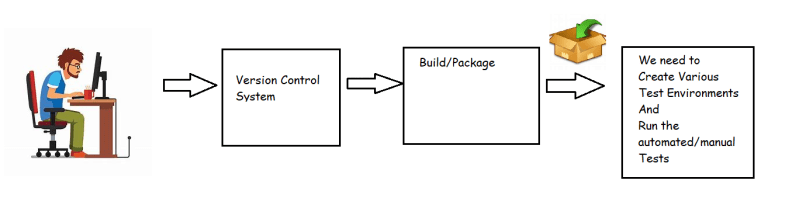
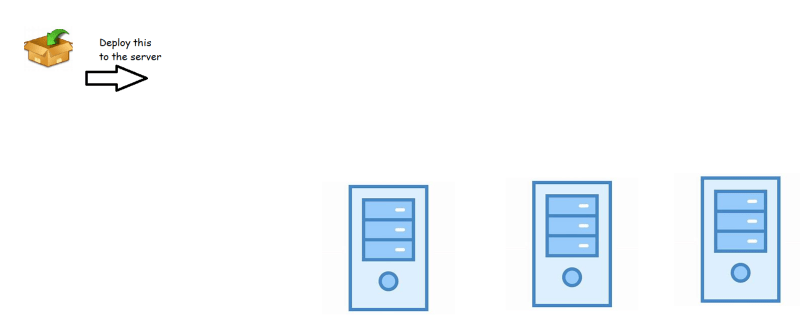
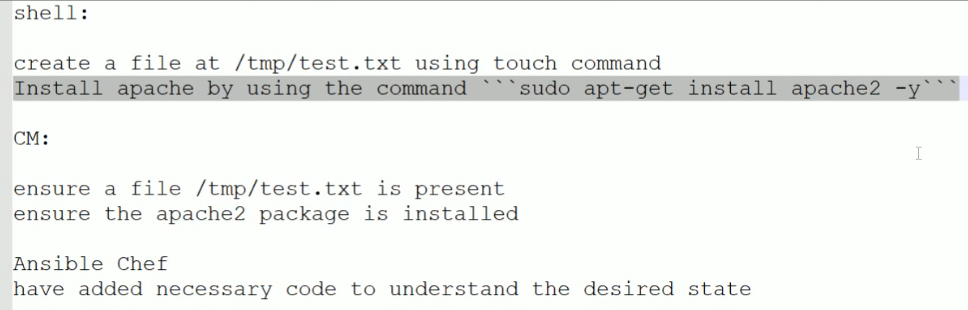
25/May/2021-chef-1

DevOps Pipeline and Configuration Management

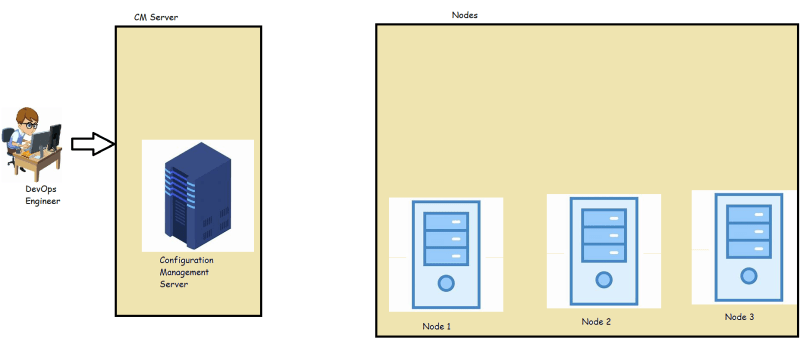
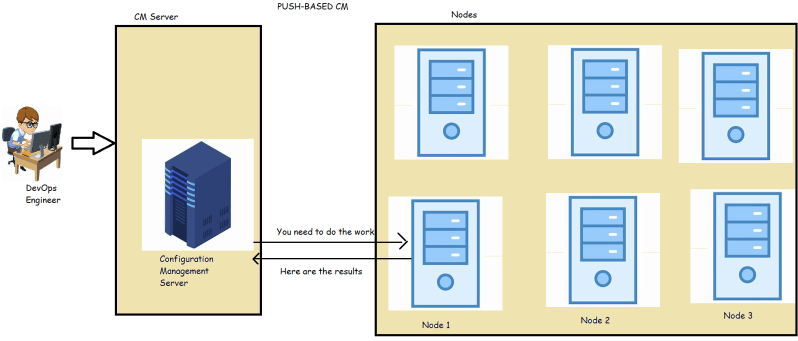
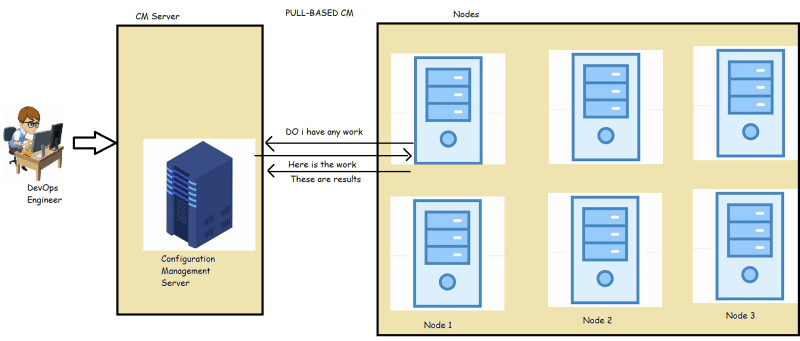
* Once the Developer commits the code to the VCS, the build system picks the code and generates the software Package. 
* We need to take this software package and then create the Test Environments 
* In Enterprise:
  + Lot of developers will be committing the code and then we need make lot of software deployments in the servers before we go for the live environment
  + We need to deploy the package into various environments
    - Environment for System Tests
    - Environment for Performance Tests
    - Environment for UAT
    - Live/Production Environment
* To Summarize:
  + We need to do this activity of deploying and configuring application repetitively
  + Manual deployments are ruled out

Possible Solutions for automating deployments



* Solution 1: Write a Shell/PowerShell Script to deploy and configure the application
  + The effectiveness of the script largely depends on the developer
  + Shell Scripts are not readable
  + It becomes difficult to handle changes and maintaining the solution
  + Making the script work on different Linux Distributions is slightly complex
* Solution 2: Configuration Management (CM)
  + In Shell Scripts, we script on the basic philosophy How to do” but CM works on philosophy “what has to be done” i.e. we specify a desired state.

