Internal training
INTRODUCTION TO ANSIBLE

Author / manager: Lev Goncharov / Ilya Semerhanov

Lection #2 – First playbook

Course Content

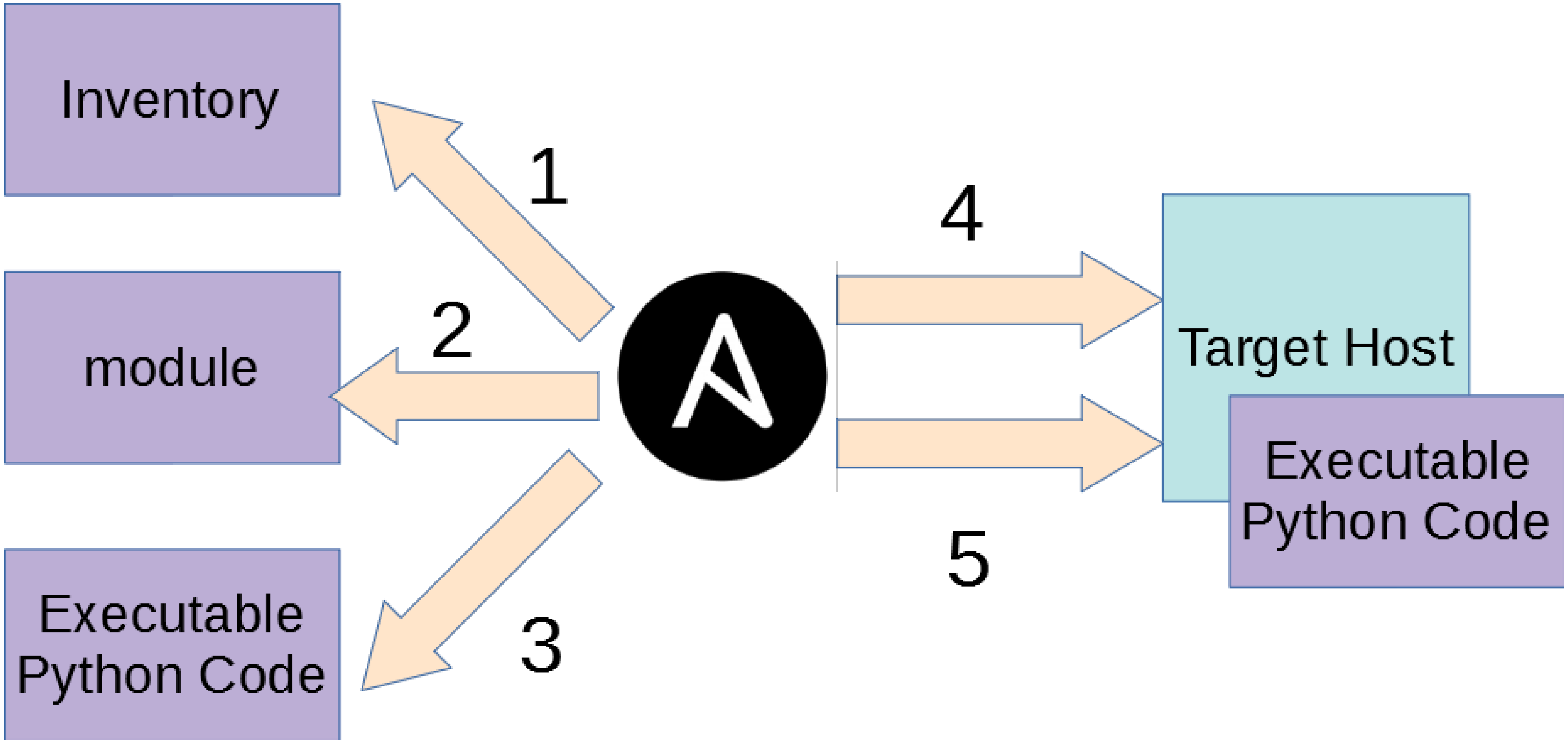
1. Lection #1: Introduction
   1. Configuration management
   2. Ansible. How it works?
   3. Vagrant. Training env
2. Lection #2: First playbook
   1. First playbook
   2. Ansible modules
   3. Facts & variables
3. Lection #3: Base features
   1. Jinja2 templating
   2. Conditions
   3. Loops
4. Lection #4: Plugins & modules
   1. Plugins
   2. Modules
   3. Handlers
5. Lection #5: Best practices
   1. Roles
   2. Working with inventory
   3. Repository structure
6. Lection #6: Usecases
   1. Golden images
   2. Management infrastructure
   3. CI/CD integration

