Red Hat
Ansible Automation
Platform

Bank United
Ansible Lunch and Learn

A quick introduction to writing playbooks

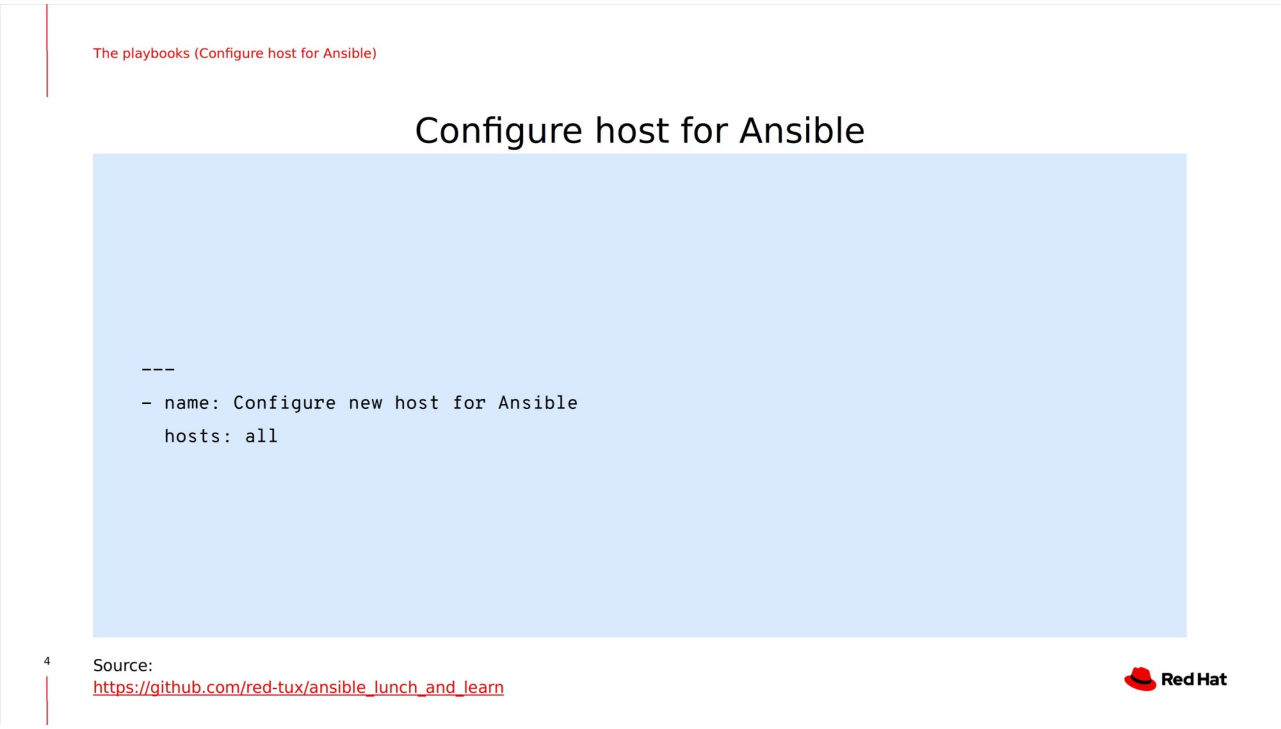
Andrew Nelson
Consulting Architect

Overview

- What are the playbooks we’ll be looking at
- Ensuring variables are set
- Templating files
- Modifying files
- Using handlers
- Tying it all together with Tower

- Two basic playbooks
  - Configure host for Ansible
  - Configure host to the environment baseline
- Custom roles are used for specific components
  - Network time
  - Banner/MOTD messages
- Available for download at:
  - https://github.com/red-tux/ansible_lunch_and_learn

Source:
https://github.com/red-tux/ansible_lunch_and_learn



The playbooks (Configure host for Ansible)

