Using Ansible

Jack Desert, Developer for Hire jdbip@riseup.net

See also Ansible Docs: docs.ansible.com/ansible

1. Get a (paper) list of IP addresses

These machines are for you to use.
Choose one of them to be your master box.

2. Log in to the master box

```
ssh ubuntu@ip-address-from-master-box
```

password: rayndrop
(notice the misspelling)

3. Install Ansible on Master Box

```
sudo apt install ansible
```

4. Set up Hosts File on Master Box

```
sudo vi /etc/ansible/hosts
```

alternatively:

```
sudo nano /etc/ansible/hosts
```

Put all the IP addresses EXCEPT your master box in the file.
Format goes like this:

<IP>
<IP>
<IP>
etc.

5. Ping your hosts

```
ansible all -m ping
```

6. What was set up already to allow this to happen?

(Optional) Run these commands if you want to see

```
cat ~/.ansible.cfg         # on master
cat ~/.ssh/authorized_keys # on hosts
ls -al /usr/bin/python     # on hosts
```

7. Ad-Hoc Commands

Ad-Hoc commands are useful for running one-off tasks on all hosts
Try this:

```
ansible all -a "ls /tmp"
```

Using the same pattern, complete the following:

* View the last ten lines of /var/log/nginx/access.log on each host
* Check how much disk space available on each host
* Check which version of git is installed on each host

Find the ansible docs for Ad-Hoc commands, and see if you can:

Apt module:
* Install nodejs on each host

Copy module:
* Copy /etc/passwd from the master machine to /tmp/passwd on each host

User module:
* Create a user named “fred” on each host

8. Flashy New Website

Load the IP address of one of your hosts in a web browser.

Now use the docs for the Service module to stop the service named “flask-demo” on each host. Then refresh the web browser to see a 502 Bad Gateway error from Nginx. And finally, start the service and refresh again to see the page working again.

9. Playbooks

Playbooks are useful when you want something repeatable.

Save this basic playbook as *my_playbook.yml*. It checks out a different branch of the Flashy New Website and then restarts flask-demo.

```
- hosts: all
  become: yes
  become_user: ubuntu
  tasks:
    - git:
        repo: http://git@github.com/jackdesert/ansible-presentation-app
        dest: /home/ubuntu/ansible-presentation-app
        version: green

- hosts: all
  tasks:
    - systemd:
        enabled: yes
        state: restarted
        daemon_reload: yes
        name: flask-demo.service
```

Invoke this playbook as:

```
ansible-playbook my_playbook.yml --become
```

Note this playbook run with the --become flag so it can do superuser things for systemd.
Also note that the git task needs to run as the normal ubuntu user, that's why it has the "become" and "become_user".

After you've got that running, load the Flashy New Website on one of your hosts. Background will be green now.

10. Extra

Expand *my_playbook.yml* so that it deploys the "redis-counter" branch of ansible-presentation-app.

This branch requires you to pip install the "redis" package to the virtualenv at /home/ubuntu/ansible-presentation-app/env. (See docs for the “pip” ansible module.)

This branch also requires you to install "redis" via apt. (See docs for the “apt” ansible module.)

Once deployed, the Flashy Website will count how many times the pages has been visited.