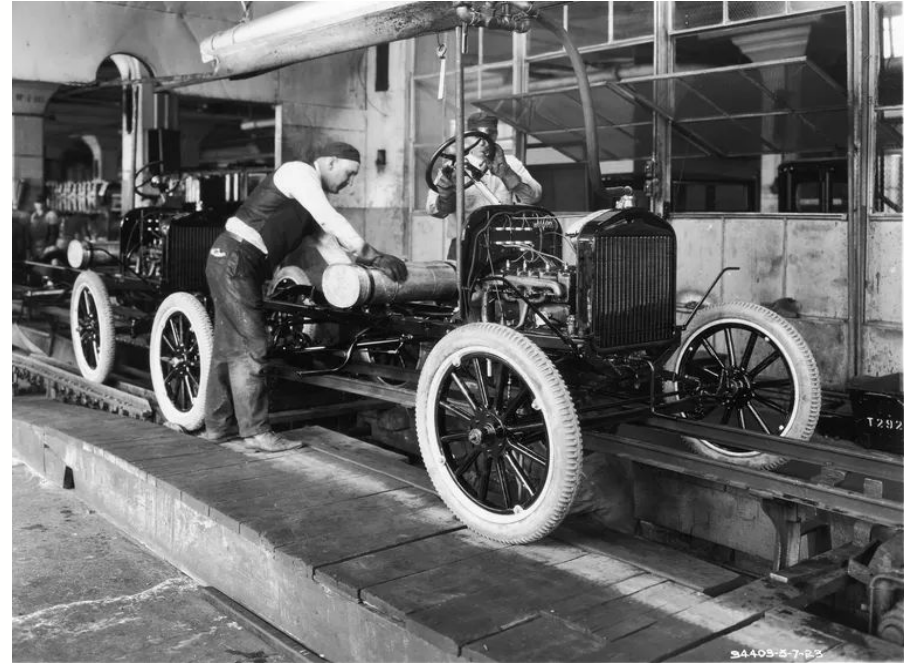ARISTA

Risky Deployments, human errors, downtime?
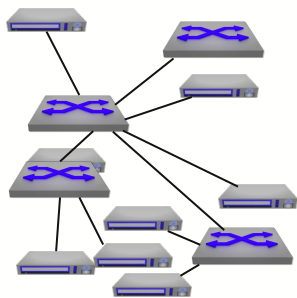
Let's talk "Infrastructure as Code"

Claudiu Captari, Systems Engineering Manager ANZ, Arista Networks

A story of mass production … The assembly line

ARISTA

100 years later… Typical Network

Complex Designs

CLI
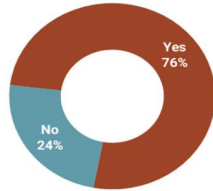
Manual changes
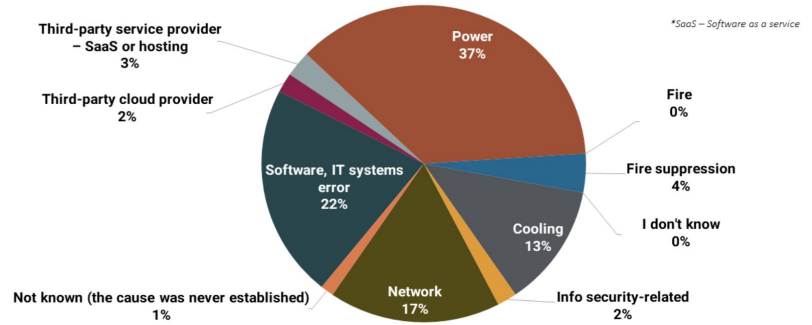
Pray it works

No visibility

ARISTA

Has your organization had any IT service outage over the past 3 years?

