ADAM STEVKO

ANSIBLE FOR NETWORK ENGINEERS

HOLA, I'M ADAM

- Slovakia (that small country east of Austria)
- System engineering background
- Ansible user
- Ansible contributor
- Github: @xen0l / Twitter: @xen0l

ORGANIZATION OF THIS WORKSHOP

- Survey
- Introduction
- Workshop 1: Inventory & fact gathering
- Workshop 2: Orchestration
- Workshop 3: Configuration Management
- Have a question? Raise your hand!

LITTLE SURVEY

WHAT IS ANSIBLE?

ANSIBLE

- Agentless provisioning
- Configuration management
- Application deployment
- **Network automation**

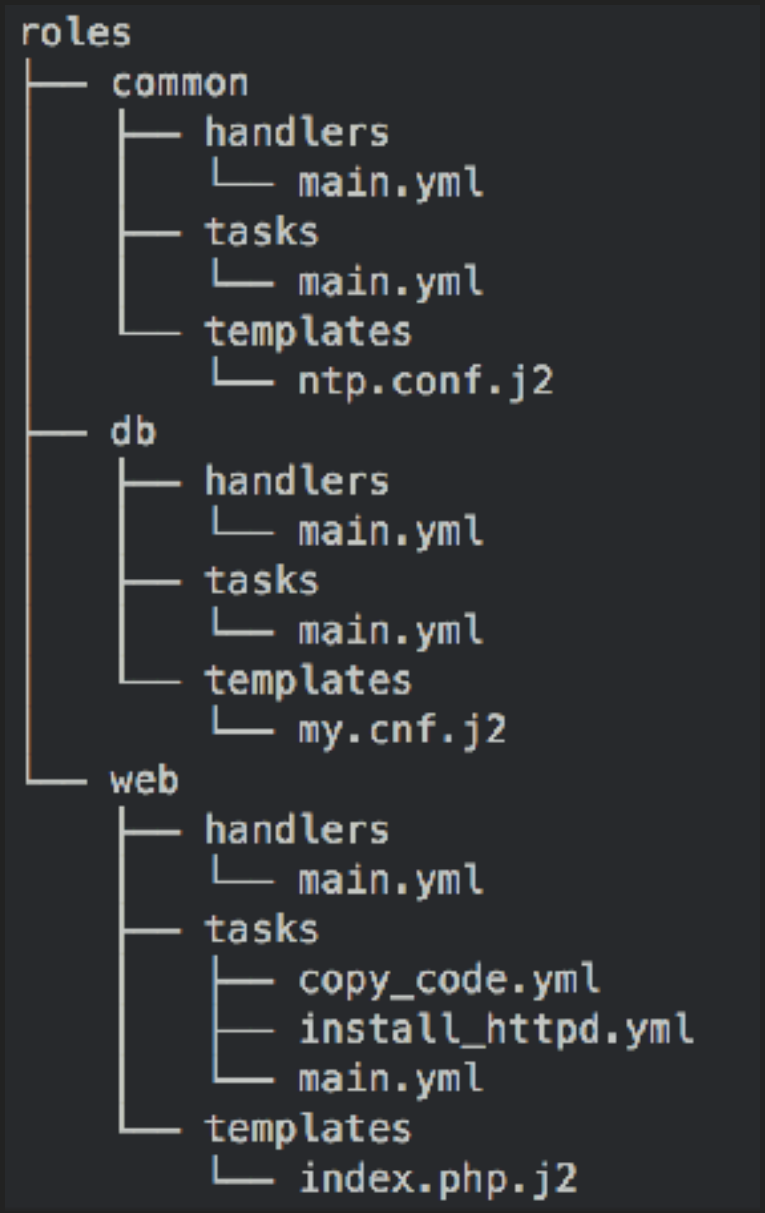
CONCEPTS

- Simplicity
  - Python
  - Jinja2
  - YAML
  - SSH
- Idempotence

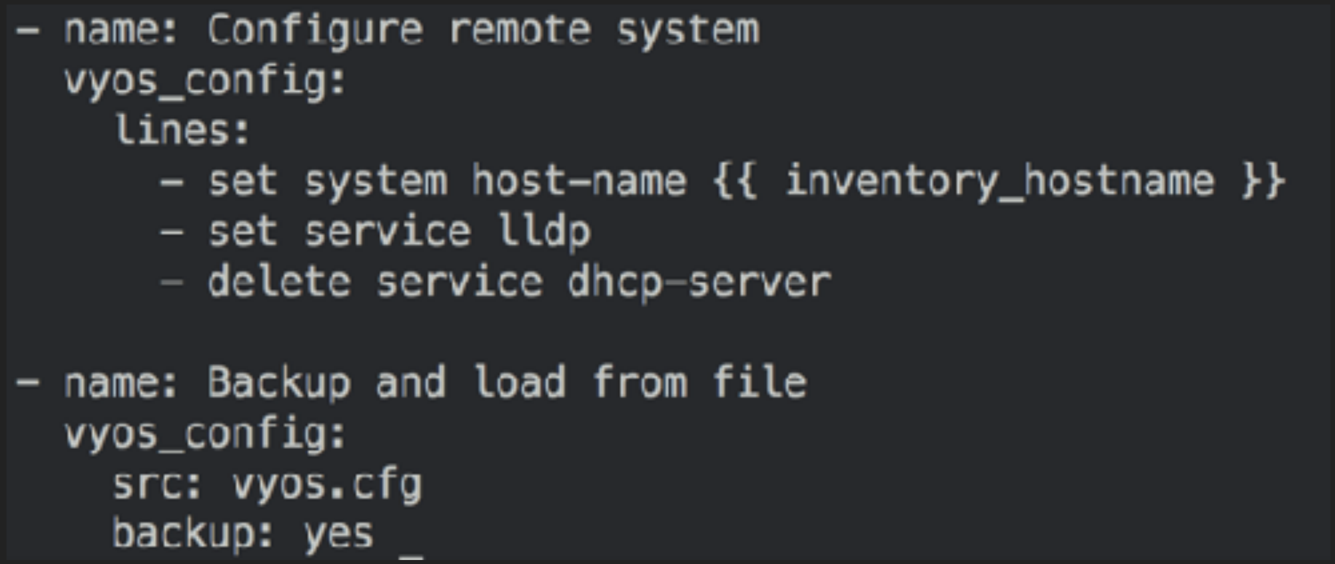
PLAYBOOKS

- Tasks
  - Specifies action
- Roles
  - Groups actions
- Plays
  - Maps actions to hosts
