# Internal training
# INTRODUCTION TO ANSIBLE

Author / manager: Lev Goncharov / Ilya Semerhanov

Lection #1 – Configuration management

# 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

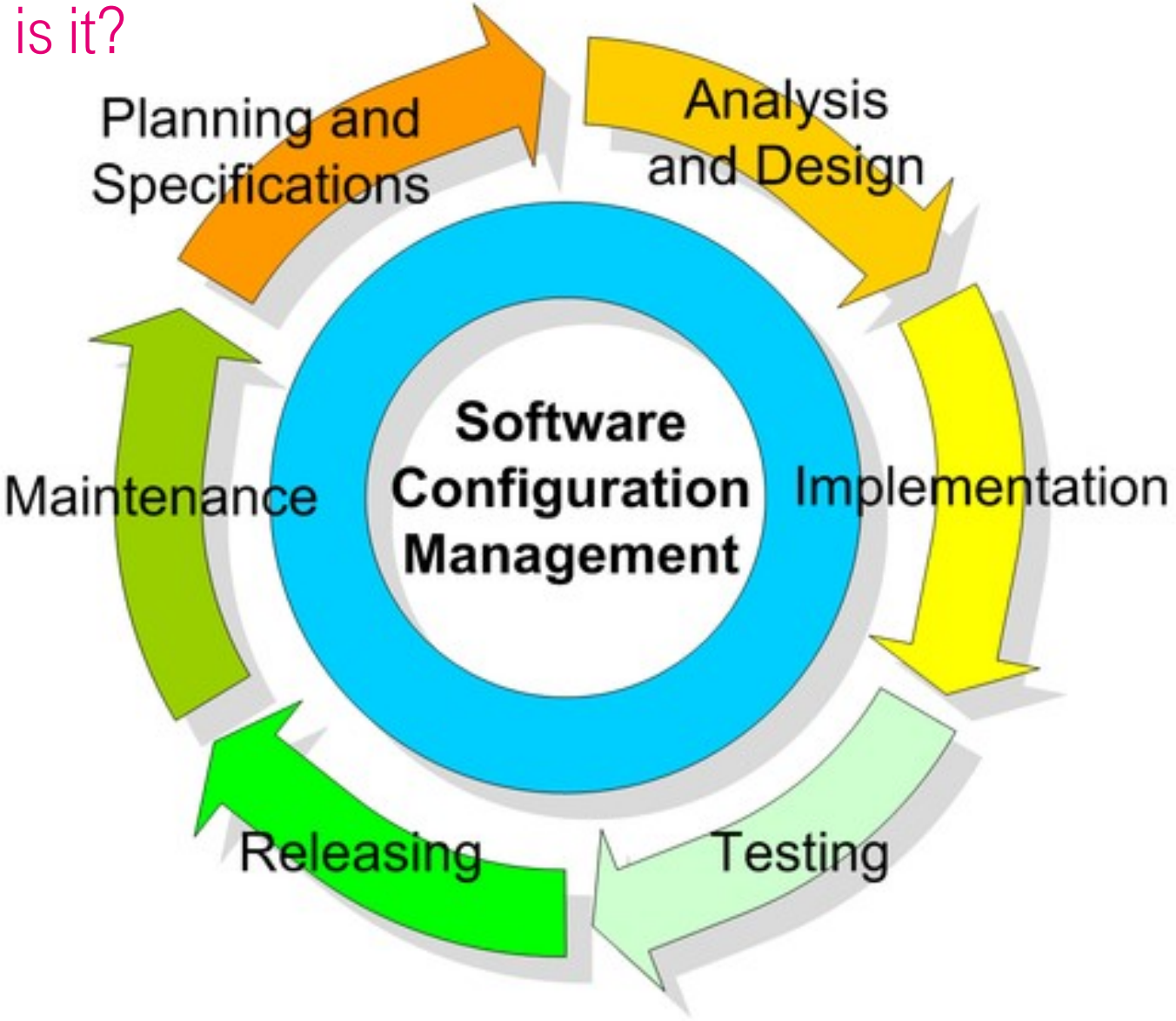
4. Lection #4: Plugins & modules
   1. Plugins
   2. Modules
   3. Handlers

