# Red Hat Enterprise Linux Automation with Ansible

FEEDBACK TRANSLATIONS CERTIFICATE OF ATTENDANCE

P (/rol/app/courses/rh294-8.4/pages/pr01)
(/rol/app/courses/rh294-8.4/pages/pr01)
(/rol/app/courses/rh294-8.4/pages/pr01s02)
(/rol/app/courses/rh294-8.4/pages/ch01s02)
(/rol/app/courses/rh294-8.4/pages/ch01s03)
(/rol/app/courses/rh294-8.4/pages/ch01s04)
(/rol/app/courses/rh294-8.4/pages/ch01s05)
2 (/rol/app/courses/rh294-8.4/pages/ch02)
(/rol/app/courses/rh294-8.4/pages/ch02)
(/rol/app/courses/rh294-8.4/pages/ch02s02)
(/rol/app/courses/rh294-8.4/pages/ch02s03)
(/rol/app/courses/rh294-8.4/pages/ch02s04)
(/rol/app/courses/rh294-8.4/pages/ch02s05)
(/rol/app/courses/rh294-8.4/pages/ch02s06)
(/rol/app/courses/rh294-8.4/pages/ch02s07)
(/rol/app/courses/rh294-8.4/pages/ch02s08)
(/rol/app/courses/rh294-8.4/pages/ch02s09)
(/rol/app/courses/rh294-8.4/pages/ch02s10)
(/rol/app/courses/rh294-8.4/pages/ch02s11)
(/rol/app/courses/rh294-8.4/pages/ch02s12)
3 (/rol/app/courses/rh294-8.4/pages/ch03)
(/rol/app/courses/rh294-8.4/pages/ch03)
(/rol/app/courses/rh294-8.4/pages/ch03s02)
(/rol/app/courses/rh294-8.4/pages/ch03s03)
(/rol/app/courses/rh294-8.4/pages/ch03s04)
(/rol/app/courses/rh294-8.4/pages/ch03s05)
(/rol/app/courses/rh294-8.4/pages/ch03s06)
(/rol/app/courses/rh294-8.4/pages/ch03s07)
(/rol/app/courses/rh294-8.4/pages/ch03s08)
4 (/rol/app/courses/rh294-8.4/pages/ch04)
(/rol/app/courses/rh294-8.4/pages/ch04)
(/rol/app/courses/rh294-8.4/pages/ch04s02)
(/rol/app/courses/rh294-8.4/pages/ch04s03)
(/rol/app/courses/rh294-8.4/pages/ch04s04)
(/rol/app/courses/rh294-8.4/pages/ch04s05)
(/rol/app/courses/rh294-8.4/pages/ch04s06)
(/rol/app/courses/rh294-8.4/pages/ch04s07)
(/rol/app/courses/rh294-8.4/pages/ch04s08)
5 (/rol/app/courses/rh294-8.4/pages/ch05)
(/rol/app/courses/rh294-8.4/pages/ch05)
(/rol/app/courses/rh294-8.4/pages/ch05s02)
(/rol/app/courses/rh294-8.4/pages/ch05s03)
(/rol/app/courses/rh294-8.4/pages/ch05s04)
(/rol/app/courses/rh294-8.4/pages/ch05s05)
(/rol/app/courses/rh294-8.4/pages/ch05s06)
6 (/rol/app/courses/rh294-8.4/pages/ch06)
(/rol/app/courses/rh294-8.4/pages/ch06)
(/rol/app/courses/rh294-8.4/pages/ch06s02)
(/rol/app/courses/rh294-8.4/pages/ch06s03)
(/rol/app/courses/rh294-8.4/pages/ch06s04)
(/rol/app/courses/rh294-8.4/pages/ch06s05)
(/rol/app/courses/rh294-8.4/pages/ch06s06)
7 (/rol/app/courses/rh294-8.4/pages/ch07)
(/rol/app/courses/rh294-8.4/pages/ch07)
(/rol/app/courses/rh294-8.4/pages/ch07s02)
(/rol/app/courses/rh294-8.4/pages/ch07s03)
(/rol/app/courses/rh294-8.4/pages/ch07s04)
(/rol/app/courses/rh294-8.4/pages/ch07s05)
(/rol/app/courses/rh294-8.4/pages/ch07s06)
(/rol/app/courses/rh294-8.4/pages/ch07s07)
(/rol/app/courses/rh294-8.4/pages/ch07s08)
(/rol/app/courses/rh294-8.4/pages/ch07s09)
(/rol/app/courses/rh294-8.4/pages/ch07s10)
(/rol/app/courses/rh294-8.4/pages/ch07s11)
(/rol/app/courses/rh294-8.4/pages/ch07s12)
8 (/rol/app/courses/rh294-8.4/pages/ch08)
(/rol/app/courses/rh294-8.4/pages/ch08)
(/rol/app/courses/rh294-8.4/pages/ch08s02)
(/rol/app/courses/rh294-8.4/pages/ch08s03)
(/rol/app/courses/rh294-8.4/pages/ch08s04)
(/rol/app/courses/rh294-8.4/pages/ch08s05)
(/rol/app/courses/rh294-8.4/pages/ch08s06)
9 (/rol/app/courses/rh294-8.4/pages/ch09)
(/rol/app/courses/rh294-8.4/pages/ch09)
(/rol/app/courses/rh294-8.4/pages/ch09s02)
(/rol/app/courses/rh294-8.4/pages/ch09s03)
(/rol/app/courses/rh294-8.4/pages/ch09s04)
(/rol/app/courses/rh294-8.4/pages/ch09s05)
(/rol/app/courses/rh294-8.4/pages/ch09s06)
(/rol/app/courses/rh294-8.4/pages/ch09s07)
(/rol/app/courses/rh294-8.4/pages/ch09s08)
(/rol/app/courses/rh294-8.4/pages/ch09s09)
(/rol/app/courses/rh294-8.4/pages/ch09s10)
(/rol/app/courses/rh294-8.4/pages/ch09s11)
(/rol/app/courses/rh294-8.4/pages/ch09s12)
10 (/rol/app/courses/rh294-8.4/pages/ch10)
(/rol/app/courses/rh294-8.4/pages/ch10)
(/rol/app/courses/rh294-8.4/pages/ch10s02)
(/rol/app/courses/rh294-8.4/pages/ch10s03)
(/rol/app/courses/rh294-8.4/pages/ch10s04)
A (/rol/app/courses/rh294-8.4/pages/apa)
(/rol/app/courses/rh294-8.4/pages/apa)
A (/rol/app/courses/rh294-8.4/pages/apb)
(/rol/app/courses/rh294-8.4/pages/apb)