- Playbooks
  - Groups plays

ROLE AND TASK EXAMPLE



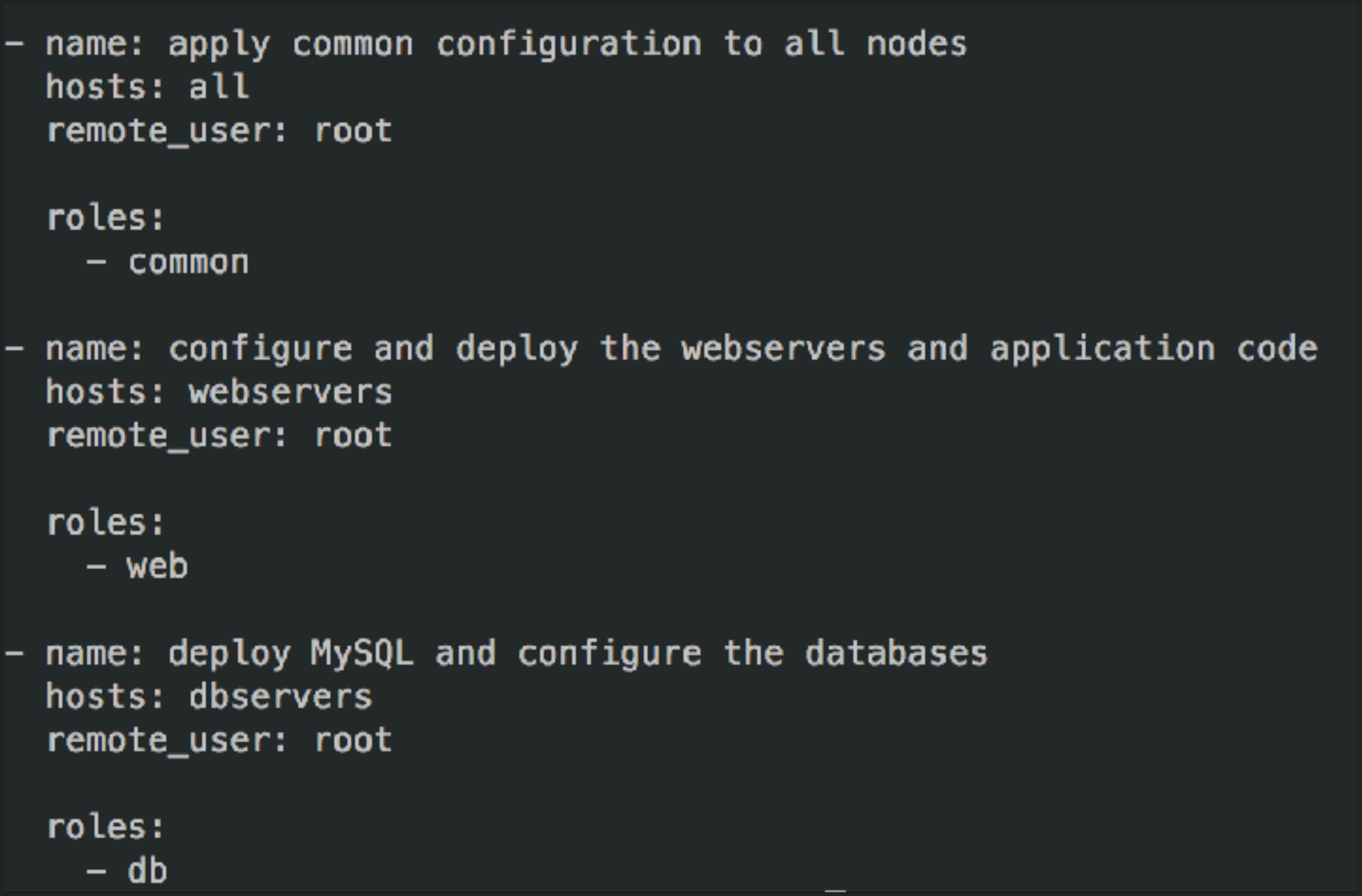
```
roles
├── common
│   ├── handlers
│   │   └── main.yml
│   ├── tasks
│   │   └── main.yml
│   └── templates
│       └── ntp.conf.j2
├── db
│   ├── handlers
│   │   └── main.yml
│   ├── tasks
│   │   └── main.yml
│   └── templates
│       └── my.cnf.j2
└── web
    ├── handlers
    │   └── main.yml
    ├── tasks
    │   ├── copy_code.yml
    │   ├── install_httpd.yml
    │   └── main.yml
    └── templates
        └── index.php.j2
```



```
- name: Configure remote system
  vyos_config:
    lines:
      - set system host-name {{ inventory_hostname }}
      - set service lldp
      - delete service dhcp-server

- name: Backup and load from file
  vyos_config:
    src: vyos.cfg
    backup: yes _
```

