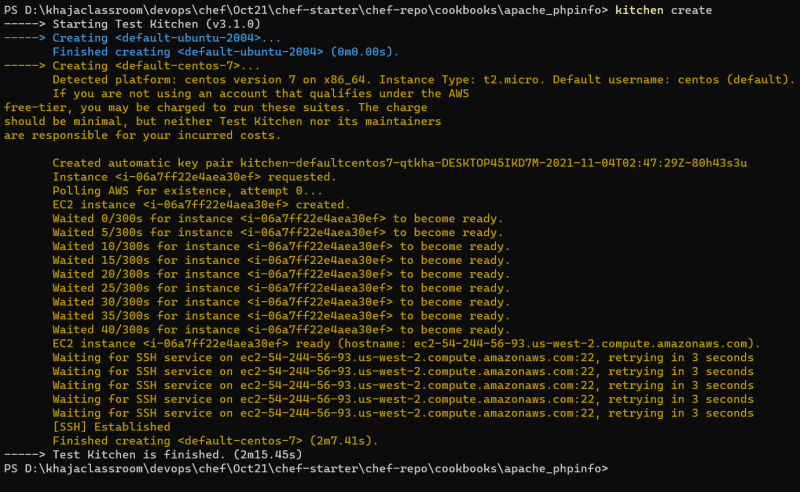
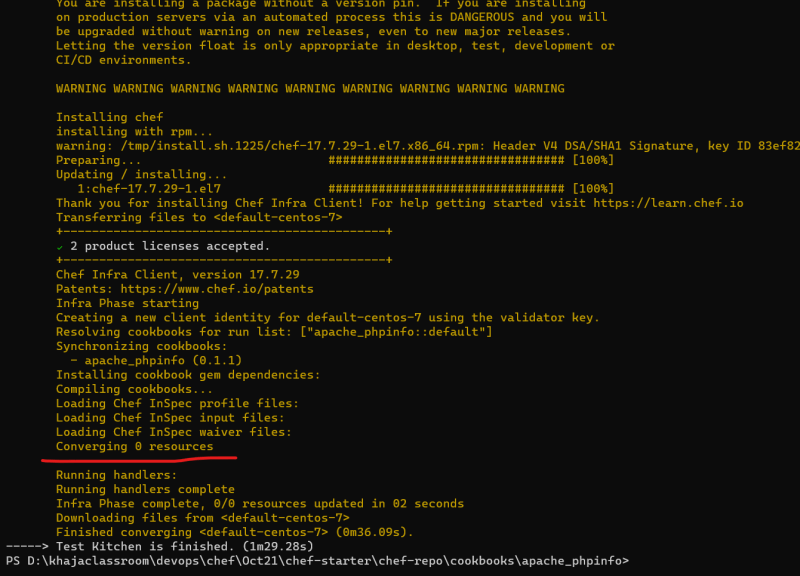
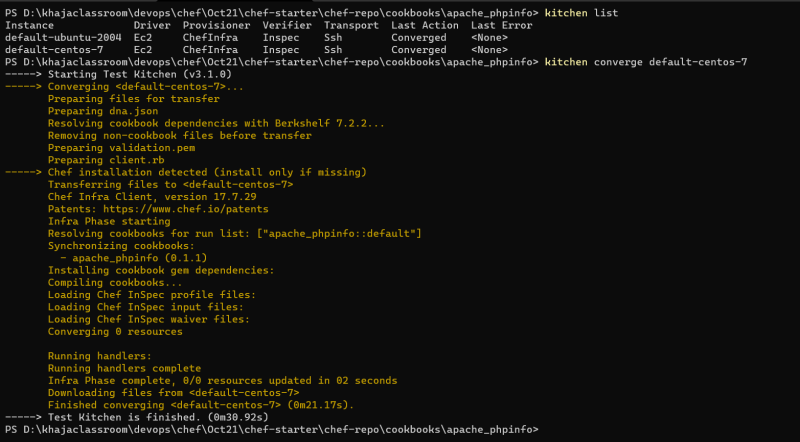
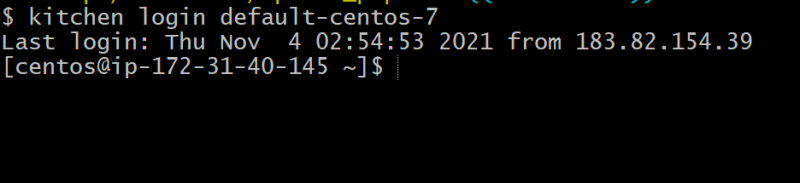
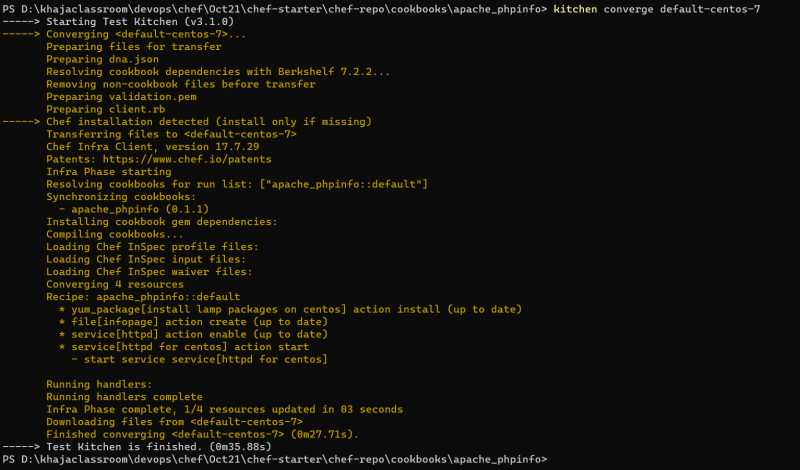
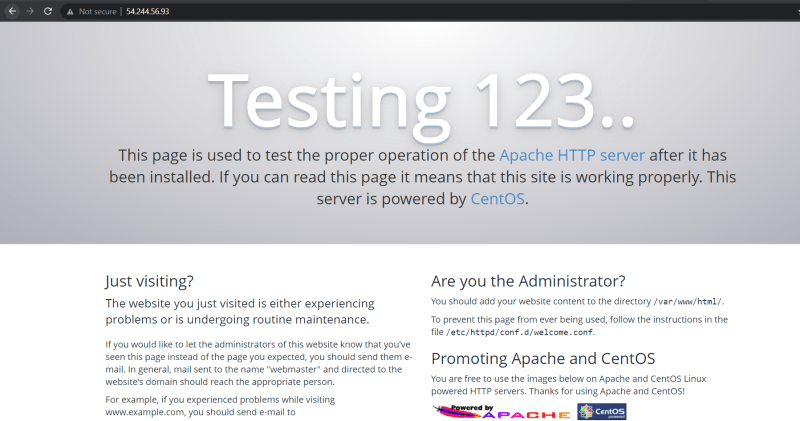
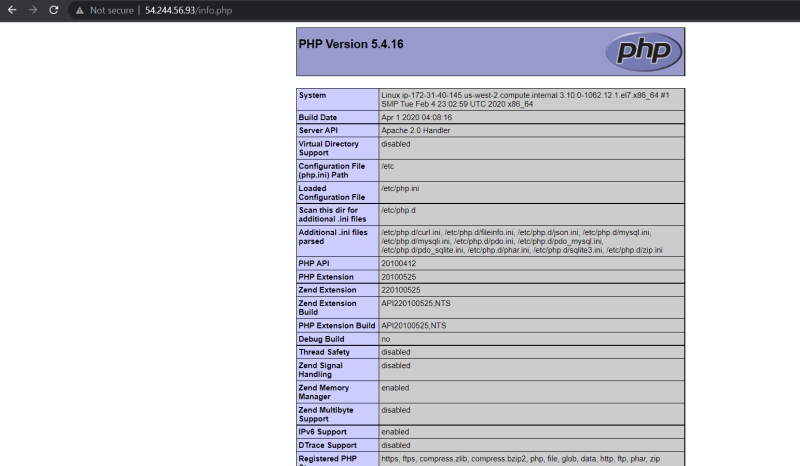
sudo systemctl restart httpd

* When we want to run the recipe on different distributions of linux we should have a way to identify the way to find the information about the node on which recipe is executing.
* For doing this in chef there is tool called as ohai which gets installed on the node and informs the chef server about the details of the node

NOVEMBER 4, 2021

# DevOps Classroom Series – 04/Nov/2021

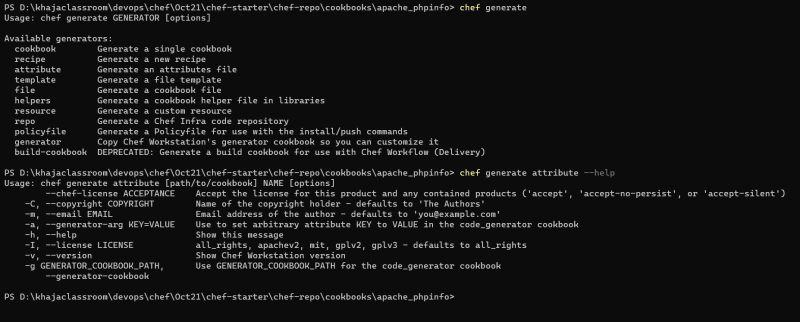
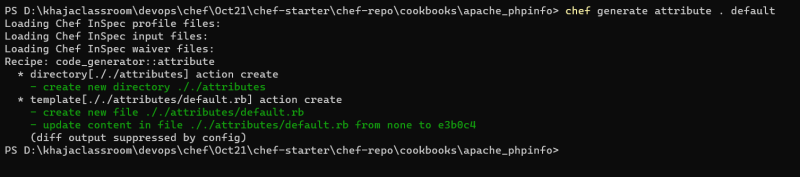
## Ohai contd

