Ansible 101 Fred Hsu, Distinguished Solutions Engineer fredhsu@arista.com @fredhsu Julio Perez, Solutions Engineer julioperez@arista.com @Julio_PDX

Ansible Automation Platform

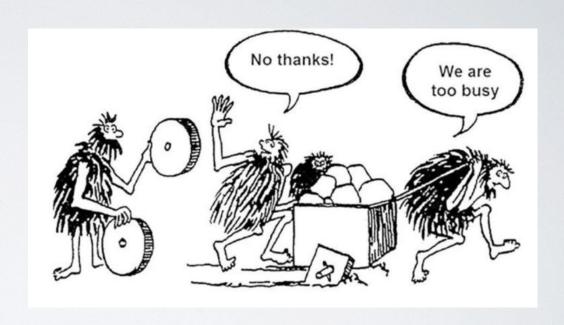
IT automation engine

- Cloud provisioning
- Configuration management
- Application deployment
- Service orchestration
- Multi-vendor



Why Ansible

- Easy to read and get started
- Decrease time to implement services
- Increase time available to focus on complex tasks
- Validate service deployments
- Reduce errors/more reliable
- No agent required on managed hosts



Quick Detour...



Git != GitHub

"Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency." - git-scm.com

"Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world." github.com/about

GitHub





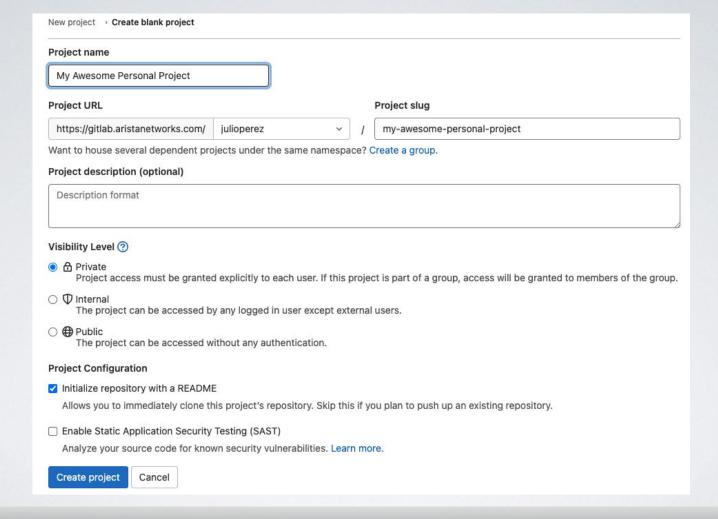


Benefits of Git

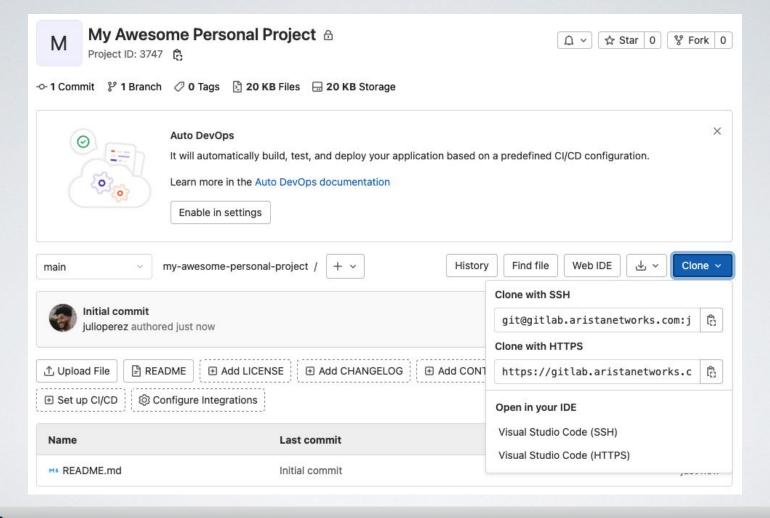
- Allow version control of the network (rollback)
- Changes are tracked by Author "Who done it?"
- Central location for desired state of the network
- Flexible

Concurrent development of network infrastructure (branches/PRs)









