



Managing network devices like servers

Mircea Ulinic
Cloudflare, London

San Jose, CA
October 2017

Mircea Ulinic

- Network engineer at Cloudflare
- Member and maintainer at NAPALM Automation
- Integrated NAPALM in Salt
- OpenConfig representative
- <https://mirceaulinic.net/>

GitHub: mirceaulinic

Twitter: @mirceaulinic

How are servers managed?

- (Typically) many more servers than network devices
- SRE teams are generally large
- Automation methodologies for many years
- Variety of tools, including:
 - Salt
 - Ansible
 - Chef
 - Puppet
- *Many* features already implemented
- Plenty of success stories

Network platforms that can be managed like servers

- White box devices
 - Arista EOS
 - Cumulus
 - etc.
- Containerised solutions
 - Cisco IOS-XR (64 bit only)
 - Cisco NX-OS
 - etc.

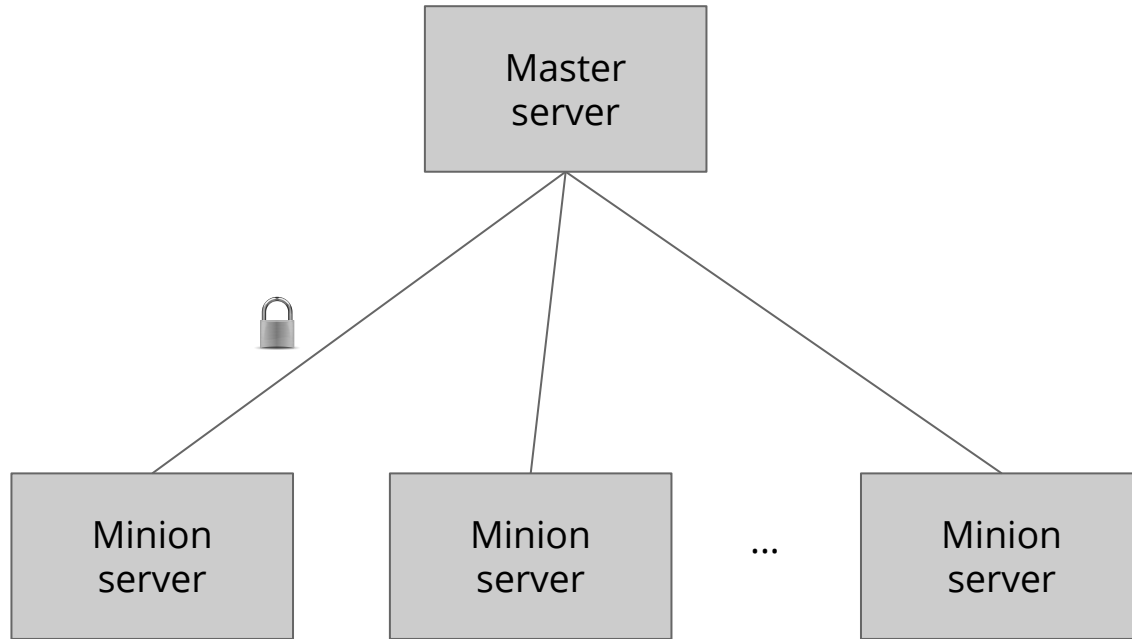
Network platforms that *can't* be managed like servers

- Junos
- Cisco IOS-XR, 32 bit
- Cisco IOS-XE, IOS
- Many many others...

Automation framework example: Salt

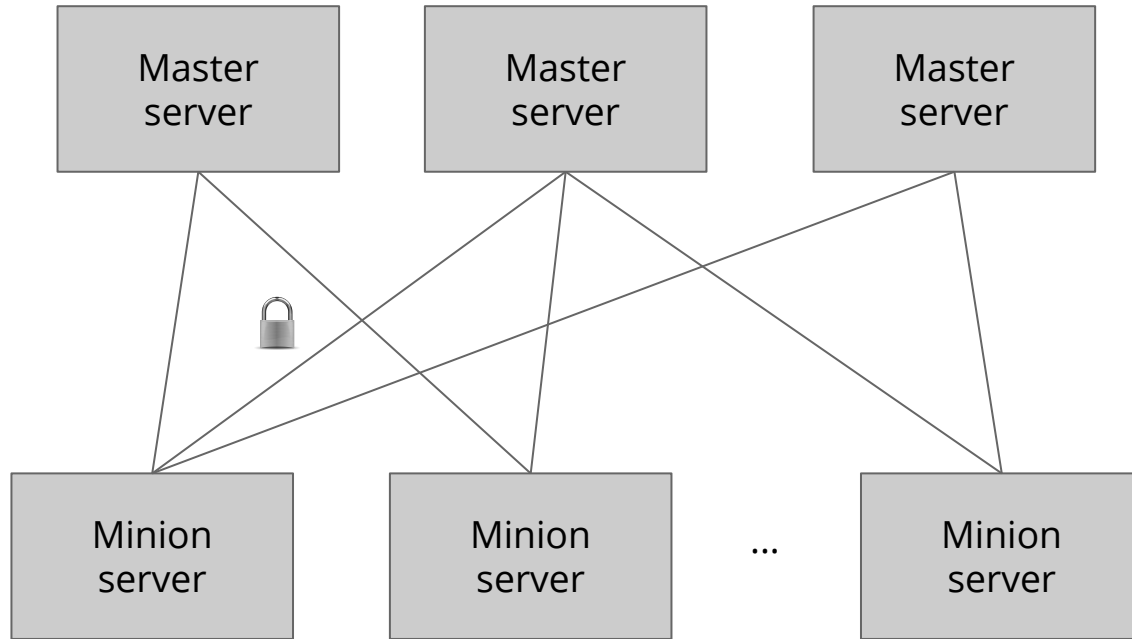
- Very scalable
 - e.g., [LinkedIn 70,000 servers](#)
- Concurrency
- Event-driven
- Easily configurable & customizable
- Native caching and drivers for useful tools

