

Ansible for GENI Experimenters

GEC 22
Tim Upthegrove
March 24, 2015



Experimenter

- Private key is stored on local machine
- Ansible client installed
- Inventory file describes substrate



- Public key is stored on each remote machine

Ansible Characteristics

- **Push-based** from a machine of choice
 - **Manage nodes** on-demand **from any host** with an ssh client
 - Can set up a server with recurring Ansible task
 - **Facts** are gathered for managed nodes before making changes to the node
- **Agentless** on managed nodes
 - ssh is used for transport of config and code
 - Modules are run on nodes based on a **playbook**
- Ansible modules are **idempotent**

Bottom Line: Characteristics

There is no need for you to do anything special to your GENI slice in order for you to use Ansible.

Ansible Components

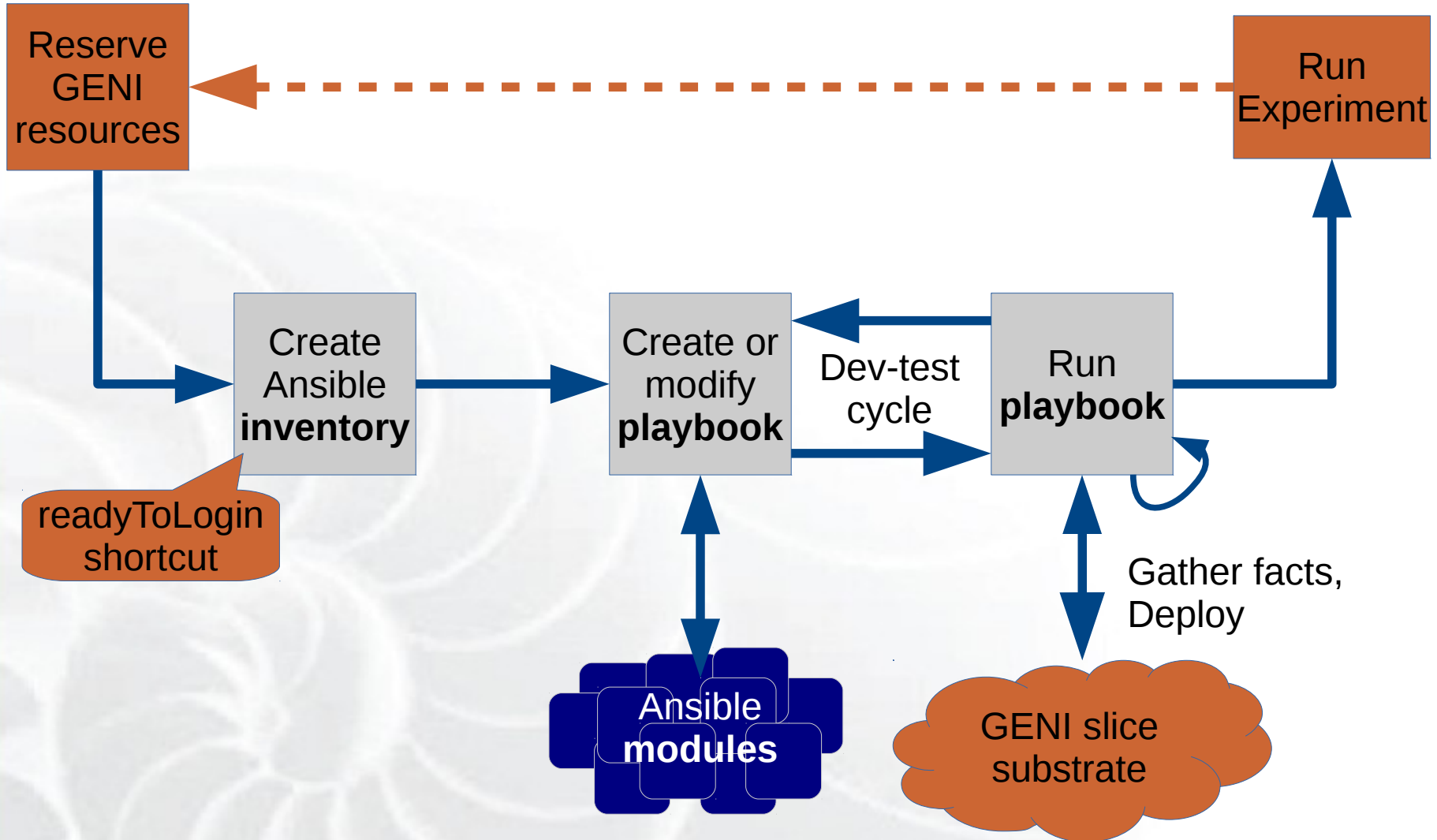
- **Inventory** file
 - Substrate ssh information, aliases, and groups
 - File format similar to **ini**
- **Ansible modules**
 - Idempotent functional building blocks
 - Many existing core and third-party modules
 - Users can write their own modules
 - **No specific language** required
 - Shortcuts exist for Python
- **Playbooks**
 - Glue modules together and map to substrate
 - Mostly written in **YAML** with module-specific details
 - Template files written in **Jinja2**