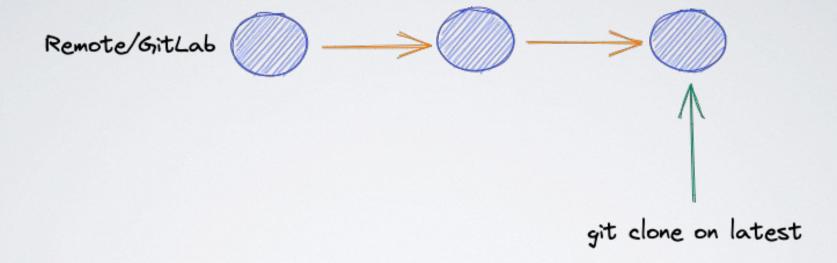


Git clone

```
> cd repos
~/repos
> git clone https://gitlab.aristanetworks.com/julioperez/my-awesome-personal-project.git
Cloning into 'my-awesome-personal-project'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
> cd my-awesome-personal-project
~/repos/my-awesome-personal-project on main
```



In the Beginning

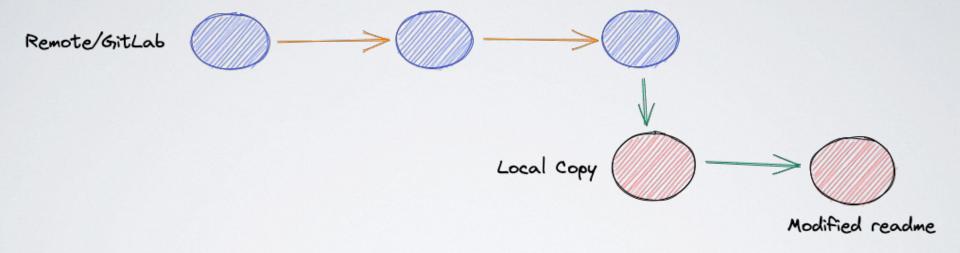


Git status

```
• • •
> git status
On branch main
Your branch is up to date with 'origin/main'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:
                README.md
no changes added to commit (use "git add" and/or "git commit -a")
~/repos/my-awesome-personal-project on main !1
                                                                        took 5s
```



Where Are We



Git add < filename > or Git add.

```
• • •
> git add .
> git status
On branch main
Your branch is up to date with 'origin/main'.
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified: README.md
~/repos/my-awesome-personal-project on main +1
                                                                        took 5s
```

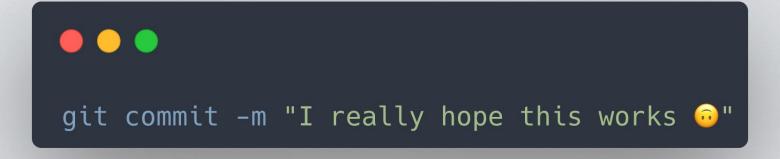


Git commit -m "Updated readme top heading"

```
> git commit -m "Updated readme top heading"
[main 8b05036] Updated readme top heading
1 file changed, 1 insertion(+), 1 deletion(-)
> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working tree clean
~/repos/my-awesome-personal-project on main >1
```



