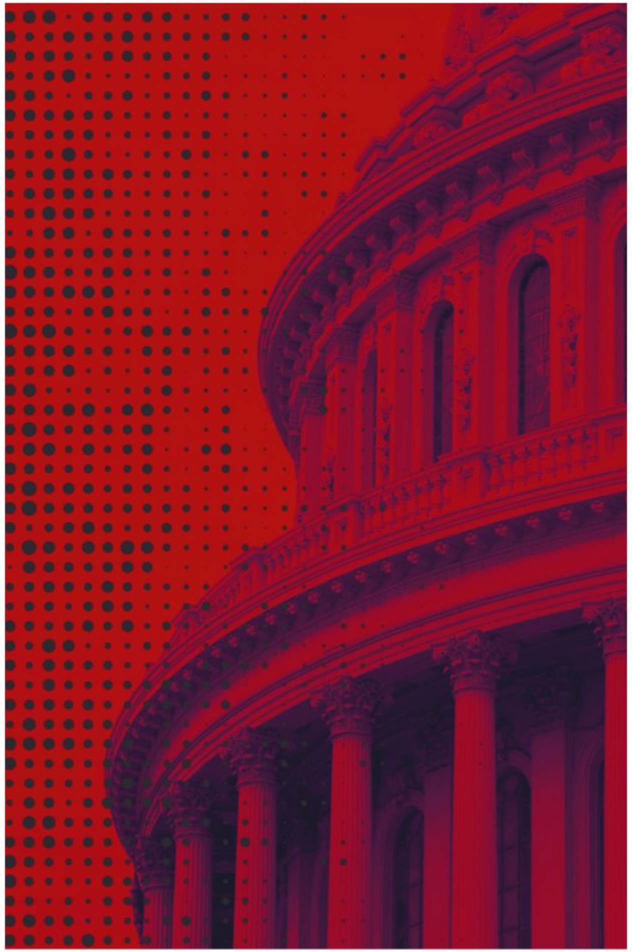


DevNation Federal 2021:

Learn More. Code More. Share more. Join the Nation.

Service Mesh Workshop

Instructors:

Jason Dudash, Principal Solutions Architect, Red Hat (@dudashtweets)

Chris Kang, Staff Solutions Architect, Red Hat (@theckang)

June 14th, 2021

DEVNATION FEDERAL

Who are we?

Dudash

Focused on emerging technology & modern applications & innovation. Specifically how to do those things with open source software

@dudashtweets

github.com/dudash

Chris

Focused on cloud native development and AI/ML best practices using open source software for government