PLAYBOOK EXAMPLE



```
- name: apply common configuration to all nodes
  hosts: all
  remote_user: root

  roles:
    - common

- name: configure and deploy the webservers and application code
  hosts: webservers
  remote_user: root

  roles:
    - web

- name: deploy MySQL and configure the databases
  hosts: dbservers
  remote_user: root

  roles:
    - db
```

WORKSHOP

REQUIREMENTS

- VAGRANT 2.0
- VIRTUALBOX
- ANSIBLE
- https://github.com/bcn-sdn-meetup/ansible-workshop

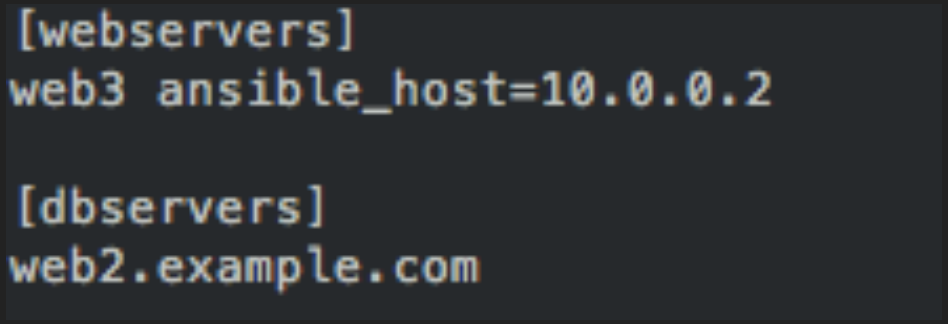
WORKSHOP 1:
INVENTORY & FACT GATHERING

INVENTORY

- List of hosts
  - Stores information how to reach hosts
- Hosts can be members of groups
- Static (file) vs dynamic (e.g. get data from a monitoring system)