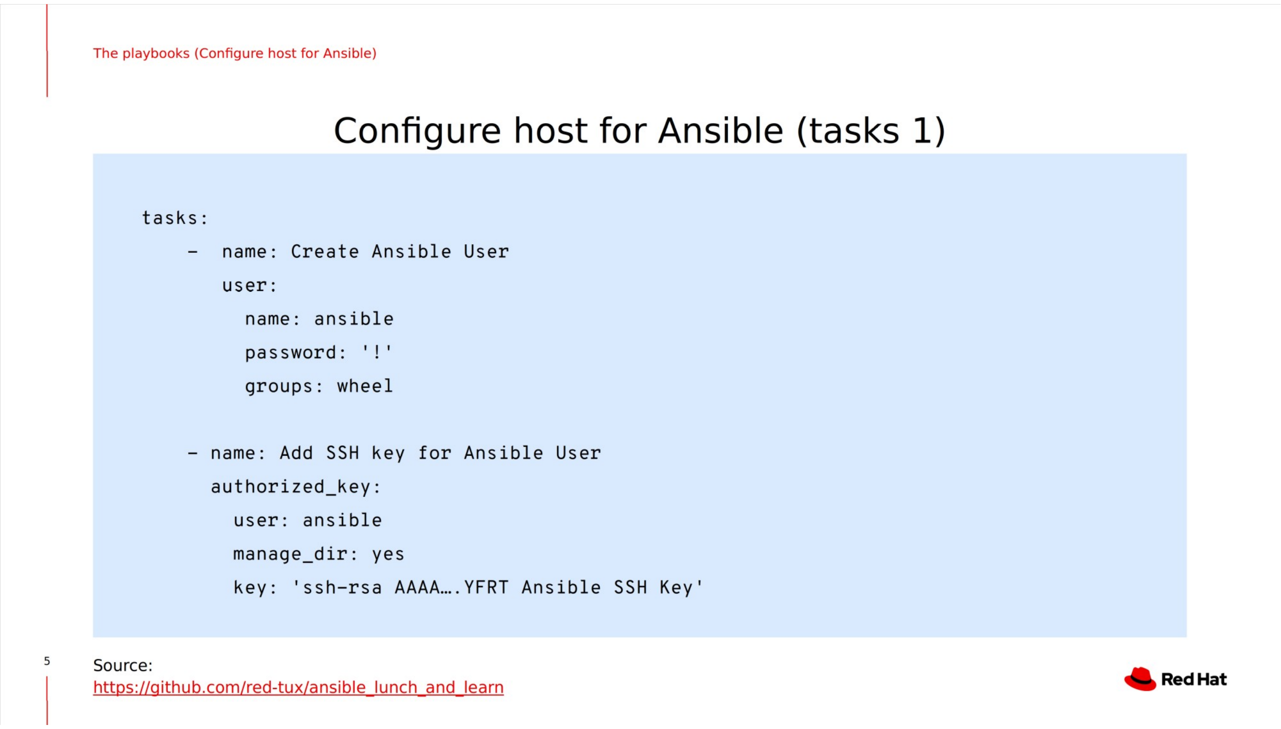
Configure host for Ansible

```
---
- name: Configure new host for Ansible
  hosts: all
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

This is the main play in the playbook to configure a host to use Ansible.
Sometimes a playbook like this is needed to configure the user Ansible will use, along with any appropriate privilege escalation rules.
It is worth noting that a playbook can contain multiple plays.

Because of the nature of this specific playbook it is intended to be run as the root user. However because the playbook is intended to be run from Ansible Tower, a login credential created in Tower is then associated with the job template for this playbook.



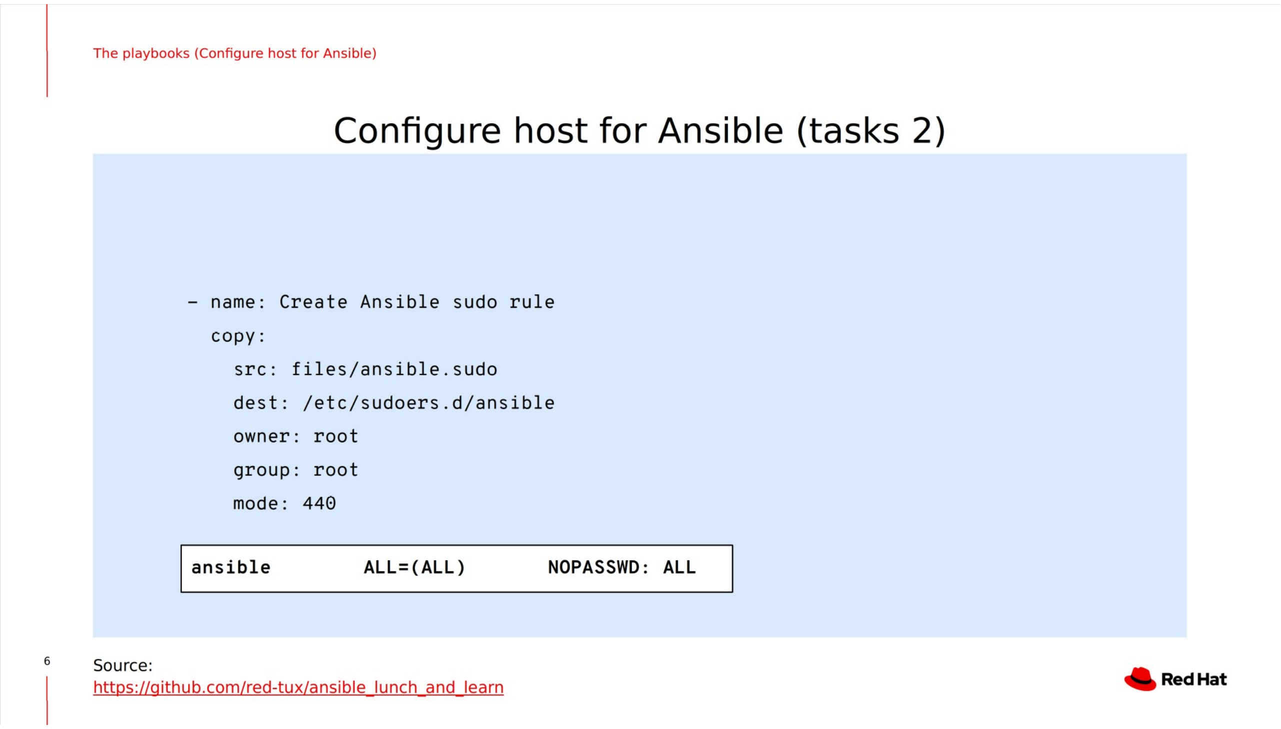
The playbooks (Configure host for Ansible)

Configure host for Ansible (tasks 1)

```
tasks:
    -  name: Create Ansible User
       user:
         name: ansible
         password: '!'
         groups: wheel

    - name: Add SSH key for Ansible User
      authorized_key:
        user: ansible
        manage_dir: yes
        key: 'ssh-rsa AAAA….YFRT Ansible SSH Key'
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

The first two tasks in the playbook. While it is difficult to tell, the tasks are indented to nest inside the play defined earlier.
The first task creates an ansible user and sets an unusable password.
The second task ensures that the public key for the Ansible user is installed



The playbooks (Configure host for Ansible)