@theckang

github.com/theckang

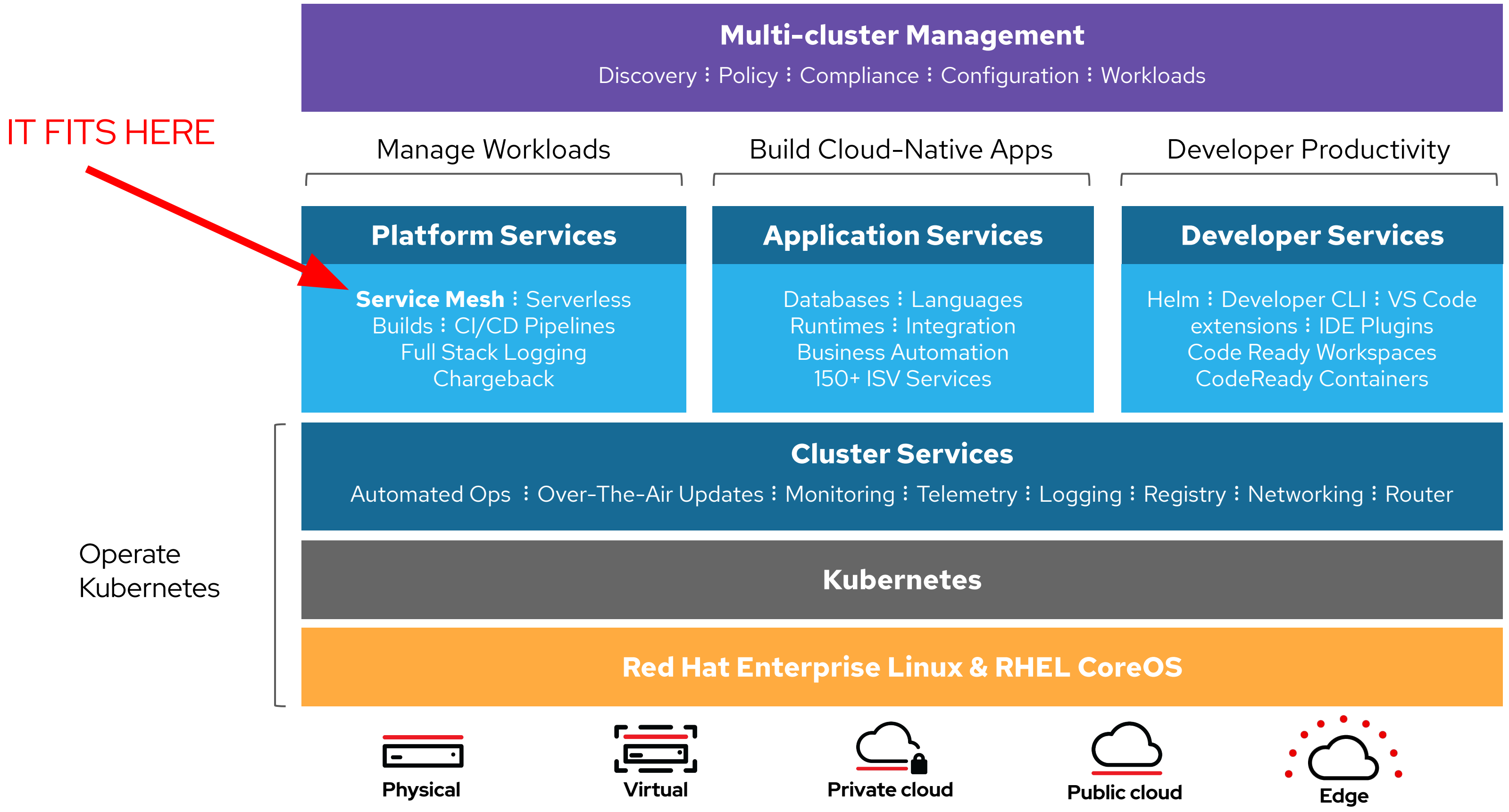
DEVNATION FEDERAL

Agenda

What's the plan for today

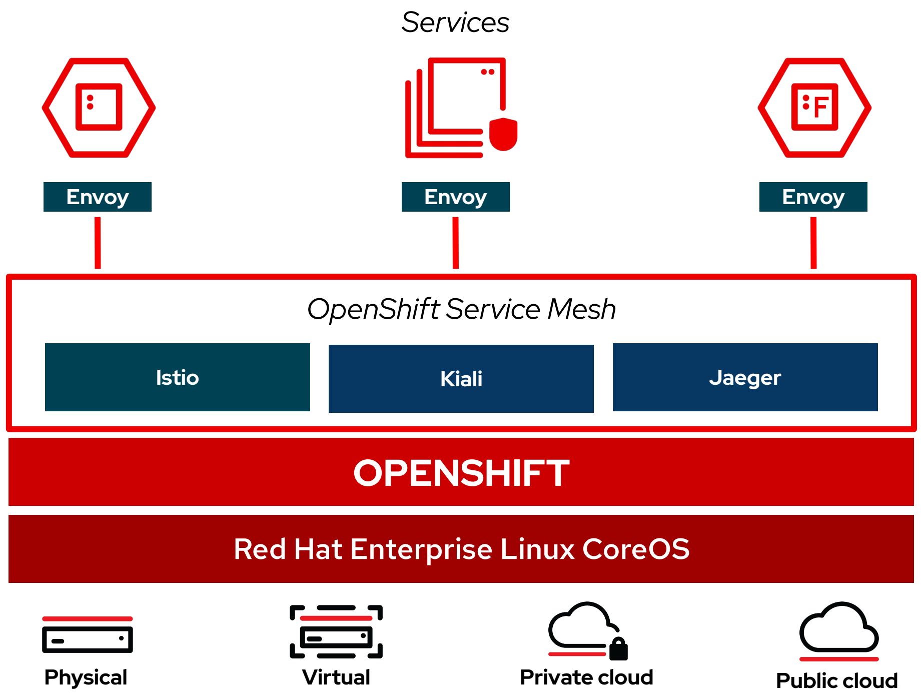
- Intros & Overview
- Part 1
  - Kickoff
  - Hands On Labs
  - Recap Poll and Breakouts
- Break
- Part 2
  - Kickoff
  - Hands On Labs
  - Poll and Concluding Discussion

Are you familiar with containers, Kubernetes, and OpenShift?