Bottom Line: Components

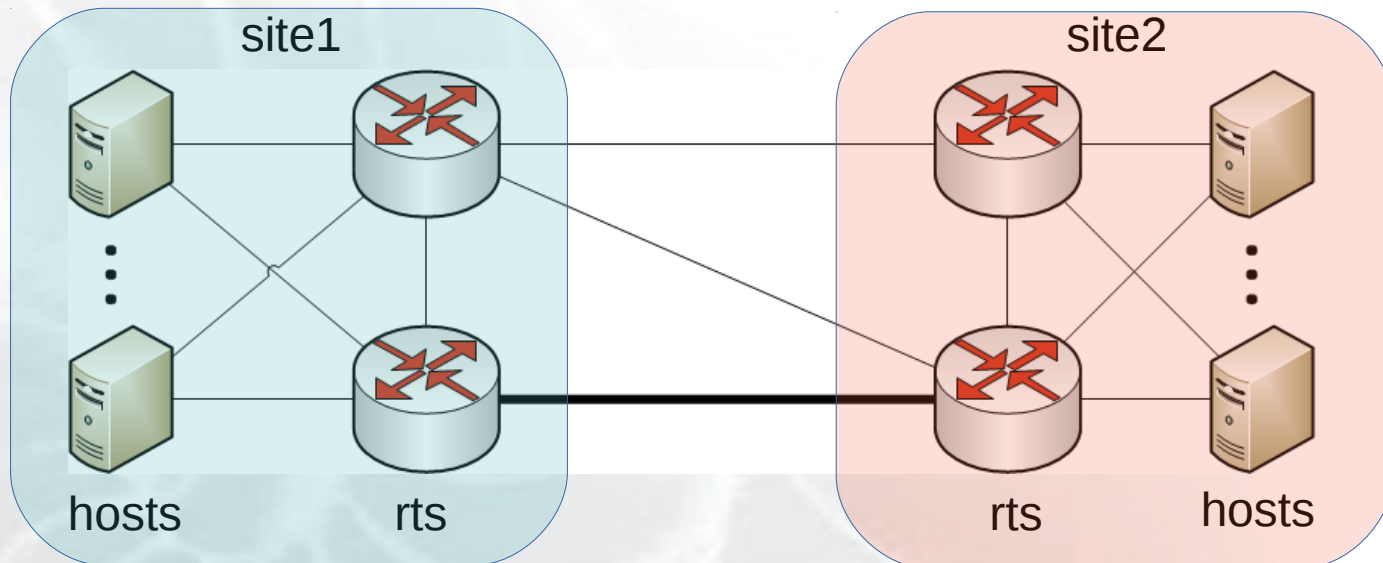
You can pick up basic Ansible by looking at examples. The languages are relatively simple and common.

If you know basic Python, you have the tools to be an Ansible power user.

GENI Slice Config Workflow



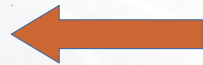
- Testing MPTCP under various conditions/topologies
 - Need to install MPTCP on all edge hosts
 - Need to configure routing on all edge hosts
- Need to install and configure code for router nodes



Example Inventory

```
site1-rt1 ansible_ssh_host=pcvm.geni.edu
site1-host1 ansible_ssh_host=pc.geni.edu ansible_ssh_port=123
...
site2-host2 ansible_ssh_host=1.2.3.4
```

```
[site1]
site1-rt1
...
site1-host2
```

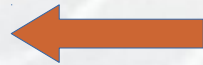


Location
groups

Define aliases for hosts.

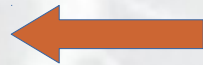
Note that ssh port can
also be specified for
nodes without public IPs.

```
[endhosts]
site1-host1
...
site2-host2
```



Function
groups

```
[instageni]
site1-rt1
...
```



Substrate
groups



Example Playbook

- **hosts:** endhosts

tasks:

First install MPTCP from the developer's repo

- **name:** Install MPTCP key

register: mptcpkey

apt_key: **url**=<http://multipath-tcp.org/mptcp.gpg.key> **state**=present

- **name:** Add MPTCP repo

when: mptcpkey|success

register: mptcrepo

apt_repository: **repo**='deb <http://multipath-tcp.org/repos/apt/debian> trusty main' **state**=present

- **name:** Install MPTCP

when: mptcrepo|success

apt: **name**=linux-mptcp **update_cache**=yes **state**=present

Then configure routing on end hosts for MPTCP

...