Course Content

1. Lection #1: Introduction
   1. Configuration management
   2. Ansible. How it works?
   3. Vagrant. Training env
2. Lection #2: First playbook
   1. First playbook
   2. Ansible modules
   3. Facts & variables
3. Lection #3: Base features
   1. Jinja2 templating
   2. Conditions
   3. Loops
4. Lection #4: Plugins & modules
   1. Plugins
   2. Modules
   3. Handlers
5. Lection #5: Best practices
   1. Roles
   2. Working with inventory
   3. Repository structure
6. Lection #6: Usecases
   1. Golden images
   2. Management infrastructure
   3. CI/CD integration

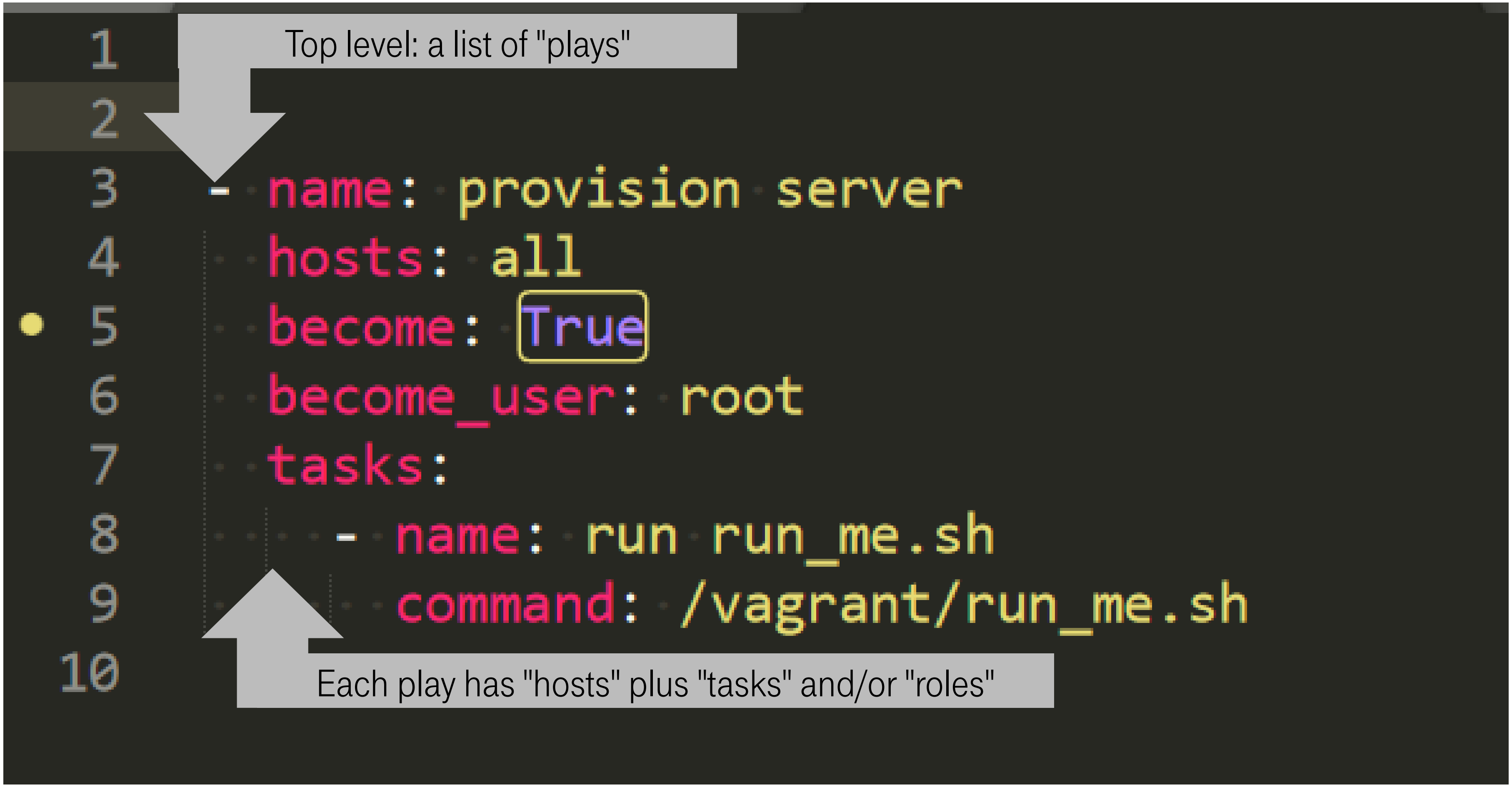
Lection #2. First playbook

First Playbook



Lection #2. First playbook

YML. provision_me.yml



```

- name: provision server
  hosts: all
  become: True
  become_user: root
  tasks:
    - name: run run_me.sh
      command: /vagrant/run_me.sh