# Guided Exercise: Running Ad Hoc Commands ☆

In this exercise, you will execute ad hoc commands on multiple managed hosts.

**Outcomes**

You should be able to execute commands on managed hosts on an ad hoc basis using privilege escalation.

You will execute ad hoc commands on `workstation` and `servera` using the `devops` user account. This account has the same `sudo` configuration on both `workstation` and `servera`.

Log in to `workstation` as `student` using `student` as the password.

On `workstation`, run the `lab deploy-adhoc start` command. This script ensures that the managed host `servera` is reachable on the network. It also creates and populates the `/home/student/deploy-adhoc` working directory with materials used in this exercise.

```
[student@workstation ~]$ lab deploy-adhoc start
```

**Procedure 2.3. Instructions**

1. Determine the `sudo` configuration for the `devops` account on both `workstation` and `servera`.

   1.1. Determine the `sudo` configuration for the `devops` account that was configured when `workstation` was built. Enter `student` if prompted for the password for the `student` account.

   ```
   [student@workstation ~]$ sudo -l -U devops
   ...output omitted...
   User devops may run the following commands on workstation:
       (ALL) NOPASSWD: ALL
   ```

   Note that the user has full `sudo` privileges but does not require password authentication.

   1.2. Determine the `sudo` configuration for the `devops` account that was configured when `servera` was built.

   ```
   [student@workstation ~]$ ssh devops@servera.lab.example.com
   [devops@servera ~]$ sudo -l
   ...output omitted...
   User devops may run the following commands on servera:
       (ALL) NOPASSWD: ALL
   [devops@servera ~]$ exit
   ```

   Note that the user has full `sudo` privileges but does not require password authentication.

2. Change directory to `/home/student/deploy-adhoc` and examine the contents of the `ansible.cfg` and `inventory` files.

```
[student@workstation ~]$ cd ~/deploy-adhoc
[student@workstation deploy-adhoc]$ cat ansible.cfg
[defaults]
inventory=inventory
[student@workstation deploy-adhoc]$ cat inventory
[control_node]
localhost

[intranetweb]
servera.lab.example.com
```

The configuration file uses the directory's `inventory` file as the Ansible inventory. Note that Ansible is not yet configured to use privilege escalation.

3. Using the `all` host group and the `ping` module, execute an ad hoc command that ensures all managed hosts can run Ansible modules using Python.

```
[student@workstation deploy-adhoc]$ ansible all -m ping
servera.lab.example.com | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
```

4. Using the `command` module, execute an ad hoc command on `workstation` to identify the user account that Ansible uses to perform operations on managed hosts. Use the `localhost` host pattern to connect to `workstation` for the ad hoc command execution. Because you are connecting locally, `workstation` is both the control node and managed host.

```
[student@workstation deploy-adhoc]$ ansible localhost -m command -a 'id'
localhost | CHANGED | rc=0 >>
uid=1000(student) gid=1000(student) groups=1000(student),10(wheel) context=unconfined_u:unconfined_r:unconfined_t:s0-s
0:c0.c1023
```

Notice that the ad hoc command was performed on the managed host as the `student` user.

5. Execute the previous ad hoc command on `workstation` but connect and perform the operation with the `devops` user account by using the `-u` option.

```
[student@workstation deploy-adhoc]$ ansible localhost -m command -a 'id' -u devops
localhost | CHANGED | rc=0 >>
uid=1001(devops) gid=1001(devops) groups=1001(devops) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Notice that the ad hoc command was performed on the managed host as the `devops` user.

6. Using the `copy` module, execute an ad hoc command on `workstation` to change the contents of the `/etc/motd` file so that it consists of the string "Managed by Ansible" followed by a newline. Execute the command using the `devops` account, but do not use the `--become` option to switch to `root`. The ad hoc command should fail due to lack of permissions.

