Objectives:

- Encrypt passwords and secrets stored in data bags

Data bag items can be encrypted using [shared secret encryption]. This allows each data bag item to store confidential information (such as a database password) or to be managed in a source control system.

1. Create a secret key for encryption:

   a. From the root directory of **chef-repo/**

   Execute:

```
$openssl rand -base64 512 | tr -d '\r\n' > .my_secret
```

   b. Then execute: `$cat .my_secret`

```
DwAP06an0E1RmvWZBwS2J3siIfu7mZLD+ccJxqQJr4/0kcIjGYHEOVvBCLaTY2oLaEFtTOHv7cAHKCinpmE2MD3R6x+Gl71JpT/wQzU
cwZWw8hqYKSqtrzBkeSKZLYUeiajBADvxYYtjibmBBQI5HssAfxXs8oMZ5IxqpxFvBEpFZR1Wb36gryc2AHCNslC3YTar+1DCNASz+w
rGRb63TnxH9YExiL2KAwkzso/0OL3nw41U5L99nCjNu5jkuiN5q3eBLaMCkWJ+owY9BiIeJ0rtTsrL66z0OanDmkUyfjKodkZU5CE0v
E01lvp630sSG6crD12gTH40Sk7bIOgNYpvOkJ/V/u+zi//5iZr2InyhF38PYDM4EsHwxaoHb27u+guNFsBFm12XYpQtJStJ4txwfYCu
vQbp4wuwenwJud01r2ocUA60+51Q/gwXn3xDzOzKzBL/llZZjUlTBp906wPKCaS75ROX6h9Fljybra8Wt6ab4LW9bYOYvXdTGNDKMup
njo1lEzyAFFaHzqj8931skGfnFWJMiE95JT3NECcL/KIREQaeoBg0jg5DMr+osTgb4BtpamzxNrmGKZbh22icDU+jypLsbr/rLotpTV
T964OiIqe0U8Vuf7OMvEnK23NK97ySKMxpY=/home/ubuntu/chef-repo $
```

2. The key above will be used to encrypt the password in **~/chef-repo/data_bags/passwords/your_name.json**

   a. **Execute:**

   $ knife data bag from file passwords data_bags/passwords/your_name.json --secret-file .my_secret --local-mode:

   (remember… -z is a nice shortcut)

   Ignore the warnings about not having a configuration file

```
/home/ubuntu/chef-repo $knife data bag from file passwords data_bags/passwords/your_name.json --secre
.my_secret --local-mode
WARNING: No knife configuration file found. See https://docs.chef.io/config_rb/ for details.
Updated data_bag_item[passwords::your_name]
/home/ubuntu/chef-repo $
```

3. Display data bag saved in passwords

   a. **Execute:**

```
$knife data bag show passwords your_name -z
```

```
WARNING: No knife configuration file found. See https://docs.chef.io/config_rb/ for details.
WARNING: Encrypted data bag detected, but no secret provided for decoding. Displaying encrypte
id:       your_name
password:
  auth_tag:       JhX5OGsFX5HO9g2ocy/DlA==

  cipher:         aes-256-gcm
  encrypted_data: s8iGOLxiXQ+R/0ZpPphpFsKz+0MnzPyqLl1xN8WPN35BZuE/IhO6R5VTQ4gp
  cjDFKuCiHAE=

  iv:             oVMPPohgOAm9oQAy

  version:        3
```

4. Display the contents of the your_name.json file by executing:

   a.

```
$cat data_bags/passwords/your_name.json
```

```
/home/ubuntu/chef-repo $cat data_bags/passwords/your_name.json
{
  "id": "your_name",
  "password": {
    "encrypted_data": "s8iGOLxiXQ+R/0ZpPphpFsKz+0MnzPyqLl1xN8WPN35BZuE/IhO6R5VTQ4gp\ncjDFKuCiHAE=\n
    "iv": "oVMPPohgOAm9oQAy\n",
    "auth_tag": "JhX5OGsFX5HO9g2ocy/DlA==\n",
    "version": 3,
    "cipher": "aes-256-gcm"
  }
```

5. Viewing an encrypted data bag with a secret file:

   a. **Execute:**

```
$knife data bag show --secret-file .my_secret  passwords your_name -z
```

```
WARNING: No knife configuration file found. See https://docs.chef.io/config_rb/ for detai
Encrypted data bag detected, decrypting with provided secret.
id:       your_name
password: $1$NLLKU4pn$nCh5VZ7IO7A5EZHicFpmu.
```

   NOTE: the difference between this output and the one in step 3 is that the output is decoded by decrypting with a provided secret.

6. Accessing data bag values with the default recipe:

   - Normally, the secret file (**.my_secret**) needs to be copied over to the node on which chef-client runs in order to unencrypt the data bag contents.

   - **When testing locally (not on a chef server), the kitchen.yml file is used as a pointer for the secret file.**

   a. Update cookbooks/users/kitchen.yml with below content:

```
suites:
  - name: default
    data_bags_path: '../../data_bags'
    encrypted_data_bag_secret_key_path: '../../.my_secret'
    sudo: true
    verifier:
      inspec_tests:
        - test/integration/default
    attributes:
```

   NOTE: The default recipe doesn't need to be updated.

Verify the encrypted data bag changes using kitchen commands:

7. From the users directory, destroy and create the kitchen

Note: the '**&&**' notation means "**IF** the first command succeeds, **THEN** execute the second command"

$ kitchen destroy && kitchen converge

```
Chef Infra Client finished, 1/1 resources updated in 21 seconds
       Finished converging <default-centos-7> (0m26.88s).
-----> Creating <default-ubuntu-2004>...
```

```
Chef Infra Client finished, 1/1 resources updated in 21 seconds
       Finished converging <default-ubuntu-2004> (0m27.42s).
-----> Test Kitchen is finished. (0m59.01s)
```

9. Execute `$kitchen list`

```
Instance             Driver  Provisioner  Verifier  Transport  Last Action  Last
default-centos-7     Dokken  Dokken       Inspec    Dokken     Converged    <None
default-ubuntu-2004  Dokken  Dokken       Inspec    Dokken     Converged    <None
```

10. Execute `$kitchen login default-centos-7`

```
[root@dokken /]#
```

11. Execute `# su your_name`, then `$ su your_name`, then enter '*hashpassword*' when prompted for a password, showing that your test environment decrypted the encrypted data bag!

```
[root@dokken /]# su your_name
[your_name@dokken /]$ su your_name
Password:
[your_name@dokken /]$
```

Using an incorrect password:

12. Try to su again, but enter an incorrect password when prompted

```
[your_name@dokken /]$ su your_name
Password:
su: Authentication failure
```

13. Exit out of the kitchen back to the workstation

Notify your instructor that you are done with the lab

END OF LAB