* All the details collected from ohai are loaded into node object (hash)  
  
* So the access the platform the syntax which we would use would be node['platform']
* Now make changes in the kitchen.yaml to accommodate centos7 and execute kitchen list  
  
* Now create the instance using kitchen create  
  
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/f4fede4d07510e18c5227a2d82a1156166995ddc) for the changes added with kitchen and recipe changes  
  
* We can converge a particular instance by using its name as shown below  
  
* Now login into centos instance and try ohai  
  
* Now using node[‘platform’] == centos we had made changes in the recipe to accommodate for the centos.
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/d17c0eeeab5d52854583dd19f85fe2cd1b3cf476) for the changes and then we had converged the centos using kitchen  
  
* Now let’s verify by accessing URL’s  
    
  
* It would be a good improvement if the separate variables from execution
* Applied the above concept and now the changes are made [Refer Here](https://github.com/asquarezone/ChefZone/commit/40475b0544c1030ed6bcfc2b002ebac82fcd9d40) for the changes
* Now apply kitchen converge

NOVEMBER 9, 2021

# DevOps Classroom Series – 09/Nov/2021

## Attributes in Chef

* If we create variables in the recipe, changes to the variables might need change in cookbook versions, if we want to handle the variables in such a way that without uploading new version of cookbook if we can change variable values it would be better.
* In chef for this we have attributes.
* Attributes in Chef can be set from
  + recipe
  + attribute file in cookbook
  + roles
  + environment
* Creating an attribute file in chef cookbook
  + Exploring command line 
  + Lets create the attribute file 
* For adding attributes we need to follow the following syntax

<attribute\_type>['attribute-name']='value'

* For now we will be using only one attribute type default and we will be following the convention

default['cookbook\_name']['attribute\_name'] = 'value'

* To use the attributes in the recipe file replace default with node

node['cookbook\_name']['attribute\_name']

* [Refer Here](https://github.com/asquarezone/ChefZone/commit/0e16c6b9e7f9d909ecc10e0618ccc3d49a8fa010) this change set for the changes done to accommodate attributes.

## Chef Infra Language checking platforms

* [Refer Here](https://docs.chef.io/infra_language/checking_platforms/) for the official documentation
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/6f330b811a36853282db0cdbcedc51f0ea5ead72) for the changes using chef infra language checking platforms
* [Refer Here](https://docs.chef.io/infra_language/checking_platforms/#value_for_platform) for returning a value based on platform & its version and [Refer Here](https://github.com/asquarezone/ChefZone/commit/77baa9eafd87e70624c4cd33f010eb554007848a) for the change set containing the changes

## Common Resource Functionality: Guards

* Every resource in chef can have two properties only\_if and not\_if which gets executed during converge
* A guard property accepts either string value of ruby block
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/ff1401fa20a987c3758b5ac092a10d6f25af080d) for the change set and [Refer Here](https://docs.chef.io/resource_common/#guards) for the documentation.

## Common Resource Functionality: Notifications

* [Refer Here](https://docs.chef.io/resource_common/#notifications) for the official docs
* The syntax for notification
  + notifies: `notifies :action, ‘resource[name]’, :timer
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/6fb7c7f74ca0a135159de46bede3f1203890e602) for usage of notifies

NOVEMBER 10, 2021

# DevOps Classroom Series – 10/Nov/2021

## Notifications

* A resource may listen to another resource and then take action if the state of the resource being listened changes

file '/var/www/html/info.php' do

content '<h1> hello </h1>'

end

service 'apache2' do

action :nothing

subscribes :restart, 'file[/var/www/html/info.php]', :immediately

end

file '/tmp/apache2' do

action :nothing

subscribes :touch, 'file[/var/www/html/info.php]', :immediately

end

## Activity: Installing tomcat9 on ubuntu 20.04

* [Refer Here](https://linuxize.com/post/how-to-install-tomcat-9-on-ubuntu-20-04/) for the manual steps
* Let’s execute manual steps on the machine created using test kitchen
* Create a new cookbook tomcat9
* Initially lets automate till tomcat installation and enabling tomcat service
* Improvements:
  + Writing all the resources in one recipe is not a good idea. So lets create different recipes.
  + Let’s start by using attributes
* Best Practices:
  + default.rb in recipes should just have resources to fail if they are running on unsupported platform or platform family and it should call other recipes to automate the deployment
* Recipe name => Generally recipe name is <cookbook-name>::<recipe-name-with-out-extension>. If we do not pass recipe name chef will assume it to be default <cookbook-name> => <cookbook-name>::default
* Let’s generate a recipe for java installation
* Attributes so far

if platform?('ubuntu')

default['tomcat9']['java\_package'] = 'openjdk-11-jdk'

end

* recipe => java

#

# Cookbook:: tomcat9

# Recipe:: java

#

# Copyright:: 2021, The Authors, All Rights Reserved.

apt\_update 'update ubuntu packages' do

ignore\_failure true

action :update

only\_if { platform?('ubuntu') }

end

package node['tomcat9']['java\_package'] do

action :install

end

* recipe => default

#

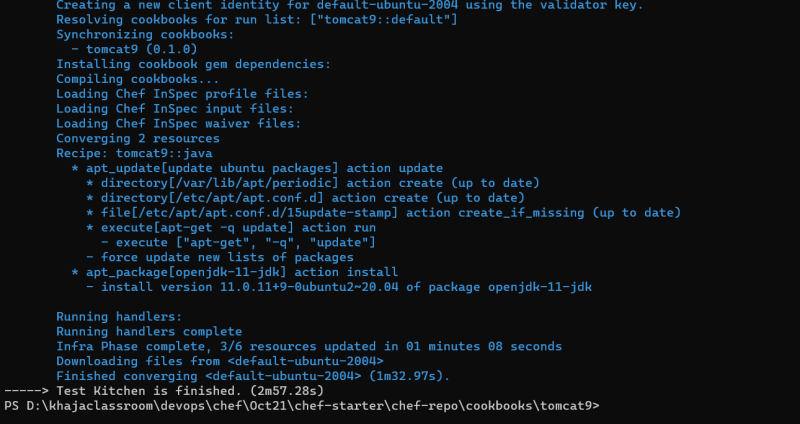
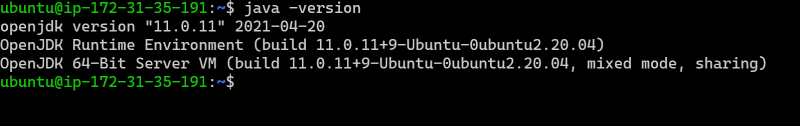
# Cookbook:: tomcat9

# Recipe:: default

#

# Copyright:: 2021, The Authors, All Rights Reserved.

include\_recipe 'tomcat9::java'

* [Refer Here](https://github.com/asquarezone/ChefZone/commit/53996d120eff952951553c7c7ddffd5d3e354657) for the changes done
* Now lets try to converge to check if the java 11 is getting installed or not
  + kitchen converge 
  + login & check for java installation 
* Next step to be automated

sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat

-m, --create-home

-d, --home-dir /opt/tomcat

-U, --user-group create a group with the same name as the user => tomcat

-s, --shell SHELL login shell of the new account /bin/false

* [Refer Here](https://docs.chef.io/resources/user/) for the user resource
* For this lets create a new recipe configure
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/57b4879fcf4b2989b768edfcb73dd7dcd76e370b) for the changes done
* Now add this recipe to default.rb in recipes

include\_recipe 'tomcat9::java'

include\_recipe 'tomcat9::configure'

* Fixed the group not existing issue and added configure to be included from default in this changeset [Refer Here](https://github.com/asquarezone/ChefZone/commit/b98ad57041d8329e37cec51839effcd632086e5c)
* Exercise: Try to automate the steps which we have done for ubuntu in this cookbook to make it work for centos7 following the documentation [Refer Here](https://linuxize.com/post/how-to-install-tomcat-9-on-centos-7/)

NOVEMBER 11, 2021

# DevOps Classroom Series – 11/Nov/2021

## Tomcat Deployment Continued

* The link to download tomcat for version 9.0.54 is <https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.54/bin/apache-tomcat-9.0.54.tar.gz>
* To download the file [Refer Here](https://docs.chef.io/resources/remote_file/) for the remote\_file resource and to extract the tar file [Refer Here](https://docs.chef.io/resources/archive_file/)
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/849657b2fd965bb3d715eac319f7d688e0c643ff) the changeset which contains download and extract tomcat
* link resource in chef is used to create symbolic and hard links [Refer Here](https://docs.chef.io/resources/link/)
* Used a link resource which is failing in convergence and we will fix this in our next session [Refer Here](https://github.com/asquarezone/ChefZone/commit/a4eb4a9b8d3825b8ddffb96b858e7e4456ffb9cf)

NOVEMBER 12, 2021

# DevOps Classroom Series – 12/Nov/2021

## Tomcat Installation contd..

* As link resource was giving some issue for now we have replaced with execute resource and made the changes [Refer Here](https://github.com/asquarezone/ChefZone/commit/d1d02ccfc631d5ac341efd27e129866a73030b0c) for the changes done [Refer Here](https://docs.chef.io/resources/execute/#syntax) for the execute resource documentation.
* Now we need to create a file for tomcat service at ‘/etc/systemd/system/tomcat.service’ with the following content

[Unit]

Description=Tomcat 9 servlet container

After=network.target

[Service]

Type=forking

User=tomcat

Group=tomcat

Environment="JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64"