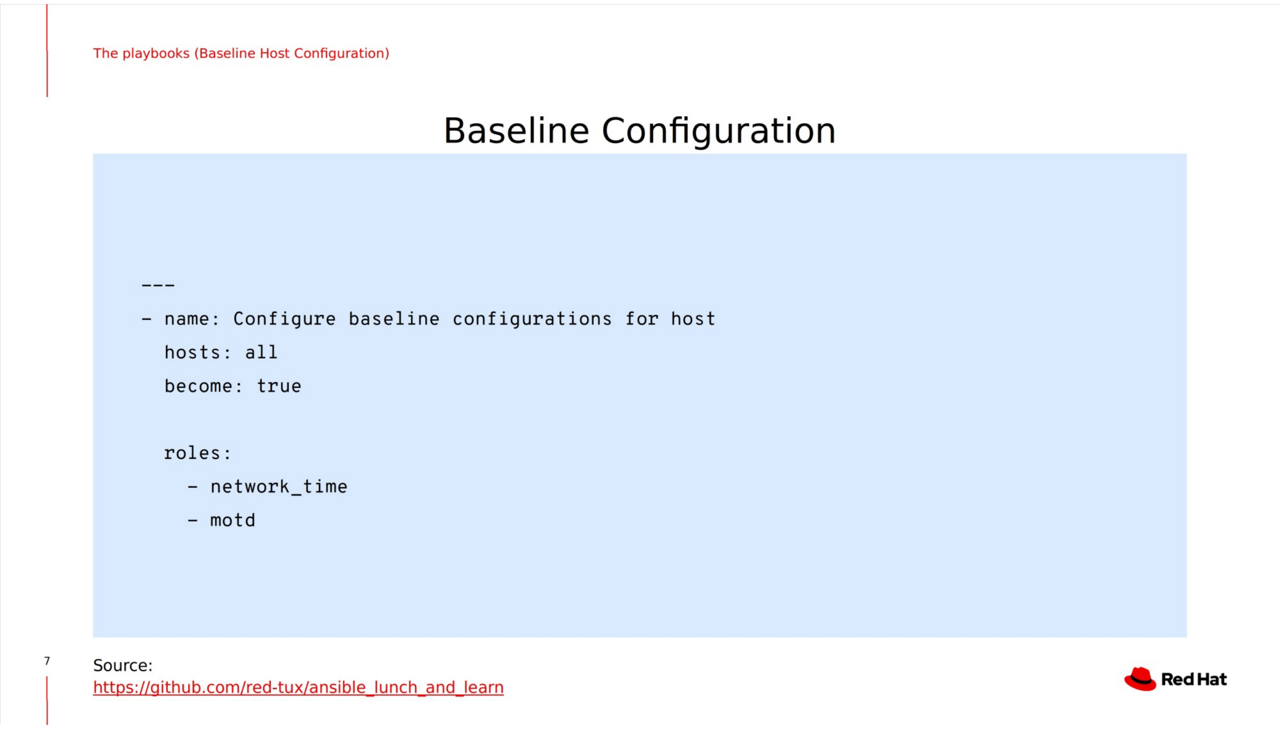
Configure host for Ansible (tasks 2)

```
- name: Create Ansible sudo rule
  copy:
    src: files/ansible.sudo
    dest: /etc/sudoers.d/ansible
    owner: root
    group: root
    mode: 440
```

```
ansible        ALL=(ALL)       NOPASSWD: ALL
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

The third task ensures there is a sudoers rule on the host to allow ansible to become root without password.
Rather than edit the main sudoers file it is recommended to make use of the /etc/sudoers.d directory. This allows for modular sudoers configurations.



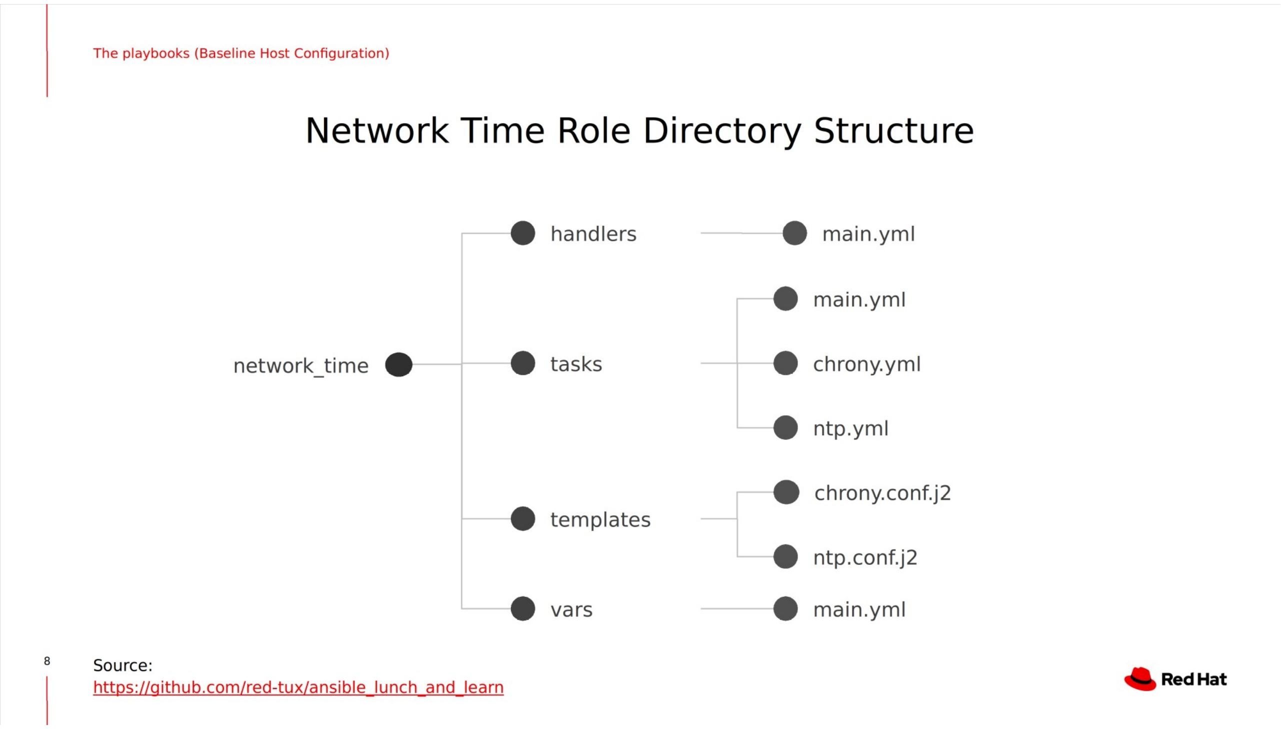
The playbooks (Baseline Host Configuration)

Baseline Configuration