```

Top level: a list of "plays"

Each play has "hosts" plus "tasks" and/or "roles"

Lection #2. First playbook

YML

1. A way of storing structured data as text
2. Conceptually similar to JSON
   1. String and numeric values
   2. Lists: ordered sequences
   3. Hashes: unordered groups of key-value pairs
3. String values don't normally need quotes
4. Lists and hashes can be nested
5. Indentation used to defne nesting

Lection #2. First playbook

YML. list

- Single line form

```
[birth, taxes, death]
```

- Multi-line form

Space after dash required

```
- birth
- taxes
- death
```

Lection #2. First playbook
YML. hash

- Single line form

```
{item: shirt, colour: red, size: 42
```

- Multi-line form

```
item: shirt
colour: red
size: 42
description: |
  this is a very long multi-line
  text field which is all one value
```

Lection #2. First playbook

YML. List of hashes

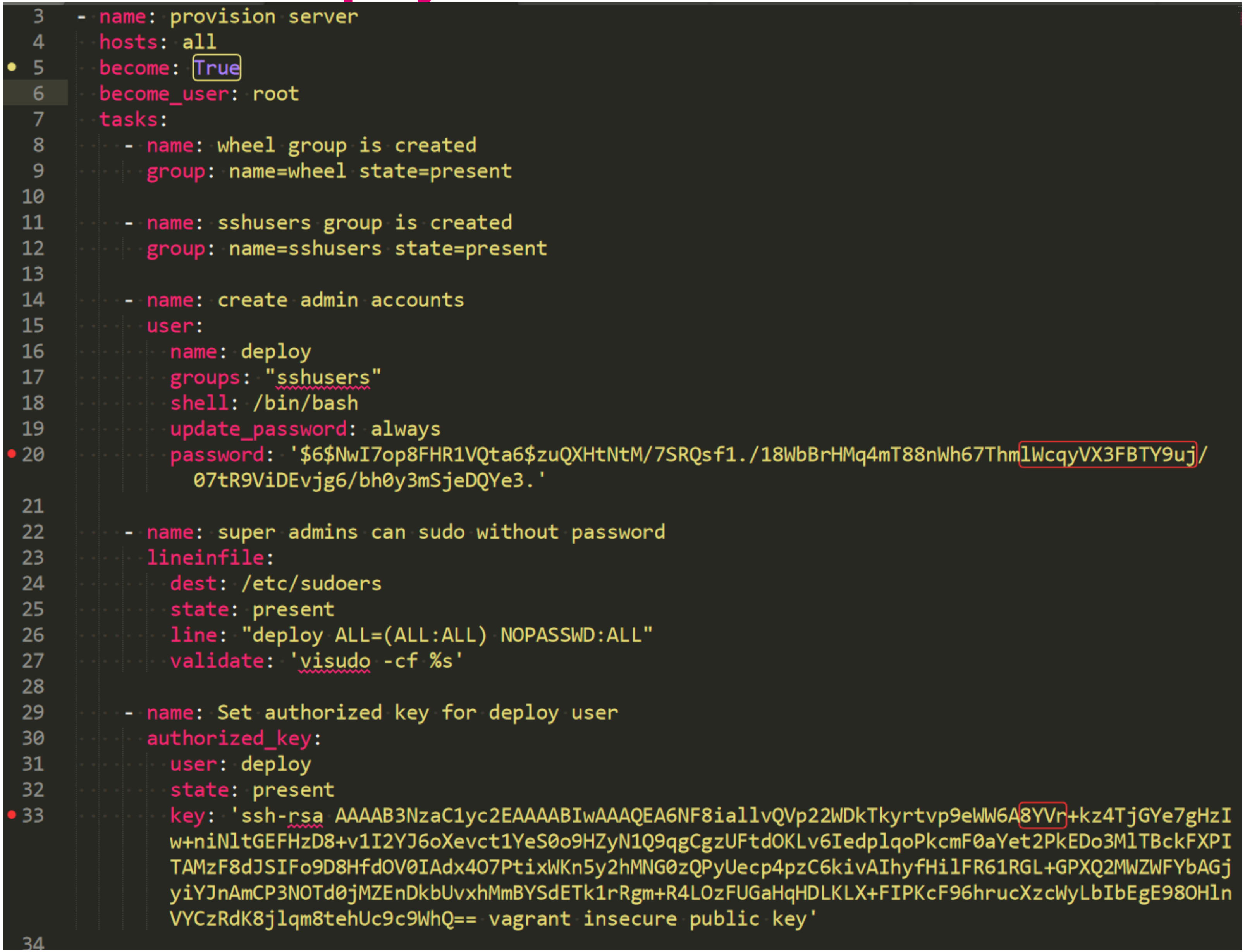
- Compact

```
- {item: shirt, colour: red, size: 42}
- {item: shirt, colour: blue, size: 44}
```