IT FITS HERE
Multi-cluster Management
Discovery ⁝ Policy ⁝ Compliance ⁝ Configuration ⁝ Workloads
Manage Workloads
Build Cloud-Native Apps
Developer Productivity
Platform Services
Service Mesh ⁝ Serverless
Builds ⁝ CI/CD Pipelines
Full Stack Logging
Chargeback
Application Services
Databases ⁝ Languages
Runtimes ⁝ Integration
Business Automation
150+ ISV Services
Developer Services
Helm ⁝ Developer CLI ⁝ VS Code extensions ⁝ IDE Plugins
Code Ready Workspaces
CodeReady Containers
Cluster Services
Automated Ops ⁝ Over-The-Air Updates ⁝ Monitoring ⁝ Telemetry ⁝ Logging ⁝ Registry ⁝ Networking ⁝ Router
Operate Kubernetes
Kubernetes
Red Hat Enterprise Linux & RHEL CoreOS
Physical
Virtual
Private cloud
Public cloud
Edge

OpenShift Service Mesh

- A software infrastructure layer between Kubernetes and your services for managing communications
- **Handles common “microservice” challenges, so that developers don’t have to:**
  - Encryption, Authn, Authz
  - Monitoring & Observability
  - Application Resilience
  - Upgrades, Rollouts & A/B Testing
  - Network Traffic Management
  - And more...



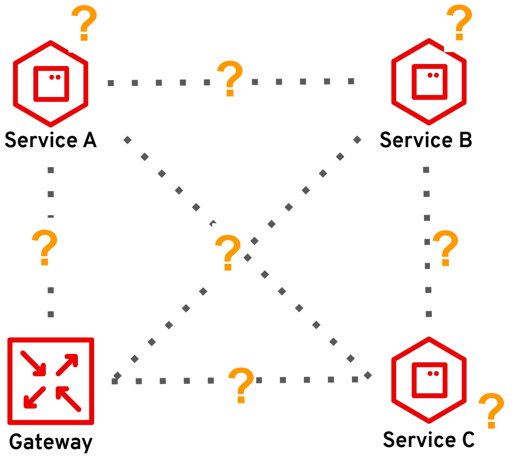
Why Service Mesh?

Because distributed systems are hard

The fallacies of distributed computing

This is was established well before microservices

- These challenges are a result of the fallacies of distributed computing:
    - The network is reliable.
    - Latency is zero.
    - Bandwidth is infinite.
    - The network is secure.
    - Topology doesn't change.
    - There is one administrator.
    - Transport cost is zero.
    - The network is homogeneous.

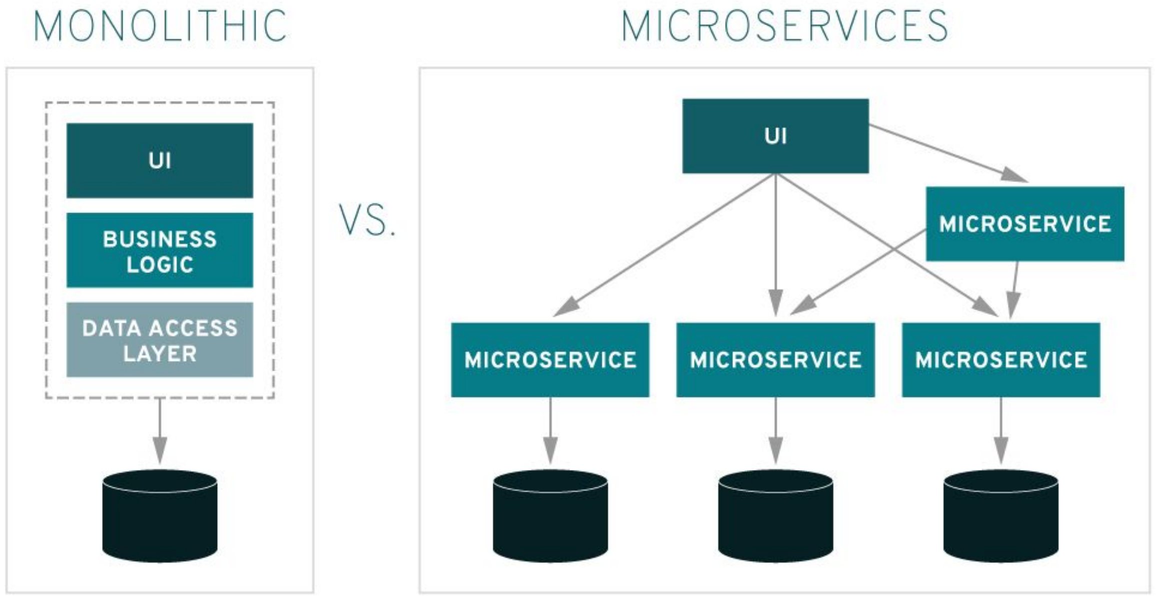


Microservices?