Yes 76%

No 24%

Source: Uptime Institute Data Center Resiliency Survey 2021

Uptime Institute | INTELLIGENCE

*SaaS — Software as a service

Power 37%

Third-party service provider – SaaS or hosting 3%

Third-party cloud provider 2%

Software, IT systems error 22%

Fire 0%

Fire suppression 4%

I don't know 0%

Cooling 13%

Not known (the cause was never established) 1%

Network 17%

Info security-related 2%

What was the primary cause of your organization's most recent significant incident or outage? Choose one.

Source: Uptime Institute Global Survey of IT and Data Center Managers 2020 (n=152)

Uptime Institute | INTELLIGENCE

Configuration/change management failure	58%
Third-party network provider failure	45%
Firmware/software error	36%
Natural disaster (e.g., public works, line breakages)	31%
Network overload/congestion error	28%
Corrupted firewall rules/routing table issues	20%
Malicious cyber-attack (e.g., DDoS, network partitioning)*	18%
Other networking-related failure	1%

*DDoS — Distributed denial of service

What were the most common root causes for the networking-related IT outages at your organization's data centers over the past three years? (Choose no more than three)
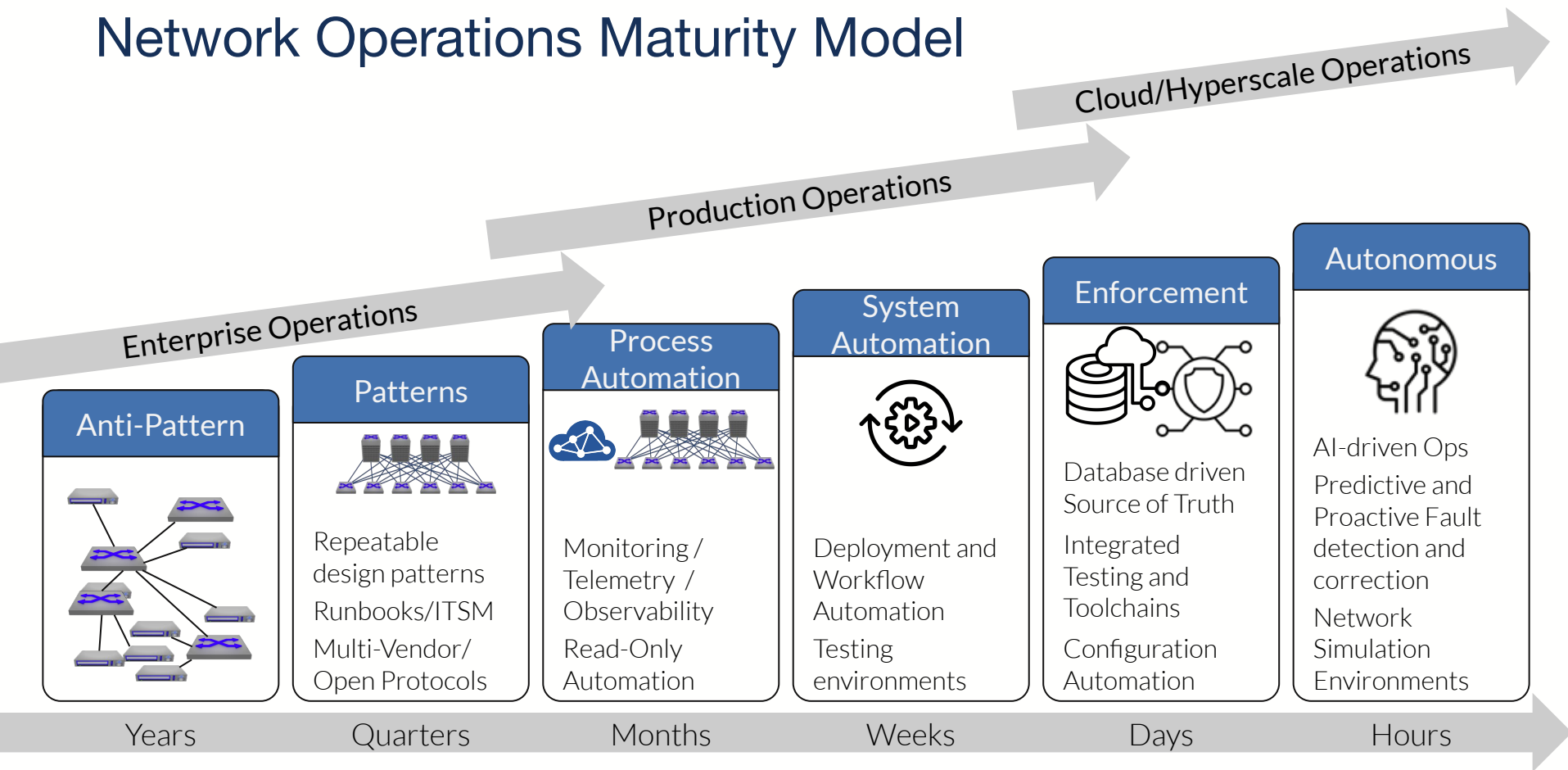
Source: Uptime Institute Data Center Resiliency Survey 2021 (n=130)

