Data Bags

Working with Custom Data Sets

CHEF

Objectives

After completing this module, you should be able to

- Explain how to use and manage a Data Bag
- Create and upload a Data Bag to Chef Server
- Query Data Bag information with the CLI
- Generate the ‘myusers’ cookbook
- Add the ‘myusers’ cookbook to company_web policyfile's run list
- Update and push the policyfile to the Chef Infra Server
- Converge web nodes

E.g. User data or Env related data

Where should we store data that each node might need access to?

Data Bags

A data bag is a container for items that represent information about your infrastructure that is **not tied to a single node.**

Examples:

- Users
- Groups
- Application Release Information
- Passwords (in an encrypted data bag)

https://docs.chef.io/data_bags.html

Data Bags

- Data bags store global variables as JSON data.
- Data bags are indexed for searching and can be loaded by a cookbook or accessed during a search.
- Can be created manually or using knife

https://docs.chef.io/data_bags.html

Group Lab: Custom Data Sets

We can store sets of JSON data on our Chef Server, accessible by a node with search

Objective:

- ❑ Create “users” data bags
- ❑ Upload data bags to Chef Server
- ❑ Use CLI to query information about data bags

GL: What Can 'knife data bag' Do?

```
$ cd ~/chef-repo
$ knife data bag --help
```

```
** DATA BAG COMMANDS **
knife data bag create BAG [ITEM] (options)
knife data bag delete BAG [ITEM] (options)
knife data bag edit BAG ITEM (options)
knife data bag from file BAG FILE|FOLDER [FILE|FOLDER..] (options)
knife data bag list (options)
knife data bag show BAG [ITEM] (options)
```


CHEF

GL: Run 'knife data bag list'

```
$ knife data bag list
```

GL: Create a data_bags Directory

Create directory to hold all data_bags in workstation

```
$ mkdir data_bags
```

Create directory to hold data bag {users} in workstation

```
$ mkdir data_bags/users
```

Create data bag {users} in server

```
$ knife data bag create users
```

GL: Create centos_user.json

~/chef-repo/data_bags/users/centos_user.json

```
{
  "id": "centos_user",
  "comment": "I am a centos_user user",
  "platform": "centos"
}
```

GL: Create windows_user.json

~/chef-repo/data_bags/users/windows_user.json

```
{
  "id": "windows_user",
  "comment": "I am a windows user",
  "platform": "windows"
}
```

Group Lab: Custom Data Sets

We can store sets of JSON data on our Chef Server, accessible by a node with search

Objective:

- ✓ Create “users” data bags
- ❑ Upload data bags to Chef Server
- ❑ Use CLI to query information about data bags

GL: Upload Data Bag Items to Chef Server

```
$ knife data bag from file users data_bags/users/centos_user.json
data_bags/users/windows_user.json
```

```
Updated data_bag_item[users::centos_user]
Updated data_bag_item[users::windows_user]
```

You can also do data_bags/users/*

You should still be in ~/chef-repo when running this command.

CHEF

GL: Upload Data Bag Items to Chef Server

```
$ knife data bag from file myusers data_bags/users/centos_user.json
```

This data bag do not exist in server

```
ERROR: The object you are looking for could not be found
Response: No data bag 'myusers' could be found. Please create this data bag
before adding items to it.
```


CHEF

GL: Validate Chef Server received items

```
$ knife data bag list
```

```
users
```

```
$ knife data bag show users
```

```
centos_user
windows_user
```


CHEF

Group Lab:Custom Data Sets

We can store sets of JSON data on our Chef Server, accessible by a node with search

Objective:

- ✓ Create “users” data bags
- ✓ Upload data bags to Chef Server
- ❑ Use CLI to query information about data bags

GL: View Details of centos_user

```
$ knife data bag show users centos_user
```

```
comment:  I am a centos_user user
id:       centos_user
platform: centos
```


CHEF

GL: Search the users Index

```
$ knife search users "*:*"
```

```
2 items found

chef_type: data_bag_item
comment:   I am a centos_user user
data_bag:  users
id:        centos_user
platform:  centos

chef_type: data_bag_item
comment:   I am a windows user
data_bag:  users
id:        windows_user
platform:  windows
```