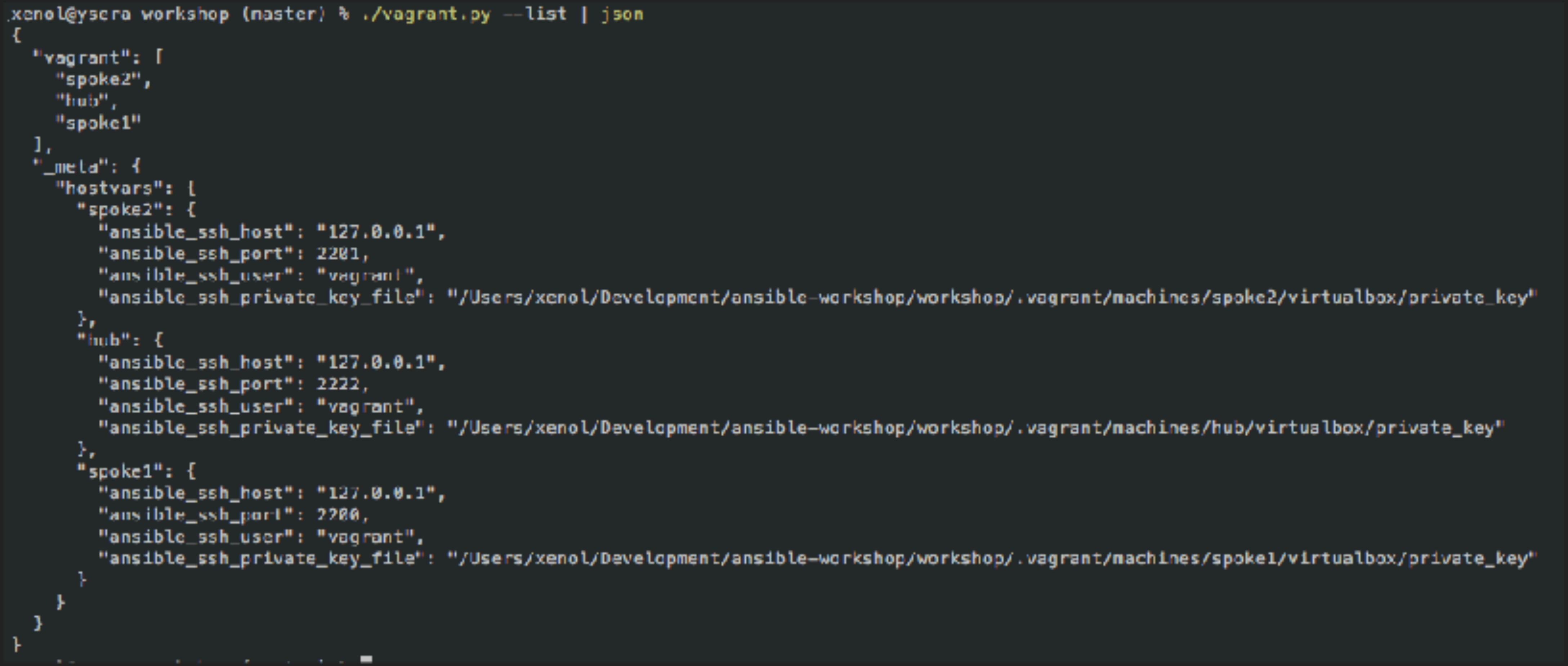
INVENTORY EXAMPLE

- Static
- Dynamic



```
[webservers]
web3 ansible_host=10.0.0.2

[dbservers]
web2.example.com
```