Configuration Management

* An effective configuration management solution can benefit your environment in a number of ways including
  + Saving Time
  + Improving Availability
  + Improve Control
  + Allow us to Do more with Less
* Configuration Management Architectures   
* Ansible, Salt uses Push Based CM
* Chef, Puppet follows the Pull Based CM

PULL BASED CM

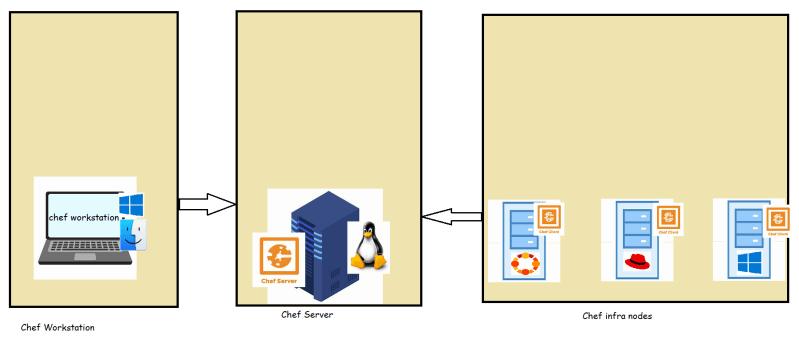
* In Pull based CM for the nodes to communicate with the CM server, there will be a software agent (chef agent, puppet agent) which will trigger the communication from node to CM Server
* DevOps/IT Engineers need to express the desired state in the supported format by the CM Server and upload the configurations to the server
* So let’s learn about how to write the configurations in Chef and upload them to the chef CM Server and make that configuration work on some node.

====================================

# CHEF-2 26/May/2021

## **Chef Infra Overview**

* The following image shows the relationship b/w various components of Chef infra, included the nodes, the server and the workstation



### Chef Infra Server

* This can be installed on Linux machines (Virtual/Physical)
* The software that needs to be installed on the Linux Machine is Chef Server software “https://downloads.chef.io/tools/infra-server”
* The chef infra server i.e. installed on Linux Machines is free for usage till 10 Nodes
* Chef also offers a hosted chef server (chef pre-installed) manage.chef.io which is free for usage till 5 nodes
* Chef Server can store
  + cookbooks
  + run\_list: This is list of recipes to be applied on each node. Chef Server maintains one run\_list per node
  + secrets
  + policy

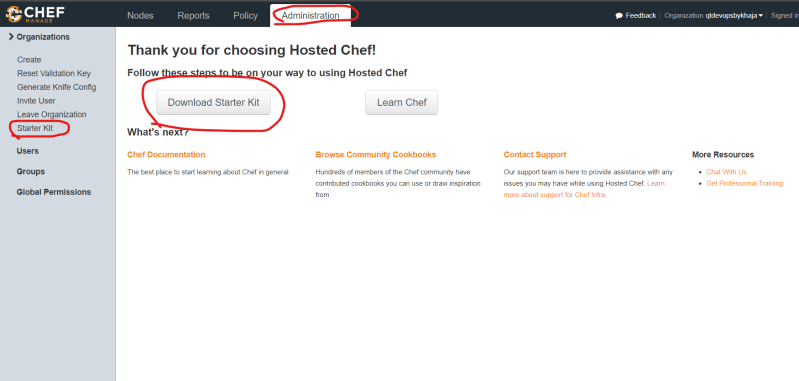
### Chef Workstation

* In Chef Architecture, we can have multiple workstation
* In Chef, we develop cookbooks
* Cookbook is collection of recipes
* Each recipe will have resources that needs to be arranged in some order.
* Using Chef Workstation, we can author, test and maintain cookbooks
* Chef Workstation also provides tools to test your cookbooks
* Chef Workstation is packaged with tools such as
  + Chef infra client
  + Chef InSpec
  + Test Kitchen
  + Cook style
  + Command line tools:
    - Knife
    - Chef

### Nodes

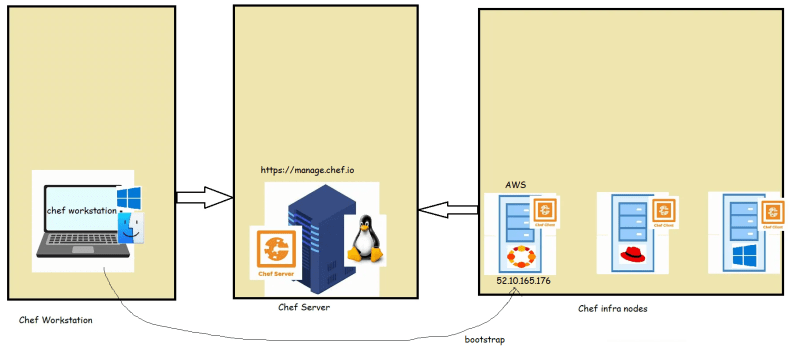
* A node is any device- Physical, virtual, cloud, network device etc.
* On the nodes the chef-infra client software should be running and it should communicate with the chef server.
* To make any Linux/windows system as Chef node, we need to install and configure chef-client. This process of installing chef-client on a node is called as bootstrapping

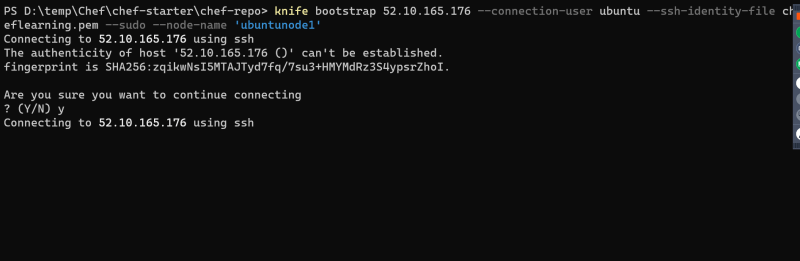
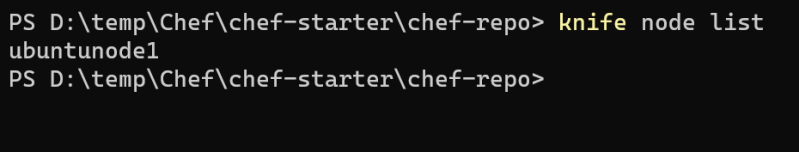
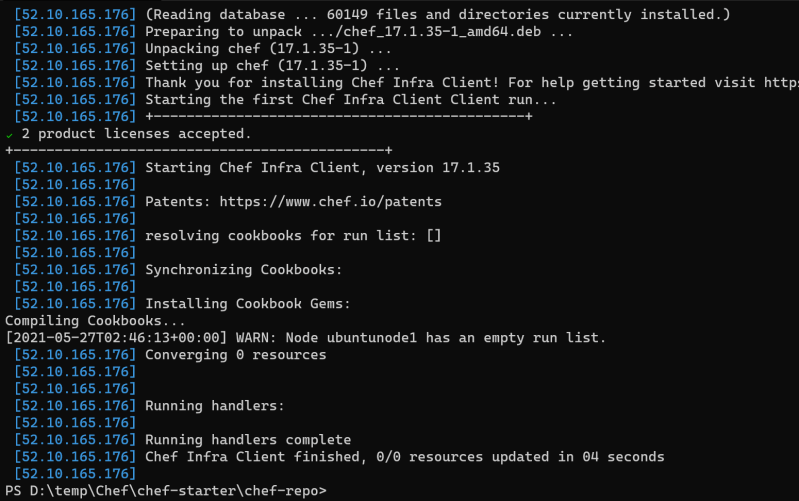
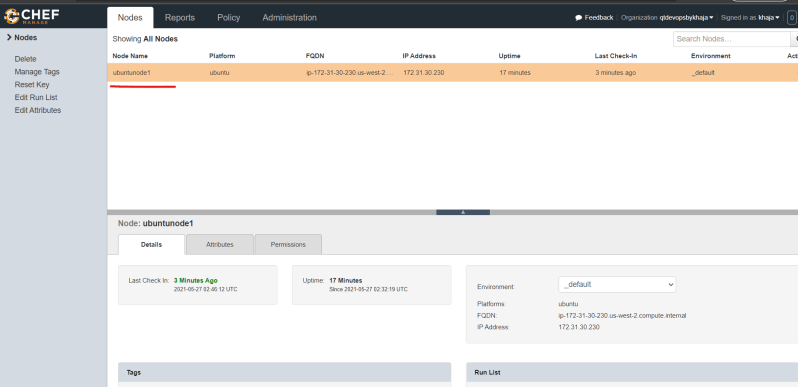
### Chef Workflow Practical View