- Multi-line

```
- item: shirt
  colour: red
  size: 42
- item: shirt
  colour: blue
  size: 44
```

Lection #2. First playbook



```
- name: provision server
  hosts: all
  become: True
  become_user: root
  tasks:
    - name: wheel group is created
      group: name=wheel state=present

    - name: sshusers group is created
      group: name=sshusers state=present

    - name: create admin accounts
      user:
        name: deploy
        groups: "sshusers"
        shell: /bin/bash
        update_password: always
        password: '$6$NwI7op8FHR1VQta6$zuQXHtNtM/7SRQsf1./18WbBrHMq4mT88nWh67ThmlWcqyVX3FBTY9uj/
          07tR9ViDEvjg6/bh0y3mSjeDQYe3.'

    - name: super admins can sudo without password
      lineinfile:
        dest: /etc/sudoers
        state: present
        line: "deploy ALL=(ALL:ALL) NOPASSWD:ALL"
        validate: 'visudo -cf %s'

    - name: Set authorized key for deploy user
      authorized_key:
        user: deploy
        state: present
        key: 'ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NF8iallvQVp22WDkTkyrtvp9eWW6A8YVr+kz4TjGYe7gHzI
        w+niNltGEFHzD8+v1I2YJ6oXevct1YeS0o9HZyN1Q9qgCgzUFtdOKLv6IedplqoPkcmF0aYet2PkEDo3MlTBckFXPI
        TAMzF8dJSIFo9D8HfdOV0IAdx4O7PtixWKn5y2hMNG0zQPyUecp4pzC6kivAIhyfHilFR61RGL+GPXQ2MWZWFYbAGj
        yiYJnAmCP3NOTd0jMZEnDkbUvxhMmBYSdETk1rRgm+R4LOzFUGaHqHDLKLX+FIPKcF96hrucXzcWyLbIbEgE98OHln
        VYCzRdK8jlqm8tehUc9c9WhQ== vagrant insecure public key'
```

Lection #2. First playbook

Playbook

- Tasks are modules called with specifc arguments
- Handlers are triggered when something changes

  e.g. restart daemon when a confg fle is changed
- Roles are re-usable bundles of tasks, handlers and templates
- All defned using YAML

Lection #2. First playbook

Playbook

- Each play contains a list of tasks.
- Tasks are executed in order one by one
- One task at a time, against all machines matched by the host pattern, before moving on to the next task.

Course Content

1. Lection #1: Introduction
   1. Configuration management
   2. Ansible. How it works?
   3. Vagrant. Training env
2. Lection #2: First playbook
   1. First playbook
   2. Ansible modules
   3. Facts & variables
3. Lection #3: Base features
   1. Jinja2 templating
   2. Conditions
   3. Loops
4. Lection #4: Plugins & modules
   1. Plugins
   2. Modules
   3. Handlers
5. Lection #5: Best practices
   1. Roles
   2. Working with inventory
   3. Repository structure
6. Lection #6: Usecases
   1. Golden images
   2. Management infrastructure
   3. CI/CD integration

Lection #2. First playbook

Modules

1. Ansible “modules” are small pieces of code which perform one function
2. Most are “idempotent”: means that they only do something when a change is required
3. Many modules supplied as standard