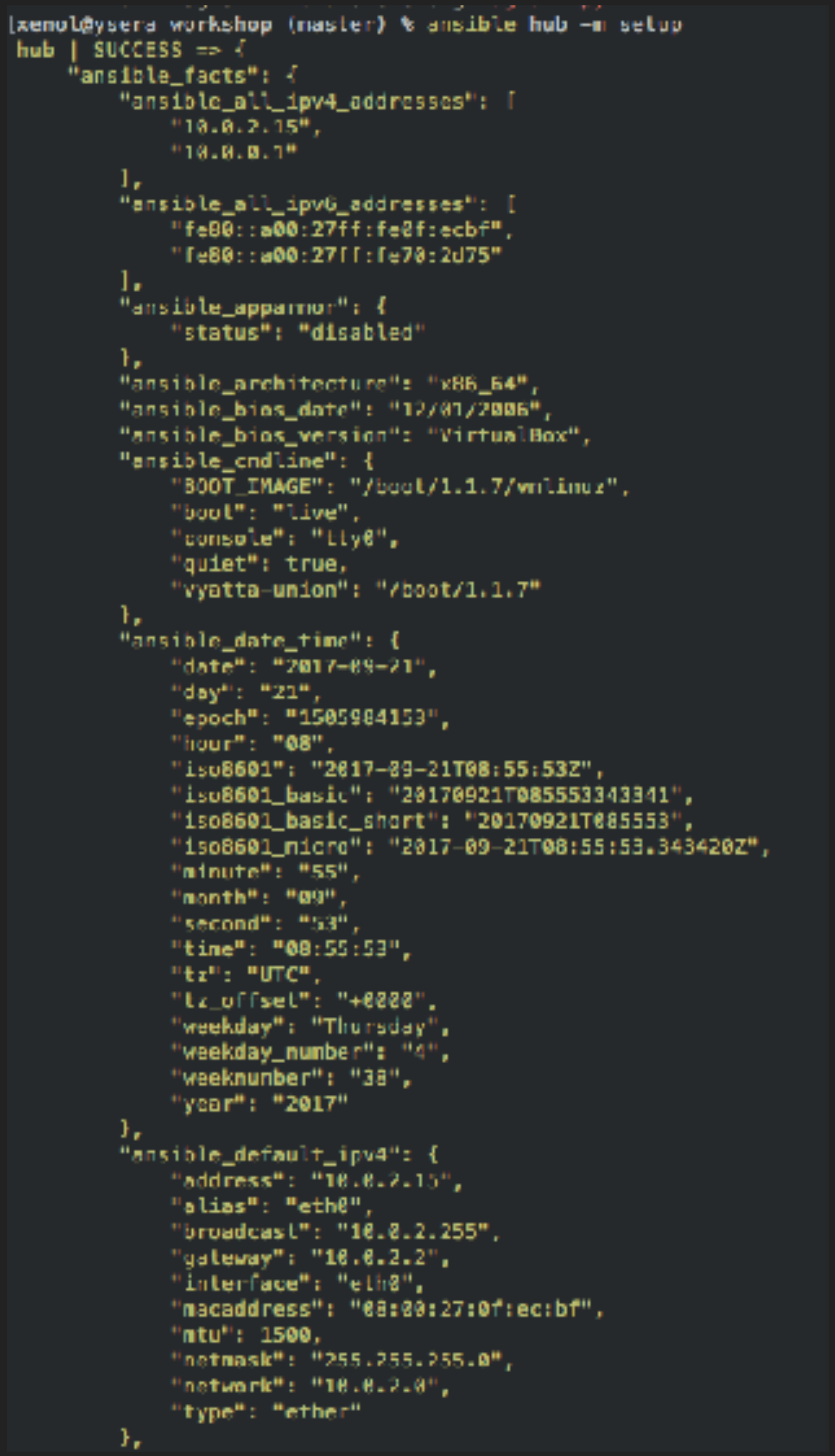


```
xenol@yscra workshop (master) % ./vagrant.py --list | json
{
  "vagrant": [
    "spoke2",
    "hub",
    "spoke1"
  ],
  "_meta": {
    "hostvars": {
      "spoke2": {
        "ansible_ssh_host": "127.0.0.1",
        "ansible_ssh_port": 2201,
        "ansible_ssh_user": "vagrant",
        "ansible_ssh_private_key_file": "/Users/xenol/Development/ansible-workshop/workshop/.vagrant/machines/spoke2/virtualbox/private_key"
      },
      "hub": {
        "ansible_ssh_host": "127.0.0.1",
        "ansible_ssh_port": 2222,
        "ansible_ssh_user": "vagrant",
        "ansible_ssh_private_key_file": "/Users/xenol/Development/ansible-workshop/workshop/.vagrant/machines/hub/virtualbox/private_key"
      },
      "spoke1": {
        "ansible_ssh_host": "127.0.0.1",
        "ansible_ssh_port": 2200,
        "ansible_ssh_user": "vagrant",
        "ansible_ssh_private_key_file": "/Users/xenol/Development/ansible-workshop/workshop/.vagrant/machines/spoke1/virtualbox/private_key"
      }
    }
  }
}
```

FACTS

- Information about hosts (e.g. network, disks, vendor, machine type…)
- **setup** module
- Local facts
- gather_facts: true

FACTS EXAMPLE



```
[xenul@ysera workshop (master) % ansible hub -m setup
 hub | SUCCESS => {
     "ansible_facts": {
         "ansible_all_ipv4_addresses": [
             "10.0.2.15",
             "10.0.0.7"
         ],
         "ansible_all_ipv6_addresses": [
             "fe80::a00:27ff:fe0f:ecbf",
             "fe80::a00:27ff:fe70:2d75"
         ],
         "ansible_apparmor": {
             "status": "disabled"
         },
         "ansible_architecture": "x86_64",
         "ansible_bios_date": "12/01/2006",
         "ansible_bios_version": "VirtualBox",
         "ansible_cmdline": {
             "BOOT_IMAGE": "/boot/1.1.7/vmlinuz",
             "boot": "live",
             "console": "tty0",
             "quiet": true,
             "vyatta-union": "/boot/1.1.7"
         },
         "ansible_date_time": {
             "date": "2017-09-21",
             "day": "21",
             "epoch": "1505984153",
             "hour": "08",
             "iso8601": "2017-09-21T08:55:53Z",
             "iso8601_basic": "20170921T085553343341",
             "iso8601_basic_short": "20170921T085553",
             "iso8601_micro": "2017-09-21T08:55:53.343420Z",
             "minute": "55",
             "month": "09",
             "second": "53",
             "time": "08:55:53",
             "tz": "UTC",
             "tz_offset": "+0000",
             "weekday": "Thursday",
             "weekday_number": "4",
             "weeknumber": "38",
             "year": "2017"
         },
         "ansible_default_ipv4": {
             "address": "10.0.2.15",
             "alias": "eth0",
             "broadcast": "10.0.2.255",
             "gateway": "10.0.2.2",
             "interface": "eth0",
             "macaddress": "08:00:27:0f:ec:bf",
             "mtu": 1500,
             "netmask": "255.255.255.0",
             "network": "10.0.2.0",
             "type": "ether"
         },
```