* Create an account in manage.chef.io as we get the hosted chef server.
* Download the starter kit. 
* Download the chef workstation on your system [Refer Here](https://downloads.chef.io/tools/workstation) (<https://downloads.chef.io/tools/workstation>)  and install it

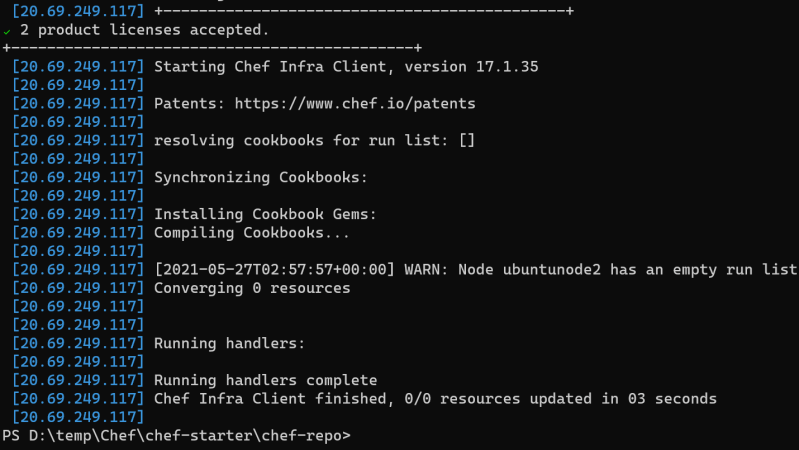
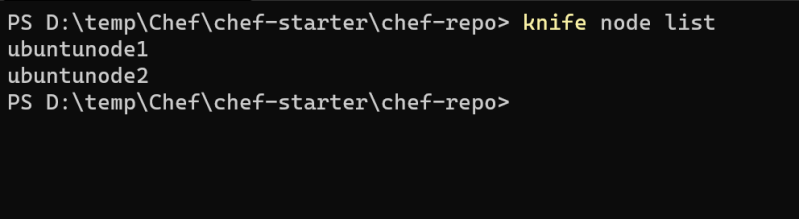
# CHEF-3 27/May/2021

### Chef Node Setup and bootstrapping

* We have configured the chef workstation and we have create account in manage.chef.io which is a hosted chef server
* Now we need to create a node.
* Note:
  + Creating a VM in Azure [Refer Here](https://www.youtube.com/watch?v=P9X-4Z-NeGg&list=PLuVH8Jaq3mLuqXuGs6aeqxhuvCYSzB1kT&index=2) <https://www.youtube.com/watch?v=P9X-4Z-NeGg&list=PLuVH8Jaq3mLuqXuGs6aeqxhuvCYSzB1kT&index=3>
  + Create a VM in AWS [Refer Here](https://www.youtube.com/watch?v=me2s3mTNwGo&list=PLuVH8Jaq3mLszrC7lv68a0VcrDripW-HK&index=2) (<https://www.youtube.com/watch?v=me2s3mTNwGo&list=PLuVH8Jaq3mLszrC7lv68a0VcrDripW-HK&index=3>)
  + 
* Lab setup for the node-1
  + Create a ubuntu 20 Linux VM (EC2 instance) in AWS and then we will try to boot strap
  + Now we will try to bootstrap the VM Created in AWS using Chef Workstation
  + To do the bootstrap, we need
    - IP address of node
    - Username and password/key file
    - User needs to have sudo permissions because chef client has to be installed.
  + Steps:
    - Ensure you have SSH connection from workstation => Node
    - Workstation needs to provide some info about chef server to the node so the bootstrap should be executed from chef-repo
    - Build a bootstrap command using knife [Refer Here](https://docs.chef.io/workstation/knife_bootstrap/) (<https://docs.chef.io/workstation/knife_bootstrap/>)
  + knife bootstrap 52.10.165.176 --connection-user ubuntu --ssh-identity-file cheflearning.pem --sudo --node-name 'ubuntunode1'

* Lab setup for the node-2:
  + Create a ubuntu 20 Linux VM in Azure and then lets try to bootstrap
  + Building a bootstrap command
* knife bootstrap <public ip> --connection-user 'qtdevops' --connection-password '<your ip>' --sudo --node-name 'ubuntunode2'

* Exercise:

1. Bootstrap an ubuntu node to the hosted chef server.
2. Bootstrap a RedHat node to the hosted chef server.
3. Create a ubuntu VM and then try to install apache server and tomcat server

### Next Steps

* Creating a Cookbook
* Writing simple recipes in the cookbook
* Using resources in recipes