A quick primer

What are Microservices?

- Single purpose
- Independently deployable
- Typically bounded context to a biz domain
- Owned by a small team
- Often stateless



Benefits of Microservices

Agility
Deliver updates faster and react faster to new business demands

Highly scalable
Scale independently to meet temporary traffic increases, complete batch processing, or other business needs

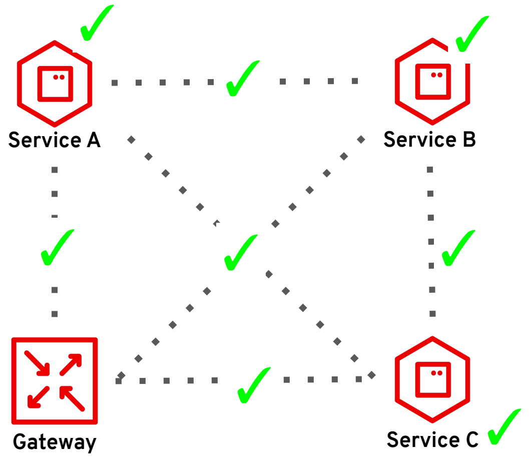
Can be purpose-built
Use the languages and frameworks best suited for the service's domain

Resilience
Improved fault isolation restricts service issues, such as memory leaks or open database connections, to only affect that specific service

So... What do we mean by “hard”?

Consider what **development** looks like

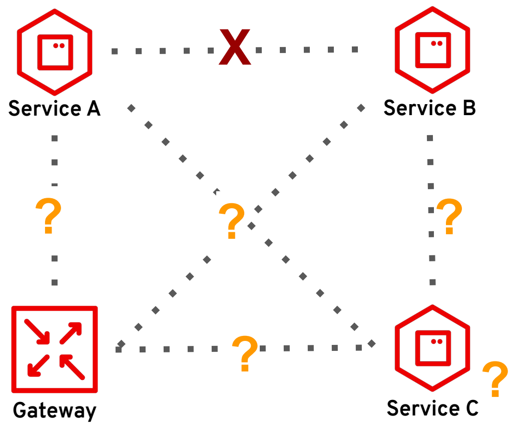
- A common process when developing microservices
- In Development:
  - New services are written
  - They are tested locally - looks good!
  - The are tested in a staging cluster - looks good!
- LGTM, Ship it!



Now that the microservices are in **production**

- In production, things become less predictable:
  - Sporadic delays and failures are seen
  - Performance is not as expected
  - Security holes may be discovered
  - Services are scaled, but performance doesn't improve
  - Fixes are made, but upgrades cause further issues

- **No clear way to troubleshooting the distributed system!**

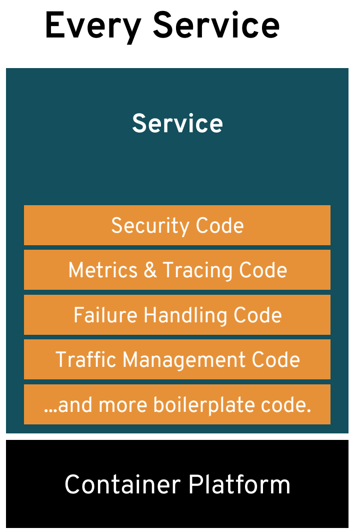


You could include code in each service to troubleshoot

- These challenges are often mitigated with:
  - Code to handle failures between services
  - Logs, metrics and traces in source code
  - 3rd party libraries for managing deployments, security and more

- A wide range of open source libraries exist to managing these challenges (Netflix are best known)