The data bag {name} becomes SOLR index

CHEF

GL: Return Users with “platform:centos”

```
$ knife search users "platform:windows"
```

```
1 items found


chef_type: data_bag_item
comment:   I am a windows user
data_bag:  users
id:        windows_user
platform:  windows
```


CHEF

Group Lab: Custom Data Sets

We can store sets of JSON data on our Chef Server, accessible by a node with search

Objective:

- ✓ Create “users” data bags
- ✓ Upload data bags to Chef Server
- ✓ Use CLI to query information about data bags

GL: Create Users from a Data Bag

Dynamically search through the Chef Server under the 'users' index to create users

Objective:

- ❑ Generate the 'myusers' cookbook
- ❑ Create users based on data bag contents within default recipe
- ❑ Add the 'myusers' cookbook to company_web policyfile's run list
- ❑ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes

Where are the users?

Because our users are now indexed on the Chef Server, we have a centralized source of truth for information regarding these users.

We can search through this information inside of our recipes.

GL: Generate the myusers Cookbook

```
$ cd ~/chef-repo
$ chef generate cookbook cookbooks/myusers
```

```
Generating cookbook myusers
- Ensuring correct cookbook content
- Committing cookbook files to git

Your cookbook is ready. To setup the pipeline, type `cd cookbooks/myusers`,
then run `delivery init`
```

GL: Create Users from a Data Bag

Dynamically search through the Chef Server under the 'users' index to create users

Objective:

- ✓ Generate the 'myusers' cookbook
- ❑ Create users based on data bag contents within default recipe
- ❑ Add the 'myusers' cookbook to company_web policyfile's run list
- ❑ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes

GL: Create users recipe

~/chef-repo/cookbooks/myusers/recipes/users.rb

```
system_users = search("users", "platform:#{node['platform']}")


system_users.each do |user_data|
  user user_data['id'] do
    comment user_data['comment']
    action :create
  end
end
```

Gets {ohai platform} for the machine where it is running and queries for data bags. For centos VMs, this becomes platform:centos For Win VMs, this becomes platform:windows

Thus, users defined for each platform will be created using data bag appropriately

Using {user} resource to create users

CHEF

GL: Include users recipe within default recipe

~/chef-repo/cookbooks/myusers/recipes/default.rb

```
#
# Cookbook:: myusers
# Recipe:: default
#
# Copyright:: 2018, The Authors, All Rights Reserved.


include_recipe 'myusers::users'
```

GL: Create Users from a Data Bag

Dynamically search through the Chef Server under the 'users' index to create users

Objective:

- ✓ Generate the 'myusers' cookbook
- ✓ Create users based on data bag contents within default recipe
- ❑ Add the 'myusers' cookbook to company_web policyfile's run list
- ❑ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes

GL: Change to the cookbooks/myusers Directory

```
$ cd cookbooks/myusers
```


CHEF

GL: Add myusers Cookbook to the company_web Policyfile

~/chef-repo/company_web.rb

```
...skipping...

# run_list: chef-client will run these recipes in the order specified.
run_list 'mychef_client::default','company_web::default','myusers::default'

# Specify a custom source for a single cookbook:
cookbook 'company_web', path: 'cookbooks/company_web'
cookbook 'myiis', path: 'cookbooks/myiis'
cookbook 'apache', path: 'cookbooks/apache'
cookbook 'mychef_client', path: 'cookbooks/mychef_client'
cookbook 'myusers', path: 'cookbooks/myusers'
```

GL: Create Users from a Data Bag

Dynamically search through the Chef Server under the 'users' index to create users

Objective:

- ✓ Generate the 'myusers' cookbook
- ✓ Create users based on data bag contents within default recipe
- ✓ Add the 'myusers' cookbook to company_web policyfile's run list
- ❑ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes

GL: Ensure You are in chef-repo

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


CHEF

GL: Update the Policyfile

```
$ chef update company_web.rb
```

```
Attributes already up to date
Building policy company_web
Expanded run list: recipe[mychef_client::default], recipe[company_web::default],
recipe[myusers::default]
Caching Cookbooks...
Installing company_web   >= 0.0.0 from path
Installing myiis         >= 0.0.0 from path
Installing apache        >= 0.0.0 from path
Installing mychef_client >= 0.0.0 from path
Installing myusers       >= 0.0.0 from path
Using       chef-client  11.2.0
Using       cron         6.2.1
Using       logrotate    2.2.0

Lockfile written to /Users/sdelfante/chef-repo/company_web.lock.json
Policy revision id: 14a342b5d2134516984b878c1f6be83a51427e6f84594c4dd9e57456b1312582
```


CHEF

GL: Push the Policyfile to Chef Infra Server

```
$ chef push prod company_web.lock.json
```

```
Uploading policy company_web (14a342b5d2) to policy group prod
Using    apache          0.1.0   (1388ab3a)
Using    chef-client     11.2.0  (0b49a3a8)
Using    company_web     0.1.0   (c1b26cb5)
Using    cron            6.2.1   (08676b5c)
Using    logrotate       2.2.0   (53e09234)
Using    mychef_client   0.1.0   (10d082a4)
Using    myiis           0.2.1   (cd0db3ed)
Uploaded myusers         0.1.0   (bebccee3)
```

Note that:
new policy is uploaded to prod only

CHEF

GL: Create Users from a Data Bag

Dynamically search through the Chef Server under the 'users' index to create users

Objective:

- ✓ Generate the 'myusers' cookbook
- ✓ Create users based on data bag contents within default recipe
- ✓ Add the 'myusers' cookbook to company_web policyfile's run list
- ✓ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes

GL: Converge All Web Nodes

```
$ knife ssh "policy_name:company_web" -x centos -i ~/aws.pem "sudo chef-client"
```

```
$ knife winrm "name:iis_web" -a cloud.public_ipv4 -x USER -P PWD "chef-
client"
```

GL: Check Local Users for Apache Server

```
$ knife ssh "policy_name:company_web AND policy_group:prod" -x
centos -i ~/aws.pem "cat /etc/passwd"
```

```
ec2-52-15-221-52.us-east-2.compute.amazonaws.com
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
ec2-52-15-221-52.us-east-2.compute.amazonaws.com
chrony:x:998:995::/var/lib/chrony:/sbin/nologin
ec2-52-15-221-52.us-east-2.compute.amazonaws.com
centos:x:1000:1000:Cloud User:/home/centos:/bin/bash
ec2-52-15-221-52.us-east-2.compute.amazonaws.com
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
ec2-52-15-221-52.us-east-2.compute.amazonaws.com
centos_user:x:1001:1001:I am a centos_user
user:/home/centos_user:/bin/bash
```

```
> knife winrm "name:iis_web" -a cloud.public_ipv4 -x USER -P PWD "net user windows_user"
```


CHEF

GL: Check Local Users for Apache Server

```
$ knife ssh "policy_name:company_web AND policy_group:acceptance" -
x centos -i ~/aws.pem "cat /etc/passwd"
```

```
You won't find centos_user

This will happen because in the last chapter, we changed one of the
webserver node to {acceptance} policy_group, and we applied the new
{company_web} policy only to {prod} policy_group
```

GL: Create Users from a Data Bag

Dynamically search through the Chef Server under the 'users' index to create users

Objective:

- ✓ Generate the 'myusers' cookbook
- ✓ Create users based on data bag contents within default recipe
- ✓ Add the 'myusers' cookbook to company_web policyfile's run list
- ✓ Update and push the policyfile to the Chef Infra Server
- ✓ Converge web nodes

Optional Lab: Managing Groups

- ❑ Create a ‘groups’ data bag on your Chef Server
- ❑ Create centos_group.rb and windows_group.rb files with: "id", "members", and "platform" fields and upload to the 'groups' data bag
- ❑ Create a 'groups.rb' recipe file that creates the group with corresponding members based on the node's platform and include this recipe in the default.rb recipe
- ❑ Update the metadata.rb file with a minor version change
- ❑ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes
- ❑ Verify the new group on apache_web with: **cat /etc/group**
- ❑ Verify the new group on iis_web with: **net localgroup GROUP_NAME**