http://www.ansibleworks.com/docs/modules.html

Lection #2. First playbook

Modules

All modules

Cloud modules

Clustering modules

Commands modules

Crypto modules

Database modules

Files modules

Identity modules

Inventory modules

Messaging modules

Monitoring modules

Net Tools modules

Network modules

Notification modules

Packaging modules

Remote Management modules

Source Control modules

Storage modules

System modules

Utilities modules

Web Infrastructure modules

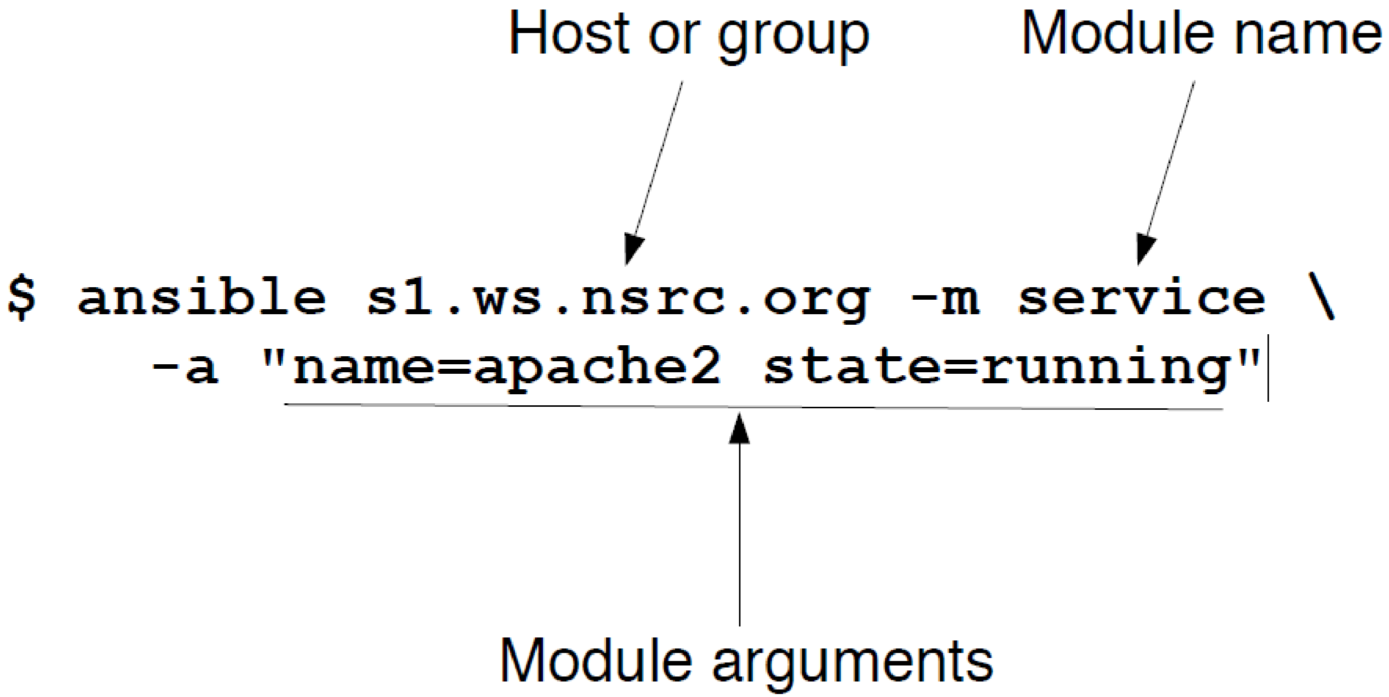
Windows modules

https://docs.ansible.com/ansible/latest/modules/modules_by_category.html

Lection #2. First playbook

Modules

Invoking modules from shell



Host or group

Module name

```
$ ansible s1.ws.nsrc.org -m service \
    -a "name=apache2 state=running"
```

Module arguments

Course Content

1. Lection #1: Introduction
   1. Configuration management
   2. Ansible. How it works?
   3. Vagrant. Training env

2. Lection #2: First playbook
   1. First playbook
   2. Ansible modules
   3. Facts & variables

3. Lection #3: Base features
   1. Jinja2 templating
   2. Conditions
   3. Loops

4. Lection #4: Plugins & modules
   1. Plugins
   2. Modules
   3. Handlers

5. Lection #5: Best practices
   1. Roles
   2. Working with inventory
   3. Repository structure

6. Lection #6: Usecases
   1. Golden images
   2. Management infrastructure
   3. CI/CD integration

Lection #2. First playbook
Facts

1. Facts are variables containing information collected automatically about the target host
2. Things like what OS is installed, what interfaces it has, what disk drives it has
3. Can be used to adapt roles automatically to the target system
4. Gathered every time Ansible connects to a host (unless playbook has "gather_facts: no")

Lection #2. First playbook
Facts

1. Information discovered from systems
2. Ansible provides many facts about the system, automatically
3. Provide by the setup module