===========================================

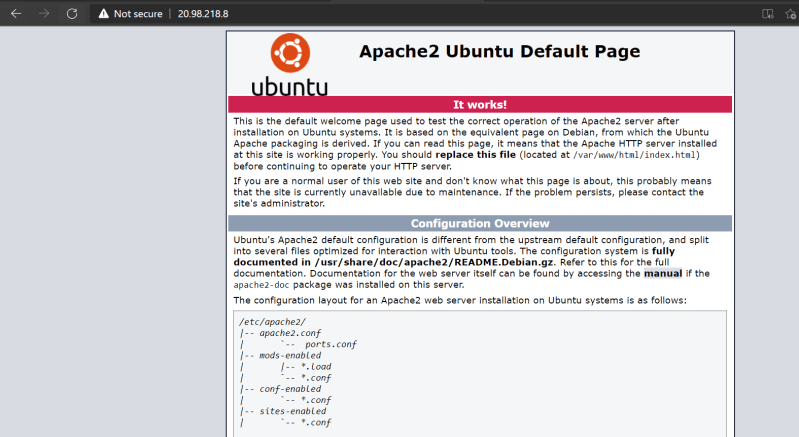
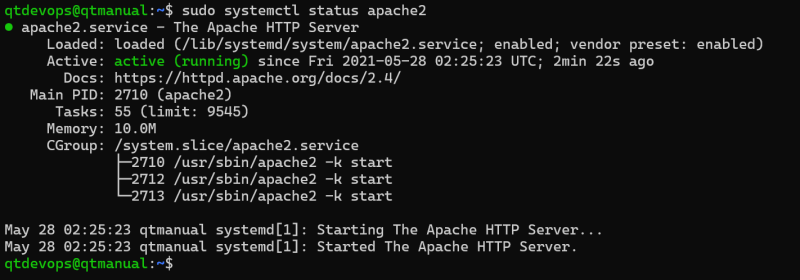
# CHEF-4 28/May/2021

### Installing apache server on ubuntu

* Launch a ubuntu VM
* Login into the VM and execute the following commands

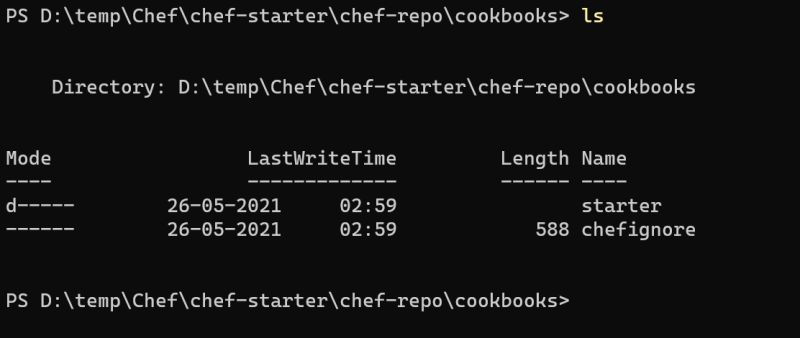
sudo apt update

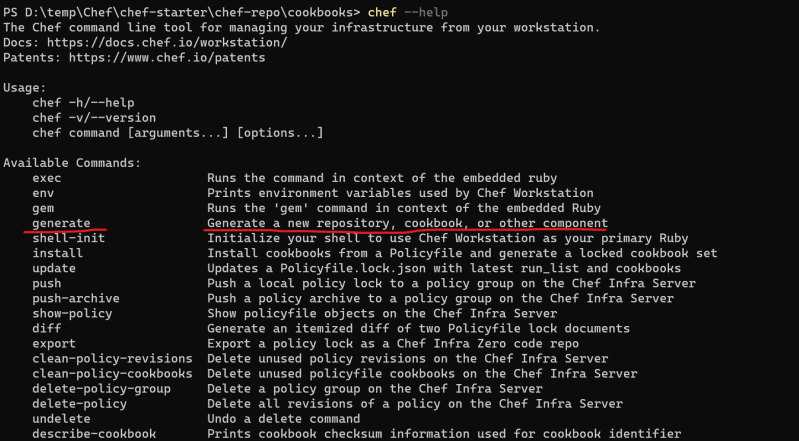
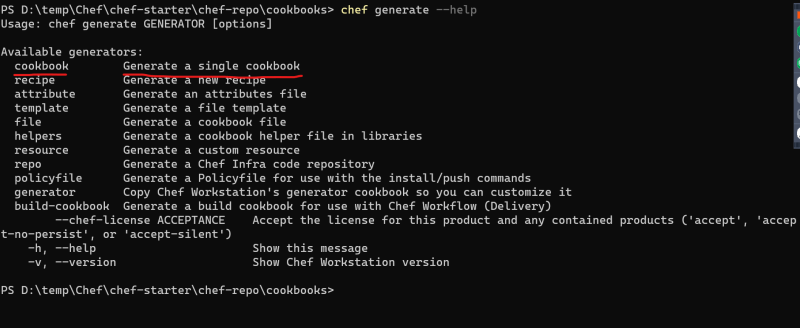
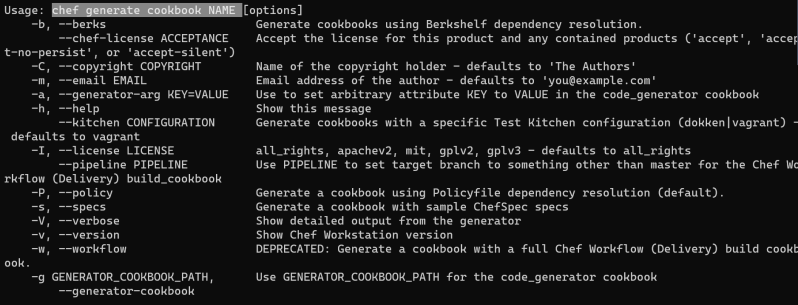
sudo apt install apache2 -y

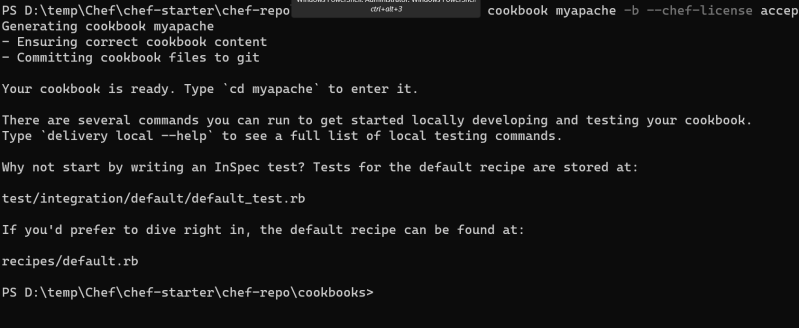
* Verify the installation by navigating to public IP
* 
* Verify the installation from command line
* 

