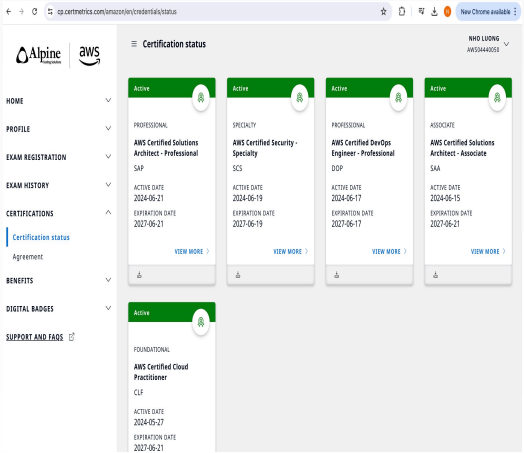
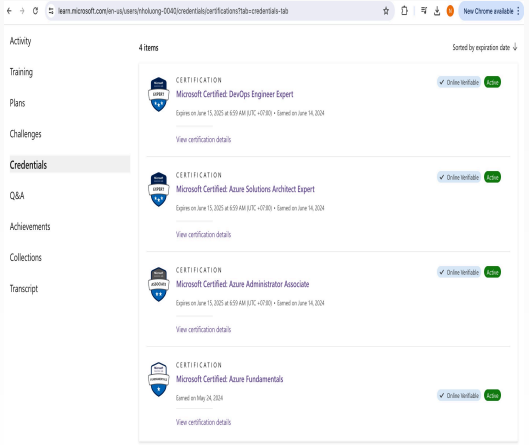
# BootStrap Node

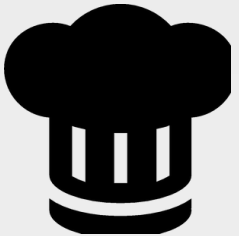
Author: Nho Luong

Skill: DevOps Engineer Lead







# Hosted Chef

*Adding nodes to your Chef Server*

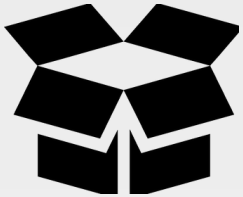
Objective:

- √ Create a Hosted Chef Account Upload your
- √ cookbooks to the Hosted Chef Server Bootstrap a
- √ node and update its runlist

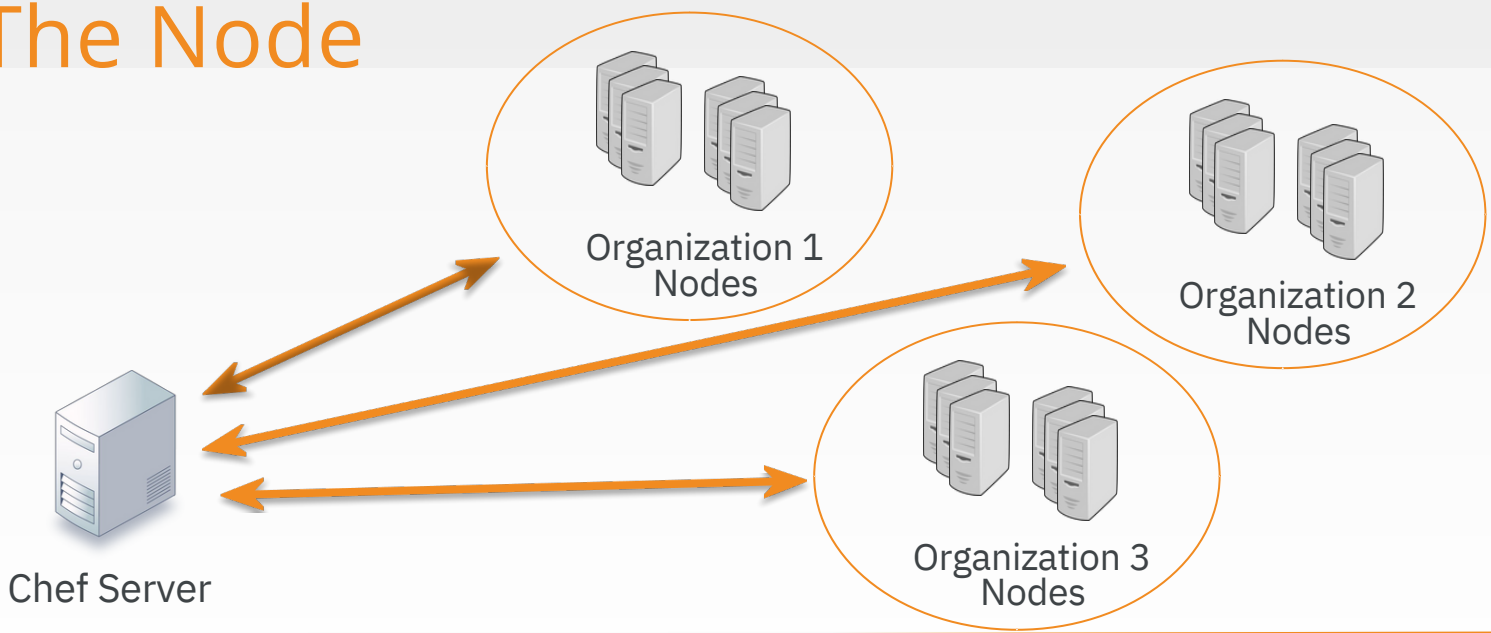
Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