Lab: Create a data_bags/groups Directory

```
$ cd ~/chef-repo
$ mkdir data_bags/groups
```

Lab: Create groups Data Bag on Chef Server

```
$ knife data bag create groups
```

```
Created data_bag[groups]
```


CHEF

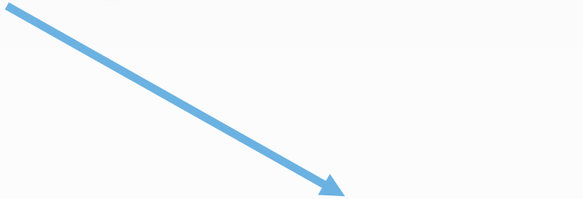
Optional Lab: Managing Groups

- ✓ Create a ‘groups’ data bag on your Chef Server
- ❑ Create centos_group.json and windows_group.json files with: "id", "members", and "platform" fields and upload to the 'groups' data bag
- ❑ Create a 'groups.rb' recipe file that creates the group with corresponding members based on the node's platform and include this recipe in the default.rb recipe
- ❑ Update the metadata.rb file with a minor version change
- ❑ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes
- ❑ Verify the new group on apache_web with: **cat /etc/group**
- ❑ Verify the new group on iis_web with: **net localgroup GROUP_NAME**

Lab: Create centos_group.json

~/chef-repo/data_bags/groups/centos_group.json

```
{
  "id": "centos_group",
  "members": ["centos_user"],
  "platform": "centos"
}
```



Lab: Create windows_group.json

~/chef-repo/data_bags/groups/windows_group.json

```
{
  "id": "windows_group",
  "members": ["windows_user"],
  "platform": "windows"
}
```

Lab: Upload Data Bag Items to Chef Server

```
$ knife data bag from file groups data_bags/groups/*
```

```
Updated data_bag_item[groups::centos_group]
Updated data_bag_item[groups::windows_group]
```

Lab: Validate Chef Server Received Items

```
$ knife data bag show groups
```

```
centos_group
windows_group
```

Lab: View details of centos_group

```
$ knife data bag show groups centos_group
```

```
id:        centos_group
members:   centos_user
platform:  centos
```


CHEF

Lab: Return groups with “platform:windows”

```
$ knife search groups "platform:windows"
```

```
1 items found


chef_type: data_bag_item
data_bag:  groups
id:        windows_group
members:   windows_user
platform:  windows
```

Optional Lab: Managing Groups

- Create a 'groups.rb' recipe file that creates the group with corresponding members based on the node's platform and include this recipe in the default.rb recipe
- Update the metadata.rb file with a minor version change
- Update and push the policyfile to the Chef Infra Server
- Converge web nodes
- Verify the new group on apache_web with: **cat /etc/group**
- Verify the new group on iis_web with: **net localgroup GROUP_NAME**

Lab: Create groups recipe

~/chef-repo/cookbooks/myusers/recipes/groups.rb

```
system_groups = search("groups", "platform:#{node['platform']}")

system_groups.each do |x|
  group x['id'] do
    members x['members']
    action :create
  end
end
```

Lab: Update the default recipe

~/chef-repo/cookbooks/myusers/recipes/default.rb

```
#
# Cookbook:: myusers
# Recipe:: default
#
# Copyright:: 2018, The Authors, All Rights Reserved.


include_recipe 'myusers::users'
include_recipe 'myusers::groups'
```

Lab: Version the myusers metadata.rb

~/chef-repo/cookbooks/myusers/metadata.rb

```
name              'myusers'
maintainer        'The Authors'
maintainer_email  'you@example.com'
license           'all_rights'
description       'Installs/Configures myusers'
long_description  'Installs/Configures myusers'
version           '0.2.0'
```

Optional Lab: Managing Groups

- ✓ Create a 'groups.rb' recipe file that creates the group with corresponding members based on the node's platform and include this recipe in the default.rb recipe
- ✓ Update the metadata.rb file with a minor version change
- ❑ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes
- ❑ Verify the new group on apache_web with: **cat /etc/group**
- ❑ Verify the new group on iis_web with: **net localgroup GROUP_NAME**