Uptime Institute | INTELLIGENCE

	2019 (n=420)	2020 (n=146)
Over $1 million	10%	16%
$100,000 - $1 million	28%	40%
Under $100,000	60%	45%
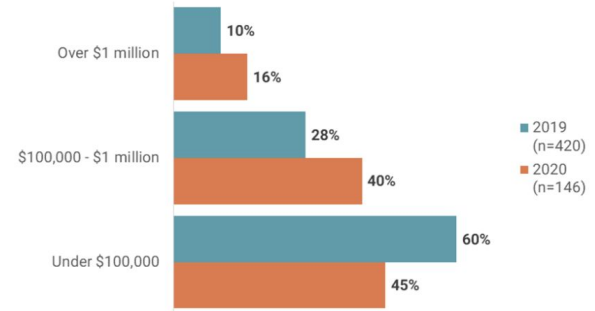
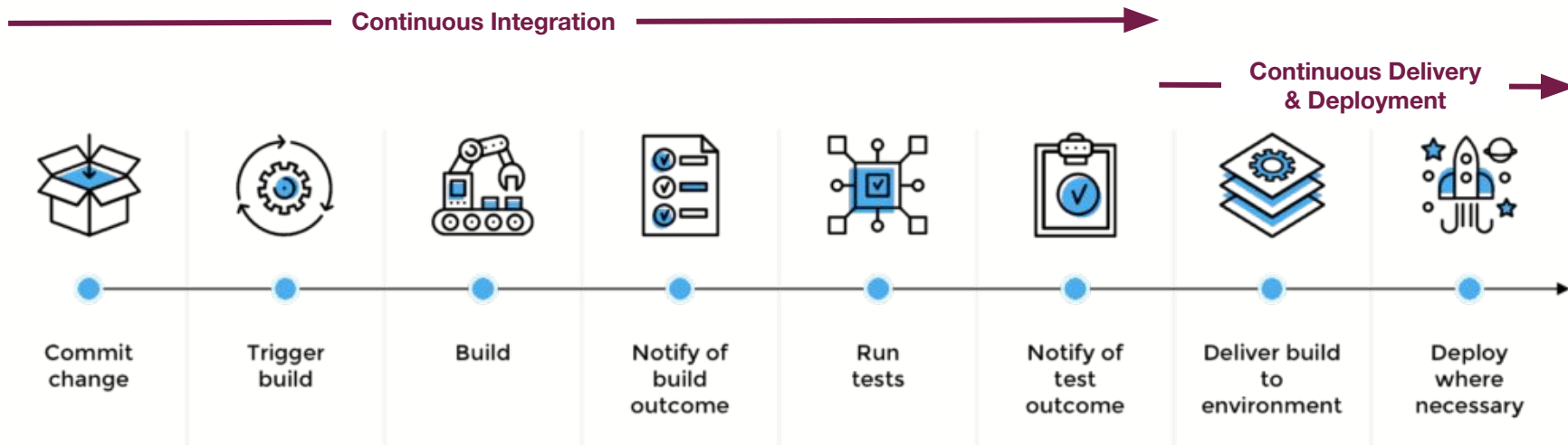**Please estimate the total cost of this downtime incident (from outage to full recovery) for your organization, including direct, opportunity and reputation costs using the following options. Choose one.***