Example Output - Failure

```
tupty@weentop-work: ~/sandbox/mptcp
tupty@weentop-work:~/sandbox/mptcp$ ansible-playbook -i inventory -s install_mptcp.yml

PLAY [endhosts] *****

GATHERING FACTS *****
The authenticity of host '[pc2.geni.case.edu]:30778 ([192.171.20.82]:30778)' can't be established.
RSA key fingerprint is 90:76:3a:91:b4:ce:7e:e0:49:e5:49:d6:08:0d:30:62.
Are you sure you want to continue connecting (yes/no)? yes
The authenticity of host '[pc2.geni.case.edu]:30779 ([192.171.20.82]:30779)' can't be established.
RSA key fingerprint is 90:76:3a:91:b4:ce:7e:e0:49:e5:49:d6:08:0d:30:62.
Are you sure you want to continue connecting (yes/no)? yes
ok: [site1-host2]
ok: [site1-host1]

TASK: [Install MPTCP key] *****
changed: [site1-host2]
changed: [site1-host1]

TASK: [Add MPTCP repo] *****
failed: [site1-host1] => {"failed": true}
msg: Could not import python modules: pycurl. Please install python-pycurl package.
failed: [site1-host2] => {"failed": true}
msg: Could not import python modules: pycurl. Please install python-pycurl package.
FATAL: all hosts have already failed -- aborting

PLAY RECAP *****
to retry, use: --limit @/home/tupty/install_mptcp.retry

site1-host1      : ok=2    changed=1    unreachable=0    failed=1
site1-host2      : ok=2    changed=1    unreachable=0    failed=1

tupty@weentop-work:~/sandbox/mptcp$
```

Example Playbook

- **hosts:** endhosts

tasks:

```
# Install dependencies
```

```
- name: Install pycurl
```

```
  apt: name=python-pycurl state=present
```

```
# First install MPTCP from the developer's repo
```

```
- name: Install MPTCP key
```

```
  register: mptcpkey
```

```
  apt_key: url=http://multipath-tcp.org/mptcp.gpg.key state=present
```

```
- name: Add MPTCP repo
```

```
  when: mptcpkey|success
```

```
  register: mptcrepo
```

```
  apt_repository: repo='deb http://multipath-tcp.org/repos/apt/debian trusty main' state=present
```

```
- name: Install MPTCP
```

```
  when: mptcrepo|success
```

```
  apt: name=linux-mptcp update_cache=yes state=present
```

...

Example Output - Success

```
tupty@weentop-work: ~/sandbox/mptcp
tupty@weentop-work:~/sandbox/mptcp$ ansible-playbook -i inventory -s install_mptcp.yml

PLAY [endhosts] *****
GATHERING FACTS *****
ok: [site1-host1]
ok: [site1-host2]
TASK: [Install pycurl] *****
changed: [site1-host2]
changed: [site1-host1]
TASK: [Install MPTCP key] *****
ok: [site1-host1]
ok: [site1-host2]
TASK: [Add MPTCP repo] *****
changed: [site1-host1]
changed: [site1-host2]
TASK: [Install MPTCP] *****
changed: [site1-host1]
changed: [site1-host2]
PLAY RECAP *****
site1-host1      : ok=5    changed=3    unreachable=0    failed=0
site1-host2      : ok=5    changed=3    unreachable=0    failed=0

tupty@weentop-work:~/sandbox/mptcp$
```

Example Output – No-op

```
tupty@weentop-work: ~/sandbox/mptcp
tupty@weentop-work:~/sandbox/mptcp$ ansible-playbook -i inventory -s install_mptcp.yml

PLAY [endhosts] *****
GATHERING FACTS *****
ok: [site1-host1]
ok: [site1-host2]
TASK: [Install pycurl] *****
ok: [site1-host1]
ok: [site1-host2]
TASK: [Install MPTCP key] *****
ok: [site1-host1]
ok: [site1-host2]
TASK: [Add MPTCP repo] *****
ok: [site1-host1]
ok: [site1-host2]
TASK: [Install MPTCP] *****
ok: [site1-host2]
ok: [site1-host1]
PLAY RECAP *****
site1-host1      : ok=5    changed=0    unreachable=0    failed=0
site1-host2      : ok=5    changed=0    unreachable=0    failed=0

tupty@weentop-work:~/sandbox/mptcp$
```

Questions?

Map GENI Tasks to Ansible

Q: How do I scale my experiment?

A: Modify the inventory file with new aliases, new groups, and new alias-to-group mappings. Then **run your playbooks.**

Q: How do I run my experiment multiple times?

A: Delete and recreate your GENI resources using your favorite GENI tool. Then update your inventory file aliases and **run your playbooks.**

Q: How do I add new features to my nodes?

A: Create new playbook, or modify an existing playbook. Then **run your playbooks.**

Reusable Ansible Playbooks

- Can be single file for simple playbooks
- Larger playbooks broken into **roles** for modularity
- Playbooks map nodes or groups to roles
- Roles are comprised of:
 - **tasks**: list of “plays” to run for a given role
 - **handlers**: list of “plays” to run if a notification occurs
 - **templates**: files managed from a template
 - **files**: managed files
 - **vars**: variables defined for a role
 - **defaults**: default variable values with lower priority
 - **meta**: role dependencies

Adding Functionality

- What if existing modules don't have some functionality needed for an experiment?
 - Easiest path: search for third-party modules
 - Easy path:
 - Use the command modules:
command, raw, script, and shell
 - Run shell commands directly
 - Must build idempotency in
 - Ansible cannot detect if these result in changes
 - Harder path:
 - Build your own Ansible module
 - Call like any normal Ansible module
 - Must build idempotency in
 - Ansible can detect if these result in changes