Interacting with Minions

Jeremy Willden
AUTOMATION ENGINEER

@WilldenJeremy www.constellationlabs.com

Overview

Targeting a subset of Minions

Minions are not configured identically

- Web servers
- Database servers
- Load balancers

Different software, files, configurations

Targeting Minions

Minion ID

staging-web-*

Operating System

os:Ubuntu

Compound Matchers

Mix and combine

IP Address

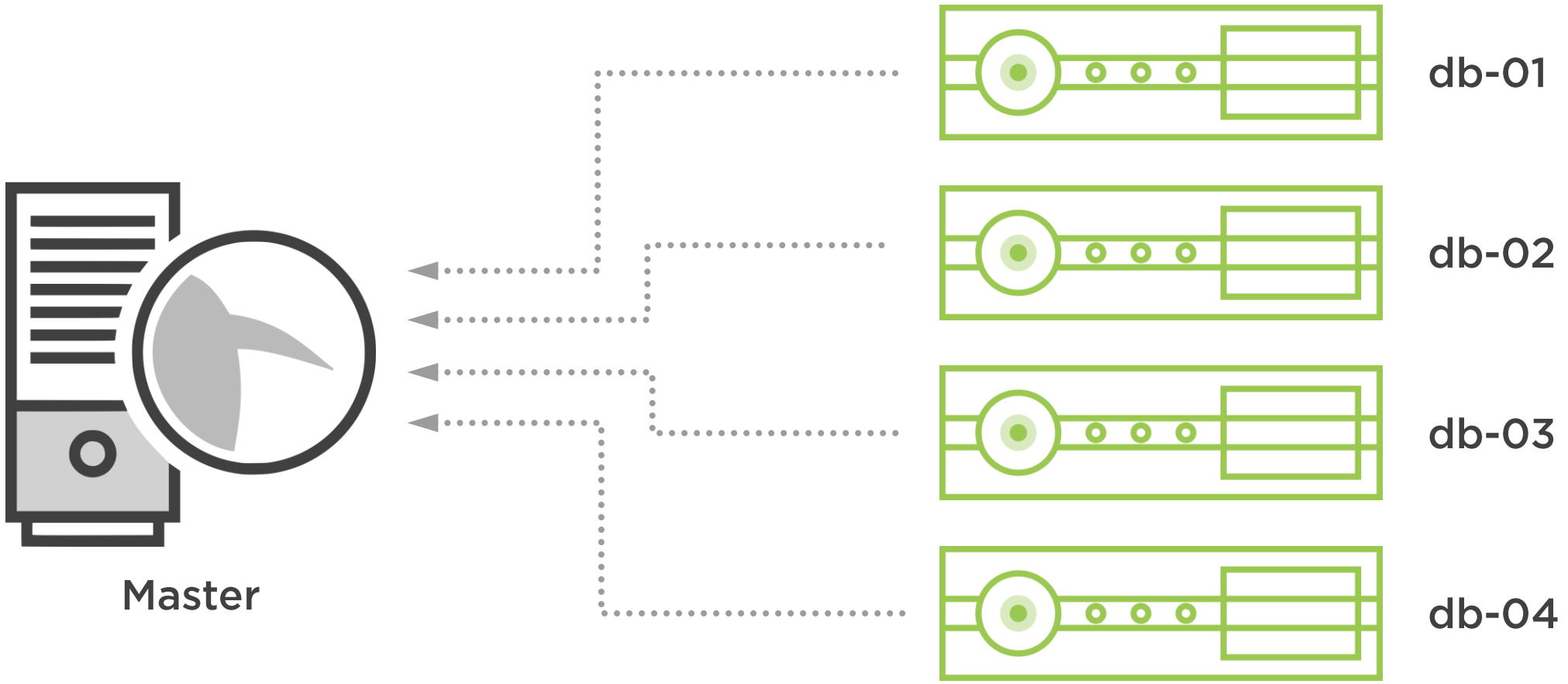