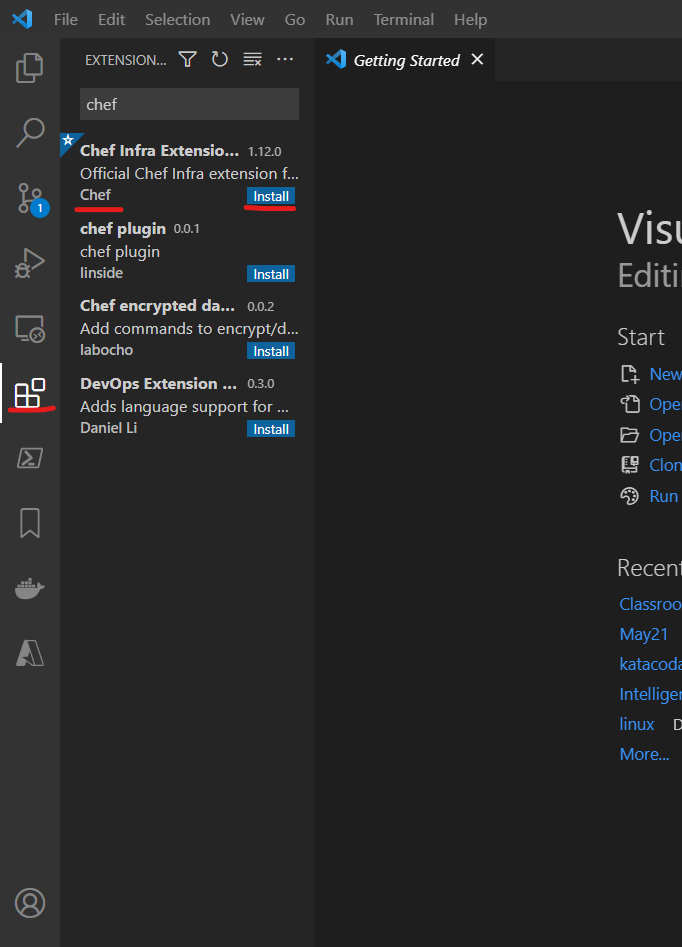
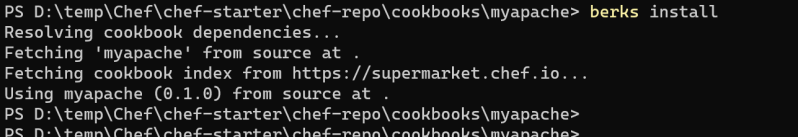
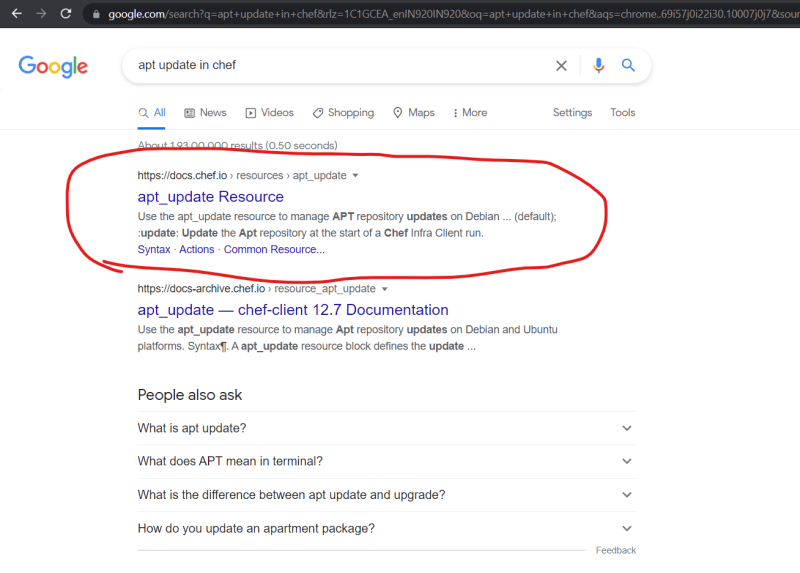
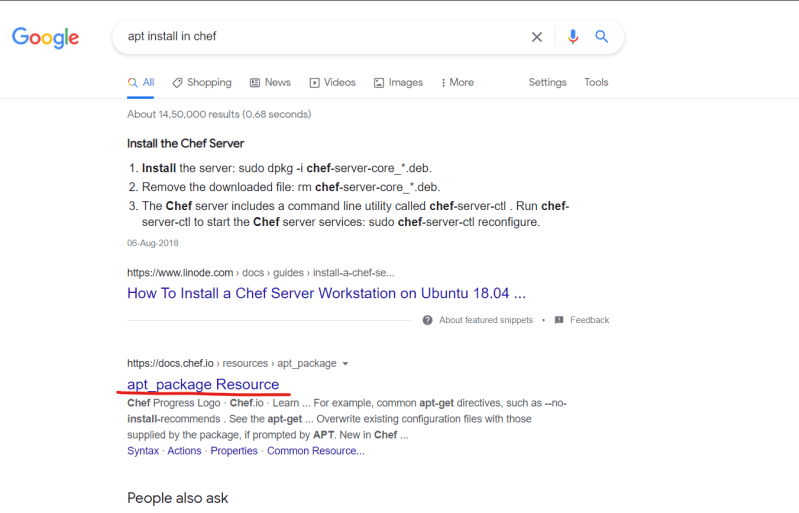
### Lets do this in Chef

* Now lets try to create a cookbook in chef which automates the above activity
* We need a cookbook:
  + Navigate to chef-repo\cookbooks folder



* + Lets create a chef cookbook [Refer Here](https://docs.chef.io/workstation/ctl_chef/#chef-generate-cookbook) (<https://docs.chef.io/workstation/ctl_chef/#chef-generate-cookbook>)  
  + I will be creating a cookbook for installing apache using the following
* chef generate cookbook myapache -b --chef-license accept



* A new directory will be created. will have some files and folders
* Chef is developed using ruby language. Chef has its own DSL (Domain Specific Language). If required we can directly use ruby in chef cookbooks.
* Launch the cookbook in visual studio code, install chef extension 
* Now execute the command berks install 
* In Chef cookbook there is already a recipe created called as default (recipes\default.rb)
* In chef recipe we will be using resources to automate the manual steps  
* The resources in chef will have following syntax

<resource type> 'name' do

property1 value1

...