ASSIGNMENT

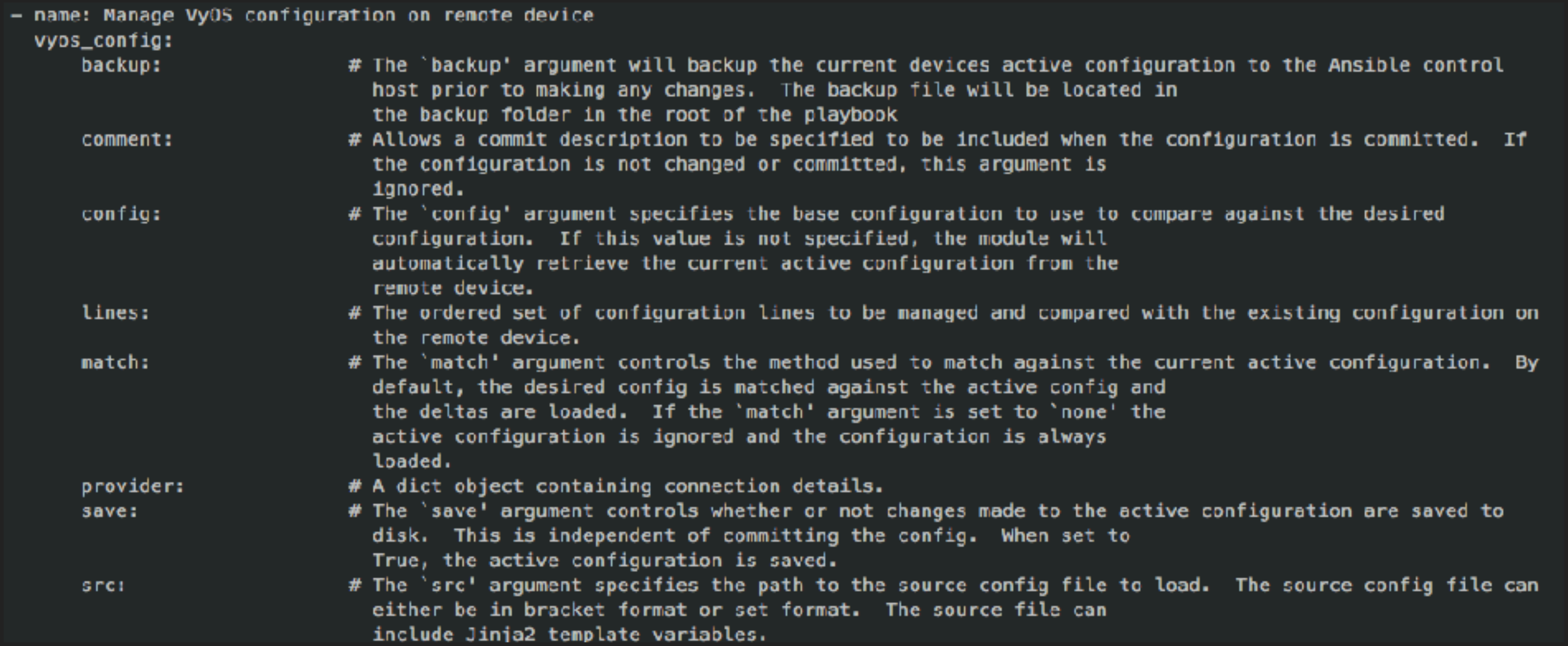
- TASK: You were provided with an sample infrastructure, try to figure out all bound network IP addresses.
- BONUS: Write a playbook to print out IP addresses. Skeleton was provided in the repository.

WORKSHOP 2: ORCHESTRATION

MODULES

- Modules do actual work
  - Modules take arguments
  - Modules are copied to the remote host and executed
- Ansible ships almost 1400 modules
- Network modules: https://docs.ansible.com/ansible/latest/list_of_network_modules.html

MODULE EXAMPLE



```
- name: Manage VyOS configuration on remote device
  vyos_config:
      backup:                # The `backup' argument will backup the current devices active configuration to the Ansible control
                               host prior to making any changes.  The backup file will be located in
                               the backup folder in the root of the playbook
      comment:               # Allows a commit description to be specified to be included when the configuration is committed.  If
                               the configuration is not changed or committed, this argument is
                               ignored.
      config:                # The `config' argument specifies the base configuration to use to compare against the desired
                               configuration.  If this value is not specified, the module will
                               automatically retrieve the current active configuration from the
                               remote device.
      lines:                 # The ordered set of configuration lines to be managed and compared with the existing configuration on
                               the remote device.
      match:                 # The `match' argument controls the method used to match against the current active configuration.  By
                               default, the desired config is matched against the active config and
                               the deltas are loaded.  If the `match' argument is set to `none' the
                               active configuration is ignored and the configuration is always
                               loaded.
      provider:              # A dict object containing connection details.
      save:                  # The `save' argument controls whether or not changes made to the active configuration are saved to
                               disk.  This is independent of committing the config.  When set to
                               True, the active configuration is saved.
      src:                   # The `src' argument specifies the path to the source config file to load.  The source config file can
                               either be in bracket format or set format.  The source file can
                               include Jinja2 template variables.
```