* All figures in US dollars

Source: Uptime Institute Global Survey of IT and Data Center Managers 2019 and 2020

Uptime Institute | INTELLIGENCE

* Annual Outage Analysis 2021 - Uptime Institute

Network Operations Maturity Model

Cloud/Hyperscale Operations

Production Operations

Enterprise Operations

Anti-Pattern

Patterns

Repeatable design patterns

Runbooks/ITSM

Multi-Vendor/ Open Protocols

Process Automation

Monitoring / Telemetry / Observability

Read-Only Automation

System Automation

Deployment and Workflow Automation

Testing environments

Enforcement

Database driven Source of Truth

Integrated Testing and Toolchains

Configuration Automation

Autonomous

AI-driven Ops

Predictive and Proactive Fault detection and correction

Network Simulation Environments

Years	Quarters	Months	Weeks	Days	Hours

ARISTA

CI/CD - A Software Assembly Line

Continuous Integration

**Continuous Delivery
& Deployment**

| Commit change | Trigger build | Build | Notify of build outcome | Run tests | Notify of test outcome | Deliver build to environment | Deploy where necessary |

A modern CI/CD pipeline is a software 'assembly line'. It enables 1000s of developers to work together to deliver software handling change control, merge conflicts, testing, and deployment. It is a *Tool and Technology* to help develop quality software.

CI/CD = Continuous Integration / Continuous Delivery

ARISTA

Obstacles to adoption
Don't get sucked into the conveyor belt

- Culture
  - DevOps is just as much about culture and process as it is about tools and technology
  - Resistance to change
  - "People think that DevOps is an org structure. To me, it's not. It is a mindset." - VP of Software Engineering and Data Science, Insurance

- Skills gap
  - Many network engineers are not comfortable with DevOps tooling

- Lack of proper tools and architectures
  - Lackluster APIs, legacy architectures
  - Tools not suited for network
  - Too many choices, fragmented

- Difficult to get started
  - "Boiling the ocean"
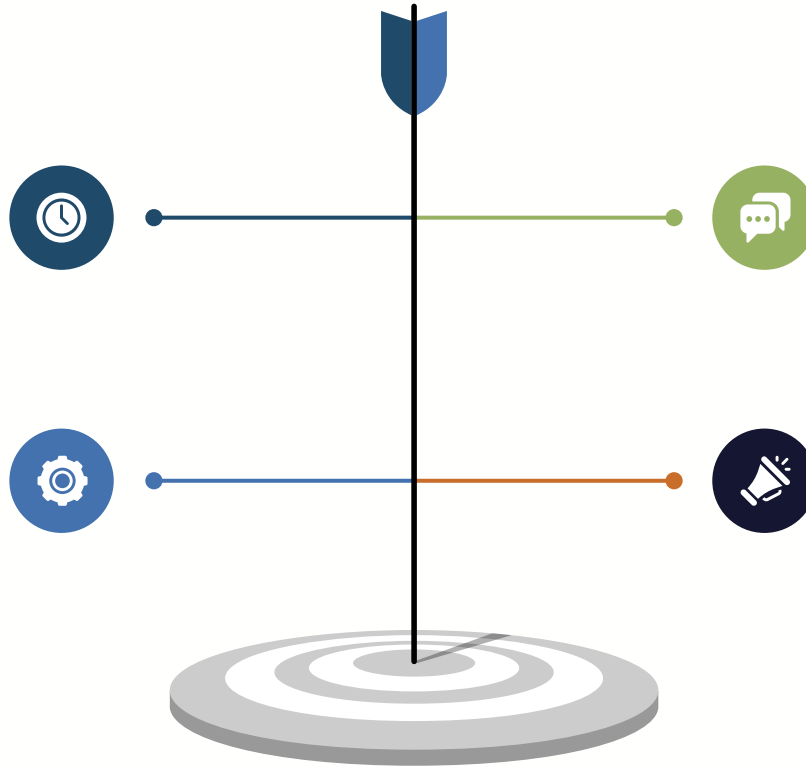 - Not sure where to begin
  - Lack of confidence

Can we not apply CI/CD principles to networks?