```
---
- name: Configure baseline configurations for host
  hosts: all
  become: true

  roles:
    - network_time
    - motd
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

This is the playbook to ensure a baseline configuration is set on the host.
Note the use of the “become” statement. This is needed as this playbook is intended to be used with the previously created ansible user. “Become” tells Ansible that privilege escalation will be required and defaults to sudo.
Next the play calls two roles to configure network time and logon banners.
It is recommended to use Ansible Collections for roles which will be included in multiple projects.
https://docs.ansible.com/ansible/latest/user_guide/collections_using.html



The playbooks (Baseline Host Configuration)

Network Time Role Directory Structure

network_time

handlers

main.yml

tasks

main.yml

chrony.yml

ntp.yml

templates

chrony.conf.j2

ntp.conf.j2

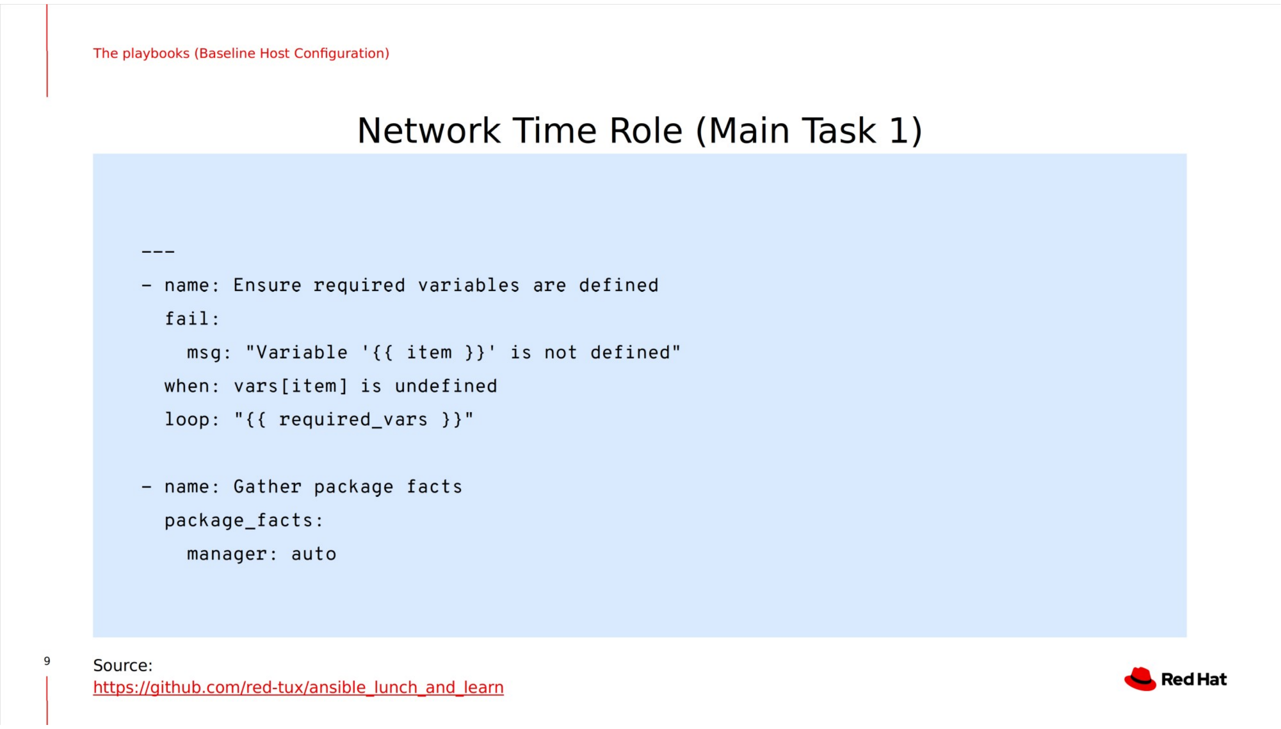
vars

main.yml

Source:
https://github.com/red-tux/ansible_lunch_and_learn

Red Hat

This is the directory structure to the Network Time role.
The motd role directory structure is similar and not shown for brevity.