Salt Architectures (1): typical hub and spoke



<https://docs.saltstack.com/en/latest/topics/topology/index.html>

Salt Architectures (2): multi master



Salt Architectures (3): masterless minions



Arista EOS Salt minion: Installation (1)

Copy the SWIX extension to the flash

```
edge01.bjm01#copy https://salt-eos.netops.life/salt-eos-latest.swix flash:  
edge01.bjm01#copy https://salt-eos.netops.life/startup.sh flash:
```

Install the SWIX extension

```
edge01.bjm01#copy flash:salt-eos-latest.swix extension:  
edge01.bjm01#extension salt-eos-latest.swix force
```

Arista EOS Salt minion: Installation (2)

Verify the installation

```
edge01.bjm01#show extensions | include salt-eos
```

```
  salt-eos-2017-7-1.swix      1.0.11/1.fc25      A, F      27
```

Enable the local unix socket

```
edge01.bjm01(config)#management api http-commands
```

```
  protocol unix-socket
```

```
  no shutdown
```

Arista EOS Salt minion: Installation (3)

Execute the Salt Minion startup script

```
edge01.bjm01#bash
```

```
#sudo /mnt/flash/startup.sh
```

Complete installation notes at:

<https://docs.saltstack.com/en/latest/topics/installation/eos.html>

Arista EOS Salt minion: Installation (3)

Execute the Salt Minion startup script

```
edge01.bjm01#bash
```

```
#sudo /mnt/flash/startup.sh
```

Complete installation notes at:

<https://docs.saltstack.com/en/latest/topics/installation/eos.html>

Cumulus Linux Salt minion: Installation

1. Download the Salt bootstrap script

```
wget -O bootstrap-salt.sh https://bootstrap.saltstack.com
```

2. Check the script!!!

3. Install the Salt minion

```
sudo sh bootstrap-salt.sh
```

Salt CLI execution: server

```
$ sudo salt 'some-server' disk.usage
```

```
some-server:
```

```
-----
```

```
/:
```

```
-----
```

```
1K-blocks:
```

```
65869280
```

```
available:
```

```
60808360
```

```
capacity:
```

```
8%
```

```
filesystem:
```

```
rootfs
```

```
used:
```

```
5060920
```

```
/dev:
```

```
-----
```

```
1K-blocks:
```

```
65902000
```

Salt CLI execution: Arista EOS minion*

```
$ sudo salt 'edge01.bjm01' disk.usage
edge01.bjm01:
-----
/:
-----
1K-blocks:
4870812
available:
4812376
capacity:
2%
filesystem:
none
used:
58436
/dev:
-----
1K-blocks:
8192
```

* This is real output collected from a device carrying Internet traffic

Need help/advice?

Join [#saltstack #napalm](https://networktocode.herokuapp.com/rooms)

By email:

- Mircea Ulinic: mircea@cloudflare.com
- Frankie Hui: frankie@cloudflare.com

Questions



By email:

- Mircea Ulinic: mircea@cloudflare.com
- Frankie Hui: frankie@cloudflare.com

How can you contribute?

- NAPALM Automation:
<https://github.com/napalm-automation>
- SaltStack
<https://github.com/saltstack/salt>

References

[Arista Software download](#)

[Authentication system](#)

[Beacons](#)

[Engines](#)

[Event System](#)

[Grains](#)

[Jinja](#)

[load_template documentation](#)

[Master config file, default](#)

[Master config file, example](#)

[Master configuration options](#)

[Master systemd file](#)

[Mine](#)

[NAPALM](#)

[NAPALM BGP execution module functions](#)

[NAPALM Grains](#)

[NAPALM Installation](#)

[NAPALM network execution module functions](#)

[NAPALM NTP execution module functions](#)

[NAPALM Proxy](#)

[NAPALM route execution module functions](#)

[NAPALM SNMP execution module functions](#)

[NAPALM users execution module functions](#)

[Nested outputter](#)

[NETAPI Modules](#)

[Netconfig state](#)

[Node Groups](#)

[NTP state](#)

[Orchestration](#)

[Output modules](#)

[Pillar](#)

[Pillar modules](#)

[Proxy config file, default](#)

[Proxy config file, example](#)

[Proxy Minion](#)

[Proxy systemd file](#)

[Reactor](#)

[REST CherryPy](#)