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 #1. Configuration Management

## CM. What is it?



# Lection #1. Configuration Management

CM

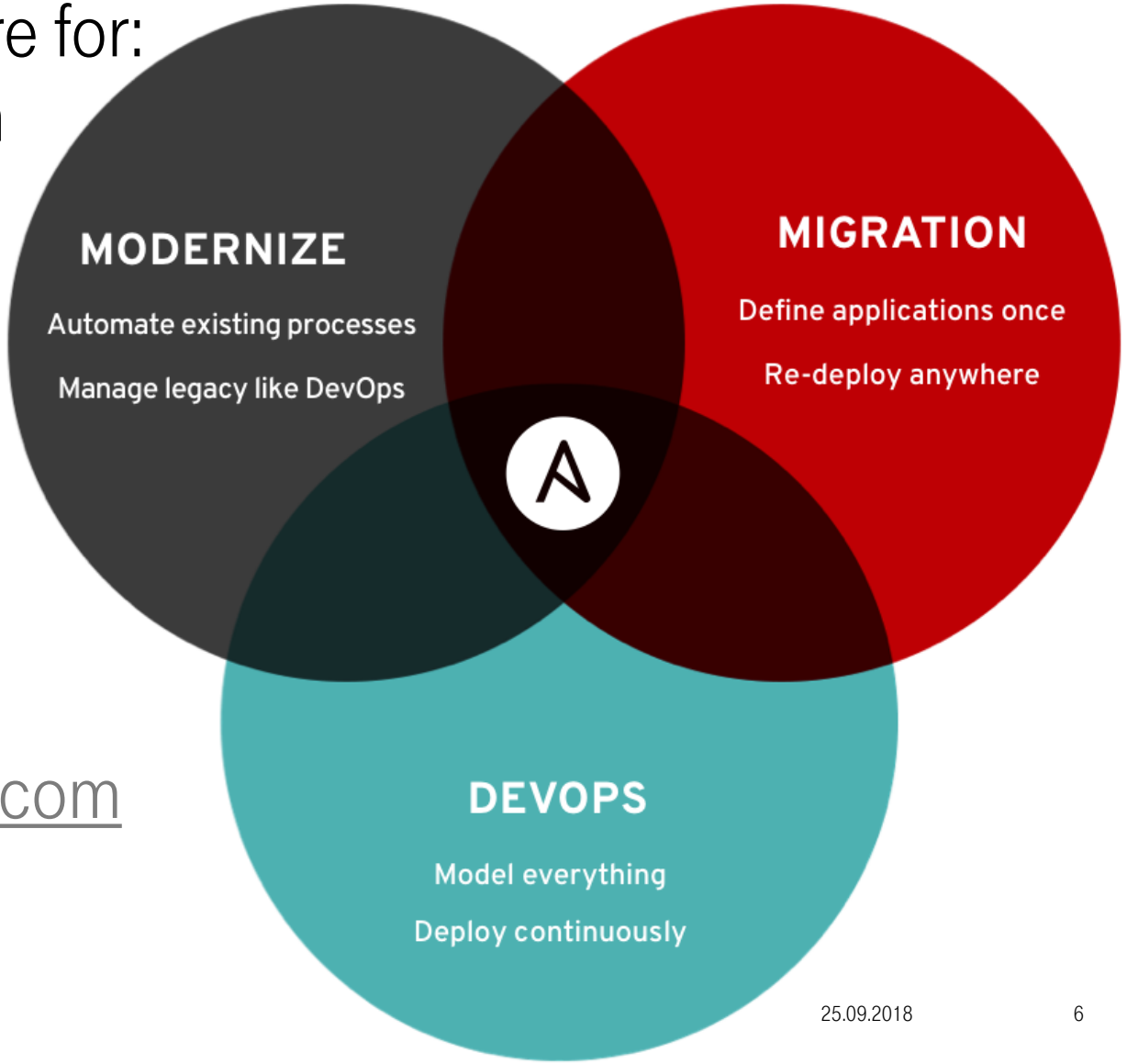
1. Infrastructure As Code
2. Easy to understand
3. Reduce busfactor