The playbooks (Baseline Host Configuration)

Network Time Role (Main Task 1)

```
---
- name: Ensure required variables are defined
  fail:
    msg: "Variable '{{ item }}' is not defined"
  when: vars[item] is undefined
  loop: "{{ required_vars }}"

- name: Gather package facts
  package_facts:
    manager: auto
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

With roles the main entry point is the main.yml file in the tasks directory.

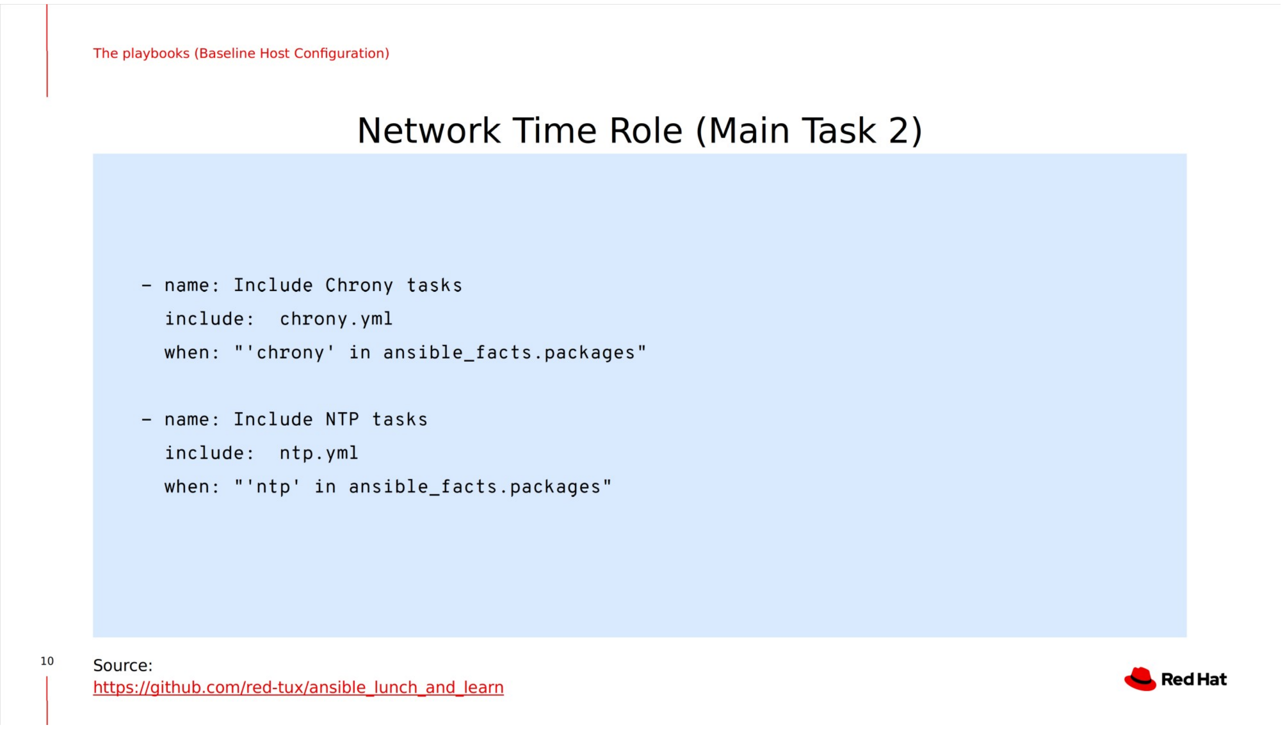
This file contains a list of tasks.

The first task iterates through a variable defined in the main.yml file in the variables directory. This variable contains a list of other variables required by this role, if any of the required variables are not defined the role will fail with a message saying which variable was not defined.

In this case, it is intended that the list of NTP servers to use is defined as a group level variable in the Ansible inventory.

It is a best practice to use defensive coding techniques such as this.

Next the role queries the remote system to determine which RPMs are installed. This module poulates the variable “ansible_facts.packages”.



The playbooks (Baseline Host Configuration)

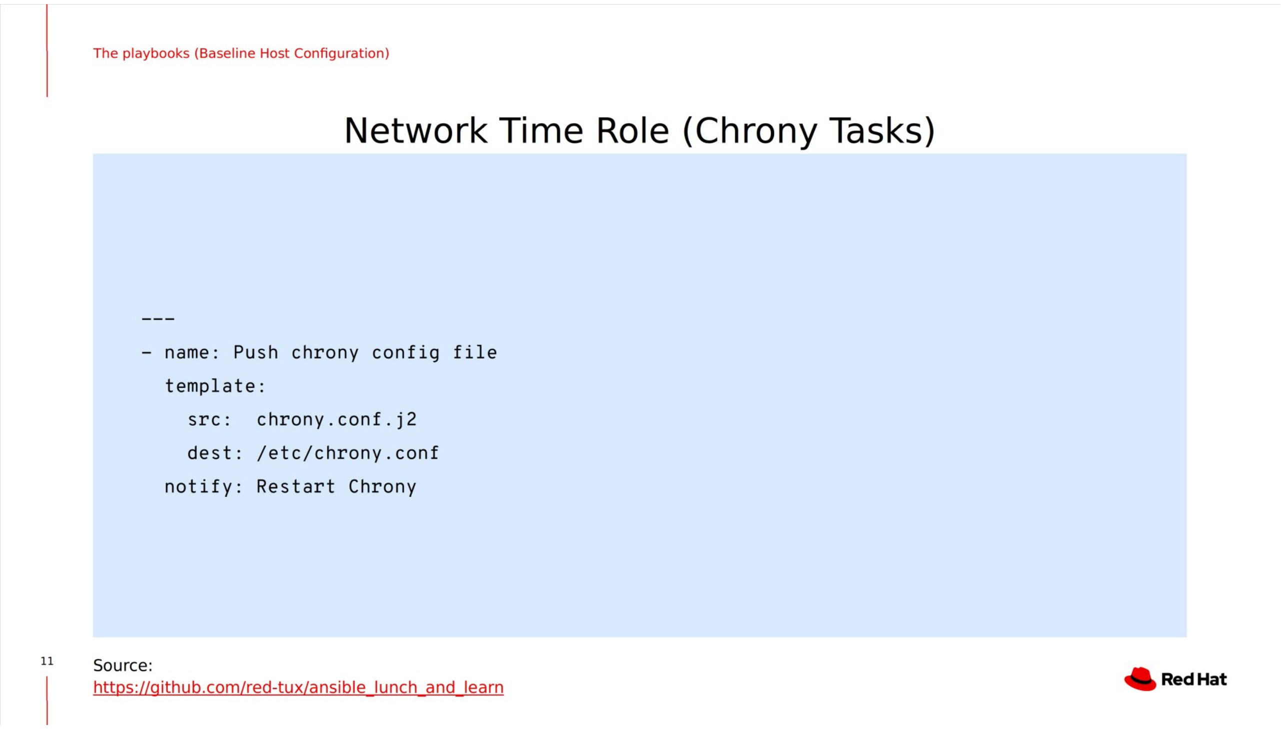
Network Time Role (Main Task 2)