propertyn valuen

action <action to acheive>

end

* In recipes\default.rb

apt\_update 'ubuntu packages' do

action :update

end

# sudo apt install apache2 -y

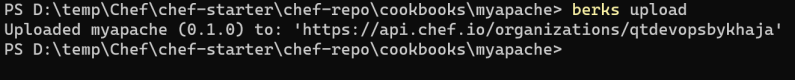
apt\_package 'apache2' do

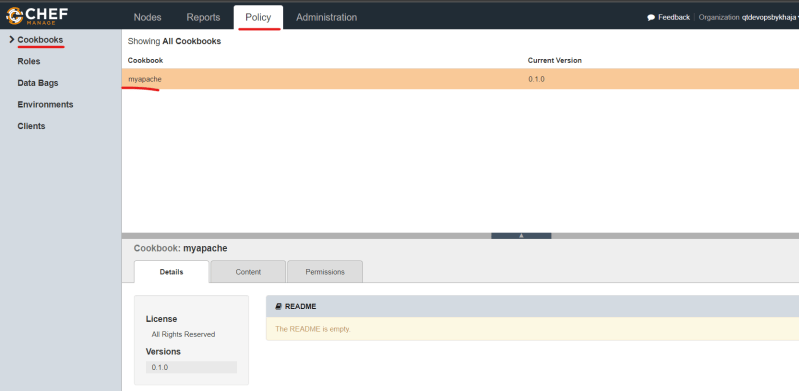
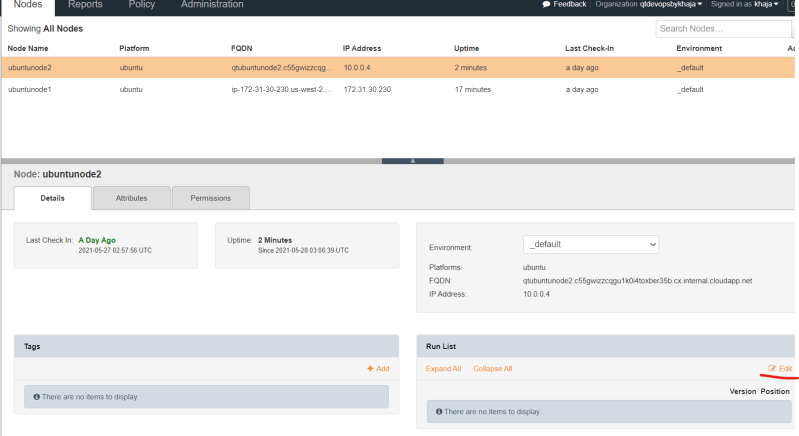
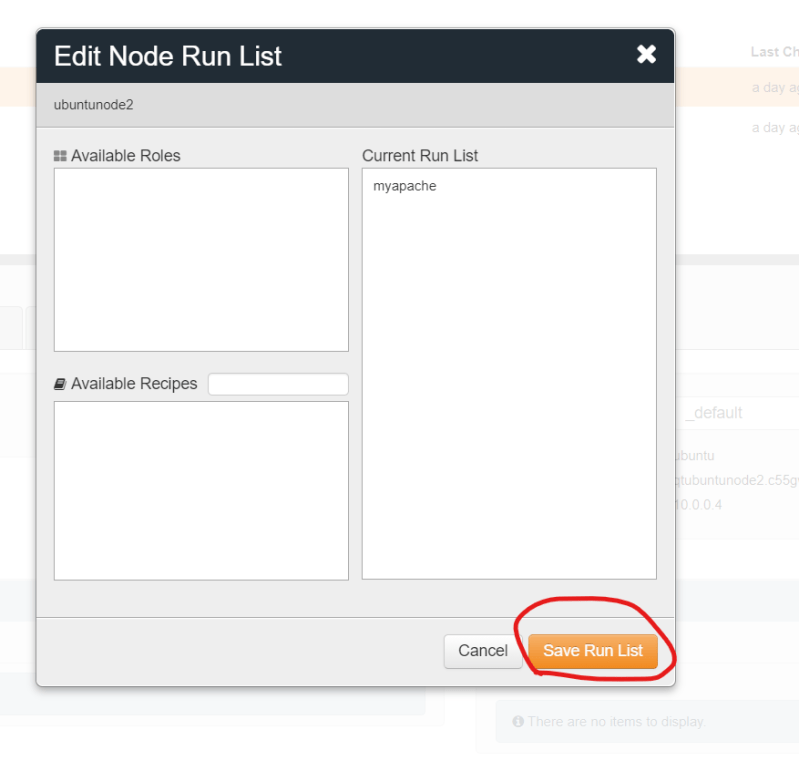
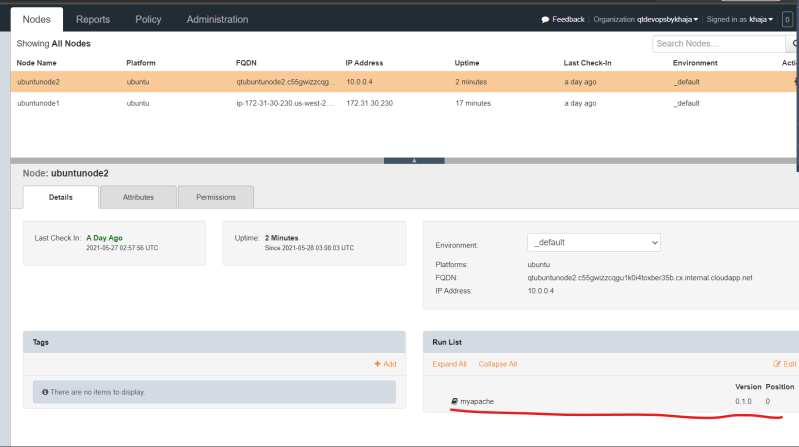
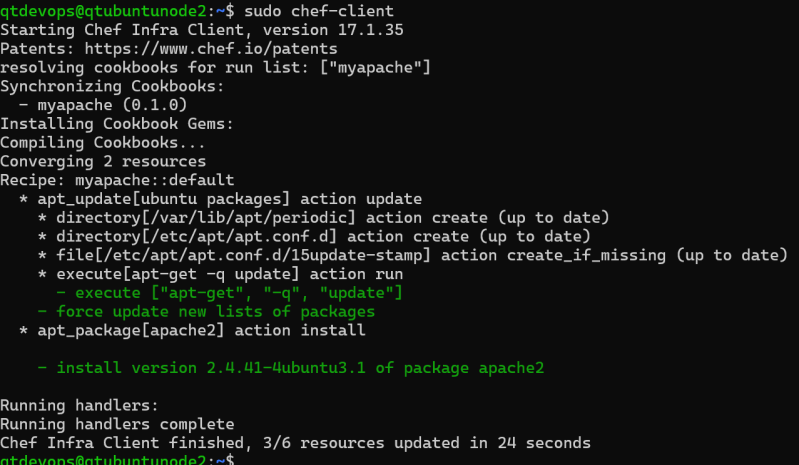
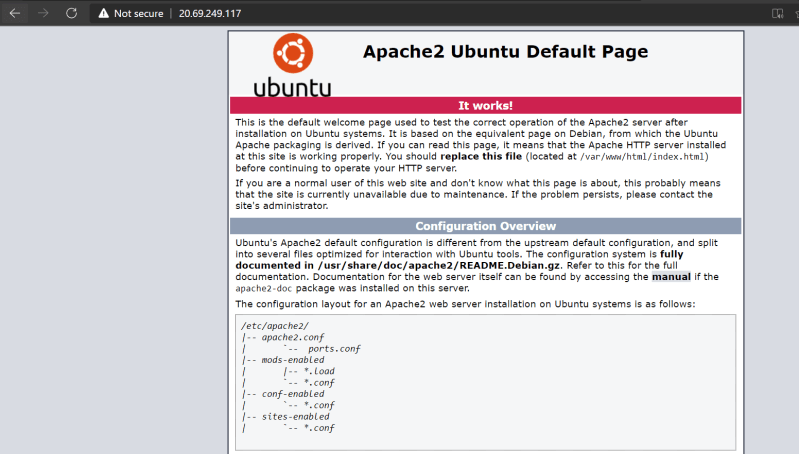
action :install