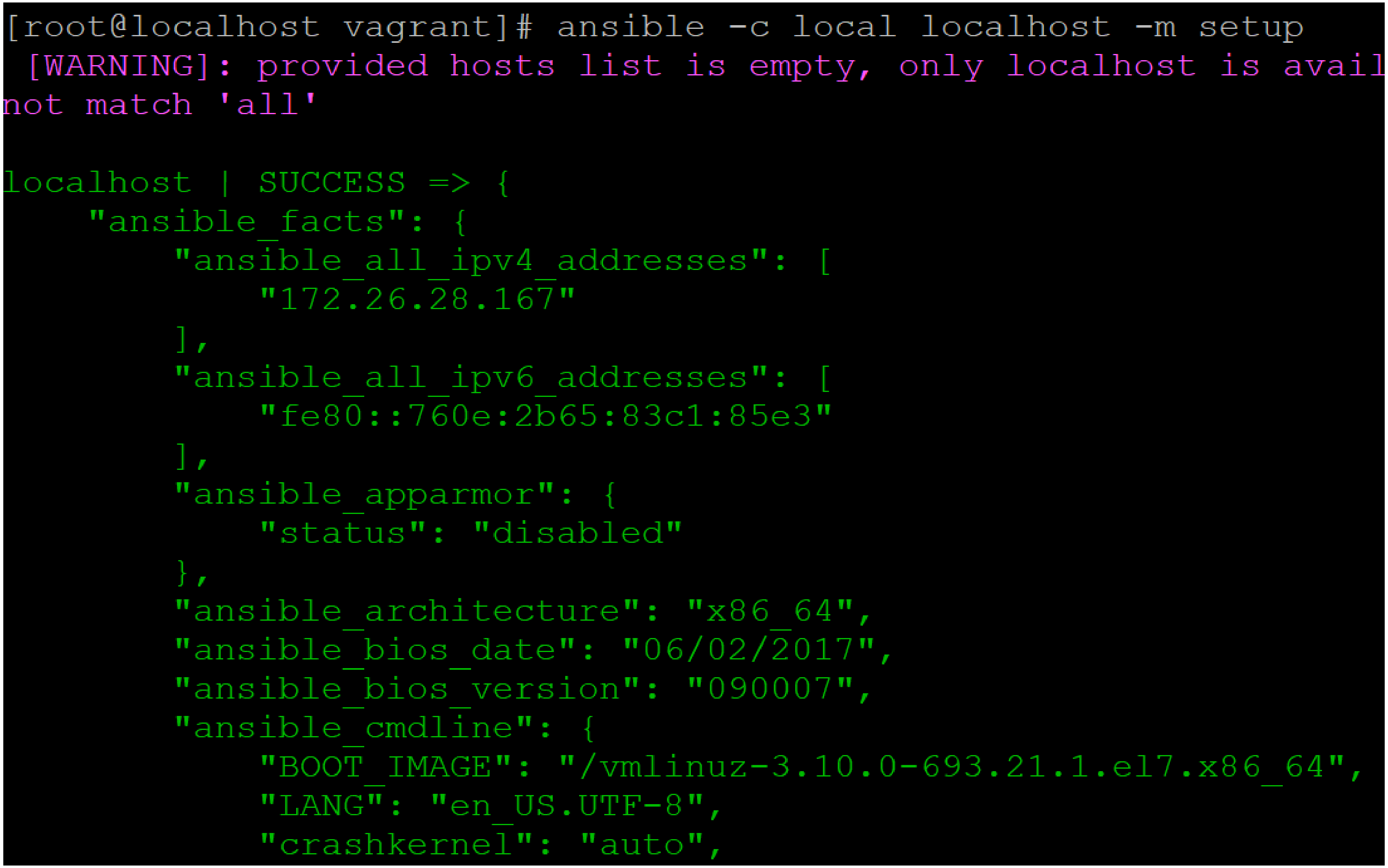
https://docs.ansible.com/ansible/latest/modules/setup_module.html

https://docs.ansible.com/ansible/2.6/user_guide/playbooks_variables.html#information-discovered-from-systems-facts

Lection #2. First playbook

Facts

```
ansible -c local localhost -m setup
```



```
[root@localhost vagrant]# ansible -c local localhost -m setup
 [WARNING]: provided hosts list is empty, only localhost is avail
not match 'all'

localhost | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "172.26.28.167"
        ],
        "ansible_all_ipv6_addresses": [
            "fe80::760e:2b65:83c1:85e3"
        ],
        "ansible_apparmor": {
            "status": "disabled"
        },
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "06/02/2017",
        "ansible_bios_version": "090007",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/vmlinuz-3.10.0-693.21.1.el7.x86_64",
            "LANG": "en_US.UTF-8",
            "crashkernel": "auto",
```

Course Content

1. Lection #1: Introduction
   1. Configuration management
   2. Ansible. How it works?
   3. Vagrant. Training env
2. Lection #2: First playbook
   1. First playbook
   2. Ansible modules
   3. Facts & variables
3. Lection #3: Base features
   1. Jinja2 templating
   2. Conditions
   3. Loops
4. Lection #4: Plugins & modules
   1. Plugins
   2. Modules
   3. Handlers
5. Lection #5: Best practices
   1. Roles
   2. Working with inventory
   3. Repository structure
6. Lection #6: Usecases
   1. Golden images
   2. Management infrastructure
   3. CI/CD integration