```
- name: Include Chrony tasks
  include:  chrony.yml
  when: "'chrony' in ansible_facts.packages"

- name: Include NTP tasks
  include:  ntp.yml
  when: "'ntp' in ansible_facts.packages"
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

Next the role will include Chrony or NTP specific tasks depending on which RPM is installed.

Exercise to the reader: What would you add to ensure that only Chrony or NTP can be installed, not both?



The playbooks (Baseline Host Configuration)

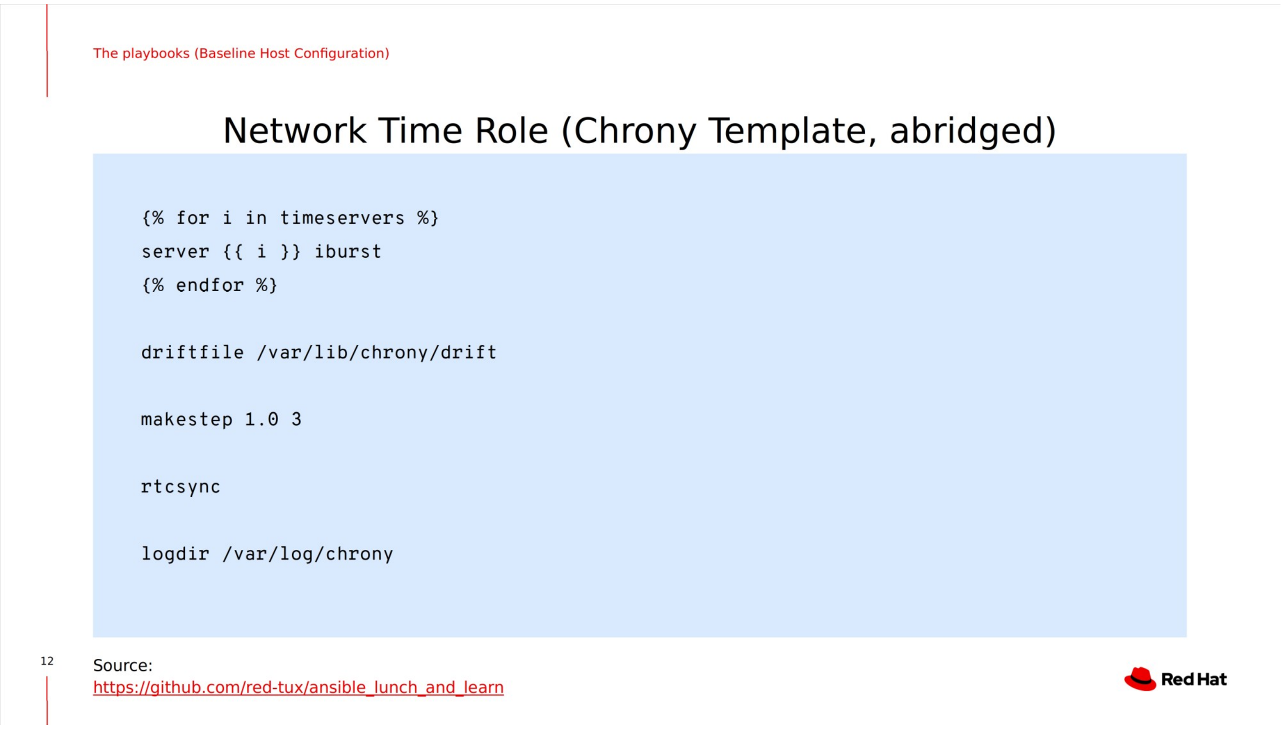
Network Time Role (Chrony Tasks)

```
---
- name: Push chrony config file
  template:
    src:  chrony.conf.j2
    dest: /etc/chrony.conf
  notify: Restart Chrony
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

These are the Chrony specific tasks.
The notify statement cause all named handlers to run should the task result in a change.

The NTP tasks are not shown for brevity but are similar.



The playbooks (Baseline Host Configuration)

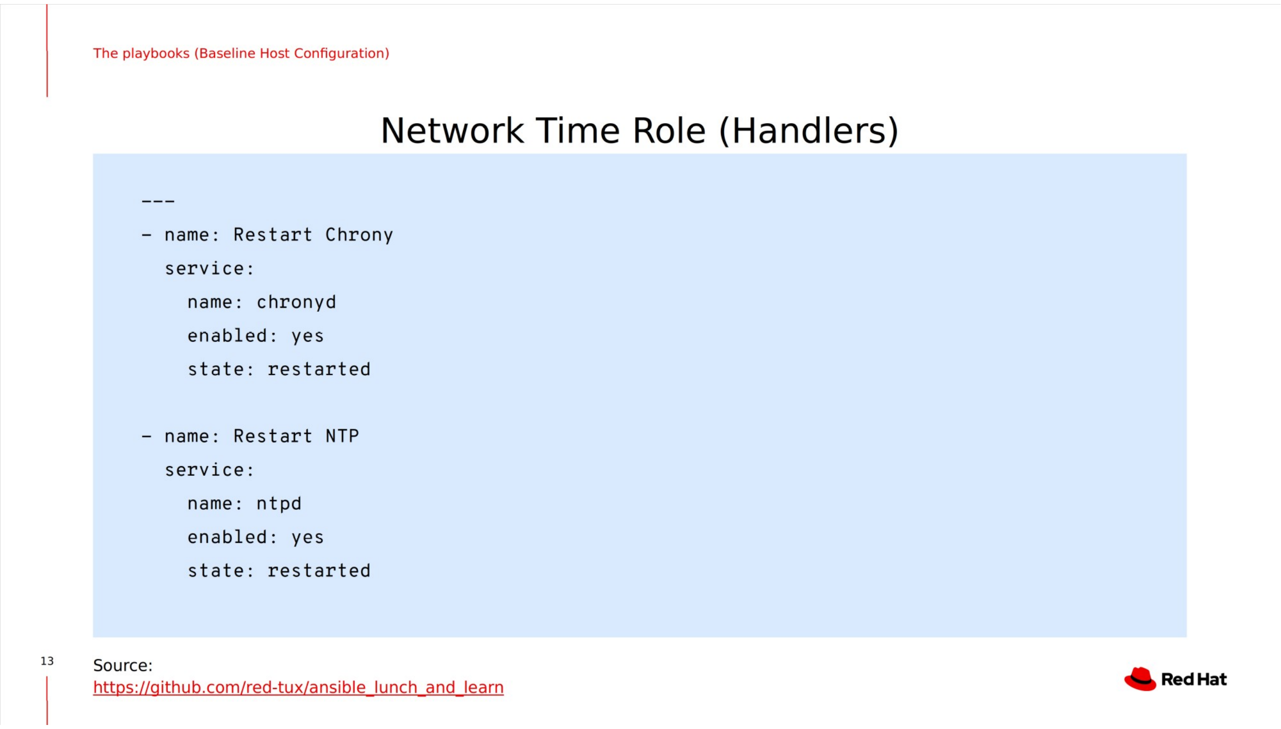
Network Time Role (Chrony Template, abridged)