```
[student@workstation deploy-adhoc]$ ansible localhost -m copy \
> -a 'content="Managed by Ansible\n" dest=/etc/motd' -u devops
localhost | FAILED! => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "checksum": "4458b979ede3c332f8f2128385df4ba305e58c27",
    "msg": "Destination /etc not writable"
}
```

The ad hoc command failed because the `devops` user does not have permission to write to the file.

7. Run the command again using privilege escalation. You could fix the settings in the `ansible.cfg` file, but for this example just use appropriate command-line options of the `ansible` command.

   Using the `copy` module, execute the previous command on `workstation` to change the contents of the `/etc/motd` file so that it consists of the string "Managed by Ansible" followed by a newline. Use the `devops` user to make the connection to the managed host, but perform the operation as the `root` user using the `--become` option. The use of the `--become` option is sufficient because the default value for the `become_user` directive is set to `root` in the `/etc/ansible/ansible.cfg` file.

```
[student@workstation deploy-adhoc]$ ansible localhost -m copy \
> -a 'content="Managed by Ansible\n" dest=/etc/motd' -u devops --become
localhost | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "checksum": "4458b979ede3c332f8f2128385df4ba305e58c27",
    "dest": "/etc/motd",
    "gid": 0,
    "group": "root",
    "md5sum": "65a4290ee5559756ad04e558b0e0c4e3",
    "mode": "0644",
    "owner": "root",
    "secontext": "system_u:object_r:etc_t:s0",
    "size": 19,
    "src": "/home/devops/.ansible/tmp/ansible-tmp-1558954193.0260043-131348380629718/source",
    "state": "file",
    "uid": 0
}
```

Note that the command succeeded this time because the ad hoc command was executed with privilege escalation.

8. Run the previous ad hoc command again on all hosts using the `all` host group. This ensures that `/etc/motd` on both `workstation` and `servera` consist of the text "`Managed by Ansible`".

```
[student@workstation deploy-adhoc]$ ansible all -m copy \
> -a 'content="Managed by Ansible\n" dest=/etc/motd' -u devops --become
servera.lab.example.com | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "checksum": "4458b979ede3c332f8f2128385df4ba305e58c27",
    "dest": "/etc/motd",
    "gid": 0,
    "group": "root",
    "md5sum": "65a4290ee5559756ad04e558b0e0c4e3",
    "mode": "0644",
    "owner": "root",
    "secontext": "system_u:object_r:etc_t:s0",
    "size": 19,
    "src": "/home/devops/.ansible/tmp/ansible-tmp-1558954250.7893758-136255396678462/source",
    "state": "file",
    "uid": 0
}
localhost | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "checksum": "4458b979ede3c332f8f2128385df4ba305e58c27",
    "dest": "/etc/motd",
    "gid": 0,
    "group": "root",
    "mode": "0644",
    "owner": "root",
    "path": "/etc/motd",
    "secontext": "system_u:object_r:etc_t:s0",
    "size": 19,
    "state": "file",
    "uid": 0
}
```

You should see `SUCCESS` for `localhost` and `CHANGED` for `servera`. However, `localhost` should report `"changed": false` because the file is already in the correct state. Conversely, `servera` should report `"changed": true` because the ad hoc command updated the file to the correct state.

9. Using the `command` module, execute an ad hoc command to run `cat /etc/motd` to verify that the contents of the file have been successfully modified on both `workstation` and `servera`. Use the `all` host group and the `devops` user to specify and make the connection to the managed hosts. You do not need privilege escalation for this command to work.

```
[student@workstation deploy-adhoc]$ ansible all -m command \
> -a 'cat /etc/motd' -u devops
servera.lab.example.com | CHANGED | rc=0 >>
Managed by Ansible

localhost | CHANGED | rc=0 >>
Managed by Ansible
```

### Finish

On `workstation`, run the `lab deploy-adhoc finish` script to clean up this exercise.

```
[student@workstation ~]$ lab deploy-adhoc finish
```

This concludes the guided exercise.

← PREVIOUS (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S05) → NEXT (/ROL/APP/COURSES/RH294-8.4/PAGES/CH02S07)

Privacy Policy (http://s.bl-1.com/h/cZrgWbQn?url=https://www.redhat.com/en/about/privacy-policy?extIdCarryOver=true&sc_cid=701f2000001D8QoAAK)

Red Hat Training Policies (http://s.bl-1.com/h/cZrb2DXG?url=https://www.redhat.com/en/about/red-hat-training-policies)

Terms of Use (https://www.redhat.com/en/about/terms-use)

All policies and guidelines (https://www.redhat.com/en/about/all-policies-guidelines)

Release Notes (https://learn.redhat.com/t5/Red-Hat-Learning-Subscription/Red-Hat-Learning-Subscription-Release-Notes/ba-p/22952)

Cookie Preferences