REGISTER

- **register: variable**
  - keyword, which will register output in variable
  - useful for getting output from commands

ASSIGNMENT

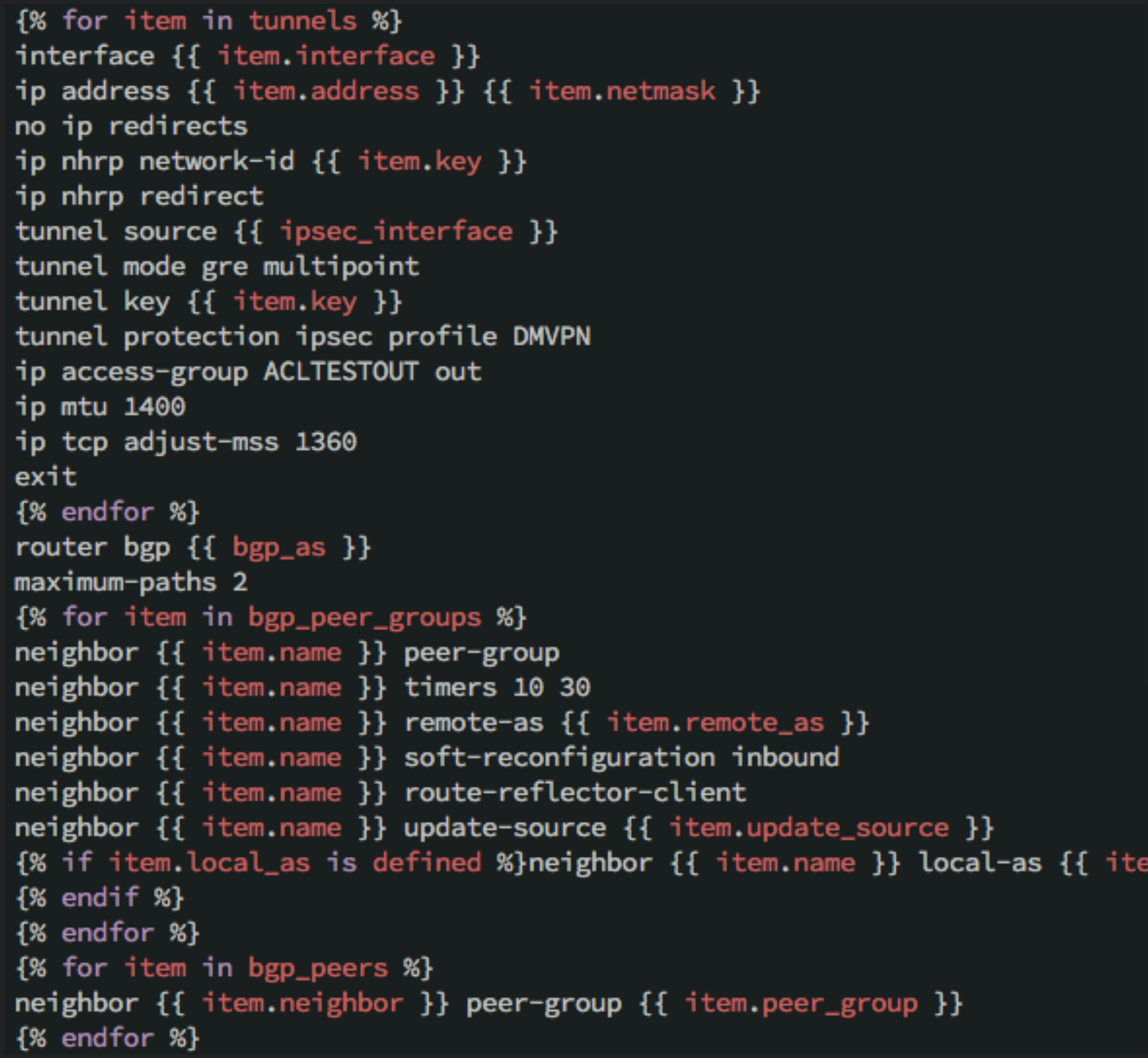
- TASK: You were provided with an sample infrastructure, try to add random static route.
- BONUS: Write a playbook to configure different option.
- BONUS 2: Proof that we don’t need a task to check the configuration from the previous assignment.