Infra-as-Code

Treat network changes and deployments as software - apply SDLC models to infrastructure

Source of Truth

Deploy and derive configurations from a flexible authoritative source of truth

Unit/System Testing

Buy-down risk by executing extensive pre and post-deployment testing and validation

Configuration Patterns

Enable large-scale pattern-based deployments. Manage network(s) as version-controlled entities.
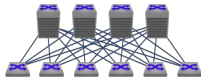
Things to work towards

- Smaller more frequent changes are easier to rationalize and have a smaller blast radius
  - Reduce friction to deployment, the more friction, the greater the tendency to batch changes
- Having good guard rails builds confidence, and allows for smaller approval chains
  - Block invalid operations, like deploying directly to prod without testing
  - Pre and post validation
  - Strong monitoring, change control, and logging
- Transparency encourages collaboration between teams and gives users the information they need to make change decisions

Where do you even get started?

ARISTA

CI Pipeline - An opinionated Architecture

Vendor Validated Designs		**Leaf Spine EVPN**	**L3 Fabrics**	**L2 Fabrics**	**Border Leaf**	**Campus MDF/IDF**

Development / Authoring / Review	Change Mgmt / Orchestration	Pre-Deployment Testing	Release / Deploy	Operate / Observe
Intelligent tools to reduce input errors.

Peer review, 'quorum or M-of-N' rules for high risk changes. | Create, Submit, Approve Change Request.

Workflow orchestrator to run the pipeline. | Test and evaluate changes before putting them into production.

De-risk deployments | Rollout changes with knowledge of network state and topology. Apply this to inform and guide deployment. | Validate changes had desired effect. Monitor and observe to develop future changes. |
| Code Editor