Environment="JAVA\_OPTS=-Djava.security.egd=file:///dev/urandom -Djava.awt.headless=true"

Environment="CATALINA\_BASE=/opt/tomcat/latest"

Environment="CATALINA\_HOME=/opt/tomcat/latest"

Environment="CATALINA\_PID=/opt/tomcat/latest/temp/tomcat.pid"

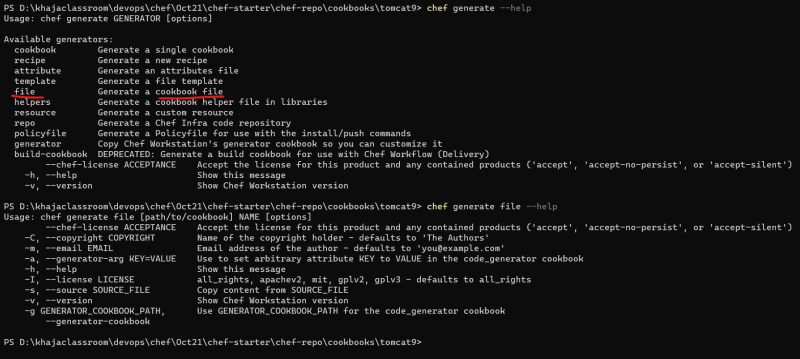
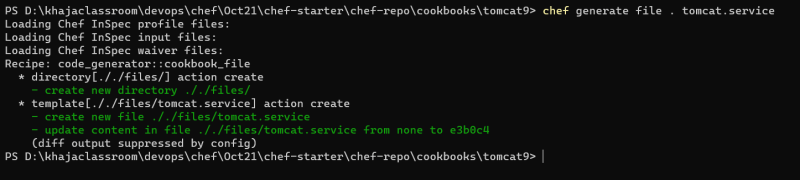
Environment="CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/latest/bin/startup.sh

ExecStop=/opt/tomcat/latest/bin/shutdown.sh

[Install]

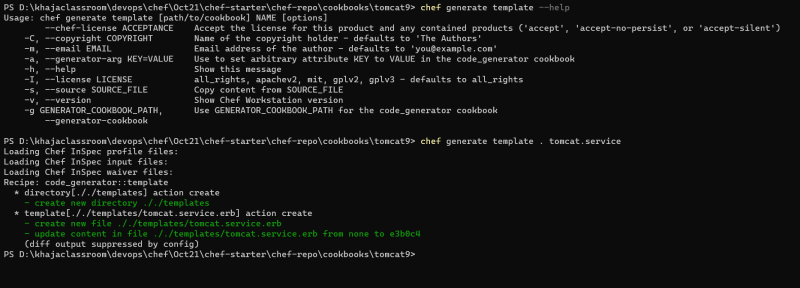
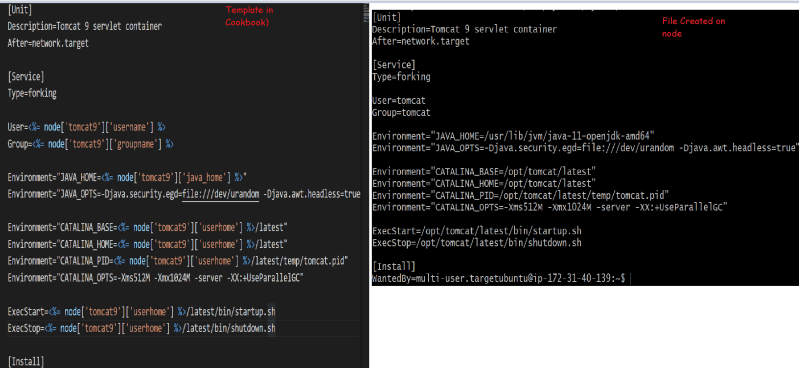
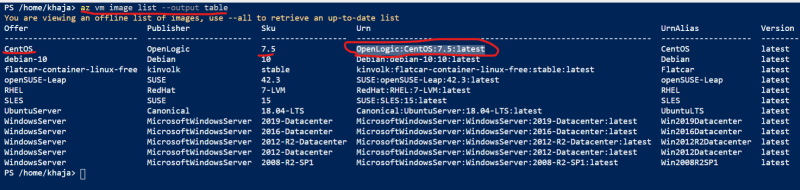
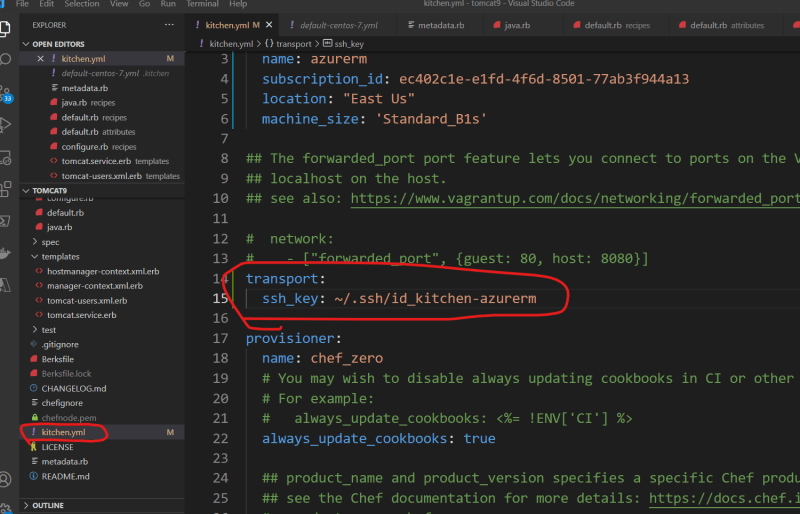
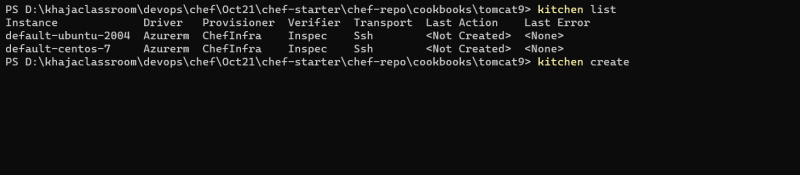
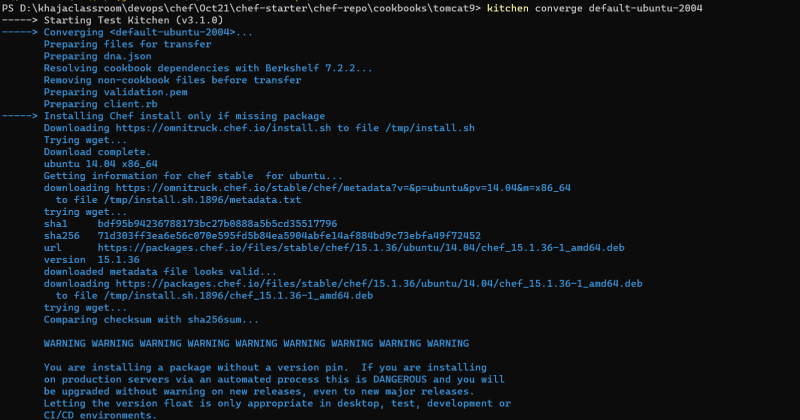
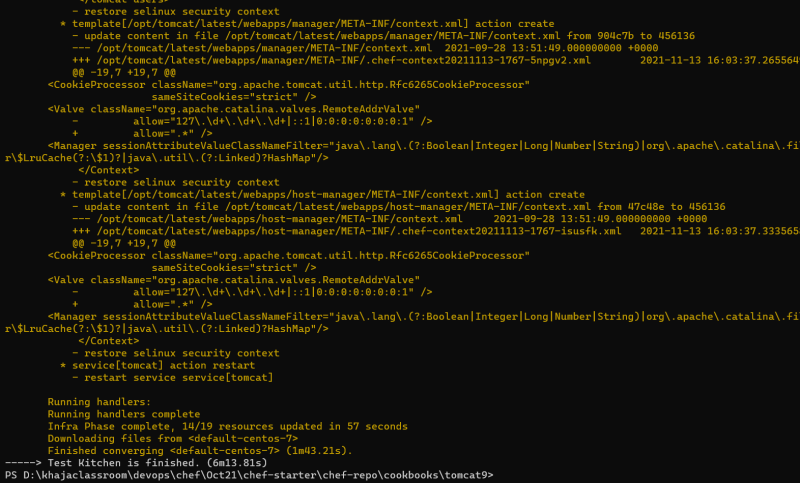
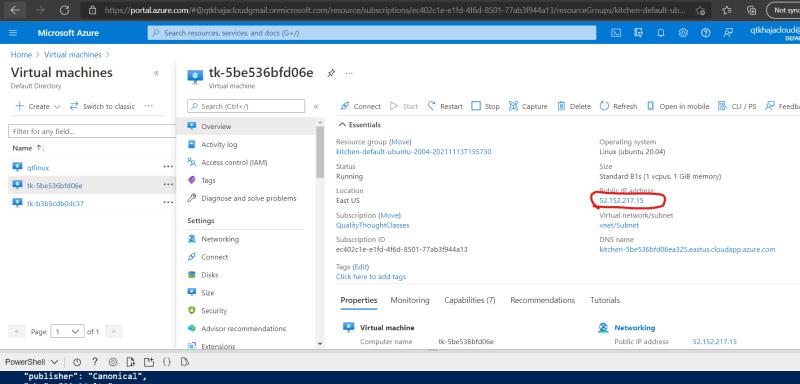
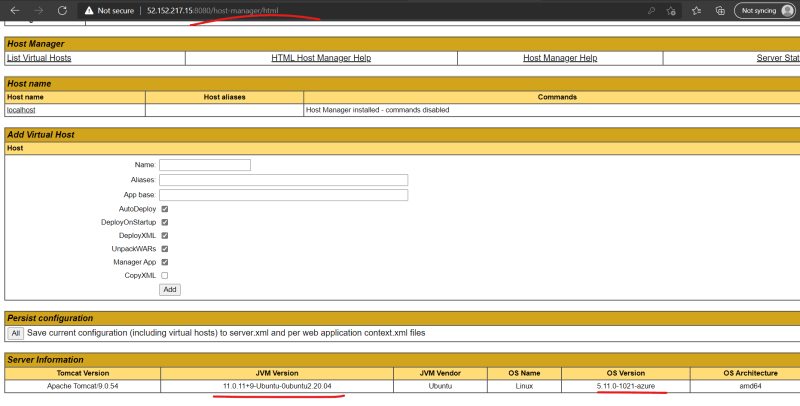
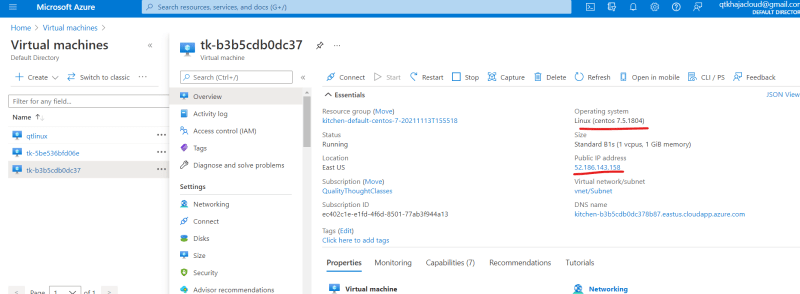
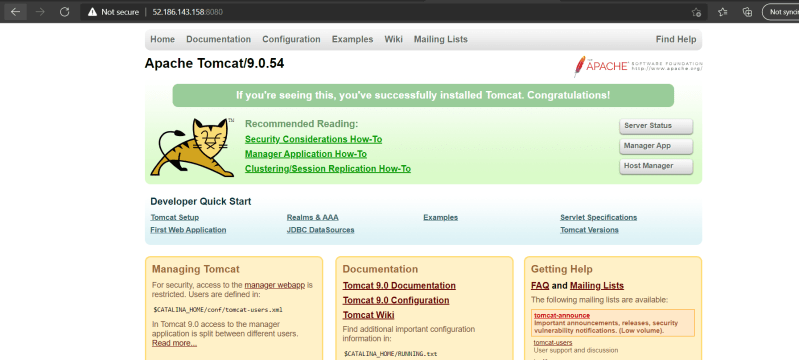
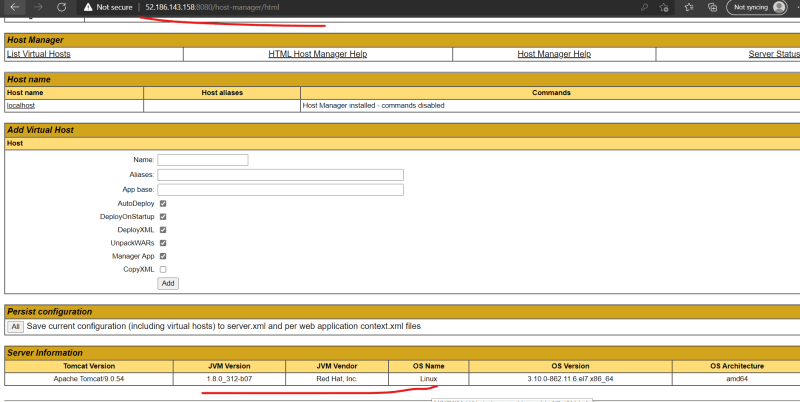
WantedBy=multi-user.target

