

Network management

Roman Kuchin
2020

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

laaC

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

IaaS?

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

Management tools

- Ansible
- Chef
- Puppet
- SaltStack
- Terraform

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
- IaaS

Ansible parts

- Control node (Where Ansible runs)
- Managed node
- Inventory
- Playbook
- Play
- Task
- Module

Docs: https://docs.ansible.com/ansible/latest/user_guide/index.html

Inventory

```
[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

- apt
- user
- hostname
- file
- ping
- ios_config
- raw

Docs: https://docs.ansible.com/ansible/2.9/modules/modules_by_category.html

Ansible for network automation

<https://docs.ansible.com/ansible/2.9/network/index.html>

Ad-hoc vs playbooks

```
ansible network-devices -m "raw" -a "show ip int br" -u cisco --ask-pass
```

```
ansible-playbook get_interfaces.yml
```