Declarative Provisioning

Vendor config authoring | Change Management

Code repos and workflow pipeline | Network modeling and config testing

Virtual twin | Multi-vendor configuration deployment

Vendor configuration deployment | Packet capture and monitoring

Telemetry and analysis for closed-loop change control |

Source of Truth / Self Documenting		

Solution Overview

ARISTA
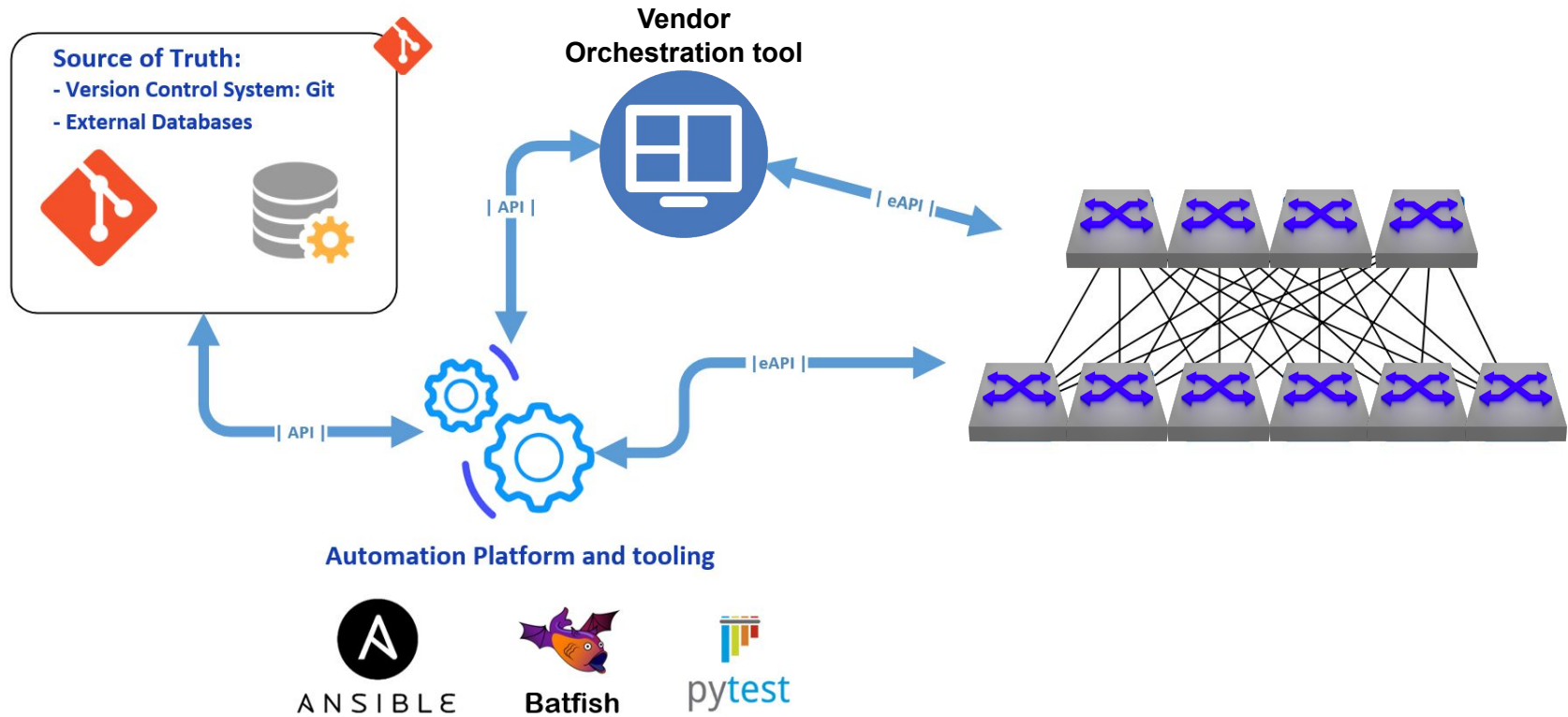
CI/CD Pipeline Demo

Batfish

An open source network configuration analysis tool

- Originally developed at Microsoft Research, UCLA, and USC

- finds errors and guarantees the correctness of planned or current network configurations

- enables network evolution, without the fear of outages or security breaches

- Builds a vendor neutral model from the imported configurations.
  The model can be queried using a series of programmatic questions

- **Example queries:**
  - Is IP address assignment correct ?
  - Are BGP peerings established ?
  - Are routes present in the routing table ?
  - does the policy permit routes to be announced ?
  - would a flow be allowed through an ACL ?

Example Batfish Query

Is the DNS server reachable?

```
In [6]:   # Check if hosts in the subnet can reach the DNS server
          dns_flow = HeaderConstraints(srcIps="10.10.10.1", # representative
                                       dstIps="218.8.104.58",
                                       applications=["dns"])
          answer = bfq.testfilters(headers=dns_flow,
                                   nodes="rtr-with-acl",
                                   filters="acl_in").answer()
          display_html(answer.frame())
```

	Node	Filter_Name	Flow	Action	Line_Content	Trace
0	rtr-with-acl	acl_in	start=rtr-with-acl [10.10.10.1:49152->218.8.104.58:53 UDP]	PERMIT	660 permit udp 10.10.10.0/24 218.8.104.58/32 eq domain	Flow permitted by 'extended ipv4 access-list' named 'acl_in', index 32: 660 permit udp 10.10.10.0/24 218.8.104.58/32 eq domain

ARISTA

Summary

Time to take back control of the service quality

Take the first steps

- Build the DevOps culture

- Experiment in a safe environment

- Get familiar with the toolset

And finally, adopt the new paradigms

ARISTA

Thank You

arista.com