

# Ansible

## rychlý úvod

Věroš Kaplan  
@verosk

<http://inuits.eu/>

<http://veroskaplan.cz/>



# Managing computers

“It’s quick change here and there...”

“And fix this, please.”

“Install this software. Fix it.”



# Deploying new servers

“Install this software on this servers.”

“Make a copy of this machine and change something.”

ASAP



2 servers ->



Image by Torkild Retvedt, CC BY SA

... a potom!



# Puppet

One manifest rules them all!

```
node base_server {
  include "verosk::ssh_keys"
  include "verosk::vimrc"
  include "verosk::packages"
  #
  class { "czechglobe::satellite_postfix": }
  augeas { "sshd": # disable SSH login
    context => "/files/etc/ssh/sshd_config",
    changes => [ 'set PermitRootLogin without-password' ],
  }
  if ($operatingsystem == "CentOS") {
    yumrepo { "czechglobe":
      name => "CzechGlobe internal repository",
      baseurl => "http://[REDACTED]",
      gpgcheck => 0,
    }
  }
}

node virtual_server inherits base_server {
}

node default {
  notify{"Default configuration applied":
    withpath=>true,
  }
  include "verosk::ssh_keys"
  file { "/etc/issue":
    ensure => present,
    content => "$operatingsystem $operatingsystemrelease\n$kernel $kernel
version on $hardwaremodel ($virtual)\nPuppet managed\n\n",
  }
}

"site.pp" 250L, 6899C written                                29,1-8                                Top
```

## **... but after some time**

- **lost in curly brackets**
- **lost in manifest dependencies**
- **lost in duplicate resource names**
  
- **a bit lost in Ruby**
- **puppet master can't be bootstrapped by  
Puppet**
- **OOM**

**DISCLAIMER: in fact, not Puppet problem,  
but my bad usage of Puppet!**



# Ansible

You can start in minutes

No agent installation

Readable recipes

Python inside

Jinja inside

<http://ansible.com/>

<http://github.com/ansible/ansible>





# Ansible

In fact, it's parallel ssh...

.... but done in smart way.

# (My) usage

- deployment of internal systems
  - owncloud, iSCSI...
  - whatever...
- automation of error-prone tasks
  - resize file system, LVM and pack it
  - mass configuration of KVM guests
  - deploy testing machine (6 times a day)
- `authorized_keys`

# Installation

```
$ yum install ansible
```

||

```
$ pip install ansible
```

# Hello, Ansible!

```
$ ansible -i hosts -m ping all
```

```
astarte | success >> {  
    "changed": false,  
    "ping": "pong"  
}
```

# Run ansible module

```
$ ansible -i hosts -m ping all
```

inventory file



module

host selection

```
# kvm host
```

```
perkele      ansible_ssh_user=root  ansible_ssh_host=192.168.1.10
```

```
# openvpn server
```

```
hades       ansible_ssh_user=root  ansible_ssh_host=192.168.1.11
```

```
# licence servers
```

```
lm-ecognition  ansible_ssh_user=root  ansible_ssh_host=192.168.1.12
```

```
lm-arcgis      ansible_ssh_user=root  ansible_ssh_host=192.168.1.13
```

```
lm-envi        ansible_ssh_user=root  ansible_ssh_host=192.168.1.14
```

```
lm-arcgis-c    ansible_ssh_user=root  ansible_ssh_host=192.168.1.15
```

```
[lm-routers]
```

```
hades
```

```
[lm-servers]
```

```
lm-ecognition
```

```
lm-arcgis
```

```
lm-arcgis-c
```

```
lm-envi
```

```
[perkele-test-servers:children]
```

```
lm-routers
```

```
lm-servers
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

# SSH key distribution

```
$ ansible -i hosts -m authorized_key \
  -a "user=root key='ssh-rsa .... verosk' " \
  all
```

...

40 hosts OK

# Simple playbook

```
$ ansible-playbook -i hosts playbook.yml -v
```

- YAML based
- list of tasks



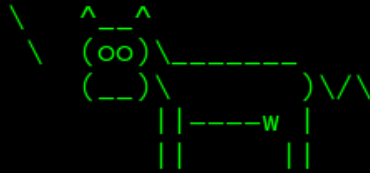
```
-- toaster@OpenShift: ~
- name: Change ssh keys
  hosts: all
  user: root
  gather_facts: no
  vars:
    user_to_change: root

  tasks:
    - name: Add keys
      authorized_key: state=present user={{user_to_change}}
                    key="{{ lookup('file', item) }}" #"
      with_fileglob: keys_to_add/*.key

    - name: Drop keys
      authorized_key: state=absent user={{user_to_change}}
                    key="{{ lookup('file', item) }}" #"
      with_fileglob: keys_to_drop/*.key