GL: Ensure You are in chef-repo

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


CHEF

GL: Update the Policyfile

```
$ chef update company_web.rb
```

```
Attributes already up to date
Building policy company_web
Expanded run list: recipe[mychef_client::default], recipe[company_web::default],
recipe[myusers::default]
Caching Cookbooks...
Installing company_web   >= 0.0.0 from path
Installing myiis         >= 0.0.0 from path
Installing apache        >= 0.0.0 from path
Installing mychef_client >= 0.0.0 from path
Installing myusers       >= 0.0.0 from path
Using       chef-client  11.2.0
Using       cron         6.2.1
Using       logrotate    2.2.0

Lockfile written to /Users/sdelfante/chef-repo/company_web.lock.json
Policy revision id: 2fb379d0bf8eeedd1c43efcc3c57b32424b1dbd9fa27850e79daba3a71c5e672
```


CHEF

GL: Push the Policyfile to Chef Infra Server

```
$ chef push prod company_web.lock.json
```

```
Uploading policy company_web (2fb379d0bf) to policy group prod
Using    apache          0.1.0  (1388ab3a)
Using    chef-client     11.2.0 (0b49a3a8)
Using    company_web     0.1.0  (c1b26cb5)
Using    cron            6.2.1  (08676b5c)
Using    logrotate       2.2.0  (53e09234)
Using    mychef_client   0.1.0  (10d082a4)
Using    myiis           0.2.1  (cd0db3ed)
Uploaded myusers         0.2.0  (0fb96da2)
```


CHEF

GL: Converge All Web Nodes

```
$ knife ssh "policy_name:company_web" -x centos -i ~/aws.pem "sudo chef-client"
```


CHEF

GL: Check Local Users for Apache Server

```
$ knife ssh "policy_name:company_web AND policy_group:prod" -x
centos -i ~/aws.pem "cat /etc/group"
```

```
ec2-52-15-221-52.us-east-2.compute.amazonaws.com chrony:x:995:
ec2-52-15-221-52.us-east-2.compute.amazonaws.com centos:x:1000:
ec2-52-15-221-52.us-east-2.compute.amazonaws.com apache:x:48:
ec2-52-15-221-52.us-east-2.compute.amazonaws.com centos_user:x:1001:
ec2-52-15-221-52.us-east-2.compute.amazonaws.com
centos_group:x:1002:centos_user
```


CHEF

GL: Check Local Users for Apache Server

```
$ knife ssh "policy_name:company_web AND policy_group:acceptance" -
x centos -i ~/aws.pem "cat /etc/group"
```

```
No centos_group found
```


CHEF

GL: Check Local Users for IIS Server

```
> knife winrm "name:iis_web" -a cloud.public_ipv4 -x USER -P PWD "net user windows_user"
```

```
34.195.38.226 User name                    windows_user
34.195.38.226 Full Name
34.195.38.226 Comment                      I am a windows user
34.195.38.226 User's comment
34.195.38.226 Country/region code          000 (System Default)
34.195.38.226 Account active               Yes
34.195.38.226 Account expires              Never
34.195.38.226
34.195.38.226 Password last set            8/7/2019 5:45:27 PM
34.195.38.226 Password expires             Never
34.195.38.226 Password changeable          8/7/2019 5:45:27 PM
34.195.38.226 Password required            Yes
34.195.38.226 User may change password     Yes
34.195.38.226
34.195.38.226 Workstations allowed         All
```

Optional Lab: Managing Groups

- ✓ Create a 'groups.rb' recipe file that creates the group with corresponding members based on the node's platform and include this recipe in the default.rb recipe
- ✓ Update the metadata.rb file with a minor version change
- ❑ Update and push the policyfile to the Chef Infra Server
- ❑ Converge web nodes
- ❑ Verify the new group on apache_web with: **cat /etc/group**
- ❑ Verify the new group on iis_web with: **net localgroup GROUP_NAME**

Review Questions

1. When should we utilize data bags instead of node attributes?

2. When creating a new data bag, what index on the Chef server does the data bag get added to?

Q&A

What questions can we help you answer?

CHEF

CHEF™