end

* Now upload the recipe to the chef server

berks upload



* Once the cookbook is uploaded, we need to specify to chef server on which nodes should the recipes be executed. 
* Now we need to change the nodes run\_list   
* Do this for the other ubuntu node
* Now we can wait till the nodes query the server about the work that needs to be done (Convergence)
* For lab purposes lets manually force the convergence
  + login into the node & execute sudo chef-client  
* Creating and making cookbooks work is the major activity in the cookbook development
* Uploading cookbook to the server adding it to run\_list etc can be done once our cookbooks are working.
* Next Steps:
  + Becoming good at writing cookbooks
  + Local development of cookbooks, testing them and upload right versions to server

## **Cookbook**

* Cookbook is fundamental unit of configuration and distribution in chef.
* A cookbook defines a scenario and contains everything needed to support that scenario
  + Recipe
  + Attributes
  + Files
  + Templates
  + Custom Resources

# CHEF-5 29/May/2021

## **Chef cookbook for installing lamp on ubuntu server**

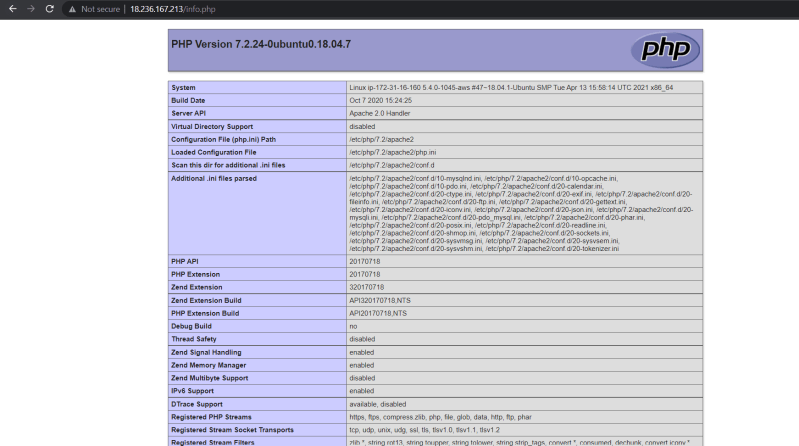
* Manual Steps for installing lamp on ubuntu 18
* [Refer Here](https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04) for the documentation (<https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-ubuntu-18-04>)
* Summary of manual steps

sudo apt update

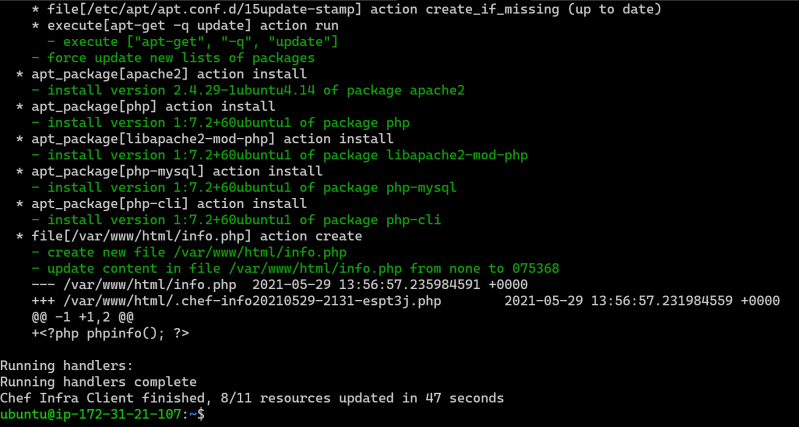
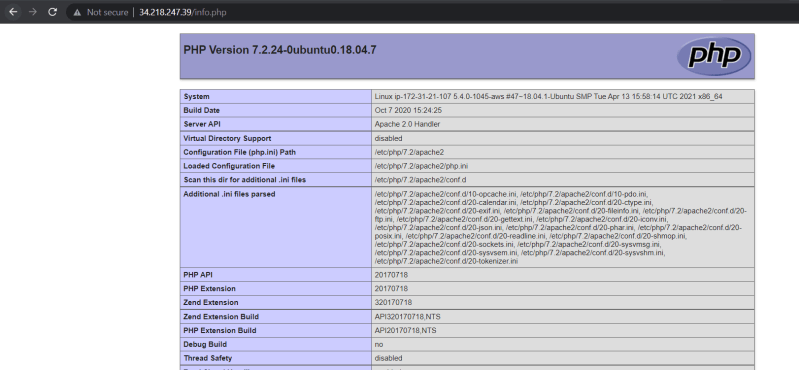
sudo apt install apache2 -y

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

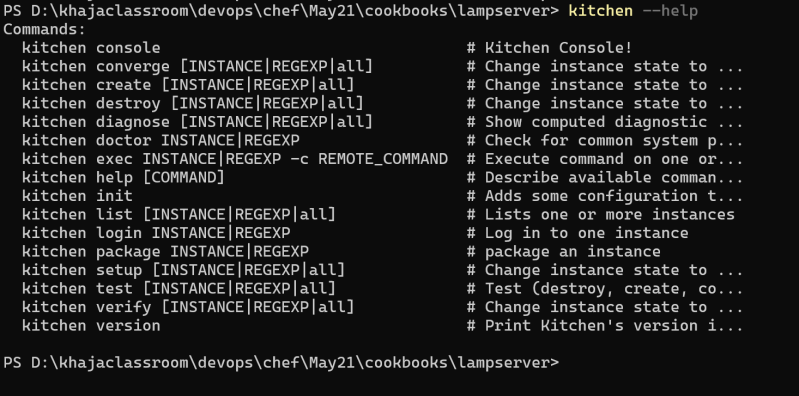
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