CONCEPT



# The Node




Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Change to the chef-repo

```
$ cd ~/chef-repo
```

LOCAL

# Run 'knife node –help'

```
$ knife node --help
```

```
** NODE COMMANDS ** knife node bulk delete REGEX (options) knife node create NODE (options) knife node delete NODE (options) knife node edit NODE (options) knife node environment set NODE ENVIRONMENT knife node from file FILE (options) knife node list (options) knife node run_list add [NODE] [ENTRY[,ENTRY]] (options) knife node run_list remove [NODE] [ENTRY[,ENTRY]] (options) knife node run_list set NODE ENTRIES (options) knife node show NODE (options)
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Run 'knife node list'

```
$ knife node list
```

LOCAL

CONCEPT

# Bootstrapping a Node

The node may not have Chef installed. It may also not have details of where the Chef Server is located or the credentials to securely talk to that Server. To add those credentials we can bootstrap that node to install all those components.

https://learn.chef.io/skills/beyond-essentials-1

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Run 'knife bootstrap –help'

## $ knife bootstrap --help

```
knife bootstrap FQDN (options)
        --bootstrap-curl-options OPTIONS
                                     Add options to curl when install chef-client
        --bootstrap-install-command
        COMMANDS                     Custom command to install chef-client
        --bootstrap-no-proxy
        [NO_PROXY_URL|NO_PROXY_IP]Do not proxy locations for the node being
bootstrapped; this option is used interna

lly by Opscode
        --bootstrap-proxy PROXY_URL  The proxy server for the node being bootstrapped
    -t TEMPLATE,                     Bootstrap Chef using a built-in or custom
template. Set to the full path of an erb

template or use one of the built-in templates.
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Bootstrap Your Node - options

```
$ knife bootstrap FQDN -x USER -P PWD --sudo -N node_name
```

```
Creating new client for node1 Creating new node for node1
Feuclt1yi nQgu taoli fieecd2 -D5o4m-1a7i5n- 46-24.compute-1.amazonaws.com
ec2-54-175-4N6a-m24e.compute-1.amazonaws.com Starting first Chef Client run...
```

user name

password

sudo flag

node name

```
ec2-54-175-46-24.compute-1.amazonaws.com Starting Chef Client, version 12.3.0

ec2-54-175-46-24.compute-1.amazonaws.com resolving cookbooks for run list: []
ec2-54-175-46-24.compute-1.amazonaws.com Synchronizing Cookbooks:

ec2-54-175-46-24.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-175-46-24.compute-1.amazonaws.com [2016-09-16T16:51:21+00:00] WARN: Node node1
has an empty run list. ec2-54-175-46-24.compute-1.amazonaws.com Converging 0 resources ec2-
54-175-46-24.compute-1.amazonaws.com  ec2-54-175-46-24.compute-1.amazonaws.com  Running
handlers:
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Verify the port and identity file for web1