WORKSHOP 3:
CONFIGURATION MANAGEMENT

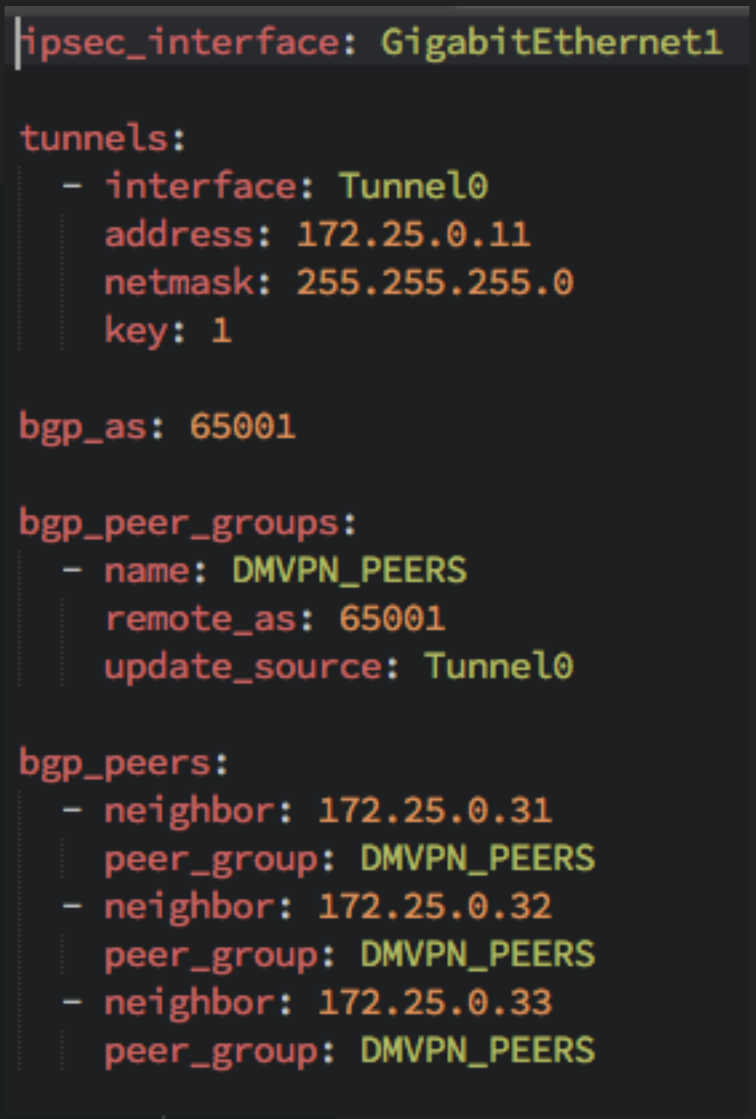
JINJA2

- Templating language
  - Used by Python ecosystem
  - Uses Ansible variables

JINJA2 EXAMPLE



```
{% for item in tunnels %}
interface {{ item.interface }}
ip address {{ item.address }} {{ item.netmask }}
no ip redirects
ip nhrp network-id {{ item.key }}
ip nhrp redirect
tunnel source {{ ipsec_interface }}
tunnel mode gre multipoint
tunnel key {{ item.key }}
tunnel protection ipsec profile DMVPN
ip access-group ACLTESTOUT out
ip mtu 1400
ip tcp adjust-mss 1360
exit
{% endfor %}
router bgp {{ bgp_as }}
maximum-paths 2
{% for item in bgp_peer_groups %}
neighbor {{ item.name }} peer-group
neighbor {{ item.name }} timers 10 30
neighbor {{ item.name }} remote-as {{ item.remote_as }}
neighbor {{ item.name }} soft-reconfiguration inbound
neighbor {{ item.name }} route-reflector-client
neighbor {{ item.name }} update-source {{ item.update_source }}
{% if item.local_as is defined %}neighbor {{ item.name }} local-as {{ item.local_as }}
{% endif %}
{% endfor %}
{% for item in bgp_peers %}
neighbor {{ item.neighbor }} peer-group {{ item.peer_group }}
{% endfor %}
```



```yaml
ipsec_interface: GigabitEthernet1

tunnels:
  - interface: Tunnel0
    address: 172.25.0.11
    netmask: 255.255.255.0
    key: 1

bgp_as: 65001

bgp_peer_groups:
  - name: DMVPN_PEERS
    remote_as: 65001
    update_source: Tunnel0

bgp_peers:
  - neighbor: 172.25.0.31
    peer_group: DMVPN_PEERS
  - neighbor: 172.25.0.32
    peer_group: DMVPN_PEERS
  - neighbor: 172.25.0.33
    peer_group: DMVPN_PEERS
```

DEMO

ASSIGNMENT

- TASK: Write a template file and configure a random static route
- BONUS: Modify a template file to configure more random static routes

RECAP & KEY TAKEAWAYS

- Automation is getting more important
- Ansible is simple, flexible, easy to get started with
- You do not have to use _ALL_ the features
- Have fun

Q&A

THANKS FOR YOUR ATTENTION!