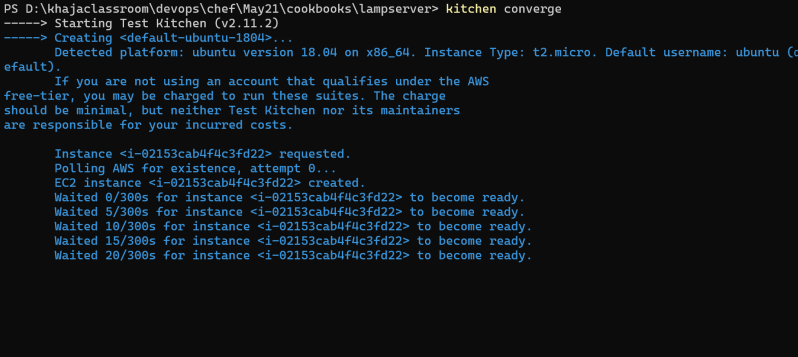
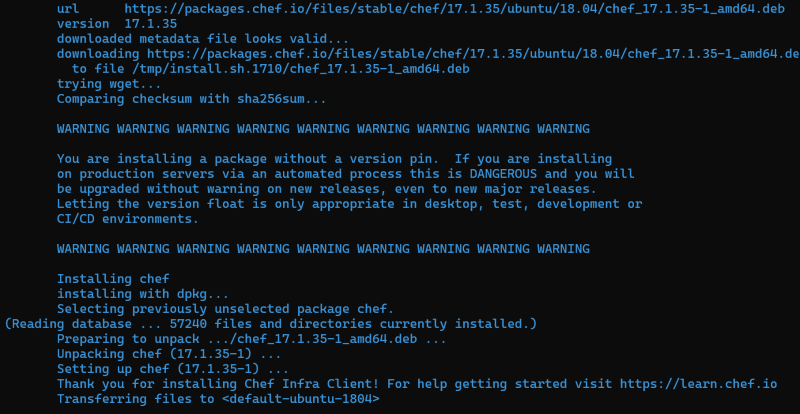
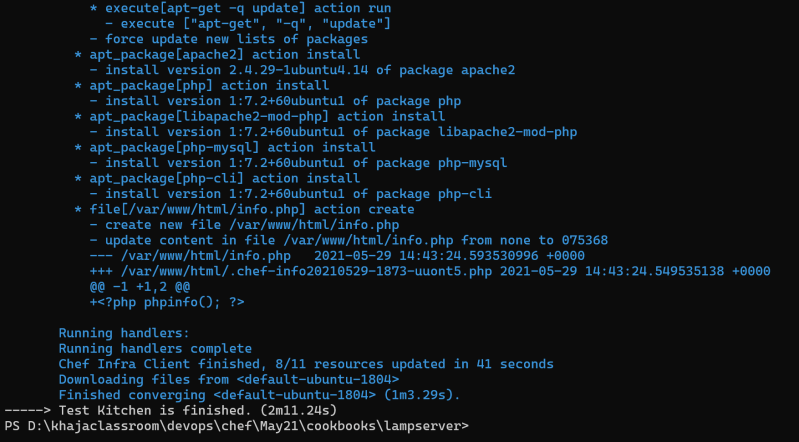
* Once we execute these steps and navigate to http://<public-ip>/info.php 
* Now Generate the cookbook

chef generate cookbook lampserver -b

* Now we need to convert each of the above mentioned manual step into a resource in a recipe file
* We already have a recipe file in recipes folder called as default.rb
* [Refer Here](https://github.com/asquarezone/ChefZone/commit/bcc7a78019e83aa24af8eff667c0f8b6d134df5f) for the cookbook created in the class (<https://github.com/asquarezone/ChefZone/commit/bcc7a78019e83aa24af8eff667c0f8b6d134df5f>)
* and [Refer Here](https://github.com/asquarezone/ChefZone/commit/bcc7a78019e83aa24af8eff667c0f8b6d134df5f#diff-495dbb08bfc494de3d9194e86564c7a8a8f72348fa97865e1bcdcc346a9b8452) for the recipe (<https://github.com/asquarezone/ChefZone/commit/bcc7a78019e83aa24af8eff667c0f8b6d134df5f#diff-495dbb08bfc494de3d9194e86564c7a8a8f72348fa97865e1bcdcc346a9b8452>)
* We have a cookbook ready, to test this cookbook
  + Create a test node and bootstrap it to the chef server (manage.chef.io)
  + Add recipe to run\_list on test node and check whether it is working or not
* When we run the chef-client on the test node we get the following result
*  
* Problems with the test node creation process
  + Bootstrapping test nodes every time
  + If my cookbook has to work on different flavours on Linux Creating VMs manually and checking the cookbook execution manually will be difficult.
  + Changes in the cook might lead to new versions uploaded to the chef server
* We need some kind of tool,
  + Which will automatically create the VM and run the recipe in the VM
  + Lets us verify the results
  + Also works with various cloud/virtual environments without need of chef server
  + No Need to change cookbook versions to verify
* In Chef we have a toolkit called as test-kitchen which can exactly do the above steps

## **Configuring Test Kitchen to Work with AWS**

* Test Kitchen can be used to automatically test cookbooks on different platforms
* To work with test kitchen
  + AWS Account
  + YAML [Refer Here](https://www.youtube.com/watch?v=ggOmHlnhPaM&list=PLuVH8Jaq3mLud3sVDvJ-gJ__0zd15wGDd&index=15) (<https://www.youtube.com/watch?v=ggOmHlnhPaM&list=PLuVH8Jaq3mLud3sVDvJ-gJ__0zd15wGDd&index=16>)
* Test kitchen can create infrastructure on different virtual environments & for that it uses a driver [Refer Here](https://docs.chef.io/workstation/kitchen/#drivers) (<https://docs.chef.io/workstation/kitchen/#drivers>)
* To set up a test kitchen [Refer Here](https://directdevops.blog/2019/04/02/test-kitchen-setup-with-aws/)(<https://directdevops.blog/2019/04/02/test-kitchen-setup-with-aws/>)
* In AWS, we need a user with which test-kitchen can login into AWS account and create ec2 instances for testing. We do this by creating an IAM user
* On the chef workstation, it is recommended to install aws cli [Refer Here](https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html) (<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html>)
* Make a note of
  + region
  + availability zone
  + vpc-id
* aws ec2 describe-vpcs --query "Vpcs[0].VpcId"
  + subnet-id
* aws ec2 describe-subnets --filters "Name=availability-zone,Values=us-west-2a" --query "Subnets[0].SubnetId"
  + security group id
* Add all of the noted information to kitchen.yaml file
* Now execute kitchen --help
* 
* Now execute kitchen converge

* [Refer Here](https://github.com/asquarezone/ChefZone/commit/ca23955b13322891ca97587a85b75762d55b79ff) for the kitchen configuration (<https://github.com/asquarezone/ChefZone/commit/ca23955b13322891ca97587a85b75762d55b79ff>)

## **Windows Terminal**

* [Refer Here](https://www.youtube.com/watch?v=qLVn2EvPsYc&list=PLuVH8Jaq3mLud3sVDvJ-gJ__0zd15wGDd&index=11) to install windows terminal. (<https://www.youtube.com/watch?v=qLVn2EvPsYc&list=PLuVH8Jaq3mLud3sVDvJ-gJ__0zd15wGDd&index=12>)

# CHEF-6 29/May/2021

====================================