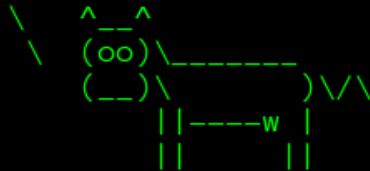
# source: https://github.com/VerosK/change-ssh-keys-ansible
~
```

< TASK: Prepare mount directory >



ok: [perkele]

< TASK: Mount the directory >



changed: [perkele]

< TASK: Stat the directory >



ok: [perkele]

< TASK: Fill the space by empty file >



TASK: [Drop the empty file] \*\*\*\*\*  
changed: [perkele]

TASK: [UMount the directory] \*\*\*\*\*  
changed: [perkele]

TASK: [Create LV swap again] \*\*\*\*\*  
changed: [perkele]

TASK: [Zero swap] \*\*\*\*\*  
failed: [perkele] => {"changed": true, "cmd": "dd if=/dev/zero of=/dev/lmenvi/swap bs=1024k ", "delta": "0:00:07.984465", "end": "2014-02-13 10:05:32.316554", "rc": 1, "start": "2014-02-13 10:05:24.332089"}  
stderr: dd: writing `'/dev/lmenvi/swap': No space left on device  
1505+0 records in  
1504+0 records out  
1577058304 bytes (1.6 GB) copied, 7.96998 s, 198 MB/s  
...ignoring

TASK: [Create the swap] \*\*\*\*\*  
changed: [perkele]

TASK: [Deativate VG] \*\*\*\*\*  
changed: [perkele]

TASK: [Drop partitions] \*\*\*\*\*  
changed: [perkele]

PLAY RECAP \*\*\*\*\*  
perkele : ok=23 changed=19 unreachable=0 failed=0

veros@snail:~/ansible/resize\_fs\$ █

# Role based playbook

- machine roles are defined
  - webserver, DB server, KVM server,...
  - role is YAML playbook
- machines in groups
  - by inventory file
- roles are applied to machines || groups
- roles depends on each other

```
---
- name: KVM guests for license routers
  hosts: lm-routers
  user: root
  roles: [ kvm-guest ]
  gather_facts: no
  tags: kvm

- name: KVM guests for license servers
  hosts: lm-servers
  user: root
  roles: [ kvm-guest ]
  gather_facts: no
  tags: kvm
  serial: 1 # install machines one-by-one

- name: Common configuration to all nodes
  hosts: all
  user: root
  roles:
    - common

- name: Configure LM router servers
  hosts: lm-routers
  user: root
  roles:
    - openvpn
    - kickstarter

- name: License servers eCognition
  hosts: lm-ecognition
  user: root
  tags: lm-ecognition
  roles: [ license-ecognition, webusage ]

- name: License servers arcgis
  hosts: [ lm-arcgis, lm-arcgis-c ]
  user: root
  tags: lm-arcgis
  roles: [ license-arcgis, webusage ]

- name: License servers envi
```

skipping: [lm-envi]

TASK: [license-envi | Install License manager] \*\*\*\*\*  
skipping: [lm-envi]

TASK: [license-envi | Make link for LSB 3.0 x64] \*\*\*\*\*  
ok: [lm-envi]

TASK: [license-envi | Update requisities] \*\*\*\*\*  
ok: [lm-envi]

TASK: [license-envi | Install license server files] \*\*\*\*\*  
ok: [lm-envi]

TASK: [license-envi | Add startup script to init.d] \*\*\*\*\*  
ok: [lm-envi]

TASK: [license-envi | Start the sript] \*\*\*\*\*  
ok: [lm-envi]

TASK: [license-envi | Order web page creation] \*\*\*\*\*  
ok: [lm-envi]

TASK: [webusage | Instal nginx server] \*\*\*\*\*  
ok: [lm-envi]

TASK: [webusage | Create nginx config] \*\*\*\*\*  
ok: [lm-envi]

TASK: [webusage | Create cronjob to update] \*\*\*\*\*  
ok: [lm-envi]

TASK: [webusage | Enable nginx server] \*\*\*\*\*  
ok: [lm-envi]

PLAY RECAP \*\*\*\*\*

hades	: ok=46	changed=2	unreachable=0	failed=0
lm-arcgis	: ok=32	changed=0	unreachable=0	failed=0
lm-arcgis-c	: ok=32	changed=1	unreachable=0	failed=0
lm-ecognition	: ok=24	changed=0	unreachable=0	failed=0
lm-envi	: ok=31	changed=1	unreachable=0	failed=0
perkele	: ok=10	changed=4	unreachable=0	failed=0

# Interesting features

handlers

register\_variable

failed\_when

ignore\_errors

notify

delegate\_to

user, sudo\_user

tags

async\_task

with\_items, with\_\*

ansible-pull

ansible-vault

**FAQ**



# Reusable roles?

<http://galaxy.ansible.com/>

# Facts

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

local facts in

`/etc/ansible/facts.d/`

# Distribution dependent playbooks

- message: Debian
  - when: ansible\_os\_distribution == 'debian'
- include\_vars: {{ansible\_os\_distribution}}

<https://galaxy.ansible.com/list#/roles/1229>

**Demo time?**

# Logstash (E+L+K)

in Vagrant

<https://github.com/VerosK/vagrant-playground-logstash-kibana>