* In chef we have files which can be used to copy static files into nodes 
* Generate a file. A new folder called as files will be created in your cookbook with a file tomcat.service 
* In chef we have a resource cookbook\_file which is used to copy the cookbook file to a specific location on the node [Refer Here](https://docs.chef.io/resources/cookbook_file/)
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/8e22570784c77b96d2acc92e6f6312c6837d1597) for the steps till tomcat service is started.
* Exercise-1: Try to make the cookbook work for centos 7 as well.
* Exercies-2: Write a chef cookbook to install nop commerce on ubuntu 20.04 (leave the database part) [Refer Here](https://docs.nopcommerce.com/en/installation-and-upgrading/installing-nopcommerce/installing-on-linux.html)

NOVEMBER 13, 2021

# DevOps Classroom Series – 13/Nov/2021

## Chef Templates

* [Refer Here](https://docs.chef.io/templates/) for the official documentation
* Chef templates is an Embedded ruby template (ERB) so in files we can have dynamic content
* Lets try to create a template 
* Dynamic expressions in embedded ruby should be <%= %>
* When we try to copy the template using template resource, the file will evaluated all the expression in <%= %> will be replaced by its dynamic values. [Refer Here](https://docs.chef.io/resources/template/) for the template resource
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/74b7693e471252a38300c76b298dba3d12a7b5f2) for the changeset 
* Now lets generate tomcat-users file template
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/2821352cf025543643546ab3b6e64d7d1664b1e2) for the changeset containing all of the steps to install tomcat9 as per documentation [Refer Here](https://linuxize.com/post/how-to-install-tomcat-9-on-ubuntu-20-04)
* Now lets also add support for centos 7
  + [Refer Here](https://github.com/asquarezone/ChefZone/commit/55090ffff72e089e05a7f0ef63fbf34dae068c1f) for the necessary changes done in the cookbook to make it work for centos 7
* Lets try to use the test kitchen with Azure [Refer Here](https://directdevops.blog/2019/04/03/test-kitchen-setup-with-azure/) for the documentation
* Lets get image urn for centos 7 
* Search for ubuntu 20 urn by executing the following command in Azure CLI az vm image list -f UbuntuServer --all --output table & the other command which we tried az vm image list -p canonical -o table --all | grep 20\_04-lts | grep -v gen2
* Fix for the issue observed in the class add the ssh\_key path in transport in kitchen.yml file 
* Now execute the kitchen commands    
* Azure Ubuntu 20.04 VM from test kitchen   
* Azure Centos 7 VM from test kitchen   
* In the changeset which i’m commiting i have removed subscription id for security reasons. [Refer Here](https://github.com/asquarezone/ChefZone/commit/8e43c465d05cef8b43be7e791a3332480ab17a80) for the changeset

### Share this:

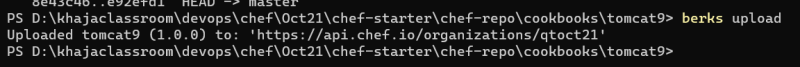
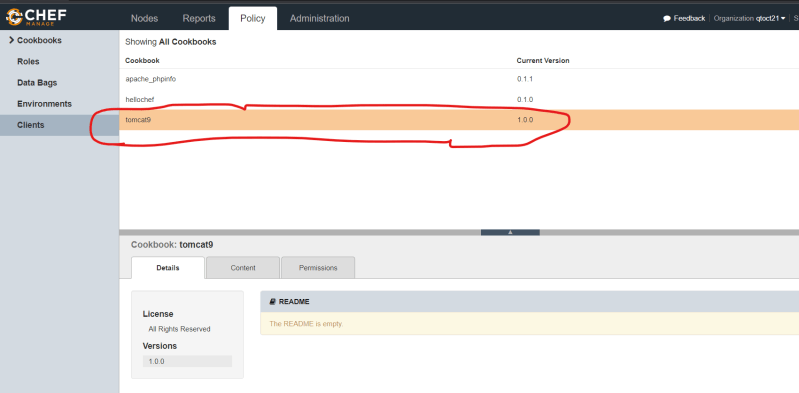
NOVEMBER 15, 2021

# DevOps Classroom Series – 14/Nov/2021

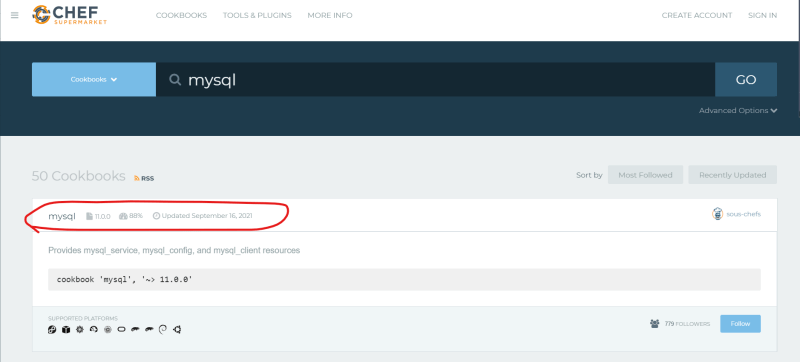
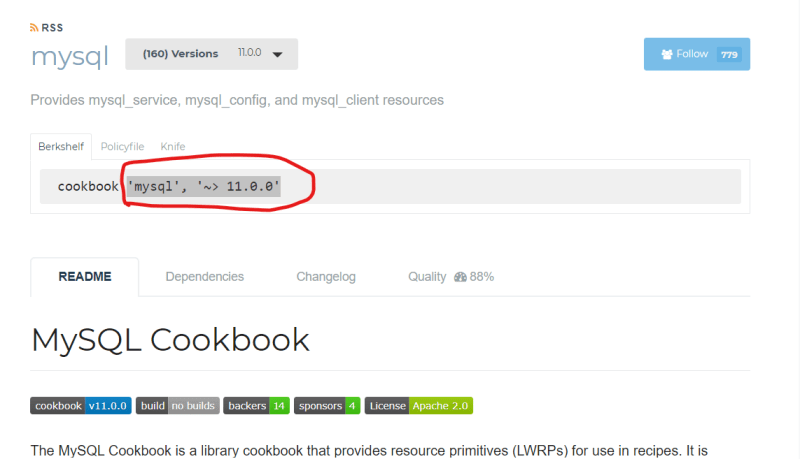
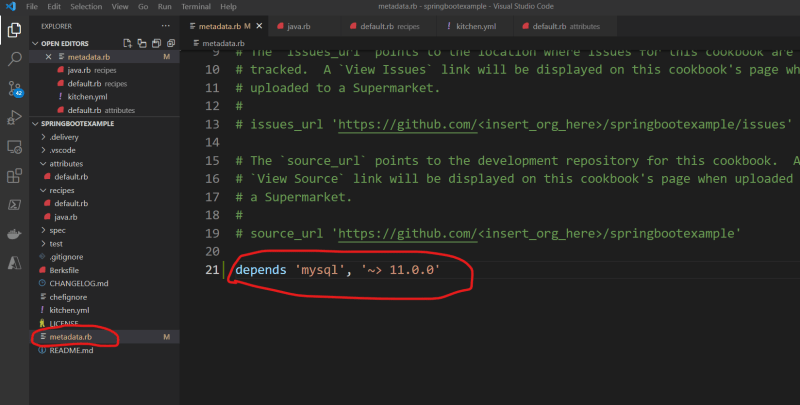
## Chef Cookstyle