# Lection #1. Configuration Management

## CM. What is Ansible?

Open source software for:

1. Software provision
2. Configuration management
3. Application deployment




https://www.ansible.com

# Lection #1. Configuration Management

## CM. Compare

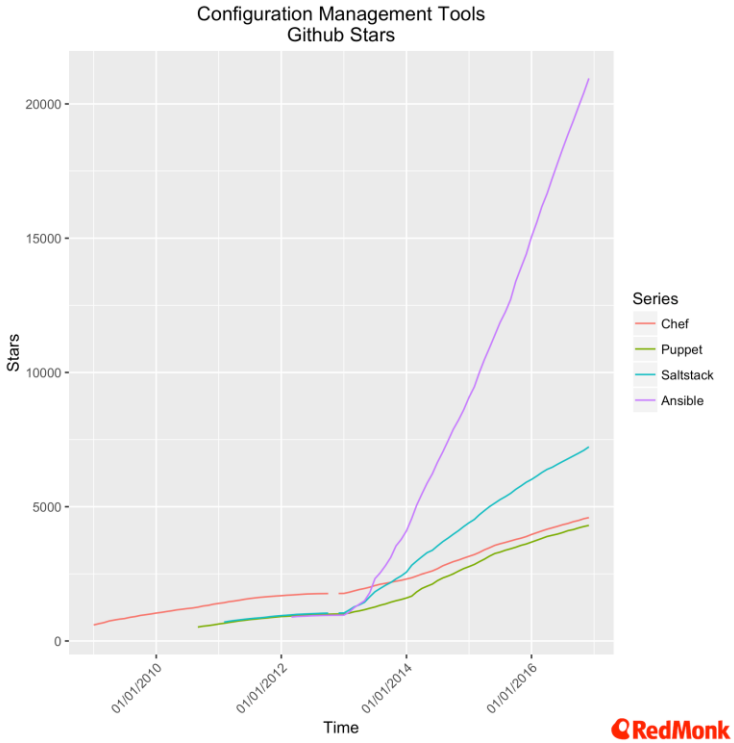
| | Puppet | Chef | Salt | Ansible |
|---|---|---|---|---|
| Initial release | 2005 | 2009 | 2011 | **2012** |
| Configuration Language | DSL | Ruby/DSL | YAML | **YAML** |
| Template Language | ERB | ERB | Jinja2 | **Jinja2** |
| Agentless | | | | ✓ |
| Simple ad-hoc task execution | | | ✓ | ✓ |
| GitHub ★s* | 2,239 | 2,729 | 3,531 | **6,202** |

*As of 2014/06/02. Just a reference point. Stars don't mean much in the grand scheme of things ;-)

# Lection #1. Configuration Management

## CM. Ansible.

- Efficient : Agentless, minimal setup
- Easy : Simple declarative language
- Scalable : Can manage thousands of nodes
- Agentless : SSH / WinRM transport
- Maturity : Production ready
- Large community