Lection #2. First playbook

Variables

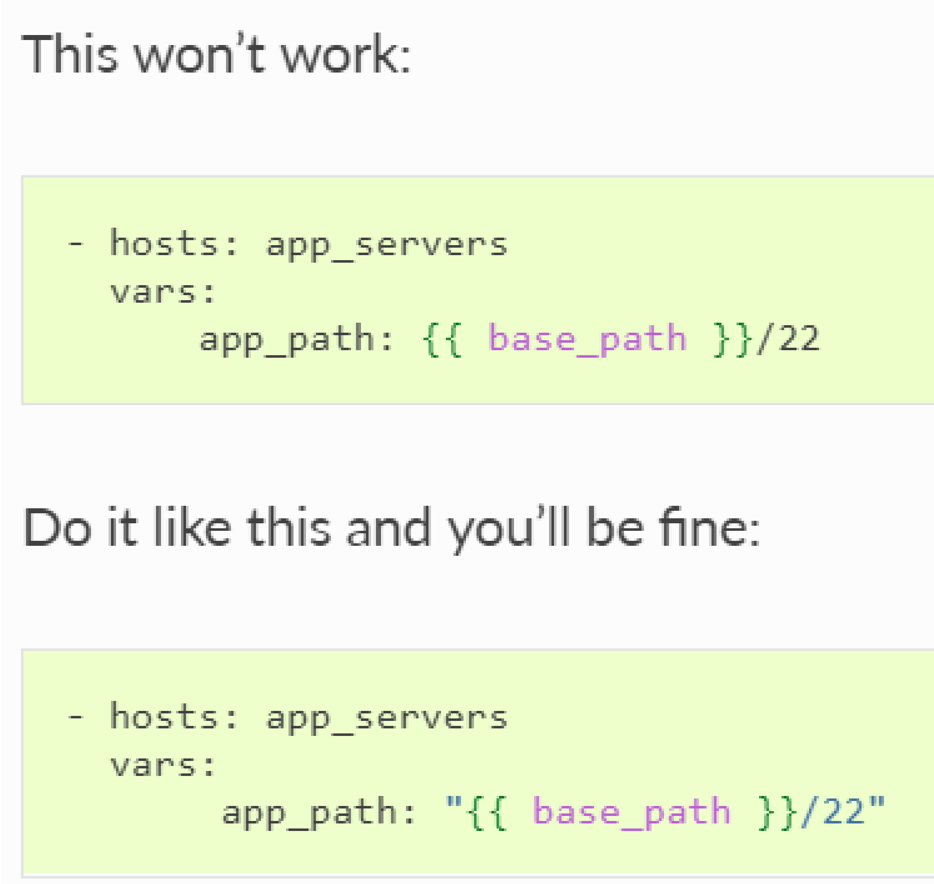
1. Variable names should be letters, numbers, and underscores. Variables should always start with a letter.
2. YAML also supports dictionaries which map keys to values
3. Can be defined in playbook/inventory/included files & roles
4. Ansible allows you to reference variables in your playbooks using the Jinja2 templating system

https://docs.ansible.com/ansible/2.6/user_guide/playbooks_variables.html

Lection #3. Base features

Jinja2 templating. Variables

Variable names should be letters, numbers, and underscores. Variables should always start with a letter.



This won't work:

```
- hosts: app_servers
  vars:
      app_path: {{ base_path }}/22
```

Do it like this and you'll be fine:

