IT Infrastructure services

Roman Kuchin
Juri Hudolejev
2019

Infrastructure parts

- Network
- Hardware (optional)
- Virtualization platform
- Container platform
- Supporting services
  - Load balancers
  - DNS
  - DHCP
  - Proxies
  - Databases(mysql, postgresql, mongodb, elastic, influxdb, couchdb)
- Services
  - Web
  - Applications

Managing infrastructure

- Manually
  - rtfm -> console -> seems working
- Infrastructure as a code
  - rtfm -> code -> staging/testing -> works

IaaC

- Takes more time at start
- Takes way more actions to start
- Slow, very slow
- No normal errors
- New languages, new surprises
- More bugs

IaaC?

- Version control
- Staging
- Code review
- Auto testing
- Scalability
- Components reuse

Management tools

- Ansible
- Chef
- Puppet
- SaltStack
- Terraform

Course rules

- All infrastructure changes should be in GitHub
- Final infrastructure should be provisioned with single button
- No passwords in repo history
- Only necessary changes in infrastructure

Ansible

- Easy to use (when compare to Chef, Puppet, Salt)
- Uses ssh, no client needed
  - Can be used in networking as well
- Support Jinja2 templates
- Lots of built-in modules
- IaaD

Ansible parts

- Control machine (Ansible server)
- Manage node
- Inventory
- Playbook
- Play
- Task
- Module

Inventory

/etc/ansible/hosts by default:

```
[switches]
1.1.1.3
1.1.1.4
[routers]
1.1.1.1
gw1.mydomain.net
[network-devices:children]
switches
routers
```

Ansible config

1. ANSIBLE_CONFIG (env)
2. ansible.cfg (current dir)
3. .ansible.cfg (~)
4. /etc/ansible/ansible.cfg

Modules

- ping
- apt
- user
- hostname
- file

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

Ad-hoc vs playbooks

ansible network-devices -m “raw” -a “show ip int br” -u cisco --ask-pass

ansible-playbook get_interfaces.yml