```
$ vagrant ssh-config web1
```

```
Host
web1  HostName 127.0.0.1 User vagrant Port 2200 UserKnownHostsFile /dev/null
  StrictHostKeyChecking no PasswordAuthentication no IdentityFile /Users/USER/chef-
  repo/.vagrant/machines/web1/virtualbox/private_key IdentitiesOnly yes LogLevel FATAL
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Bootstrap Your Node

```
$ knife bootstrap localhost --ssh-port WEB1_PORT --ssh-user vagrant --sudo --identity-file PATH_TO_KEY -N web1
```

```
Creating new client for web1 Creating new node for web1 Connecting to
localhost localhost -----> Installing Chef Omnibus (-v 12) localhost downloading
https://omnitruck-direct.chef.io/chef/install.sh


localhost       to file /tmp/install.sh.12058/install.sh
localhost trying wget... localhost el 7 x86_64 localhost Getting information for chef
stable 12 for el... localhost downloading
https://omnitruck-direct.chef.io/stable/chef/metadata?v=12&p=el&pv=7&m=x86_64


localhost       to file /tmp/install.sh.12063/metadata.txt
localhost trying wget...
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Run 'knife node list' Again

```
$ knife node list
```

```
web
1
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# View More Information About Your Node

**$ knife node show web1**

```
Node Name:   web
Environment: _default
FQDN: IP:    web1
Run List:    10.0.2.15
Roles:
Recipes:
Platform:
Tags:        centos 7.2.1511
```

Notice that the IPAddress is not what we defined in the Vagrantfile. It's the internal IP instead.

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Add a Recipe to web1's Run List

```
$ knife node run_list add web1 "recipe[workstation],recipe[apache]"
```

```
Web1
:   run_list: recipe[workstation]
    run_list: recipe[apache]
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# View More Information About Your Node

## $ knife node show web1

**Node Name:** **web**
Environment: _default
FQDN: IP: web1 10.0.2.15 recipe[workstation],
Run List: recipe[apache]
Roles:
Recipes:
Platform:
Tags: centos 7.2.1511

Your Run List for web1 should contain the workstation and apache cookbooks

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Login to web1

```
$ vagrant ssh web1
```

```
Last login: Sat Dec 31 02:59:27 2016 from 10.0.2.2
[vagrant@web1 ~]$
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Run chef-client to converge web1

```
[vagrant@web1 ~]$ sudo chef-client
```

```
Starting Chef Client, version 12.17.44 resolving cookbooks for run
list: ["workstation", "apache"] Synchronizing Cookbooks:  - apache
(0.2.1)  - workstation (0.2.1) Installing Cookbook Gems: Compiling
Cookbooks... Converging 8 resources....
```

LOCAL

# Verify the state of your web application

```
[vagrant@web1 ~]$ curl localhost
```

```
<html>
<body>
      <h1>Hello, world!</h1>
      <h2>ipaddress: 192.168.10.43</h2>
      <h2>hostname: web1</h2>
 </body>
</html>
```

LOCAL

# Return to your Workstation

```
[vagrant@web1 ~]$ exit
```

```
logout Connection to 127.0.0.1
closed.
```

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# View More Information About Your Node

## $ knife node show web1

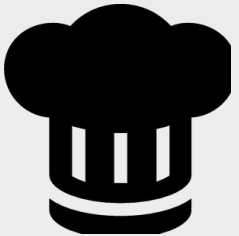
**Node Name:** **web**
Environment: _default
FQDN: web1
IP: 192.168.10.43
Run List: recipe[workstation], recipe[apache]
Roles:
Recipes: workstation, workstation::default, apache, apache::default, workstation::vagrant, workstation::setup, apache::server
Platform: centos 7.2.1511
Tags:

The IPAddress should now match what we defined in the Vagrantfile.

LOCAL

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF



# Hosted Chef

*More easily manage multiple nodes*

Objective:

- √ Create a Hosted Chef Account Upload your
- √ cookbooks to the Hosted Chef Server Add web1 as
- √ a managed node

Author: Nho Luong
Skill: DevOps Engineer Lead

CHEF

# Discussion

What is the benefit of storing cookbooks in a central repository? What is the primary tool for communicating with the Chef Server? How did you add a node to your organization?

# Thank You

Author: Nho Luong
Skill: DevOps Engineer Lead