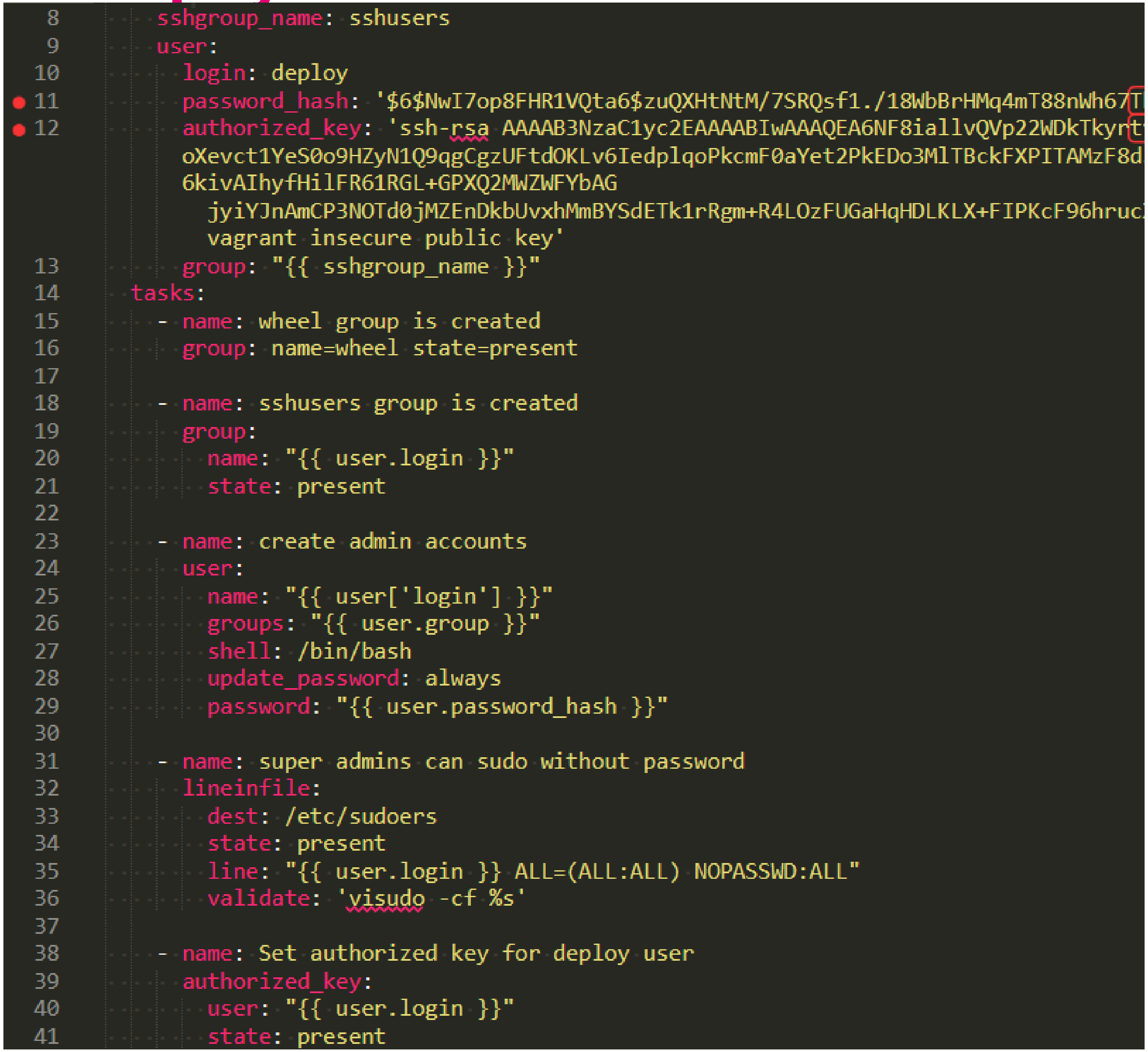
```
- hosts: app_servers
  vars:
       app_path: "{{ base_path }}/22"
```

Lection #2. First playbook

Workshop

```
$env:http_proxy='http://spbsrv-proxy2.t-systems.ru:3128'
$env:https_proxy='http://spbsrv-proxy2.t-systems.ru:3128'
git clone http://projects.t-systems.ru/lgonchar/ansible-course-public.git
cd student_files/02
vagrant up –provider hyperv
```

Lection #2. First playbook
Workshop



```
    sshgroup_name: sshusers
    user:
      login: deploy
      password_hash: '$6$NwI7op8FHR1VQta6$zuQXHtNtM/7SRQsf1./18WbBrHMq4mT88nWh67T
      authorized_key: 'ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NF8iallvQVp22WDkTkyrt
      oXevct1YeS0o9HZyN1Q9qgCgzUFtdOKLv6IedplqoPkcmF0aYet2PkEDo3MlTBckFXPITAMzF8d
      6kivAIhyfHilFR61RGL+GPXQ2MWZWFYbAG
        jyiYJnAmCP3NOTd0jMZEnDkbUvxhMmBYSdETk1rRgm+R4LOzFUGaHqHDLKLX+FIPKcF96hruc
        vagrant insecure public key'
      group: "{{ sshgroup_name }}"
  tasks:
    - name: wheel group is created
      group: name=wheel state=present

    - name: sshusers group is created
      group:
        name: "{{ user.login }}"
        state: present

    - name: create admin accounts
      user:
        name: "{{ user['login'] }}"
        groups: "{{ user.group }}"
        shell: /bin/bash
        update_password: always
        password: "{{ user.password_hash }}"

    - name: super admins can sudo without password
      lineinfile:
        dest: /etc/sudoers
        state: present
        line: "{{ user.login }} ALL=(ALL:ALL) NOPASSWD:ALL"
        validate: 'visudo -cf %s'

    - name: Set authorized key for deploy user
      authorized_key:
        user: "{{ user.login }}"
        state: present
```

ERLEBEN, WAS VERBINDET.

Lection #2. First playbook
Homework

Modify existing playbook:

- Install snmpd
- Remove default snmpd config via file module
- Configure snmpd via blockinfile module
- Open via iptables module 161 udp port
- Generate self signed cert via openssl_certificate module

THANK YOU!
Q&A

> "Use the ansible, Luke"
>
> Obi Wan Kenobi

lev.goncharov@t-systems.com