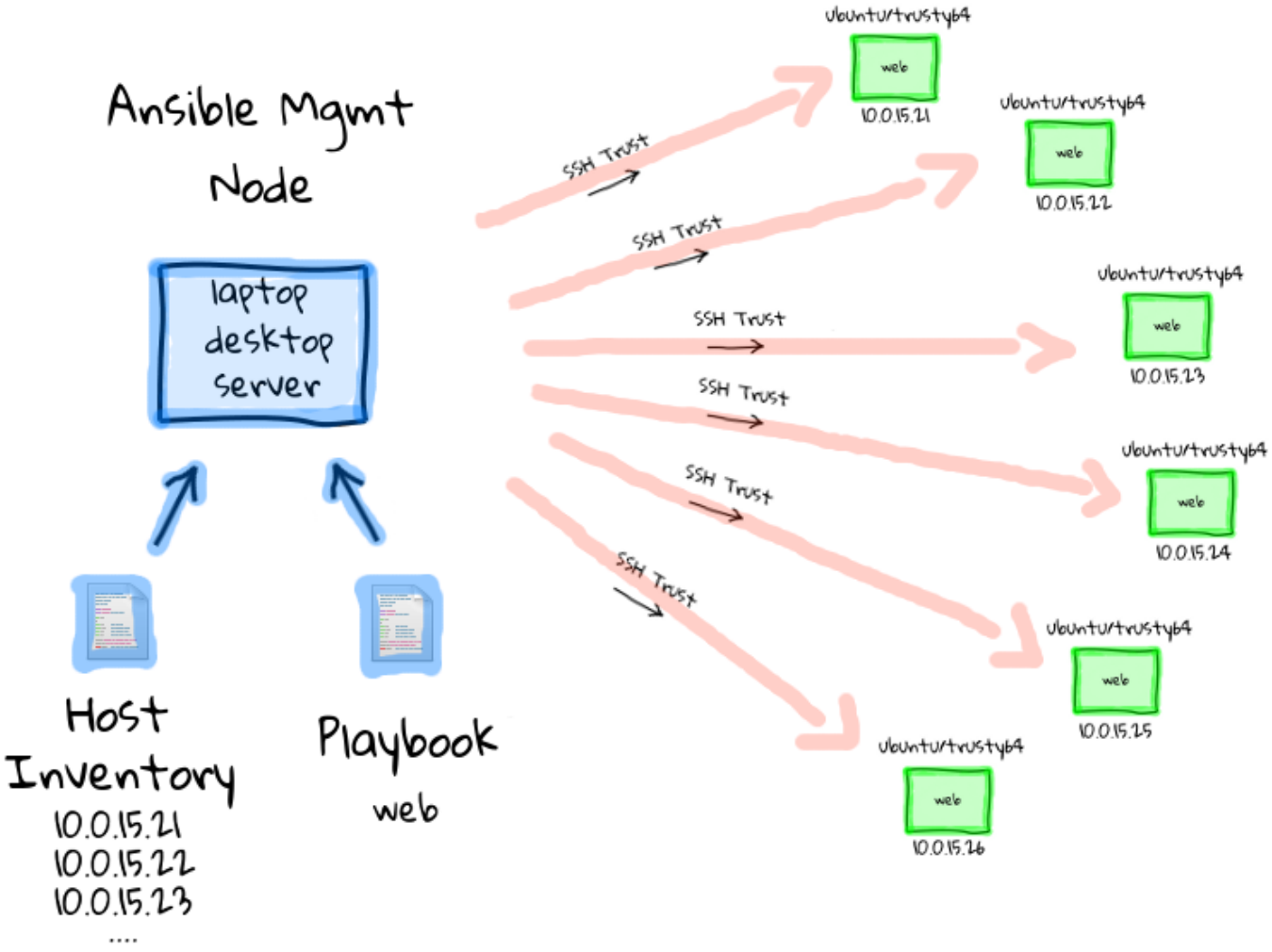


# Course Content

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

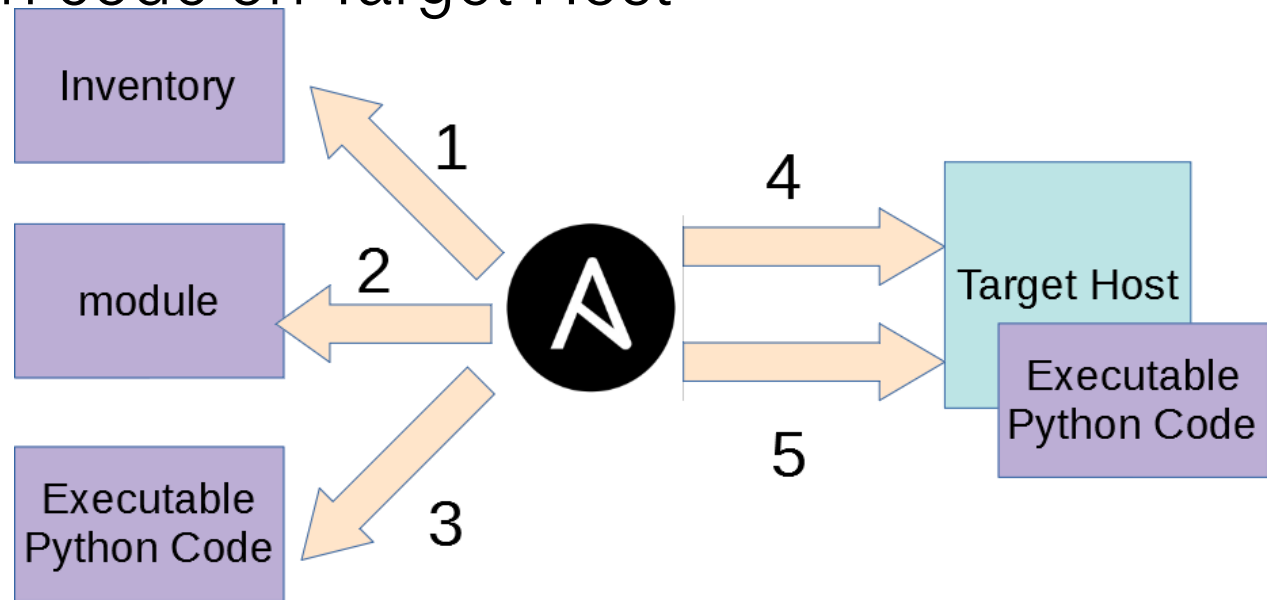
## Ansible. How it works?



# Lection #1. Configuration Management

## Ansible. How it works?

1. Lookup Target Host
2. Read Module
3. Generate executable code from Module
4. Copy Executable python code to via SCP
5. Execute python code on Target Host



# Lection #1. Configuration Management

## Ansible. Terms

- Controller Machine: The machine where Ansible is installed, responsible for running the provisioning on the servers you are managing.
- Inventory: An initialization file that contains information about the servers you are managing.
- Playbook: The entry point for Ansible provisioning, where the automation is defined through tasks using YAML format.
- Task: A block that defines a single procedure to be executed, e.g. Install a package.
- Module: A module typically abstracts a system task, like dealing with packages or creating and changing files. Ansible has a multitude of built-in modules, but you can also create custom ones.
- Role: A pre-defined way for organizing playbooks and other files in order to facilitate sharing and reusing portions of a provisioning.
- Play: A provisioning executed from start to finish is called a play. In simple words, execution of a playbook is called a play.
- Facts: Global variables containing information about the system, like network interfaces or operating system.
- Handlers: Used to trigger service status changes, like restarting or stopping a service.