v4 10.42.37.0/24
v6 1a2d:db9::/64

Nodegroups

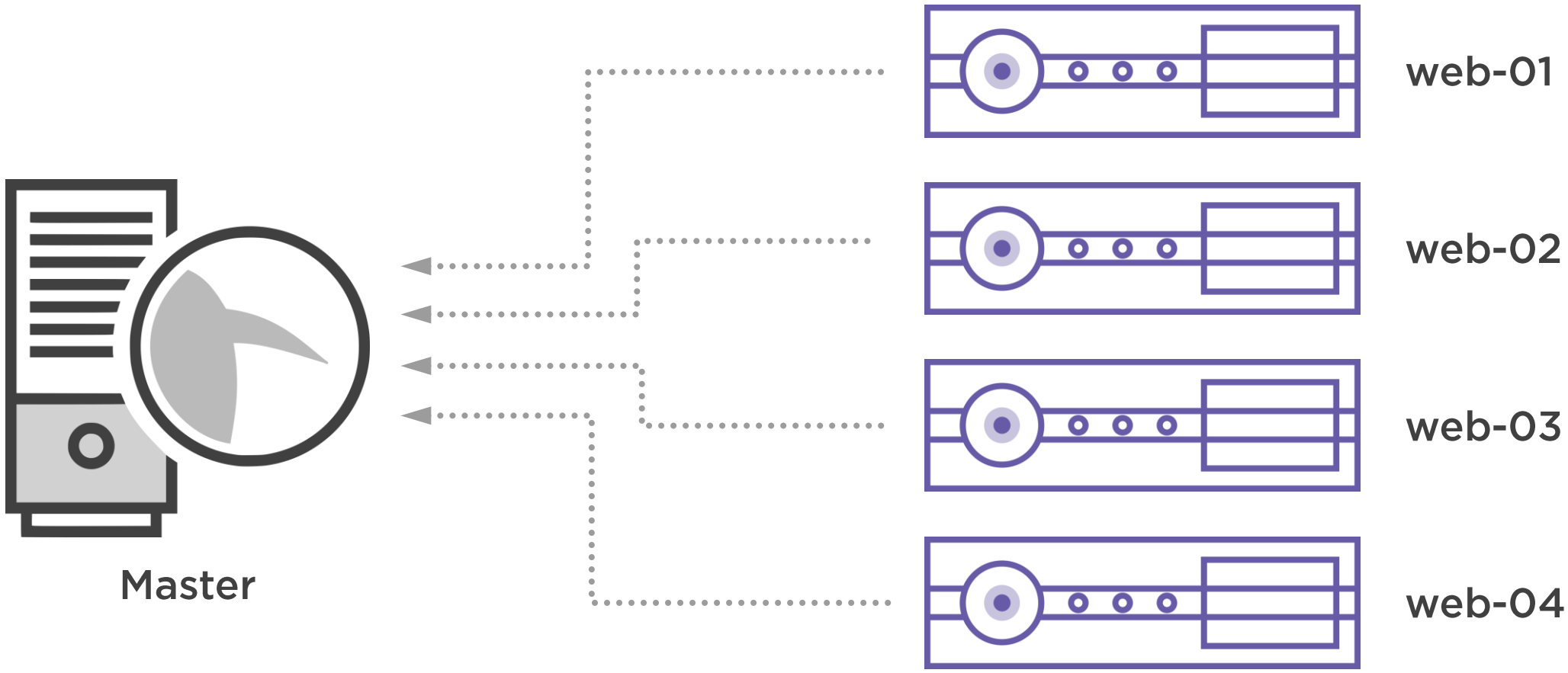
Arbitrary lists

Custom Grains

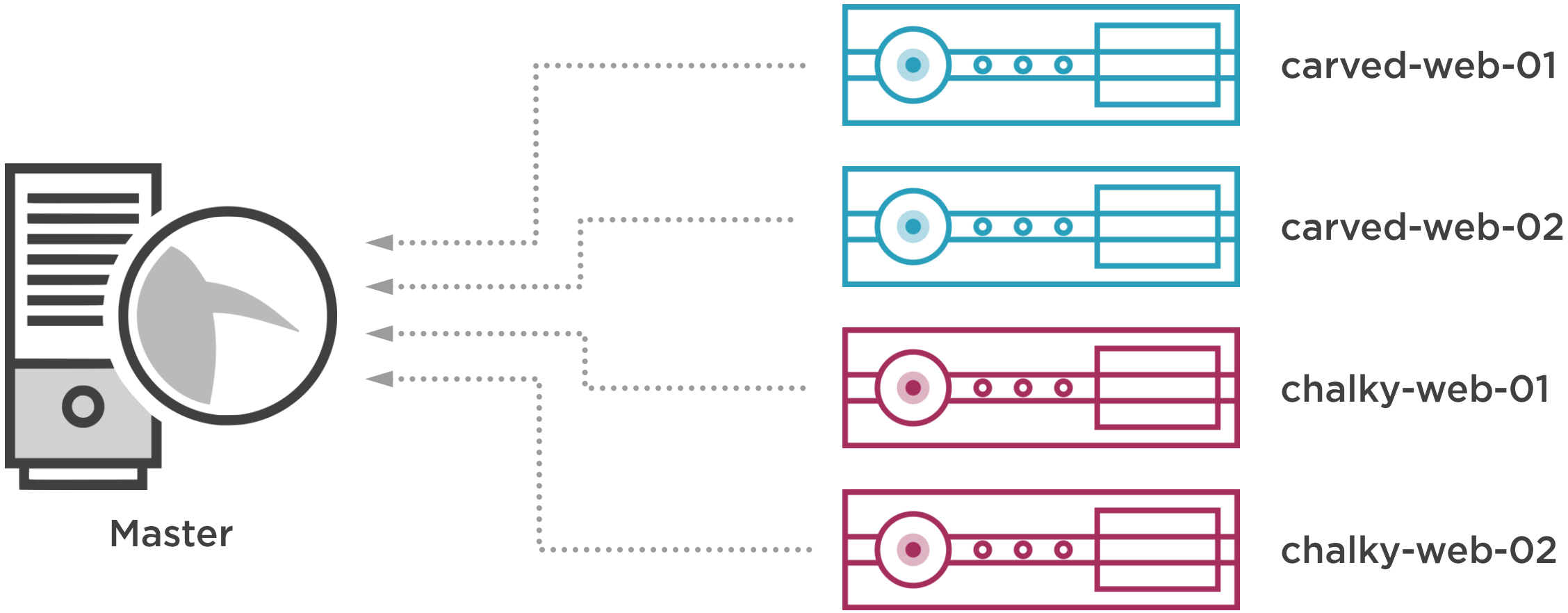
Targeting by Minion ID



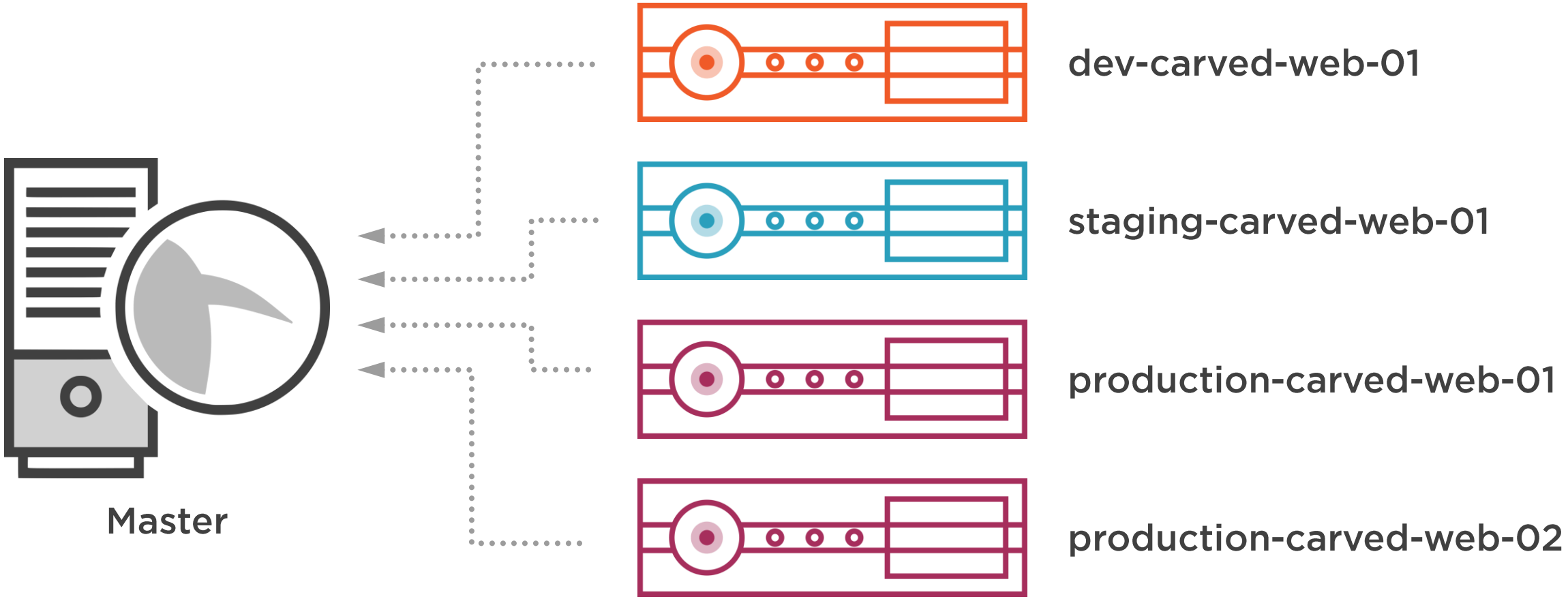
Targeting by Minion ID



Targeting by Minion ID



Targeting by Minion ID



Talking head

Targeting by grain: Operating System

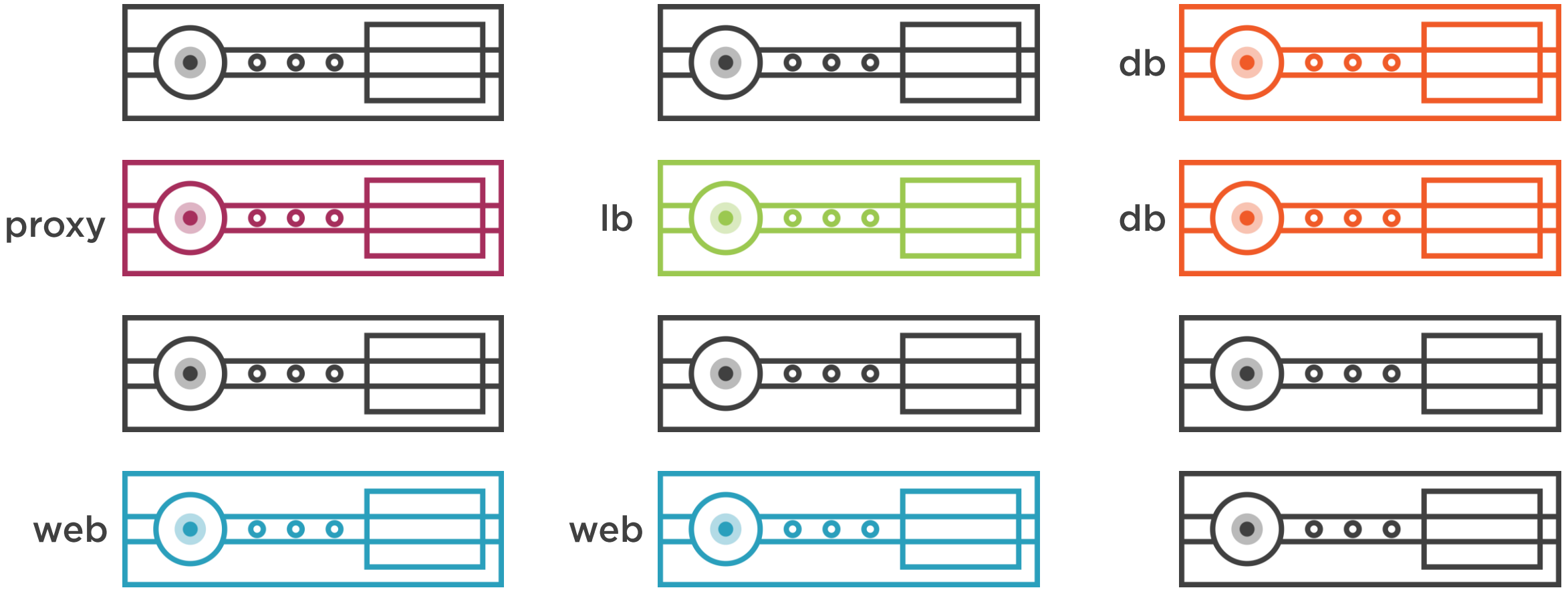
- G@os:Debian (SLS file)
- -G ‘os:Ubuntu’ (command line)

Custom grains:

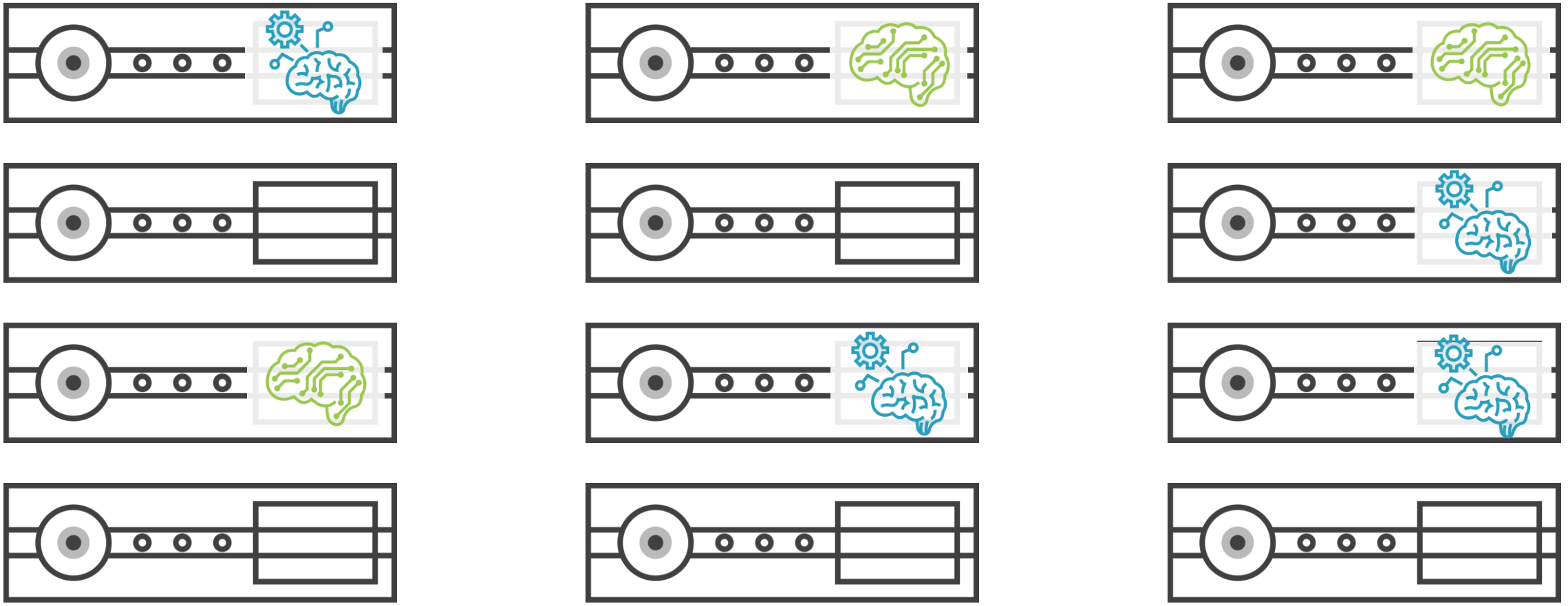
- role:webserver
- datacenter: emea_nw
- row: 15
- rack: 12
- slot: 18

Talking head

Targeting by Grain

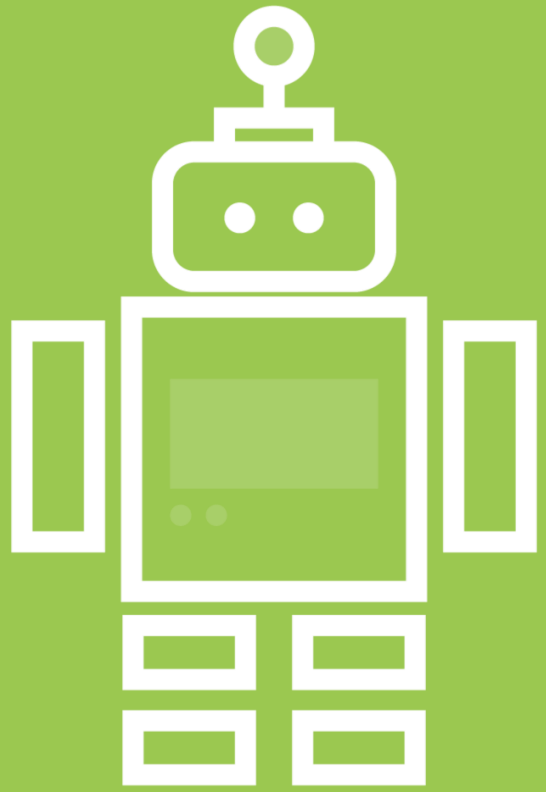


Targeting by Grain



Custom grains from the Minions are only available after applying the Highstate or after manual synchronization.

Talking head



Salt Returners

Standard and custom functions to process and take action on the results of salt activity.

- Databases
- HTML files or email
- SMS, logging, and monitoring tools

Custom Returner in Python

```
import fakedb, salt.utils.json

def returner(ret):
    '''

    Return information to a hypothetical fake database

    '''

     fakeconn = fakedb.FakeDb(host='fakeserver.example.com')

     fakeconn.sadd("%(id)s:jobs" % ret, ret['jid'])

     fakeconn.set("%(jid)s:%(id)s" % ret, salt.utils.json.dumps(ret['return']))

     fakeconn.sadd('jobs', ret['jid'])

     serv.sadd(ret['jid'], ret['id'])
```

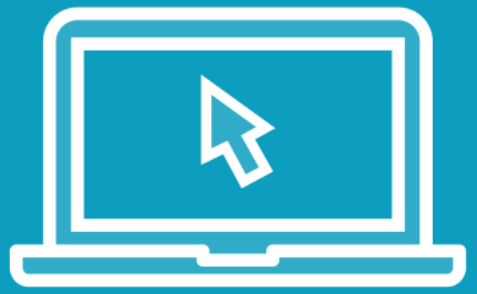
Custom Returner in Python

```
import lightcontrol, salt.utils.json

def returner(ret):
    '''

    Change the color of room lighting if there is a failure

    '''

     lightconnection = lightcontrol.LightControl(host='10.42.37.88')

     if “fail” in salt.utils.json.dumps(ret[‘return’])

         lightconnection.ChangeLights(“Red”)
```

Custom Returner in Python

```
import lightcontrol, salt.utils.json

def returner(ret):
    '''

    Change the color of room lighting if there is a failure

    '''

     lightconnection = lightcontrol.LightControl(host='10.42.37.88')

     if “fail” in salt.utils.json.dumps(ret[‘return’])

         lightconnection.ChangeLights(“Red”)
```

Demo



Targeting specific Minions

Remote command execution

Package installation

Copy files to the Minions

Returners

- Built-in: logs, databases, files
- Custom

Quick Reference: Targeting

```
$ sudo salt 'single-minion-id' test.ping
$ sudo salt 'carved-*-dev*' test.ping
$ sudo salt 'chalky-web-dev-0[1-3]' test.ping
$ sudo salt -L 'cluster-01,cluster-07,dev-web-02' test.ping
$ sudo salt -E 'carved-web-(staging|production)' test.ping
$ sudo salt -E 'cluster-(01|15|19|22)' test.ping
$ sudo salt -G 'os:Ubuntu' test.ping
$ sudo salt -G 'location:closet-16-02' test.ping
$ sudo salt -S '10.37.42.0/24' test.ping
$ sudo salt -C 'os:Debian and location:emea-main' test.ping
```

Example Module Functions

Modules	Examples
salt.modules.test	test.ping test.fib 24601
salt.modules.state	state.apply exclude=foo*
salt.states.cmd	cmd.run “my-command.sh –t –s --now”
salt.states.pkg	pkg.install vim

filewrite.py

```
def returner(ret):
    """
    Write the output of the Salt command to a file on the Minion itself.
    """
    try:
        f = open("/root/last-salt-result.txt", "w+") # use "a+" to append instead
        f.write( str(ret) )
        f.close
    except:
        pass
```

filewrite.py

```
def returner(ret):
    """
    Write the output of the Salt command to a file on the Minion itself.
    """
    with open("/root/last-salt-result.txt", "w+") as f # use "a+" to append
        f.write( str(ret) )
```

CIDR IP Notation

10.0.1.64 – 10.0.1.71

Decimal	Binary
64	0100 0000
65	0100 0001
66	0100 0010
67	0100 0011
68	0100 0100
69	0100 0101
70	0100 0110
71	0100 0111

IPv4: 32 Bit Address

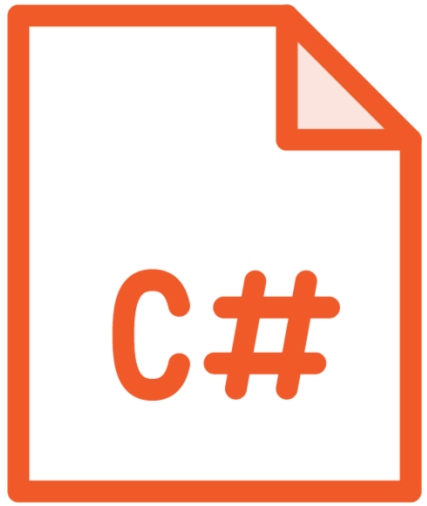
8 hosts require 3 bits,
000 through 111

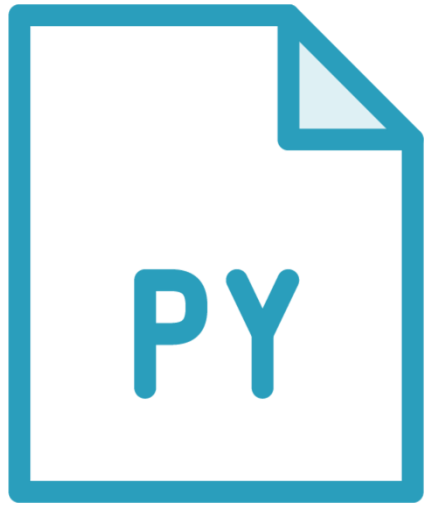
32 bits – 3 bits = 29

CIDR Notation
for these 8 hosts:

10.0.1.64/29

Custom Returners





Cython
Supports C language integration

Python
Easy setup and vast library support

Summary

This bullet list is preset with animations

Use this layout to introduce and/or summarize the module

Don’t just read a list of topics

Build excitement

Tell the viewer why this is important

- Where would they use this info on the job?