* This helps in writing better chef infra cookbooks.
* Command : cookstyle <path to cookbook>
* Also apply necessary metadata as done in the class [Refer Here](https://docs.chef.io/config_rb_metadata/)
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/e92efd12d18757b0ca416a21f06f454fdae7c24a) for the changes made.

## Tomcat 9 Installation

* Now our cookbook located at [Refer Here](https://github.com/asquarezone/ChefZone/tree/master/Oct21/chef-starter/chef-repo/cookbooks/tomcat9) is properly tested and is eligible to upload to chef server.
* Now lets upload this cookbook using berks  

## Chef Supermarket

* Chef Supermarket is a platform where open source community shares their cookbooks.
* We can use those cookbooks along with the recipes, which we develop or use them straight away in the run\_list.
* [Refer Here](https://supermarket.chef.io/) for the supermarket website
* Lets try to install mysql on ubuntu 2004 using supermarket cookbook and lets also install java 11
* Lets try to find the mysql cookbook shared by community 
* [Refer Here](https://supermarket.chef.io/cookbooks/mysql) for the documentation.
* When you use supermarket cookbooks, we need to ensure we read the documentation as we have to use the cookbook as developed by the third party
* Lets create a cookbook called as springbootexample
* Lets create a recipe for installing java called as java and create an attributes file default
* Now lets configure kitchen to create ubuntu20.04 in any cloud env (in this example i will be using aws )
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/39b3ab2ae8d02de5c51a4ccb5d2b5769d049db57) for the initial set of changes
* Now we want to install mysql suing the supermarket cookbook [Refer Here](https://supermarket.chef.io/cookbooks/mysql)
* Add dependecny to your cookbook by adding the highlighted in the below image with depends in metadata.rb  
* Now to download dependencies i.e mysql and its dependencies, berkshelf (berks) is a tool which can download the dependencies from supermarket.
* Lets install the dependencies by executing berks install 
* Generally when we use cookbooks from supermarket there are two kinds of usage
  + Cookbooks which have custom resources developed
  + Cookbooks where we need to include\_recipe or add the recipe to run\_list
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/7c15ab484fe286009d2e9f48bcfc88379b14d387) for the changes done to add mysql installation using supermarket cookbook
* Lets try to add nginx installation using supermarket cookbook [Refer Here](https://supermarket.chef.io/cookbooks/nginx).
* [Refer Here](https://github.com/sous-chefs/nginx/blob/main/documentation/nginx_install.md) for the nginx installation
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/52208d9ce74a59ab5ea1981b93200bacc730806a) for the changes.

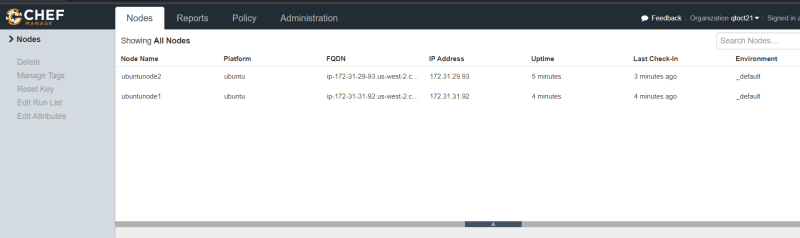
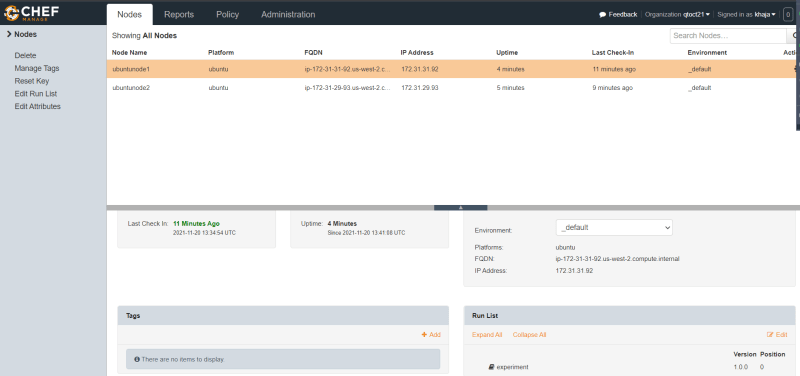
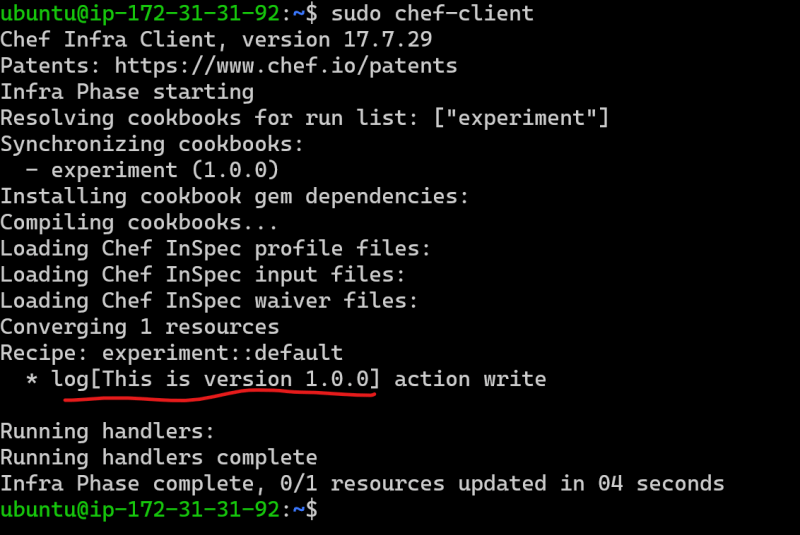
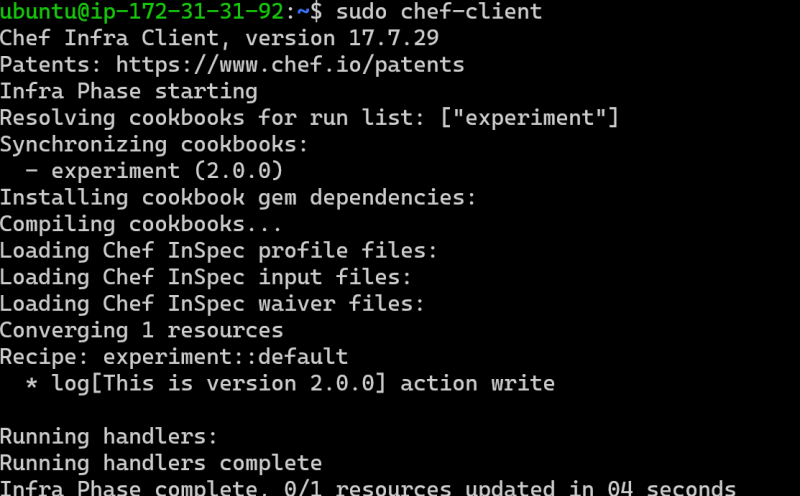
## Next Steps

* Using Roles
* Using Environments
* Using databags
* Setting Attributes
* Chef on Windows
* Installing our own chef server.