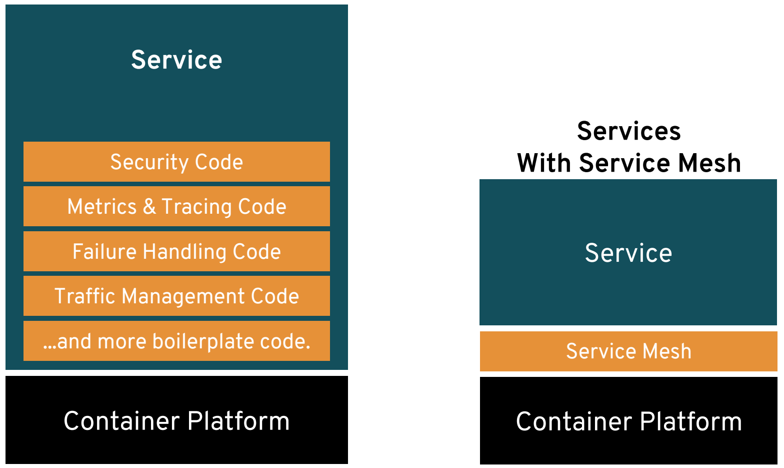
- **This results in:**
  - **Different solutions in different services**
  - **Boilerplate code**
  - **New dependencies to keep up date**



Don’t force extra work on developers

Service mesh does this at a platform level

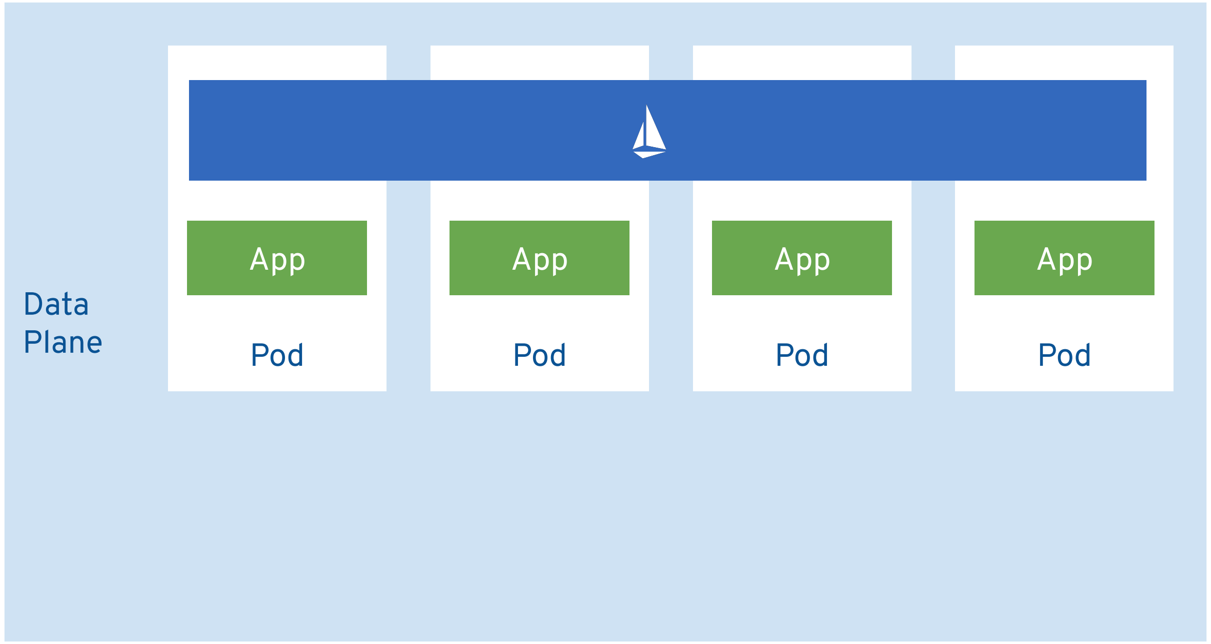
- Service Mesh solve distributed systems challenges at **a common infrastructure layer**
- This reduces boilerplate code and copy/paste errors across services
- Enforces common policies across all services
- Removes the obligation to implement cross cutting concerns from developers