```
{% for i in timeservers %}
server {{ i }} iburst
{% endfor %}

driftfile /var/lib/chrony/drift

makestep 1.0 3

rtcsync

logdir /var/log/chrony
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

This is a shortened version of the Chrony template file.
Notice the for loop at the top allowing for the use of multiple NTP servers in the "timeservers" variable.

The NTP template is not shown for brevity but is similar.

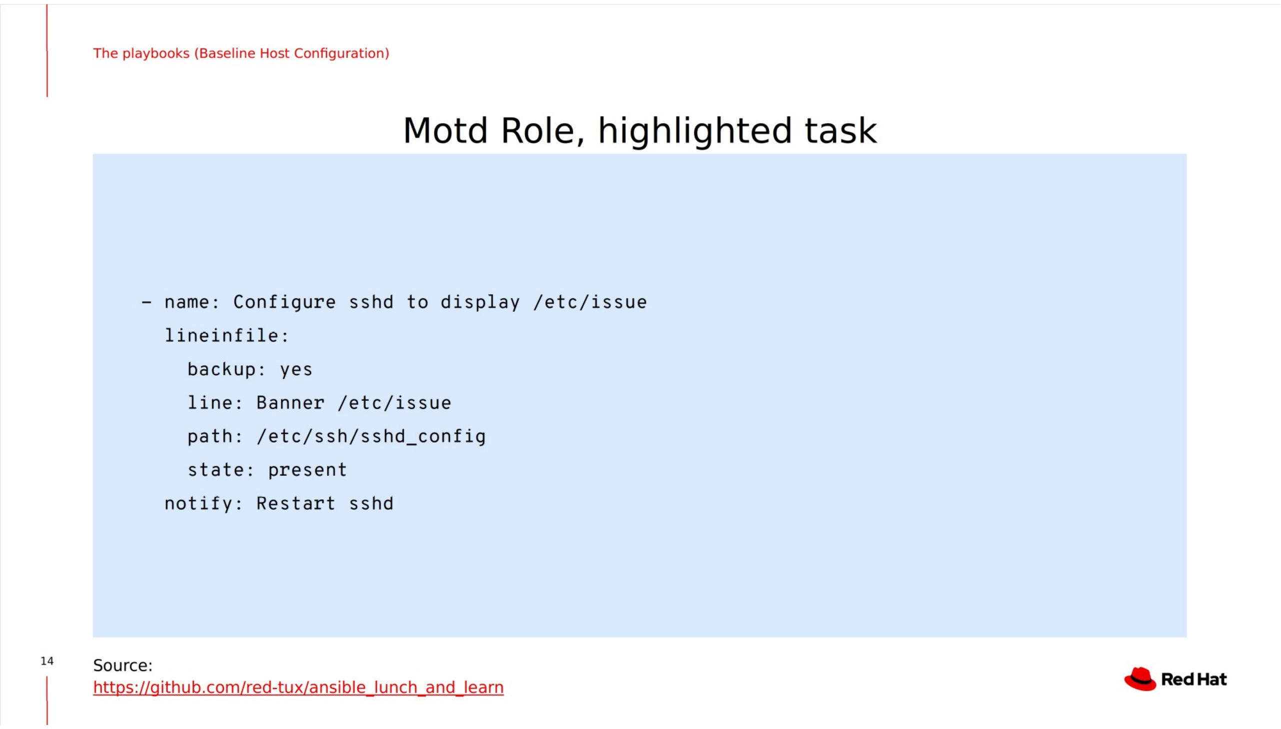


Network Time Role (Handlers)

```
---
- name: Restart Chrony
  service:
    name: chronyd
    enabled: yes
    state: restarted

- name: Restart NTP
  service:
    name: ntpd
    enabled: yes
    state: restarted
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

These are the handlers which are defined in the handlers/main.yml file.
In these cases they will restart Chrony or NTP if the configuration files change, along with ensuring they are enabled.



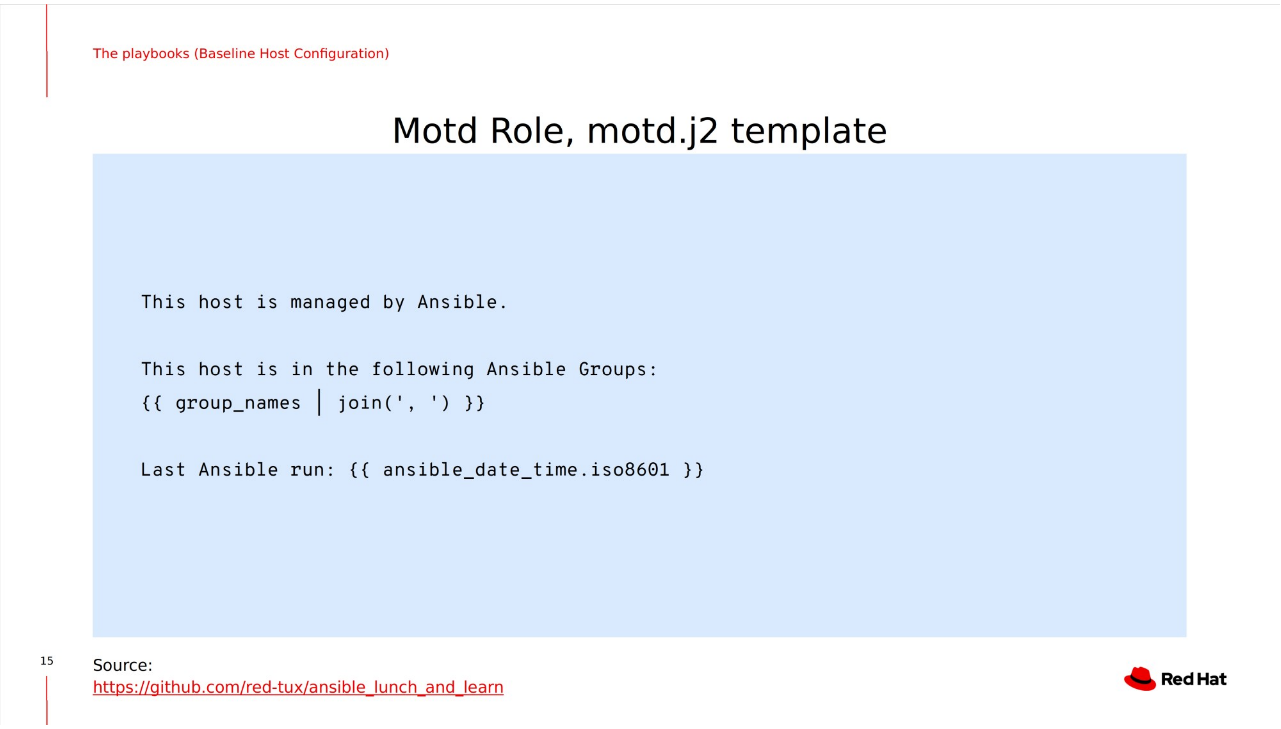
The playbooks (Baseline Host Configuration)

Motd Role, highlighted task

```
- name: Configure sshd to display /etc/issue
  lineinfile:
    backup: yes
    line: Banner /etc/issue
    path: /etc/ssh/sshd_config
    state: present
  notify: Restart sshd
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

In order for the SSH banner to be displayed the sshd_config file must be modified to include a Banner statement. The lineinfile module allows for the inserting of lines at specific places, or at the end.
Also, if there is a change, cause the handler to run to ensure sshd is restarted.



The playbooks (Baseline Host Configuration)

Motd Role, motd.j2 template

```
This host is managed by Ansible.

This host is in the following Ansible Groups:
{{ group_names | join(', ') }}

Last Ansible run: {{ ansible_date_time.iso8601 }}