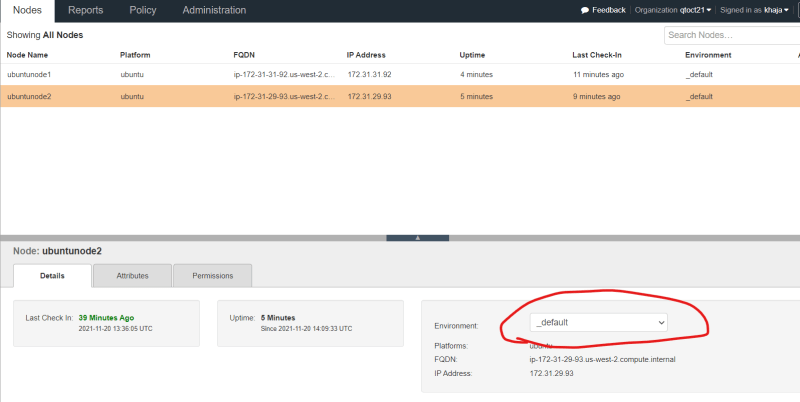
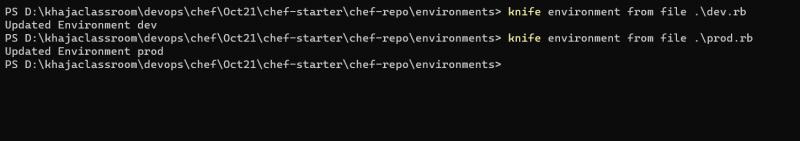
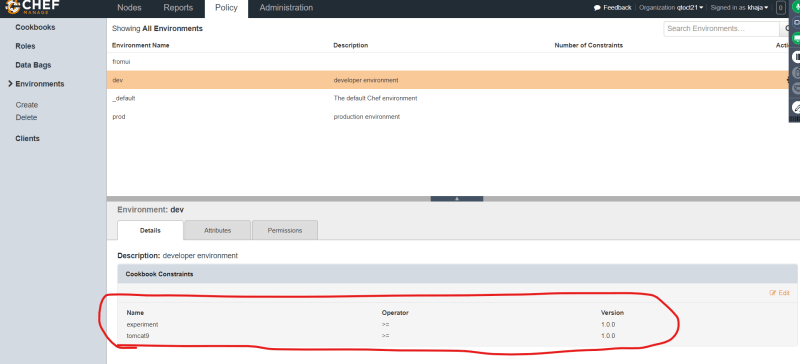
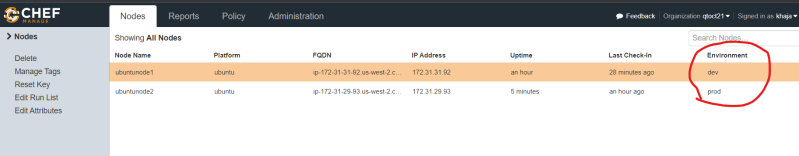
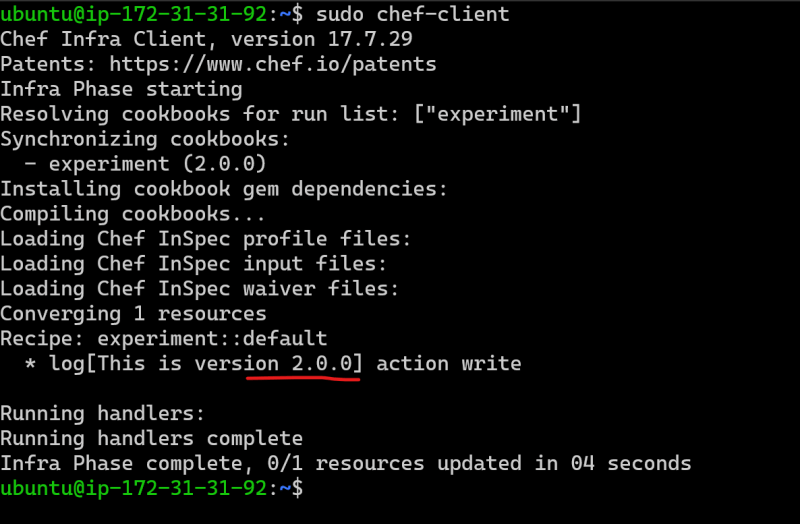
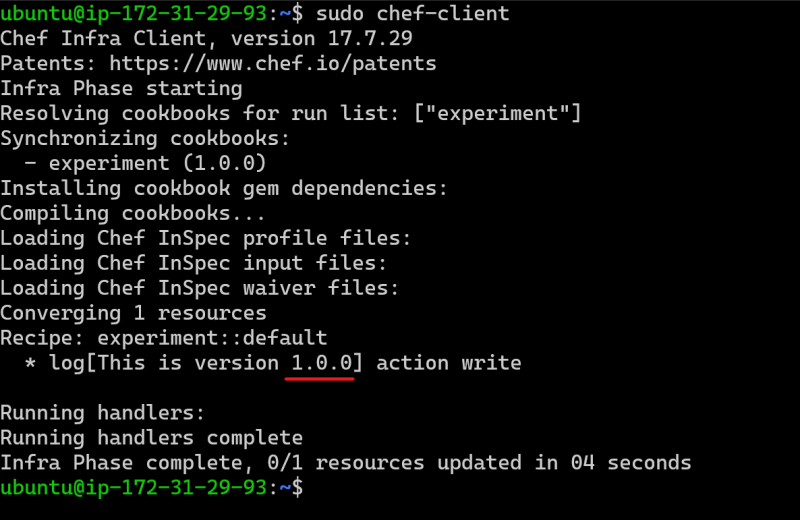
NOVEMBER 20, 2021

# DevOps Classroom Series – 20/Nov/2021

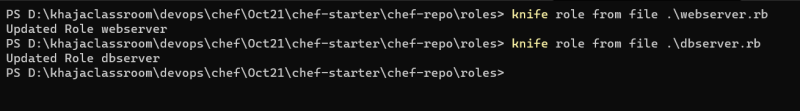
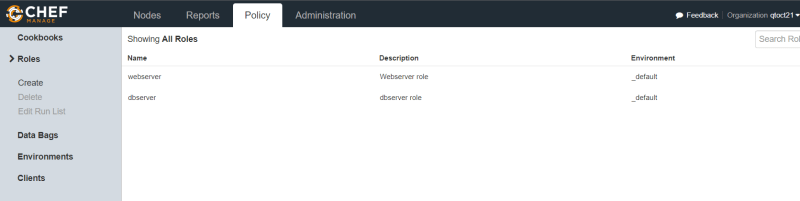
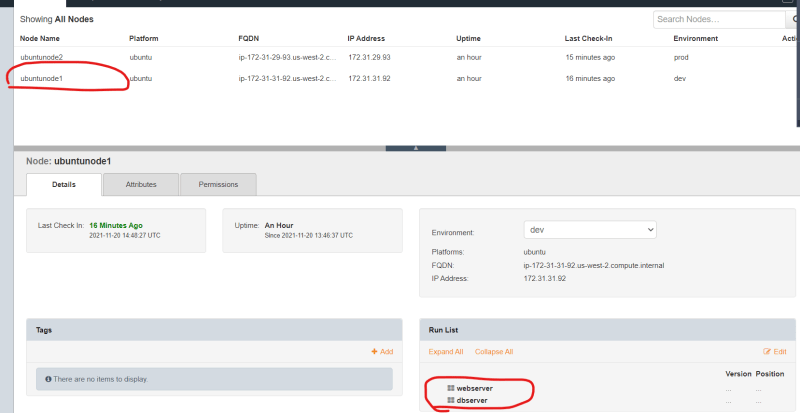
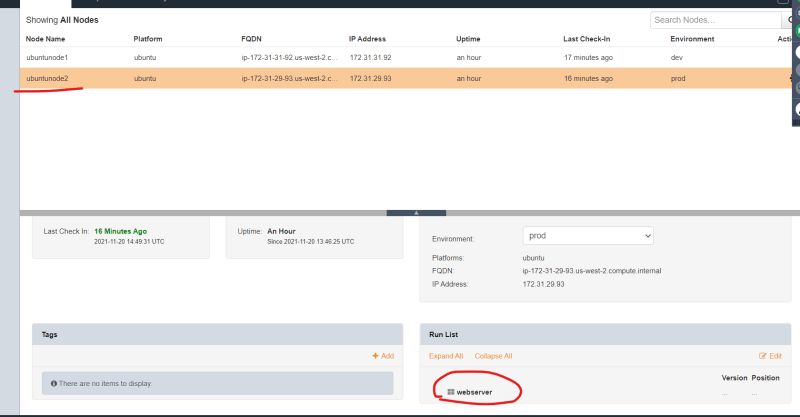
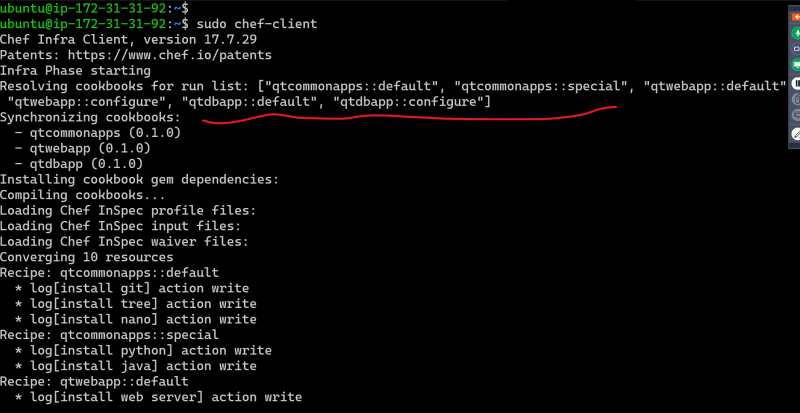
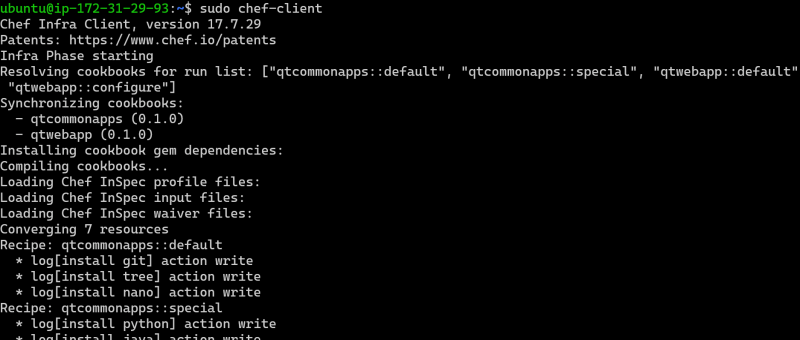
## Chef Cookbooks With Chef Server

* Lets bootstrap two nodes to the hosted chef server 
* Experiment:
  + Develop a simple cookbook with version 1.0.0 which prints this is version 1.0.0
  + use log resource [Refer Here](https://docs.chef.io/resources/log/)
  + [Refer Here](https://github.com/asquarezone/ChefZone/commit/b8006e4069b9fb4e74eeed2b2332eb3e36c81308) for the new cookbook added.
  + Now upload the recipe and this to the run\_list of the ubuntunode1 
  + Now lets login into ubuntu node 1 and manually force convergance 
  + Now lets assume we are giving uploading a new version of the cookbook (experiment 1.0.0) and this should not apply to ubuntunode1. ubuntunode1 should work with experiment with version 1.0.0
  + [Refer Here](https://github.com/asquarezone/ChefZone/commit/4e01420c8d95b4c6c50b56da4664b4d0810f59ef) for the new version of cookbook
  + Upload the new version and now login into ubuntunode1 and manually force convergence 
  + **We need to have some kind of mechanism where we need to restrict the cookbook versions to be uploaded.**
  + In Chef we can solve the problem with cookbook versions using **Environments**
  + Lets create three cookbooks
    - qtwebapp
      * default
      * configure
    - qtdbapp
      * default
      * configure
    - qtcommonapps
      * recipes:
        + default
        + special
  + Now lets assume we have three categories of nodes
    - We have db servers
    - We have web servers
    - For testing we have some nodes where we install both web and db on same node
  + To do this we need to have the following run\_lists
    - webserver => qtcommonapps::default, qtcommonapps::special, qtwebapp::default, qtwebapp::configure
    - db server => qtcommonapps::default, qtdbapp::default, qtdbapp::configure
    - web+dbserver => qtcommonapps::default, qtcommonapps::special, qtwebapp::default, qtwebapp::configure,qtdbapp::default, qtdbapp::configure
  + When we have lot of servers to deal with creating run\_lists like shown above does it make sense?
  + **We need to have some kind of a solution to manage run\_lists effectively**
  + In Chef we can solve the problem with managing run\_lists using **Roles**

## Chef Environments

* An environment is a way to map organization real-life workflow.
* With Environments we can define the below and attach environment to the node
  + cookbook versions applicable
  + attributes
* Chef has a default environment already defined \_default where there are no restrictions, 
* We can create environments like shown below and attach it to the node
  + DEV
  + QA
  + STAGING
  + PROD
* There are two ways of creating environments,
  + Using Chef Managment console: Easy but not recommended
  + Create a file in the chef-repo and upload: In this we are having a file which represents environment and we can easily track all the changes done to a file which represents environment as we version control it.
* [Refer Here](https://docs.chef.io/environments/) for the official documentation
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/be59aac4e883a6af56bf9c308295fc7e0ef2eb19) for the changeset.
* Now lets try to upload environments knife environment from file [Refer Here](https://docs.chef.io/workstation/knife_environment/#from-file) for documentation  
* Now lets apply the same runlist of experiment to ubuntunode1 and ubuntunode2.
* Now lets make the ubuntunode1 environment => dev and ubuntunode2 environment => prod 
* converge results of ubuntunode1 
* converge results of ubuntunode2 

## Roles

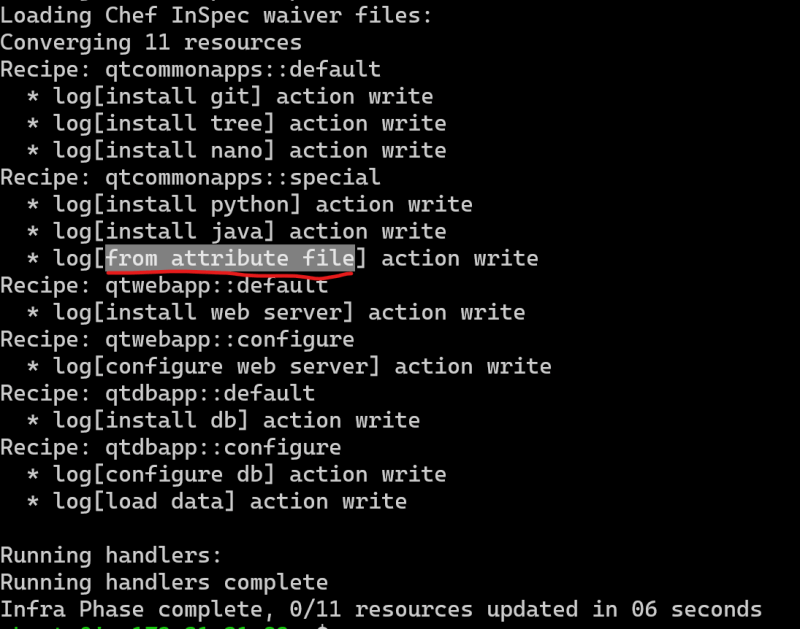
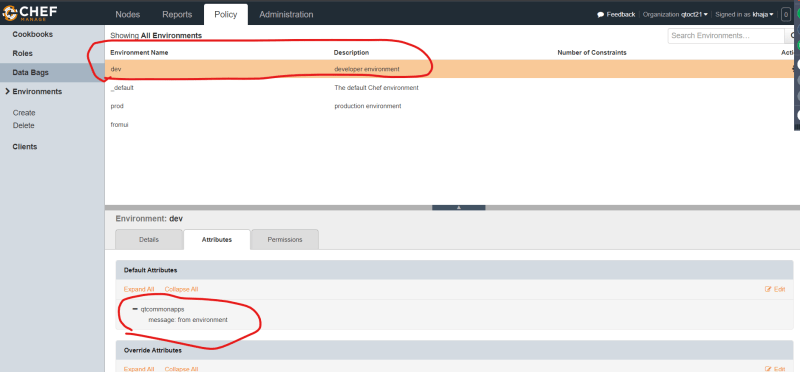
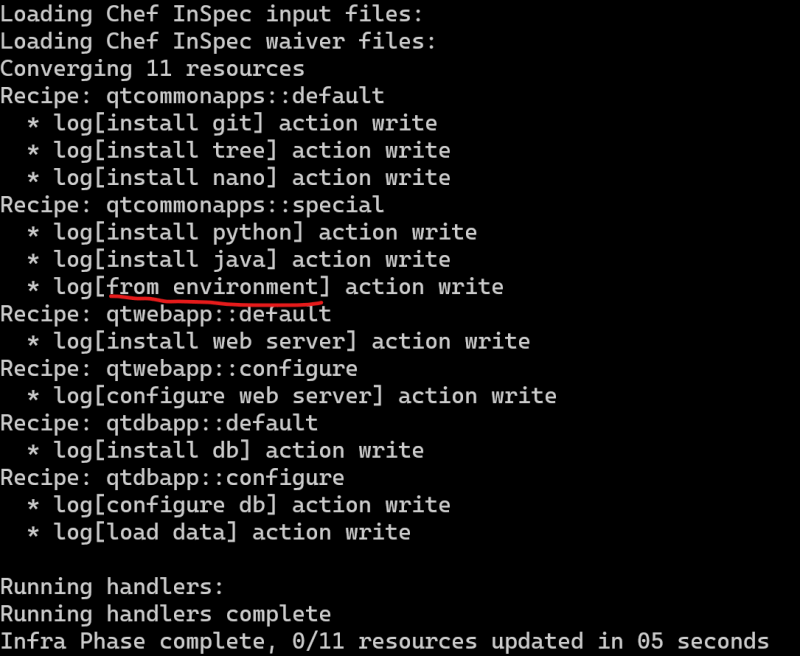
* Role consists of run\_list and attributes
* To the nodes run\_list we can add roles directly instead of individual recipes
* To do this we need to have the following run\_lists
  + webserver => qtcommonapps::default, qtcommonapps::special, qtwebapp::default, qtwebapp::configure
  + db server => qtcommonapps::default, qtdbapp::default, qtdbapp::configure
  + web+dbserver => qtcommonapps::default, qtcommonapps::special, qtwebapp::default, qtwebapp::configure,qtdbapp::default, qtdbapp::configure
* Now first lets upload qtcommonapps, qtwebapps and qtdbapss. [Refer Here](https://github.com/asquarezone/ChefZone/commit/4ba32e9e941843c168ac4e7f65166fcace8c187a) for the changeset with all the three cookbooks
* Now lets create the roles, As environments we can create using console and from file. [Refer Here](https://docs.chef.io/roles/#chef-language) for the official docs
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/2b52d97730825a97afe50a6209ab131f6ca1436a) for the changeset
* [Refer Here](https://docs.chef.io/workstation/knife_role/#from-file) for the command to upload the roles to chef server  
* Now lets add the roles
  + webserver+dbserver to the run\_list of ubuntunode1 
  + webserver to the run\_list of ubuntunode2 
* Now lets manually force the convergence
  + ubuntu node1: => webserver+dbserver 
  + ubuntu node2: => webserver 

### Share this:

NOVEMBER 22, 2021

# DevOps Classroom Series – 21/Nov/2021

## Chef Attribute Precedence

* Chef attributes can be created in
  + attribute files
  + recipes
  + Environment
  + roles
* Chef attributes have different types
  + default
  + force\_default
  + normal
  + override
  + force\_override
  + automatic
* Automatic is set to the attributes collected by ohai, we cannot set automatic type to attributes
* Depending on type and where we have an attribute an weight is associated. When the same attribute is defined at two different places, then attribute with highest wait wins during chef convergence. This is called as attribute precedence. [Refer Here](https://docs.chef.io/attribute_precedence/) for the offical docs 
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/a8142987a6b966bfd3cabcffd0fe85dab4a6cb49) for the changeset
* Upload the cookbook and run the convergence on ubuntunode1 
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/9c6e0b6515330c9e73e608a5b7cbcb1b70d1d93b) for the changeset containing environment attribute 
* Now lets manually force the convergence 