# 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 #1. Configuration Management

## Vagrant. What is it?

Vagrant – wrapper around hypervisor

1. Cross platform
   1. win / linux / macos
   2. Hyper-v / libvirt / virtualbox / AWS
2. Unified workflow

https://www.vagrantup.com/

HashiCorp
Vagrant

# Lection #1. Configuration Management

## Vagrant. What is it?

```
vm_name = 'ansible-course-01'
rsync_excl = ['.git/', '.vagrant/']

Vagrant.configure('2') do |config|
  unless ENV['http_proxy'].nil?
    raise 'Install plugin vagrant-proxyconf' unless Vagrant.has_plugin?('vagrant-proxyconf')
    config.proxy.enabled  = true
    # set values from environmental variables
    config.proxy.http     = ENV['http_proxy']
    config.proxy.https    = ENV['http_proxy']
    config.proxy.no_proxy = ENV['no_proxy']
  end

  config.vm.synced_folder './', '/vagrant', type: 'rsync', rsync__exclude: rsync_excl
  config.vm.box = 'bento/centos-7.4'
  config.vm.provision 'shell', inline: 'chmod -v 0440 /etc/sudoers.d/vagrant'
  config.vm.provision 'shell', path: 'run_me.sh'
  config.vm.provision 'ansible_local' do |ansible|
    ansible.playbook = '/vagrant/provision_me.yml'
  end

  config.vm.provider 'hyperv' do |h|
    h.vm_integration_services = {
      guest_service_interface: true,
      time_synchronization: true
    }
    h.memory  = 512
    h.vmname  = vm_name
  end
end
```

# Lection #1. Configuration Management
## Vagrant. links

1. https://www.vagrantup.com/
2. http://projects.t-systems.ru/lgonchar/vagrant_demo
3. https://t.me/ru_hashicorp

# 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 #1. Configuration Management
## Workshop

```
1 $env:http_proxy='http://spbsrv-proxy2.t-systems.ru:3128'
2 $env:https_proxy='http://spbsrv-proxy2.t-systems.ru:3128'
3 git clone http://projects.t-systems.ru/lgonchar/ansible-course-public.git
4 cd student_files/01
5 vagrant up –provider hyperv
6 vagrant rsync
7 vagrant ssh-config
```

# Lection #1. Configuration Management

## Workshop. run_me.sh

```
#!/bin/bash
set -e

grep deploy /etc/passwd || useradd -m deploy
grep deploy /etc/sudoers || echo 'deploy ALL=(ALL:ALL) NOPASSWD:ALL' >> /etc/sudoers
visudo -c
[ -d /home/deploy/.ssh ] || mkdir -v /home/deploy/.ssh
echo 'ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEA6NF8iallvQVp22WDkTkyrtvp9eWW6A8YVr+kz4TjGYe7gHzIw+niNltGEFHzD8+v
1I2YJ6oXevct1YeS0o9HZyN1Q9qgCgzUFtdOKLv6IedplqoPkcmF0aYet2PkEDo3MlTBckFXPITAMzF8dJSIFo9D8HfdOV0IAdx4O7Ptix
WKn5y2hMNG0zQPyUecp4pzC6kivAIhyfHilFR61RGL+GPXQ2MWZWFYbAGjyiYJnAmCP3NOTd0jMZEnDkbUvxhMmBYSdETk1rRgm+R4LOzF
UGaHqHDLKLX+FIPKcF96hrucXzcWyLbIbEgE98OHlnVYCzRdK8jlqm8tehUc9c9WhQ== vagrant insecure public key' > /home/
  deploy/.ssh/authorized_keys
chown -Rv deploy:deploy /home/deploy/.ssh
chmod -v 700 /home/deploy/.ssh
chmod -v 600 /home/deploy/.ssh/authorized_keys
yum -y install httpd iptables-services git net-snmp net-snmp-utils
git clone https://github.com/gabrielecirulli/2048.git
cp -Rv 2048/* /var/www/html
cat > /etc/snmp/snmpd.conf << EOL
syslocation Server Room
syscontact SysAdmin (devops@example.com)
rocommunity snmp_secret_rocommunity
EOL
systemctl start httpd snmpd
systemctl enable httpd snmpd
iptables -A INPUT -m tcp -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -m udp -p udp --dport 161 -j ACCEPT

```

# Lection #1. Configuration Management

## Workshop. provision_me.yml

```
1  ---
2
3  - name: provision server
4    hosts: all
5    become: True
6    become_user: root
7    tasks:
8      - name: run run_me.sh
9        command: /vagrant/run_me.sh
10
```

# Lection #1. Configuration Management
## Homework

Create training environmental

- Install git
- Install hyper-v or virtualbox
- Install vagrant
- Provision VM
- Review bash script & ansible playbook
- Visit provisioned web site

# THANK YOU!

Q&A

> Use the ansible, Luke
>
> Obi Wan Kenobi

lev.goncharov@t-systems.com