Please Don't Do This



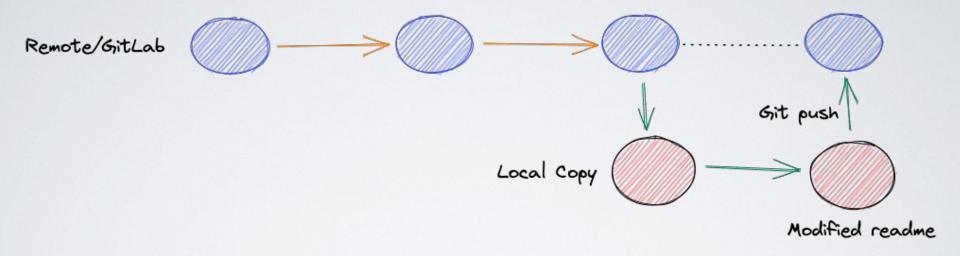


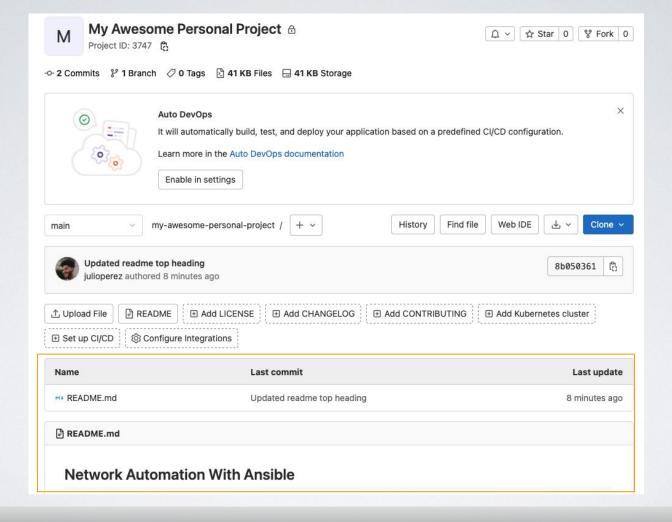
Git push

```
> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 318.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To https://gitlab.aristanetworks.com/julioperez/my-awesome-personal-project.git
   28a95c4..8b05036 main -> main
~/repos/my-awesome-personal-project on main
```



Back in Sync







YAML Ain't Markup Language (YAML)

```
. .
vendor: Arista # string
founded: 2004 # integer
is_awesome: true # boolean, many possible values
  - "DCS-7020SR-32C2"
  - DCS-7050TX3-48C8
  - DCS-7280SR3-48YC8
models_2: ["DCS-7020SR-32C2", DCS-7050TX3-48C8, DCS-7280SR3-48YC8]
   ip_address: 10.1.11.1/31
   description: Uplink to spine-11
    ip_address: 10.1.12.1/31
   description: Uplink to spine-12
```



Jinja2 Templating

Host Variables

Jinja2 Template

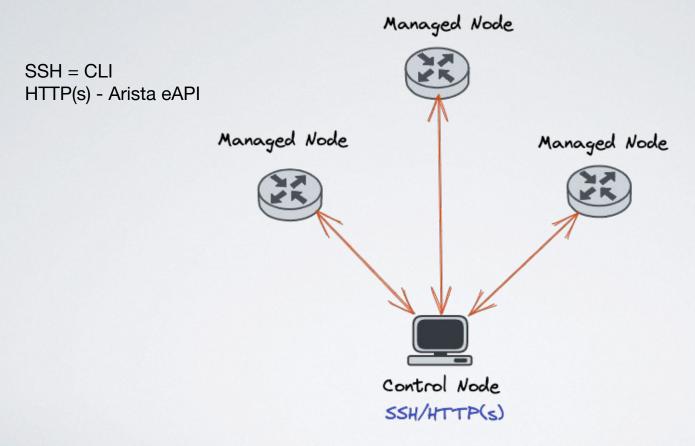
Rendered Configuration

```
hostname: leaf-101
   vrf: MGMT
 - ip: 8.8.8.8
   vrf: MGMT
   name: ELAN-10
   name: ELAN-20
```

```
hostname {{ hostname }}
{%- for server in name_servers %}
ip name-server vrf {{ server.vrf }} {{ server.ip }}
{%- endfor %}
{%- for vlan in vlans %}
vlan {{ vlan.id }}
  {{ vlan.name }}
{%- endfor %}
```

```
hostname leaf-101
!
ip name-server vrf MGMT 1.1.1.1
ip name-server vrf MGMT 8.8.8.8
!
vlan 10
ELAN-10
!
vlan 20
ELAN-20
!
```

How Ansible Works





ansible.cfg

- Optional
- Set host file location for simplicity
- Many options available see: https://tinyurl.com/jmmab6fm

```
[defaults]
host_key_checking = false
inventory = hosts.yml
deprecation_warnings = false
```



Inventories

INI

```
[DC1]
spine-11 ansible_host=192.168.100.21
leaf-101 ansible_host=192.168.100.11
leaf-102 ansible_host=192.168.100.12
[spine]
[leaf]
[DC1:vars]
ansible user = admin
ansible network os = arista.eos.eos
# Configure privilege escalation
ansible become = true
ansible_become_method = enable
# HTTPAPI configuration
ansible connection = httpapi
ansible httpapi port = 443
ansible httpapi use ssl = true
ansible httpapi validate certs = false
```

YAML

```
. . .
         ansible host: 192.168.100.11
         ansible host: 192.168.100.21
       ansible user: admin
       ansible network os: arista.eos.eos
       ansible become: true
       ansible become method: enable
       ansible_connection: httpapi
```

