Ansible Demonstration

Travis Michette

Version 1.0

Table of Contents

System Administration Demos Traditional vs. Ansible

1. Traditional Administration from the CLI

Without using an automation language or platform, administrators have traditionally created batch scripts of loop commands to administer the system from a CLI interface.

1.1. A Traditional RHCSA method to add a new user for a single system

1. Adding a new user and password from the command prompt.

   Listing 1. Creating a User from the Command Line

   ```
   useradd -c "Travis Michette" -u 5000 travis
   ```

   

   useradd *Options*

   - **-u** options allows specifying a **userid**
   - **-c** option specifies the GECOS comment which is the user's full name.

   Listing 2. Setting a User Password from the Command Line

   ```
   echo redhat | passwd travis --stdin
   ```

2. Add user to Sudoers file to allow sudo without a password.

   Listing 3. Adding a User without Requiring Password to Sudoers File from the Command Line in a Loop

   ```
   echo "travis2 ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/travis2
   ```

1.2. A Traditional RHCSA method to add a new user for multiple systems

1. Adding a new user and password on multiple system from the command prompt.

   

   Bash **for** *loops to execute commands on multiple systems*

   A **for** loop can be used to execute commands over multiple systems via SSH. This allows an administrator

Listing 4. Creating a User from the Command Line in a Loop

```
hosts="hosta hostb"
for host in $hosts
do
  echo $host
  ssh -t root@$host 'useradd -c "Travis Michette" -u 8000 travis2'
done
```

useradd *Options*

- **-u** options allows specifying a **userid**
- **-c** option specifies the GECOS comment which is the user's full name.

Listing 5. Setting a User Password from the Command Line

```
hosts="hosta hostb"
for host in $hosts
do
  echo $host
  ssh -t root@$host 'echo redhat | passwd travis2 --stdin'
done
```

2. Add user to Sudoers file to allow sudo without a password on multiple systems.

Listing 6. Adding a User without Requiring Password to Sudoers File from the Command Line in a Loop

```
hosts="hosta hostb"
for host in $hosts
do
  echo $host
  ssh -t root@$host 'echo "travis2 ALL=(ALL) NOPASSWD:ALL" > /etc/sudoers.d/travis2'
done
```

2. System Administation with Ansible

2.1. Ansible Ad-Hoc Commands

Ansible **ad-hoc** commands allow administrators to perform repetitive tasks on one or more systems by leveraging an Ansible task interactively from the CLI.

Ansible **copy** *Module Options*

It is important to validate a **sudoers** file to ensure that there are no errors that will break the ability to SUDO. VISUDO validates for you as improper entries in SUDO can kill the ability to SUDO.

Listing 7. Ansible **copy** *Module in use*

```
ansible -m copy -a 'content="travis ALL=(ALL) NOPASSWD:ALL"
dest=/etc/sudoers.d/travis validate="/usr/sbin/visudo -csf %s"'
```

Creating a user with Ansible ad-hoc commands

1. Use the **user** module to create a user on one or more remote systems

Listing 8. Creating the User

```
ansible -m user -a 'name=travis uid=9000 comment="Travis Michette" state=present'
servera.lab.example.com
```

Listing 9. Create a Password for the User

```
ansible -m shell -a 'echo redhat | passwd travis --stdin'
```

Command-Line Method

The above ad-hoc Ansible command will leverage the traditional CLI method and shell redirection to allow setting the password for the provided user without providing the interactive password verification.

2. Add the user to the Sudoers file

Listing 10. Ansible **ad-hoc** *method for updating/creating a sudoers file*

```
ansible -m copy -a 'content="travis ALL=(ALL) NOPASSWD:ALL" dest=/etc/sudoers.d/travis
validate="/usr/sbin/visudo -csf %s"'
```

2.2. Ansible Playbook

It is also possible to leverage a playbook to create the users and add them to one or more remote systems. A playbook will allow Infrastructure-as-Code and will allow for easily moving and changing users around on remote systems.

Listing 11. Ansible Playbook Snippet to Create User Password

```
## Playbook snippet for setting password
### Create the Ansible User
- name: Create Ansible User Password (if required)
  shell: echo {{ ansible_user_password }} | passwd {{ ansible_user_name }} --stdin
  when: ansible_user_password is defined
```