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

This is the motd template file. The task which deploys this template is not shown for brevity, but is similar to the previously shown template module.
In this template a filter is used to add commas to the list of groups the host is a member of. In addition the timestamp of when the last Ansible run is added.

Sample Output

```
[nelsonab@lenny-mclean-red-tux-net ~]$ ssh root@ansible-test.lab
********************************************************************
*                                                                  *
* This system is for the use of authorized users only.  Usage of   *
* this system may be monitored and recorded by system personnel.   *
*                                                                  *
* Anyone using this system expressly consents to such monitoring   *
* and is advised that if such monitoring reveals possible          *
* evidence of criminal activity, system personnel may provide the  *
* evidence from such monitoring to law enforcement officials.      *
*                                                                  *
********************************************************************
root@ansible-test.lab's password:
Last login: Wed Mar 11 08:59:10 2020 from lenny.mclean.red-tux.net
This host is managed by Ansible.

This host is in the following Ansible Groups:
group1, lab_systems

Last Ansible run: 2020-03-11T13:00:04Z

[root@ansible-test ~]#
```

Source:
https://github.com/red-tux/ansible_lunch_and_learn

Thank you

Red Hat is the world’s leading provider of

enterprise open source software solutions.

Award-winning support, training, and

consulting services make

Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/
RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat