Plumbing the Internet, BSD style.

github.com/tom-clx/bsdcan18-doc (diagram.pdf, slides.pdf)
github.com/tom-clx/bsdcan18-ansible
github.com/tom-clx/bsdcan18-config

Room Check – Show of hands

- BGP?
- ...on BSD?
- Automation?
- ...with Ansible?

Hi! My name is

```
% whois -a  JOHNS58-ARIN
Name:           Johnson, Thomas
Handle:         JOHNS58-ARIN
Company:        ClaimLynx, Inc
```

- I don’t do many slideshows...

The wheels of progress grind slowly.

- **2010: Graduate to FreeBSD.**
- **2013: Start planning for HA, BGP.**
- **early-2014: BGP at primary site.**
- **mid-2014: Turn up DR site.**
- **2015: Start integrating Ansible.**
- ***Everything else gets in the way.***
- **early-2018: Production test of DR site.**
- **...because...**

Nothing lights a fire under business continuity plans like the Superbowl moving in next door...

The Superbowl is going on across the street from my datacenter, right now.

Ansible. Embrace cattle. Death to snowflakes.

- Written in Python
- Idempotent Operation
- Agent-less model…
- Client nodes need Python, sshd, rsync.
- Up and running in 30 minutes.
  http://docs.ansible.com/
- sysutils/ansible

Ansible. Initial set-up

- **Master node(s).**
  - Install Ansible
  - Edit ansible.cfg, not much to do here.
  - DNS, or you can specify IPs.
  - SSH. Use pubkeys and config file.
  - Version control.
- **Client nodes.**
  - python2 or python3, and probably rsync
  - .ssh/authorized_keys
  - sshd_config: `PermitRootLogin without-password`
  - A bootstrap script can be helpful on the client.

./inventory/hosts

- Inventory defines hosts and groups of hosts

```
[bhyve_guests]
pzansible
pzgateway
pz-rt1
as50-rt1
as50-rt2
as50-rt3
as200-rt1
as300-rt1
as100-rt1
```

```
[os_freebsd:children]
as99999
as50
as300
as200
as100
```

Ansible. host_vars and group_vars

```yaml
# inventory/host_vars/pz-rt1
host_conf: "{{ conf_root }}/rt01"
allowed_users:
  - "tom"
loader_conf: "stem"
rc_conf: "stem"

dns_main_if: "vtnet0"
ipv4_loopback: "10.70.35.2"
ipv6_loopback: "fd00:20:ffff::2"
ansible_host: "{{ ipv6_loopback }}"

network:
  netif:
    - name: "lo1"
      clone: True
      ipv4_addr: "{{ ipv4_loopback }}"
      ipv4_plen: "32"
      ipv6_addr: "{{ ipv6_loopback }}"
      ipv6_plen: "128"
      ospf: "stub"
```

```yaml
# inventory/group_vars/os_freebsd
ansible_python_interpreter: "/usr/bin/env python3"
# vim: set ts=2 sw=2 tw=80 filetype=yaml :
```

^^^ BSD Gotcha!

Hello World.

```
> ansible -m ping pzbhyve
pzbhyve | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Playbooks, plays, tasks

- **Playbooks are sets of instructions. Contain >= 1 *plays*.**
- **Plays are a set of instructions applied to >= 1 hosts/groups.**
- **Tasks represent a single operation. Placed directly in a play, or included by one of several methods.**

```yaml
- hosts: bhyve_guests
  gather_facts: no
  strategy: free

  tasks:

  - name: Preflight checks
    assert:
      that:
        - vm_root is defined
        - bhyve_host is defined
        - src_snap is defined
        - bhyve_keep is not defined

  - name: Stat the vmm path
    delegate_to: "{{ bhyve_host }}"
    failed_when: false
    register: vmm
    stat:
      path: "/dev/vmm/{{ inventory_hostname }}"
```

Ansible roles. Reuse made easy.

```
- hosts: os_freebsd

  roles:
    - { role: r53_rrset, tags: dns }
    - { role: freebsd_config, tags: freebsd }
    - { role: pf, tags: pf }
    - { role: user_management, tags: users }
```

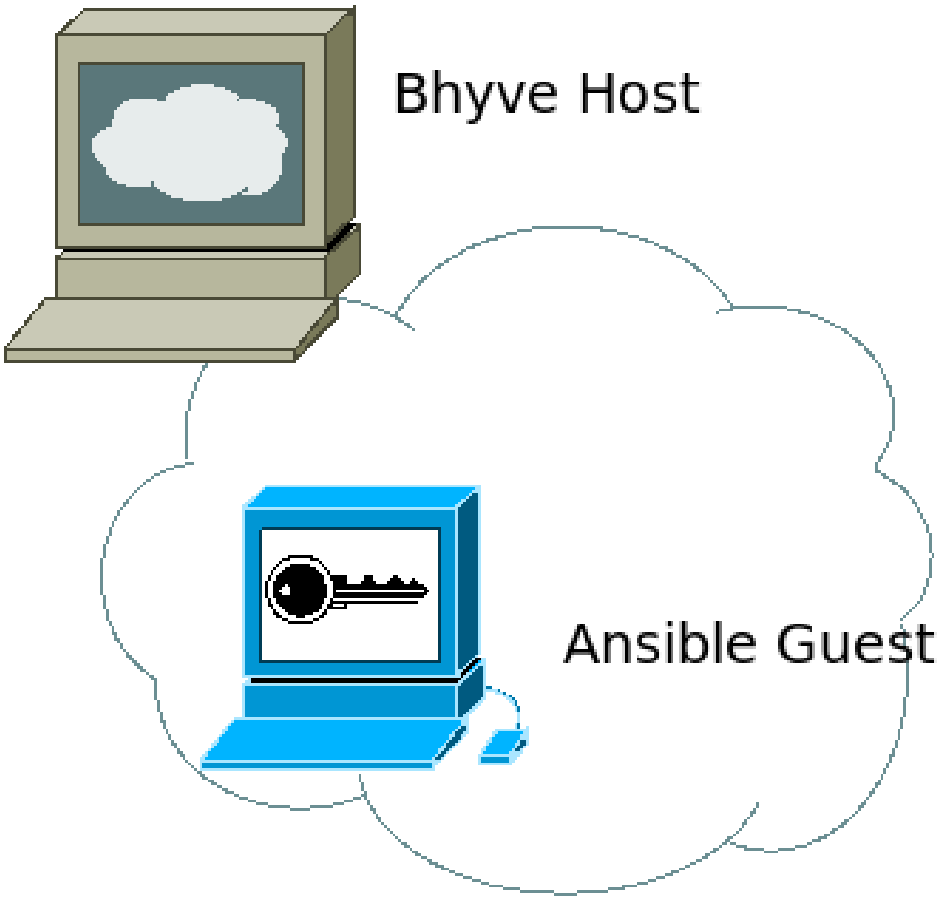
```
> find roles/freebsd_config/
roles/freebsd_config/
roles/freebsd_config/tasks
roles/freebsd_config/tasks/pkgng.yml
roles/freebsd_config/tasks/main.yml
roles/freebsd_config/tasks/boot_files.yml
roles/freebsd_config/tasks/sysctl.yml
roles/freebsd_config/tasks/syslog.yml
roles/freebsd_config/defaults
roles/freebsd_config/defaults/main.yml
roles/freebsd_config/handlers
roles/freebsd_config/handlers/main.yml
```

sysutils/vm-bhyve

https://github.com/churchers/vm-bhyve

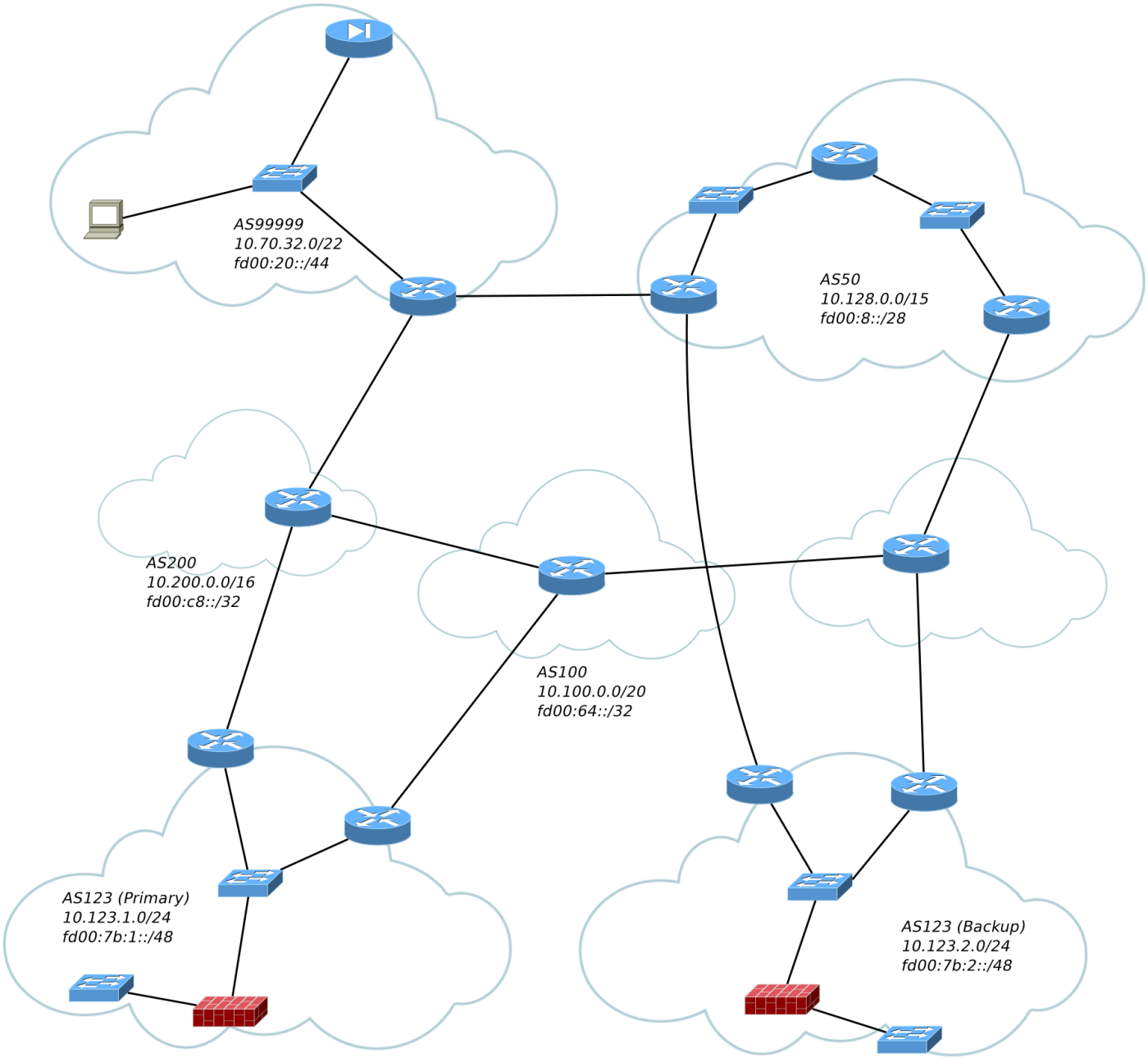
- **CLI front-end to Bhyve.**
- **Lightweight and easy.**
- **Why doesn’t FreeBSD have a tool in base?!**

Live demo (it can’t fail worse than Windows 98)



- Bhyve host with ZFS root.
- All guests cloned from a pre-built “stem cell” zvol.
- Ansible guest is persistent
- All other guests are ephemeral.

About 17 minutes later…



Ansible bootstrap process

1) Clone stem cell zvol.

```
- name: Clone the stem cell zfs
  delegate_to: "{{ bhyve_host }}"
  command: "zfs clone {{ vm_root }}/{{ src_snap }} {{ vm_root }}/{{ inventory_hostname }}/disk0"
```

2) Import zvol for bootstrapping.

```
- name: Mount the zfs at a temporary path
  delegate_to: "{{ bhyve_host }}"
  command: "zpool import -fR /tmp/{{ inventory_hostname }} -d /dev/zvol/{{ vm_root
}}/{{ inventory_hostname }} zroot_stem zroot_{{ inventory_hostname }}"
```

Ansible bootstrap process

3) Call roles for minimal host setup and routing bootstrap

```
  - name: Generate the bird.conf files
    when: "'birdies' in group_names"
    delegate_to: "{{ bhyve_host }}"
    import_role:
      name: bird_conf

  - name: Execute the rc.conf role in the context of the bhyve host
    delegate_to: "{{ bhyve_host }}"
    import_role:
      name: rc_conf
```

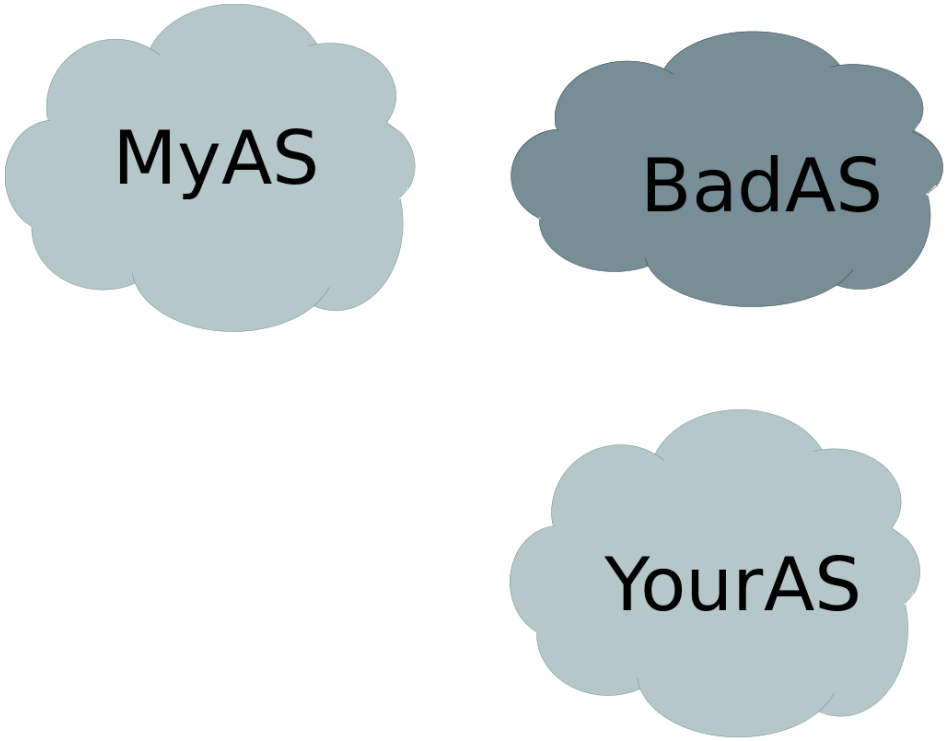
4) Boot the new guest

```
- name: Start vm via shim
  delegate_to: "{{ bhyve_host }}"
  register: reg_vm_start
  When:
    - vmm.stat.exists is defined
    - not vmm.stat.exists
  command: "/root/bin/vm_start_hax.sh {{ inventory_hostname }}"
```

The Internet is NOT a series of tubes.

(My apologies to the Senator from Alaska)

- **It is a series of autonomous systems (AS), nearly 61k as of May 19.**
- **Consisting of some networks...720k.**
- **(and that's just IPv4)**
- **Source: https://www.cidr-report.org**



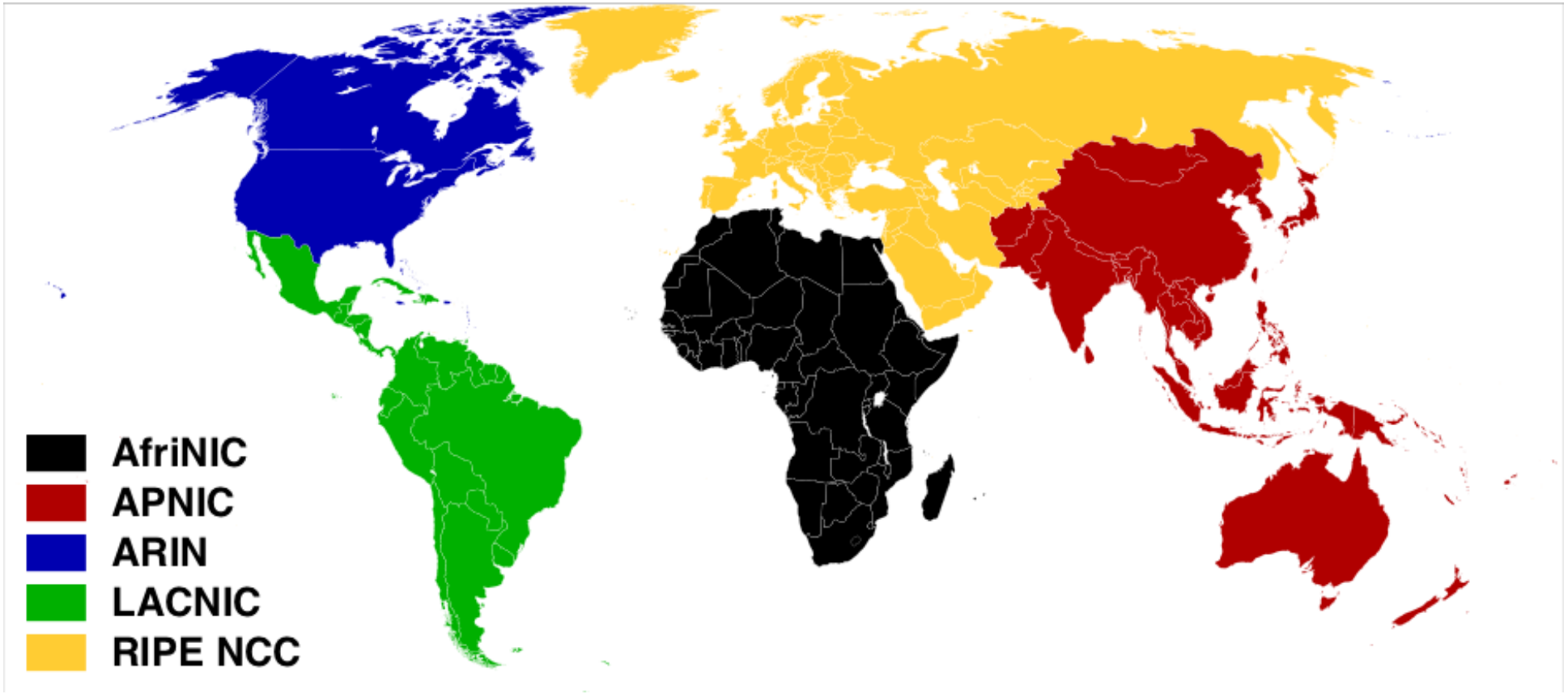
BGP is the answer.

- **Current BGP4 spec (RFC 4271) is from 2006, but older revisions date to mid-90s.**
- **Distance-Vector protocol; shortest *AS-path.***
- **Internal and External Varieties.**
- **eBGP used to communicate BETWEEN ASs.**
- **iBGP used between routers WITHIN ASs.**
- **You still need an IGP to describe your internal network.**

BGP for real. What do I need?

- **Money!**
  - USD 1050 NRC
  - USD 300 annual maintenance
- **Autonomous System Number (AS, ASN)**
- **Number resources**
  - IPv4
  - IPv6
- **Router (>1, really)**
- **Connectivity (>1, really)**

Regional Internet Registries



Source: Wikipedia

Autonomous System Numbers

- **Identifies your organization.**
- **2-byte or 4-byte.**
  - 4-byte shows up in 2007 (rfc4893)
  - Backwards-compatibility with older gear.
  - Don’t worry about it.
- **Obtain from your RIR.**
  - ARIN: USD 550 NRC, 100 annual.

IPv6. Just do it.

- **Obtain from your RIR**
- **Allocation depends on your size**
  - Single site gets /48
  - 2-12 gets /44
  - Allocations on nibble boundaries (and you should too).
- **Cost depends on size (see: ARIN fee schedule)**
  - <= /40, USD 250 NRC, 100 annual
- **Minimum announcement, /48**
- **Do not allocate < /64**

IPv4. Kickin’ it old school.

- **Legacy protocol. Everyone in this room knows about it.**
- **Exhausted! September 25, 2015**
  (https://teamarin.net/education/ipv4-depletion/)
- **Minimum announcement, /24**
- **NAT, tiny subnets, it’s all icky...but it’s the Devil we know.**

Obtaining IPv4 resources...

- **IPv4 blocks can be had.**
  - RIR waiting lists
  - Transfer market

But wait, there's more [IPv4]!

- **ARIN "4-10 rule"**
  - Formally, ARIN Number Policy Section 4.10
- **/10 set aside for "IPv6 transition"**
- **One /24 every six months, with justification**
- **If you have an IPv6 allocation, and want to dual-stack, you have justification.**

Connectivity.

Three primary methods of connecting to the world.

- **Transit**
- **Internet Exchange Points (IXPs)**
- **Peering**

Connectivity gotchas

- **Letter of Agency (LOA). Reputable carriers require you to explicitly authorize them to announce your address space.**
- **Cross-connect (XC) fees.**
- **Diverse physical routing.**

FreeBSD router in 60 seconds.

```
################################################################################
# GENERATED FROM ANSIBLE TEMPLATE. DO NOT MODIFY IN PLACE!                     #
################################################################################
hostname="as200-rt1.r53.claimlynx.net"
cloned_interfaces="lo1"
# link-local required to make OSPFv3 work.
ifconfig_lo1_alias0="inet6 fe80::1"
ifconfig_lo1="inet 10.200.10.1/32"
ifconfig_lo1_ipv6="inet6 fd00:c8:a::1/128"
ifconfig_vtnet0="inet 10.100.0.6/30"
ifconfig_vtnet0_ipv6="inet6 fd00:64:3::6/64"
ifconfig_vtnet1="inet 10.200.255.1/30"
ifconfig_vtnet1_ipv6="inet6 fd00:c8:ff00::/127"
gateway_enable="YES"
ipv6_gateway_enable="YES"
# Routing configuration
bird_enable="YES"
bird6_enable="YES"
```

To pf or not pf?

- **Firewalling is sometimes necessary on routers.**
  - Protect the host.
  - Prevent traffic leakage (IXPs).
  - Drop obvious junk.
- **Beware ‘keep state’**
  - Connection legs WILL take different paths.
  - If you track state, you WILL break connections.
  - When your connections break, you WILL lose hair troubleshooting.

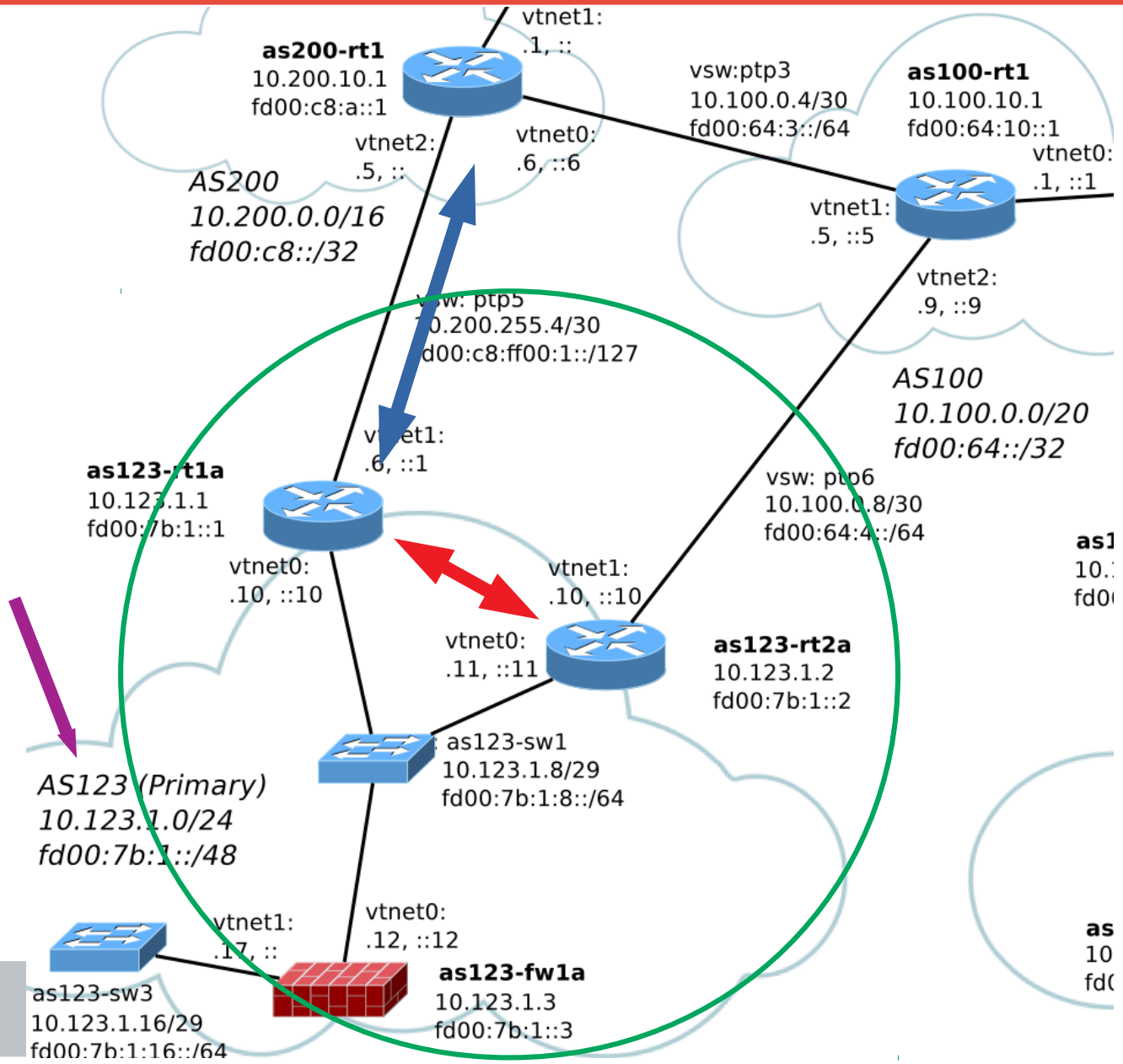
BIRD Internet Routing Daemon

http://bird.network.cz

BIRD highlights.

- Currently two versions: 1.6 and 2.0.
- Support for common protocols: BGP, OSPF, etc.
- Support for IPv4 and IPv6.
- Easy, flexible configuration structure.
- Birdc(6) utility to monitor and control.

A tour of BIRD configuration



Bird in action.

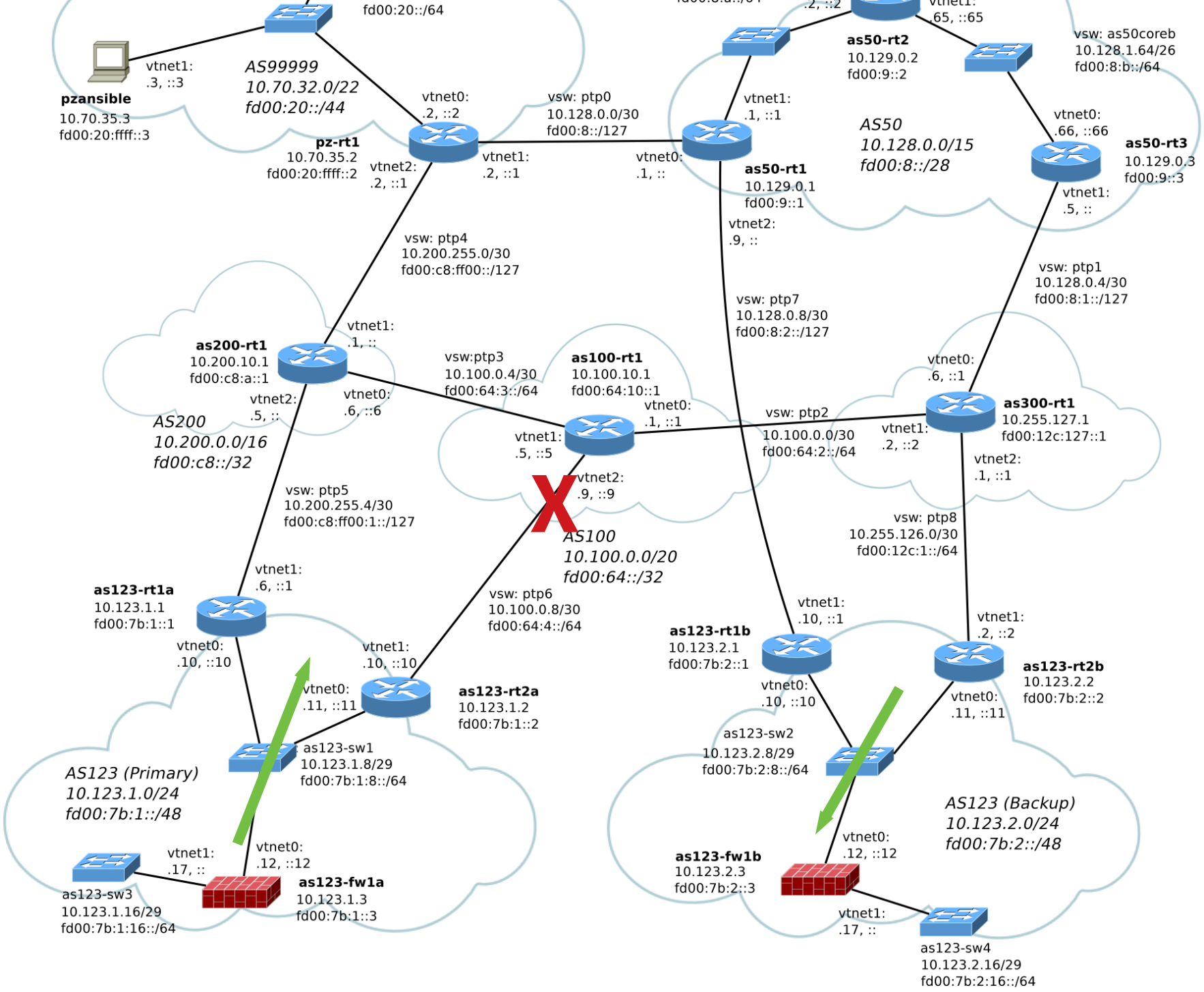
Your birdc cheat sheet…

- **show protocol all**
- **Show route all**
  - Show route for <ip>
  - Show route <prefix>
- **Show ospf neighbors**
- **Configure**
- **Echo all**
- **Restart <protocol>**

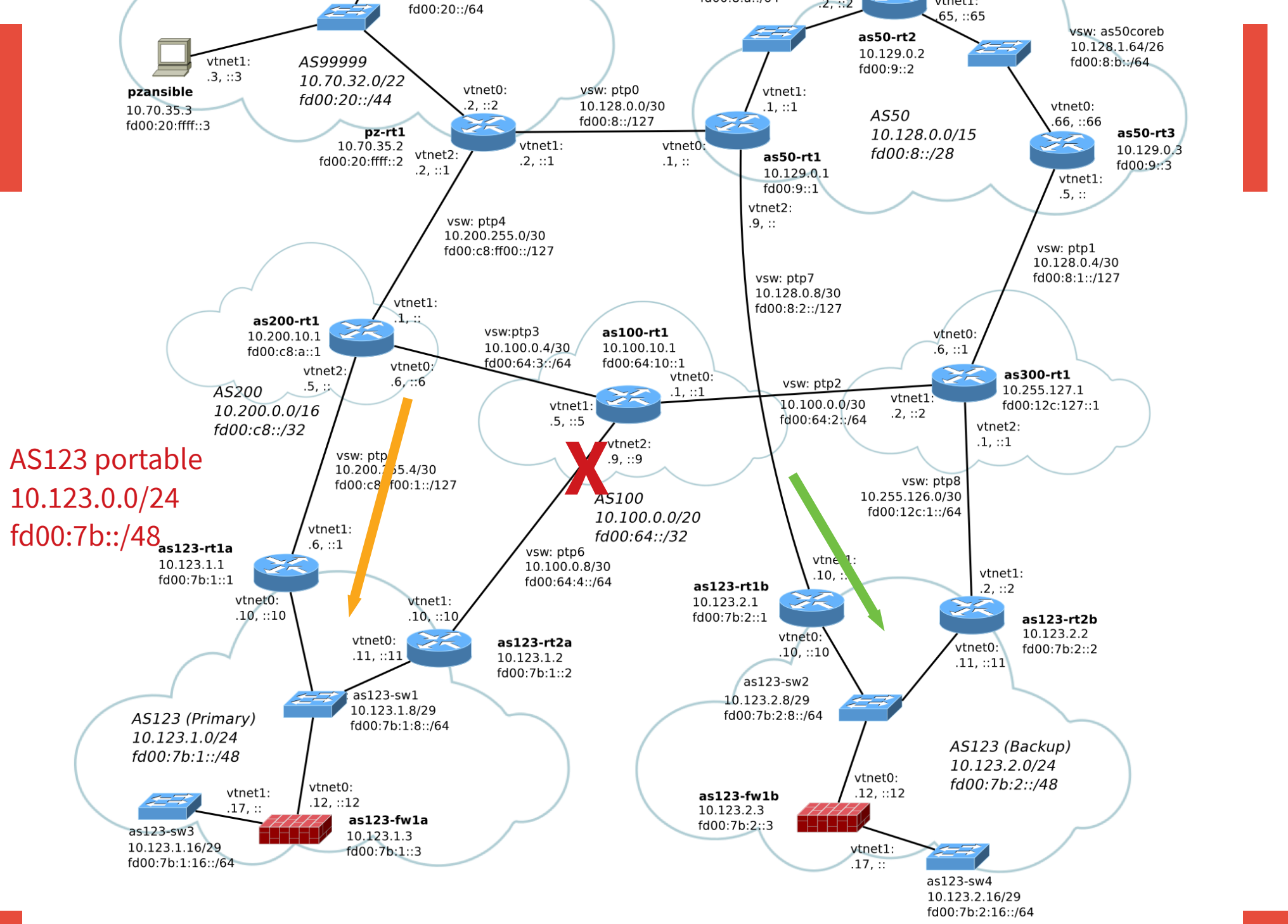
One Template to rule them all

- **Ansible uses Jinja2 templating language**
- **Inventory defines the configuration for all routers**
- **Further abstraction with groups minimizes data duplication.**
- **15 routers are deployed from 1 set of config templates.**

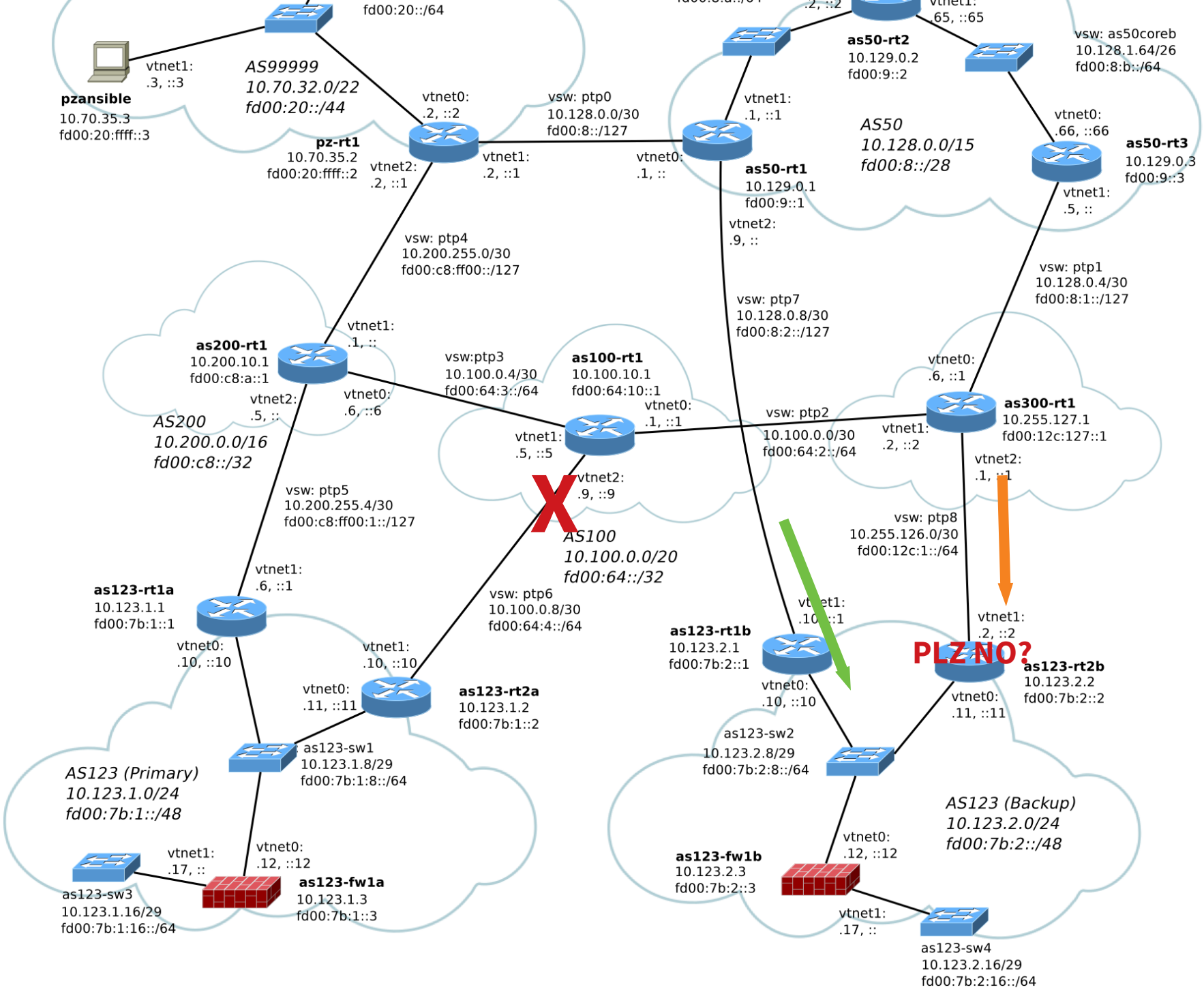
Enough theory. Let’s break stuff!


fd00:20::/64
vtnet1:
.3, ::3
pzansible
10.70.35.3
fd00:20:ffff::3
AS99999
10.70.32.0/22
fd00:20::/44
vtnet0:
.2, ::2
pz-rt1
10.70.35.2
fd00:20:ffff::2
vtnet2:
.2, ::1
vtnet1:
.2, ::1
vsw: ptp0
10.128.0.0/30
fd00:8::/127
vtnet0:
.1, ::
as50-rt1
10.129.0.1
fd00:9::1
vtnet2:
.9, ::
vtnet1:
.1, ::1
as50-rt2
10.129.0.2
fd00:9::2
vtnet1:
.65, ::65
vsw: as50coreb
10.128.1.64/26
fd00:8:b::/64
AS50
10.128.0.0/15
fd00:8::/28
vtnet0:
.66, ::66
as50-rt3
10.129.0.3
fd00:9::3
vtnet1:
.5, ::
vsw: ptp4
10.200.255.0/30
fd00:c8:ff00::/127
vsw: ptp7
10.128.0.8/30
fd00:8:2::/127
vsw: ptp1
10.128.0.4/30
fd00:8:1::/127
vtnet1:
.1, ::
as200-rt1
10.200.10.1
fd00:c8:a::1
vtnet2:
.5, ::
vtnet0:
.6, ::6
AS200
10.200.0.0/16
fd00:c8::/32
vsw:ptp3
10.100.0.4/30
fd00:64:3::/64
as100-rt1
10.100.10.1
fd00:64:10::1
vtnet1:
.5, ::5
vtnet0:
.1, ::1
vtnet2:
.9, ::9
X
AS100
10.100.0.0/20
fd00:64::/32
vsw: ptp2
10.100.0.0/30
fd00:64:2::/64
vtnet0:
.6, ::1
as300-rt1
10.255.127.1
fd00:12c:127::1
vtnet1:
.2, ::2
vtnet2:
.1, ::1
vsw: ptp8
10.255.126.0/30
fd00:12c:1::/64
vsw: ptp5
10.200.255.4/30
fd00:c8:ff00:1::/127
vtnet1:
.6, ::1
as123-rt1a
10.123.1.1
fd00:7b:1::1
vtnet0:
.10, ::10
vtnet1:
.10, ::10
vsw: ptp6
10.100.0.8/30
fd00:64:4::/64
vtnet0:
.11, ::11
as123-rt2a
10.123.1.2
fd00:7b:1::2
as123-sw1
10.123.1.8/29
fd00:7b:1:8::/64
AS123 (Primary)
10.123.1.0/24
fd00:7b:1::/48
vtnet1:
.17, ::
vtnet0:
.12, ::12
as123-sw3
10.123.1.16/29
fd00:7b:1:16::/64
as123-fw1a
10.123.1.3
fd00:7b:1::3
vtnet1:
.10, ::1
as123-rt1b
10.123.2.1
fd00:7b:2::1
vtnet0:
.10, ::10
vtnet1:
.2, ::2
as123-rt2b
10.123.2.2
fd00:7b:2::2
vtnet0:
.11, ::11
as123-sw2
10.123.2.8/29
fd00:7b:2:8::/64
AS123 (Backup)
10.123.2.0/24
fd00:7b:2::/48
vtnet0:
.12, ::12
as123-fw1b
10.123.2.3
fd00:7b:2::3
vtnet1:
.17, ::
as123-sw4
10.123.2.16/29
fd00:7b:2:16::/64

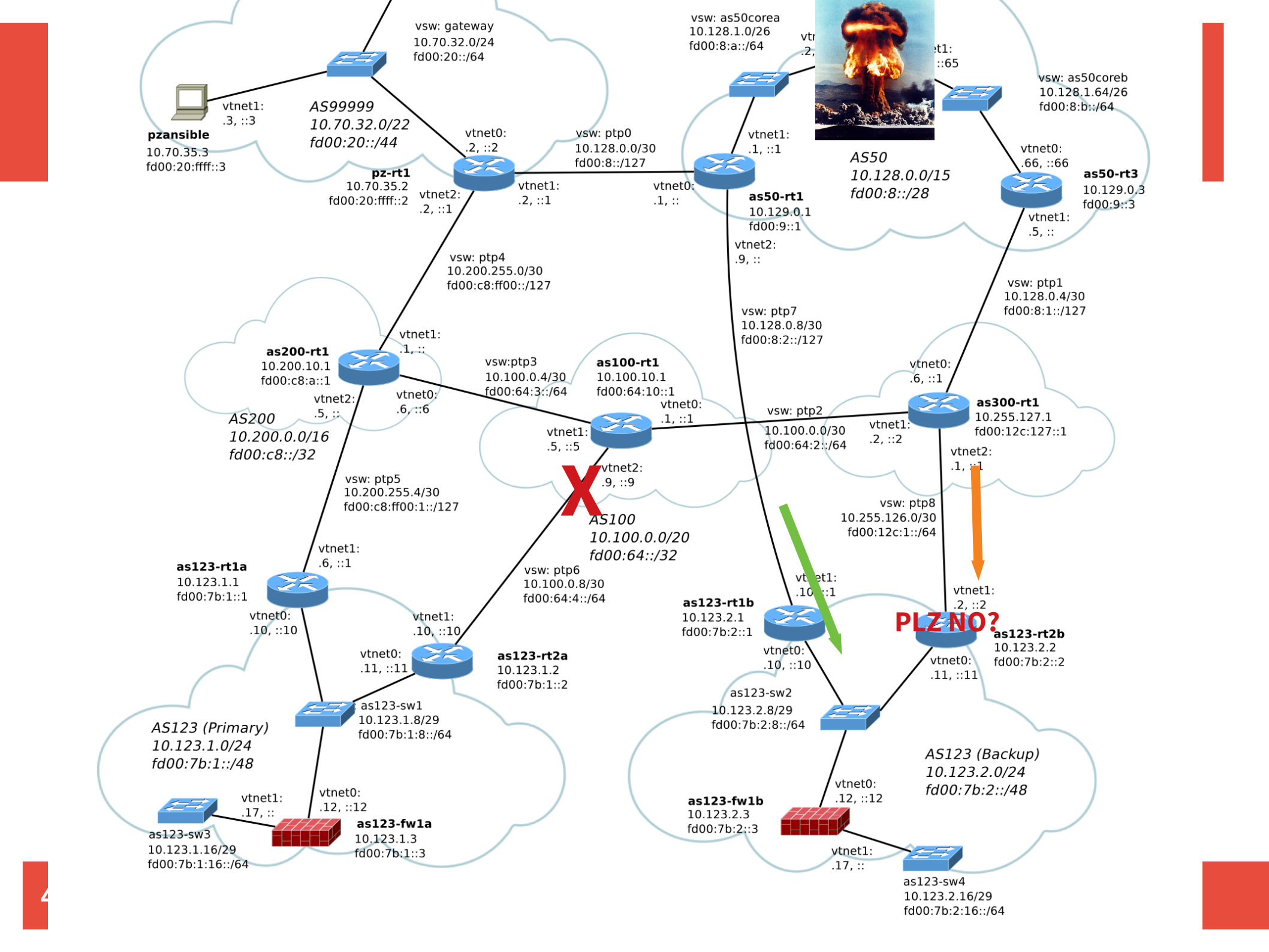
Site Failover


AS123 portable
10.123.0.0/24
fd00:7b::/48
pzansible
10.70.35.3
fd00:20:ffff::3
vtnet1:
.3, ::3
AS99999
10.70.32.0/22
fd00:20::/44
fd00:20::/64
pz-rt1
10.70.35.2
fd00:20:ffff::2
vtnet0:
.2, ::2
vtnet2:
.2, ::1
vtnet1:
.2, ::1
vsw: ptp0
10.128.0.0/30
fd00:8::/127
vtnet0:
.1, ::
as50-rt1
10.129.0.1
fd00:9::1
vtnet1:
.1, ::1
as50-rt2
10.129.0.2
fd00:9::2
vtnet1:
.65, ::65
vsw: as50coreb
10.128.1.64/26
fd00:8:b::/64
AS50
10.128.0.0/15
fd00:8::/28
vtnet0:
.66, ::66
as50-rt3
10.129.0.3
fd00:9::3
vtnet1:
.5, ::
vtnet2:
.9, ::
vsw: ptp4
10.200.255.0/30
fd00:c8:ff00::/127
vsw: ptp7
10.128.0.8/30
fd00:8:2::/127
vsw: ptp1
10.128.0.4/30
fd00:8:1::/127
as200-rt1
10.200.10.1
fd00:c8:a::1
vtnet1:
.1, ::
vtnet2:
.5, ::
vtnet0:
.6, ::6
AS200
10.200.0.0/16
fd00:c8::/32
vsw:ptp3
10.100.0.4/30
fd00:64:3::/64
as100-rt1
10.100.10.1
fd00:64:10::1
vtnet1:
.5, ::5
vtnet0:
.1, ::1
vtnet2:
.9, ::9
AS100
10.100.0.0/20
fd00:64::/32
vsw: ptp2
10.100.0.0/30
fd00:64:2::/64
vtnet0:
.6, ::1
as300-rt1
10.255.127.1
fd00:12c:127::1
vtnet1:
.2, ::2
vtnet2:
.1, ::1
vsw: ptp8
10.255.126.0/30
fd00:12c:1::/64
vsw: ptp
10.200.255.4/30
fd00:c8:ff00:1::/127
vtnet1:
.6, ::1
as123-rt1a
10.123.1.1
fd00:7b:1::1
vtnet0:
.10, ::10
vtnet1:
.10, ::10
vsw: ptp6
10.100.0.8/30
fd00:64:4::/64
vtnet0:
.11, ::11
as123-rt2a
10.123.1.2
fd00:7b:1::2
as123-sw1
10.123.1.8/29
fd00:7b:1:8::/64
AS123 (Primary)
10.123.1.0/24
fd00:7b:1::/48
vtnet1:
.17, ::
vtnet0:
.12, ::12
as123-sw3
10.123.1.16/29
fd00:7b:1:16::/64
as123-fw1a
10.123.1.3
fd00:7b:1::3
as123-rt1b
10.123.2.1
fd00:7b:2::1
vtnet1:
.10, ::
vtnet0:
.10, ::10
vtnet1:
.2, ::2
as123-rt2b
10.123.2.2
fd00:7b:2::2
vtnet0:
.11, ::11
as123-sw2
10.123.2.8/29
fd00:7b:2:8::/64
AS123 (Backup)
10.123.2.0/24
fd00:7b:2::/48
vtnet0:
.12, ::12
as123-fw1b
10.123.2.3
fd00:7b:2::3
vtnet1:
.17, ::
as123-sw4
10.123.2.16/29
fd00:7b:2:16::/64

Deprioritizing Routers


pzansible
10.70.35.3
fd00:20:ffff::3
vtnet1:
.3, ::3
AS99999
10.70.32.0/22
fd00:20::/44
fd00:20::/64
pz-rt1
10.70.35.2
fd00:20:ffff::2
vtnet0:
.2, ::2
vtnet2:
.2, ::1
vtnet1:
.2, ::1
vsw: ptp0
10.128.0.0/30
fd00:8::/127
vtnet0:
.1, ::
vtnet1:
.1, ::1
as50-rt1
10.129.0.1
fd00:9::1
as50-rt2
10.129.0.2
fd00:9::2
vtnet1:
.65, ::65
vsw: as50coreb
10.128.1.64/26
fd00:8:b::/64
AS50
10.128.0.0/15
fd00:8::/28
vtnet0:
.66, ::66
as50-rt3
10.129.0.3
fd00:9::3
vtnet1:
.5, ::
vtnet2:
.9, ::
vsw: ptp4
10.200.255.0/30
fd00:c8:ff00::/127
vsw: ptp7
10.128.0.8/30
fd00:8:2::/127
vsw: ptp1
10.128.0.4/30
fd00:8:1::/127
as200-rt1
10.200.10.1
fd00:c8:a::1
vtnet1:
.1, ::
vtnet2:
.5, ::
vtnet0:
.6, ::6
AS200
10.200.0.0/16
fd00:c8::/32
vsw:ptp3
10.100.0.4/30
fd00:64:3::/64
as100-rt1
10.100.10.1
fd00:64:10::1
vtnet1:
.5, ::5
vtnet0:
.1, ::1
vtnet2:
.9, ::9
AS100
10.100.0.0/20
fd00:64::/32
vsw: ptp2
10.100.0.0/30
fd00:64:2::/64
vtnet0:
.6, ::1
as300-rt1
10.255.127.1
fd00:12c:127::1
vtnet1:
.2, ::2
vtnet2:
.1, ::1
vsw: ptp8
10.255.126.0/30
fd00:12c:1::/64
vsw: ptp5
10.200.255.4/30
fd00:c8:ff00:1::/127
vsw: ptp6
10.100.0.8/30
fd00:64:4::/64
as123-rt1a
10.123.1.1
fd00:7b:1::1
vtnet1:
.6, ::1
vtnet0:
.10, ::10
vtnet1:
.10, ::10
vtnet0:
.11, ::11
as123-rt2a
10.123.1.2
fd00:7b:1::2
as123-sw1
10.123.1.8/29
fd00:7b:1:8::/64
AS123 (Primary)
10.123.1.0/24
fd00:7b:1::/48
as123-sw3
10.123.1.16/29
fd00:7b:1:16::/64
vtnet1:
.17, ::
vtnet0:
.12, ::12
as123-fw1a
10.123.1.3
fd00:7b:1::3
as123-rt1b
10.123.2.1
fd00:7b:2::1
vtnet1:
.10, ::1
vtnet0:
.10, ::10
PLZ NO?
as123-rt2b
10.123.2.2
fd00:7b:2::2
vtnet1:
.2, ::2
vtnet0:
.11, ::11
as123-sw2
10.123.2.8/29
fd00:7b:2:8::/64
AS123 (Backup)
10.123.2.0/24
fd00:7b:2::/48
as123-fw1b
10.123.2.3
fd00:7b:2::3
vtnet0:
.12, ::12
vtnet1:
.17, ::
as123-sw4
10.123.2.16/29
fd00:7b:2:16::/64

$mischief++;


vsw: gateway
10.70.32.0/24
fd00:20::/64
vtnet1:
.3, ::3
pzansible
10.70.35.3
fd00:20:ffff::3
AS99999
10.70.32.0/22
fd00:20::/44
vtnet0:
.2, ::2
pz-rt1
10.70.35.2
fd00:20:ffff::2
vtnet2:
.2, ::1
vtnet1:
.2, ::1
vsw: ptp0
10.128.0.0/30
fd00:8::/127
vtnet0:
.1, ::
vsw: as50corea
10.128.1.0/26
fd00:8:a::/64
vtnet1:
.1, ::1
as50-rt1
10.129.0.1
fd00:9::1
vtnet2:
.9, ::
AS50
10.128.0.0/15
fd00:8::/28
vsw: as50coreb
10.128.1.64/26
fd00:8:b::/64
vtnet0:
.66, ::66
as50-rt3
10.129.0.3
fd00:9::3
vtnet1:
.5, ::
vsw: ptp1
10.128.0.4/30
fd00:8:1::/127
vsw: ptp4
10.200.255.0/30
fd00:c8:ff00::/127
vsw: ptp7
10.128.0.8/30
fd00:8:2::/127
as200-rt1
10.200.10.1
fd00:c8:a::1
vtnet1:
.1, ::
vtnet2:
.5, ::
vtnet0:
.6, ::6
AS200
10.200.0.0/16
fd00:c8::/32
vsw:ptp3
10.100.0.4/30
fd00:64:3::/64
as100-rt1
10.100.10.1
fd00:64:10::1
vtnet0:
.1, ::1
vtnet1:
.5, ::5
vtnet2:
.9, ::9
X
AS100
10.100.0.0/20
fd00:64::/32
vsw: ptp2
10.100.0.0/30
fd00:64:2::/64
vtnet0:
.6, ::1
as300-rt1
10.255.127.1
fd00:12c:127::1
vtnet1:
.2, ::2
vtnet2:
.1, ::1
vsw: ptp8
10.255.126.0/30
fd00:12c:1::/64
vsw: ptp5
10.200.255.4/30
fd00:c8:ff00:1::/127
vtnet1:
.6, ::1
as123-rt1a
10.123.1.1
fd00:7b:1::1
vtnet0:
.10, ::10
vtnet1:
.10, ::10
vsw: ptp6
10.100.0.8/30
fd00:64:4::/64
vtnet0:
.11, ::11
as123-rt2a
10.123.1.2
fd00:7b:1::2
as123-sw1
10.123.1.8/29
fd00:7b:1:8::/64
AS123 (Primary)
10.123.1.0/24
fd00:7b:1::/48
vtnet1:
.17, ::
vtnet0:
.12, ::12
as123-sw3
10.123.1.16/29
fd00:7b:1:16::/64
as123-fw1a
10.123.1.3
fd00:7b:1::3
as123-rt1b
10.123.2.1
fd00:7b:2::1
vtnet1:
.10, ::1
vtnet0:
.10, ::10
PLZ NO?
vtnet1:
.2, ::2
vtnet0:
.11, ::11
as123-rt2b
10.123.2.2
fd00:7b:2::2
as123-sw2
10.123.2.8/29
fd00:7b:2:8::/64
AS123 (Backup)
10.123.2.0/24
fd00:7b:2::/48
vtnet0:
.12, ::12
as123-fw1b
10.123.2.3
fd00:7b:2::3
vtnet1:
.17, ::
as123-sw4
10.123.2.16/29
fd00:7b:2:16::/64

Thank You!

“...And again, the Internet is not something that you just dump something on. It's not a big truck. It's a series of tubes.”

- Sen. Ted Stevens

github.com/tom-clx/bsdcan18-doc

github.com/tom-clx/bsdcan18-ansible

github.com/tom-clx/bsdcan18-config