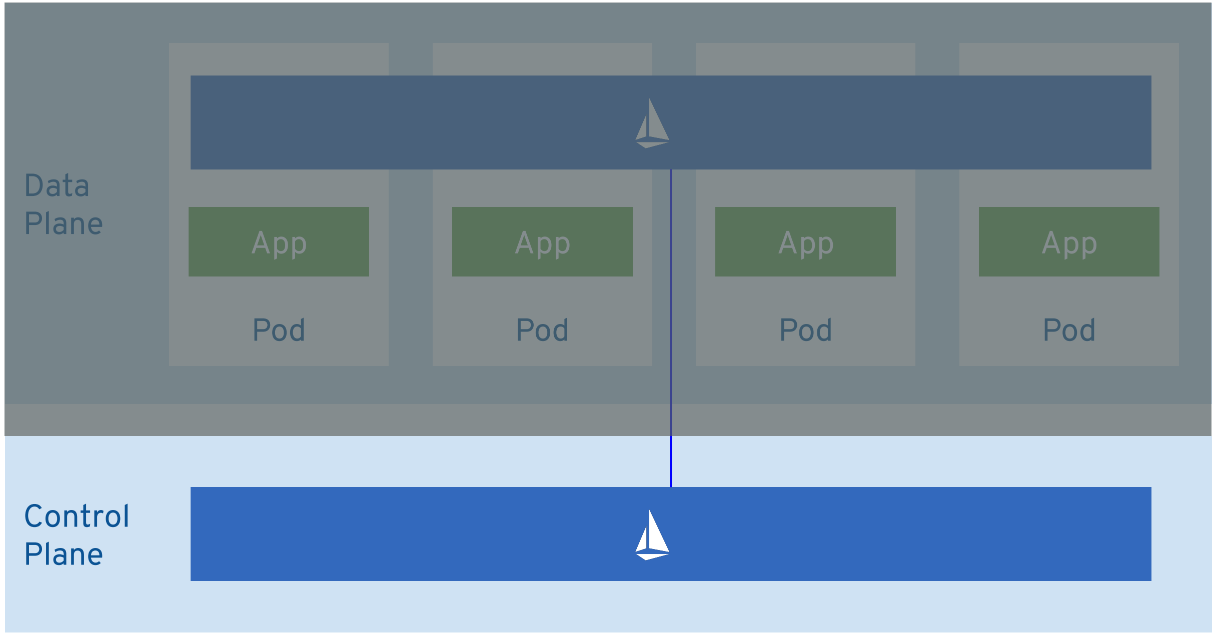
Behind the curtain

How the service mesh is architected

Your microservices are in a “data plane”



Your policy is in a “control plane”



With these you can dynamically configure microservices without code changes or redeployment

Time for labs!

Let's get hands-on

- Intros & Overview
- **Part 1**
  - Kickoff
  - Hands On Labs
  - Recap Poll and Breakouts
- Break
- Part 2
  - Kickoff
  - Hands On Labs
  - Poll and Concluding Discussion

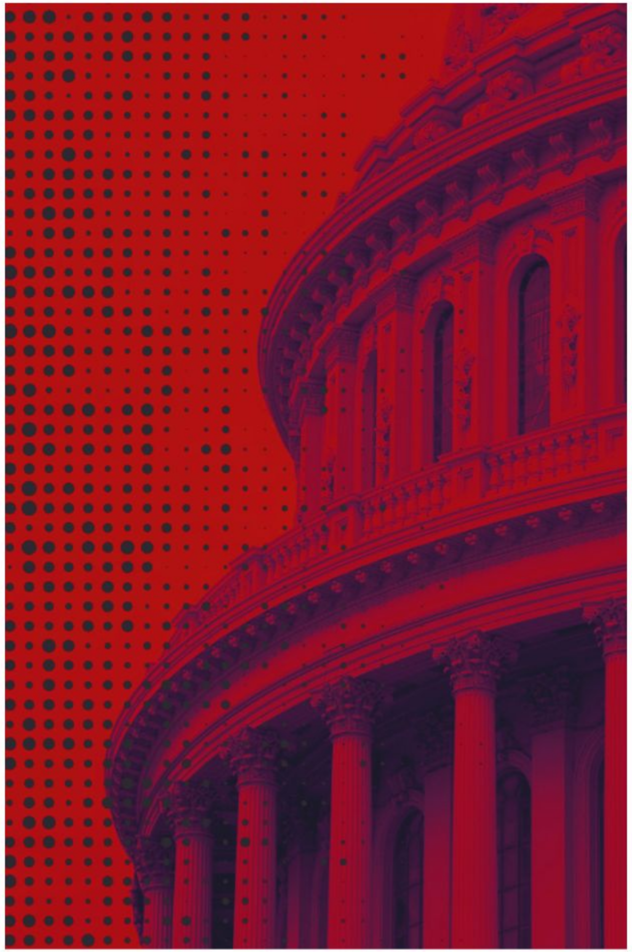
We're on break

10 min break

Time for labs!

Let's get hands-on

- Intros & Overview
- Part 1
  - Kickoff
  - Hands On Labs
  - Recap Poll and Breakouts
- Break
- **Part 2**
  - Kickoff
  - Hands On Labs
  - Poll and Concluding Discussion



DevNation Federal 2021:

Learn More. Code More. Share more. Join the Nation.

THANK YOU!

DEVNATION FEDERAL