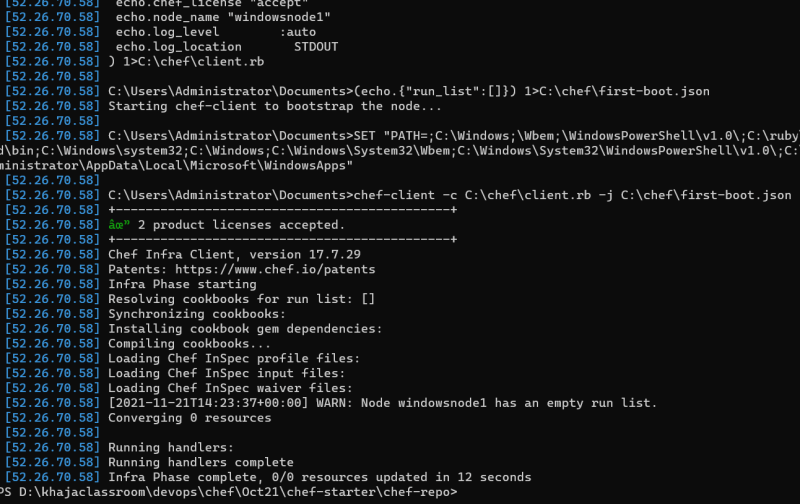
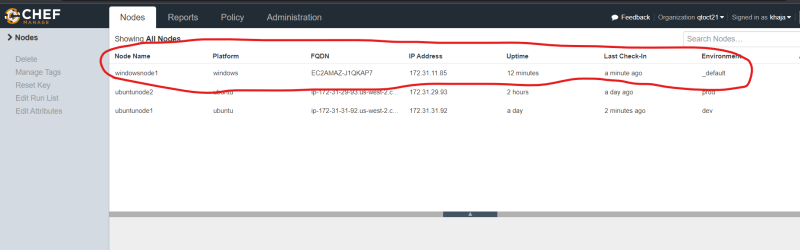
## Changing Convergence

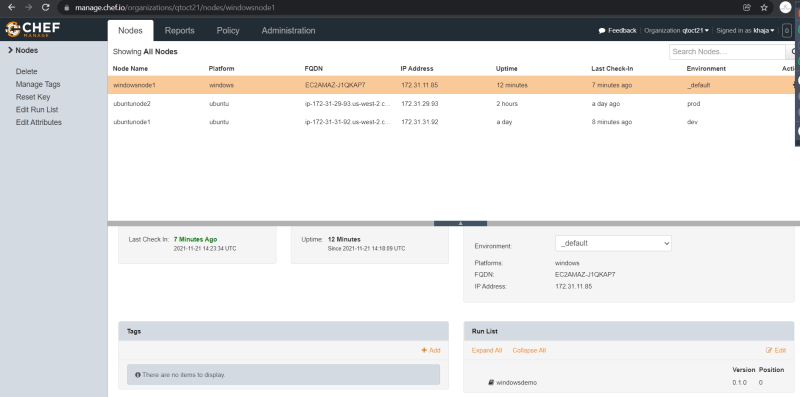
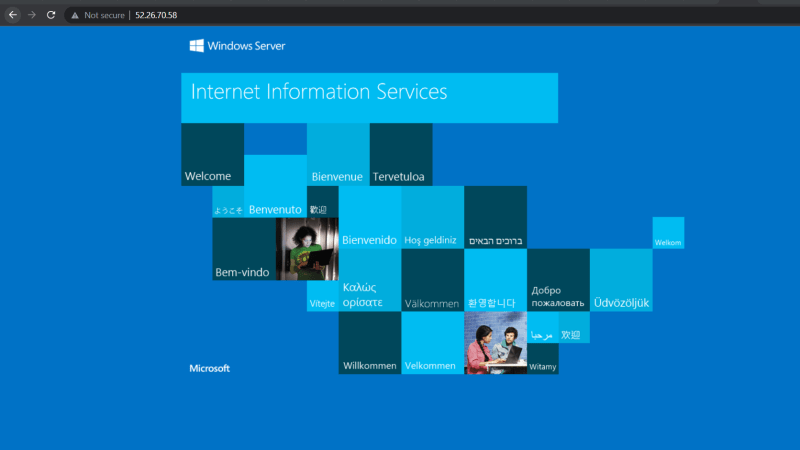
* We can use chef supermarket cookbook [Refer Here](https://supermarket.chef.io/cookbooks/chef-client) to change the convergence. We can use the resources of this cookbook to update the chef client configuration.
* In the latest version of chef from 16.0, these resources are made availbale normally [Refer Here](https://docs.chef.io/resources/#chef-client-scheduled-task-resource) for windows
* If you want multiple nodes to converge exactly at same time on all the nodes use this resource [Refer Here](https://docs.chef.io/resources/#chef-client-cron-resource) and try to give same cron expression for linux
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/6f889725399efd9a913799f57826e043f0078f9b)

## Windows Node in Chef

* Create a Windows 2016 server
* [Refer Here](https://directdevops.blog/2021/10/31/devops-classroom-series-30-oct-2021-2/) to this documentation in the Ansible with Windows Section to configure the Windows node
* After configuring winrm now execute bootstrap command
* e have bootstrapped using the following command

knife bootstrap 52.26.70.58 -U Administrator --winrm-basic-auth-only --winrm-no-verify-cert -P 'XDR5\*dyco7?ZgF@4P5ctdym?Trt@ugfQ' -N 'windowsnode1' -o winrm

* Now lets try to write a cookbook which will install iis server on windows
* To install iis server from powershell Install-WindowsFeature -name Web-Server -IncludeManagementTools
* Lets try to create one cookbook and configure iis server [Refer Here](https://github.com/asquarezone/ChefZone/commit/1692b6a753bd392057d94a45714c05ec2a04802f) for the changes 
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/67a497cd89f6e5ceecf4a7464ebf10617765a6a8) for the fix done to perform installation 

## Install Chef Server

* [Refer Here](https://docs.chef.io/server/install_server/) for the official docs
* Commands executed

sudo apt update

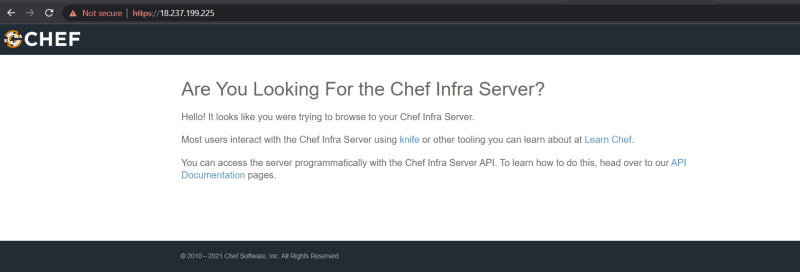
wget https://packages.chef.io/files/stable/chef-server/14.10.23/ubuntu/20.04/chef-server-core\_14.10.23-1\_amd64.deb

sudo dpkg -i chef-server-core\_14.10.23-1\_amd64.deb

sudo chef-server-ctl reconfigure

sudo chef-server-ctl user-create qtdevops QT devops qtdevops@gmail.com 'motherindia@123' --filename /home/ubuntu/qtdevops.pem

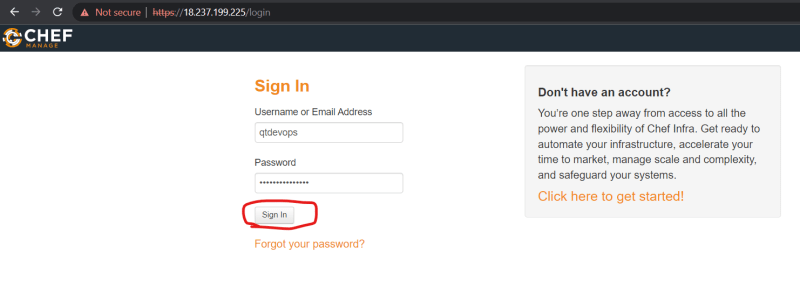
sudo chef-server-ctl org-create qthought 'Quality Thought' --association\_user qtdevops --filename /home/ubuntu/qthought-validator.pem

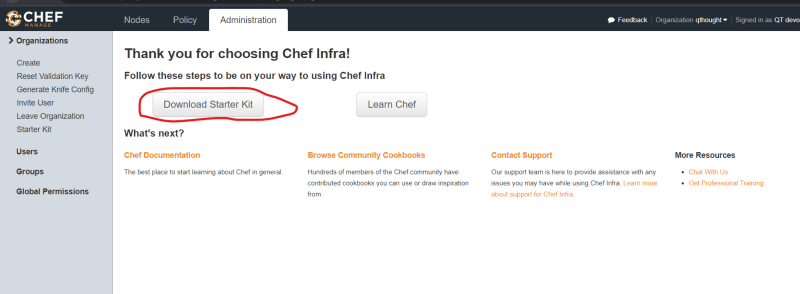
* Lets try to access server [https://publicip](https://publicip/) 
* Now execute the following commands

sudo chef-server-ctl install chef-manage

sudo chef-server-ctl reconfigure

sudo chef-manage-ctl reconfigure



* Now you can download the starter kit and proceed with cookbook development 

### Share this: