DisNOG 2

The Network is Down: Fire Drills

Jonathan Kartes

Hello!

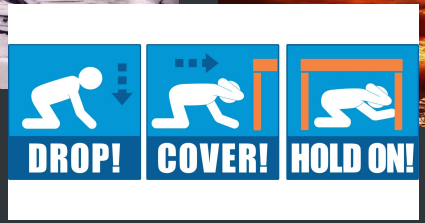
I am Jonathan Kartes

Technical Solutions Engineer (TAC) @ Arista Networks

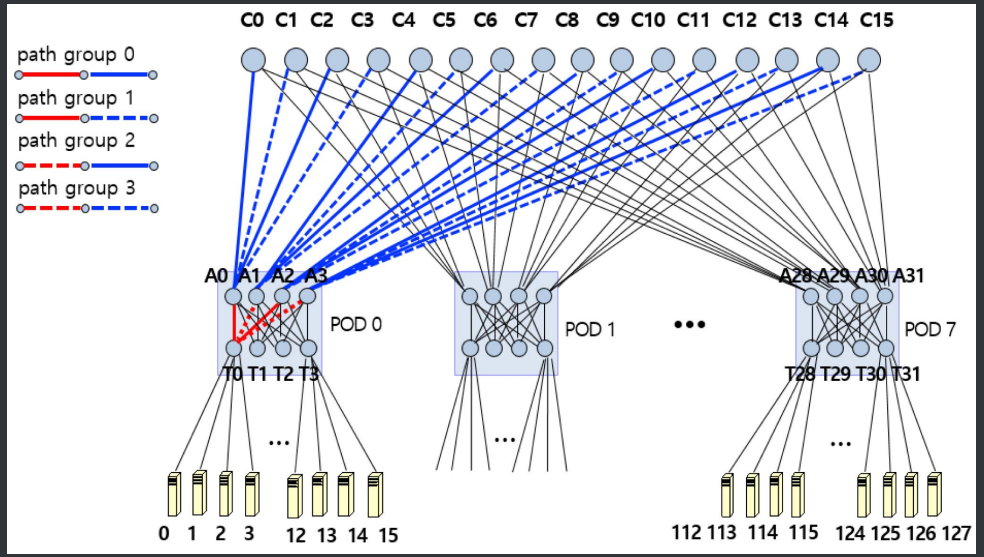
 Kvotthe#6441 (Jon)

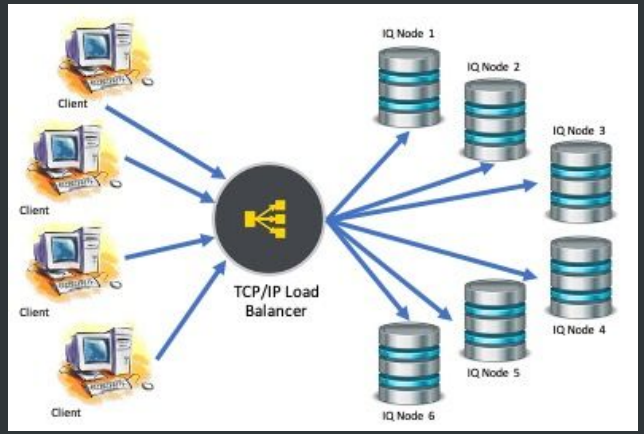
 JonathanKartes

Your business/company has drills for all sorts of emergencies, but what about your network?



Companies spend thousands, often tens and hundreds of thousands, to get N+X redundancy to prevent outages

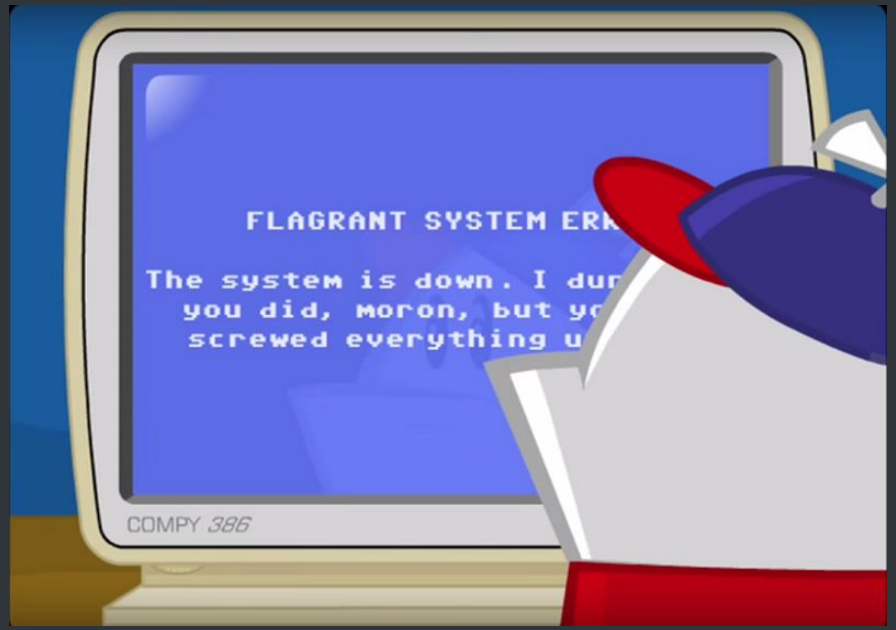




1

Have a plan - build a strong Problem Statement

- The network shouldn't just be down. Applications shouldn't just fail
  - Problem statements are how we quickly communicate what exactly is wrong to a third party. Make them count!



What is a good problem statement?

- Hosts in Vlan603, connected behind SwitchA in the NY-DC, are experiencing connectivity issues to hosts in other vlans.
  - Communication within Vlan603 is not impacted.
  - Communication between Vlan603 and all other Vlans is always impacted
  - Vlan603 is deployed in the following DCs
    - A
    - B
    - C
- The following tests have been performed
  - Ping from X to Y in DC A with 100% success
  - Ping from X to Z in DC NY with 100% failure

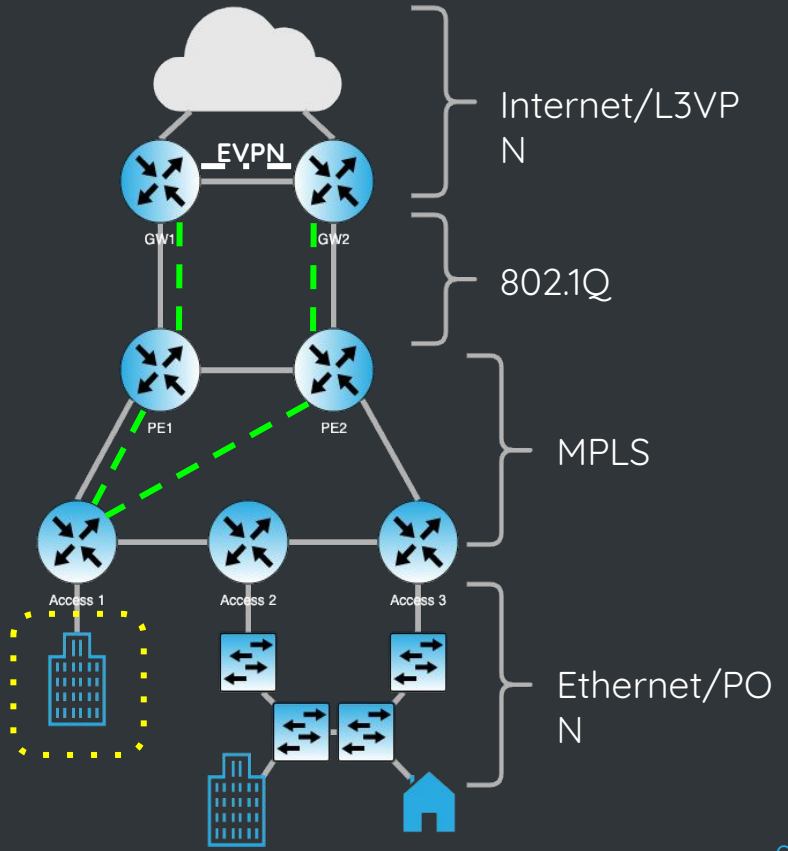
Problem Statements can and should evolve as new data is learned

The Basics - The stuff anyone can do

- What exactly is the problem?
  - Host is down
  - Application is down
  - Peering with XYZ is down
- When did it start?
  - How specific can we get? Day/Hour/Minute?
- Were there any changes?
  - Application changes are changes too
- Have we had this problem before?

The Technical - What can we do in advance?

- How is this supposed to work?
  - Topology with expected traffic flow
- Isolating the issue
  - Every test should serve a purpose. What tests prove out what failures?
- Who do we need to engage?
  - What information can we provide proactively?



What baseline of data should be gathered?

- Tech support files
- Telemetry
- Process logs/traces

- Can an alias be created that gathers and zips data into one file?
  - #alias getlogs bash cat /var/log/stuff/* >/mnt/flash/stuff.log
- Rough timeline of Outage/Troubleshooting steps

What comes after resolution?

By the time the Outage has been resolved, you should have:

- A strong problem-statement
  - Symptoms (what failed)
- Relevant data that was gathered
  - Troubleshooting steps
- History of the issue
  - First time or repeat occurrence
- Timeline of the Outage

Use this data!

- Improve upon existing processes/documentation
  - Where were the biggest gaps in the Outage
- How can the issue be avoided in the future?

4

Other Considerations

Small teams are more efficient than large groups

- There’s no one-size-fits all for conference calls, but I recommend 15 or fewer active individuals.
  - My best calls are typically resolved by 3-4 people and have no-more than 8 in the call.

- When everyone is responsible, no-one is responsible. Make sure there is an individual leading the debugging efforts *especially* when there’s multiple vendors involved.

- If there are multiple leads to chase, consider breaking into smaller groups

What if network fire-drills were a thing?

If there's a plan with guidelines and expectations, why not actually run drills?

- It builds familiarity with a deployment across multiple teams
- You can proactively see who takes initiative
- Finds gaps in documentation/processes
- Great opportunity to proving coaching
- Reduce downtime

If we're spending $thousands$ to maximize redundancy and network uptime. Let's spend 20 minutes writing a process to minimize downtime

Thank you!

Any questions?

 Kvotthe#6441 (Jon)

 JonathanKartes