Variables

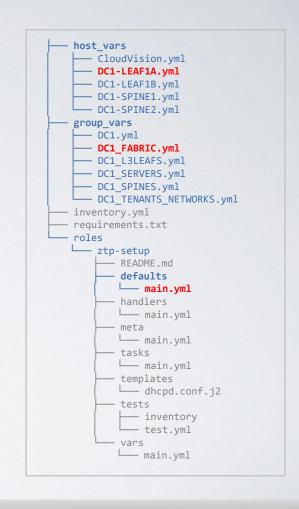
- Variable definitions are very flexible
- A variable file is a YAML(.yml/.yaml) file
- Variable files can be stored in many places
- Variables can be defined in Playbooks, inventory, modules
- Accessible from templates and Playbooks





Directory Structure for Variables

- Variable files can be in many places
 - >> host_vars folder contains the variables per host
 - >> group_vars folder contains the variables per inventory group (spines, site1, all, ...)
 - defaults subfolders of roles can be used to store variables for roles.
- One or multiple variable files per
 - ≫ Host
 - ≫ Group
- Very flexible (least specific to more specific)



Variable Precedence

- A lot of options
- Keep it simple
- https://tinyurl.com/3cbjv96t

- 1. command line values (for example, -u my_user, these are not variables)
- 2. role defaults (defined in role/defaults/main.yml) 1
- 3. inventory file or script group vars ²
- 4. inventory group_vars/all ³
- 5. playbook group_vars/all 3
- 6. inventory group_vars/* 3
- 7. playbook group_vars/* 3
- 8. inventory file or script host vars ²
- 9. inventory host_vars/* 3
- 10. playbook host_vars/* 3
- 11. host facts / cached set_facts 4
- 12. play vars
- 13. play vars_prompt
- 14. play vars_files
- 15. role vars (defined in role/vars/main.yml)
- 16. block vars (only for tasks in block)
- 17. task vars (only for the task)
- 18. include_vars
- 19. set_facts / registered vars
- 20. role (and include_role) params
- 21. include params
- 22. extra vars (for example, -e "user=my_user")(always win precedence)



Variable Exercise

bgp_as: 65000

```
TASK [View response]
bgp_as: 65100
                                                           ok: [spine-11] => {
                                                              "msg": 65100
                                                           ok: [leaf-102] => {
                                                              "msg": 65000
                             bgp_as: 65001
ok: [leaf-101] => {
                                                              "msg": 65001
```



Ansible Playbooks

- Ansible scripts are called Playbooks
- Playbooks are written in YAML
 - A Playbook is a YAML list of plays
 - A Play has a list of tasks
- A play maps a selection of hosts to a list of tasks
- Playbook runs top to bottom
- Playbook is executed with ansible-playbook command
- Option -i to specifies inventory host file (default=/etc/ansible/hosts)

```
- name: This is a hello world example
  hosts: DC1
    - name: Check connectivity on DC1
      register: ping response
    - name: View response
        msg: "{{ ping_response }}"
 name: This is a hello world example 2
  hosts: leaf-101
    - name: Check connectivity on leaf-101
      register: ping_response
    - name: View response
        msg: "{{ ping_response }}"
```

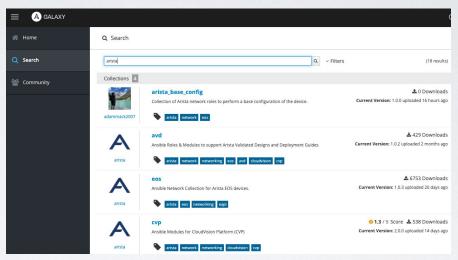
Ansible Roles

- Roles are called inside a Playbook
- Very useful to reuse the same automation content across Playbook
- A set of tasks
- An optional set of templates
- Can define and consume some variables
- •You can use the "ansible-galaxy init" command to create the skeleton of a role

```
roles/eos_config_deploy_cvp
  - README.md
    inventory-converter.py
   handlers
      absent.yml
      - present.yml
    cvp-devices.j2
      inventory
   vars
```

Ansible Collections

- Collections are a distribution format for Ansible content that can include
 - Playbooks
 - Roles
 - Modules
 - Plugins



- Can install and use collections through <u>Ansible Galaxy</u>
 - ansible-galaxy collection install my_namespace.my_collection



Ansible & Arista

- Arista EOS Modules
- Arista AVD Collection (Arista Design Patterns)

 Arista CVP Collection (management, telemetry, workflows, change control, etc...)

Arista NetDevOps Examples
(github.com/aristanetworks/netdevops-examples/tree/master/demo

Ansible Documentation



Demo!

