

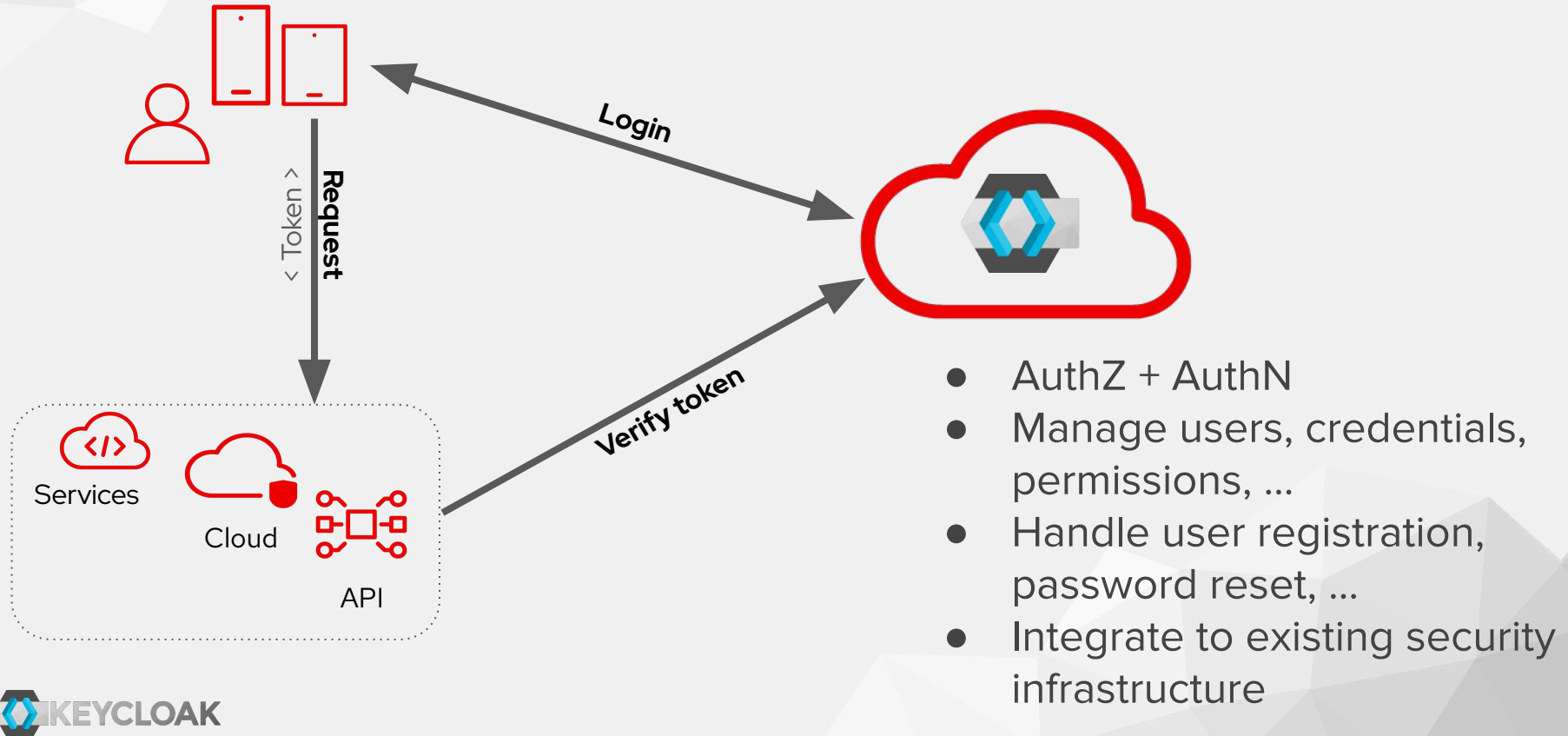


# Highly available Identity and Access Management with multi-site Keycloak deployments in the cloud

Ryan Emerson, Kamesh Akella | Principal Software Engineers | Red Hat  
KubeCon SLC, Utah | 2024-11-15

# What is Identity and Access Management (IAM), and do I need one?

# Authenticate and authorize users for services



## Day 1: Single-Sign-On is cool!

- Users need to remember only one password
- Authenticate only once per day
- Add second factor for authentication for security
- Theme the frontend to match your needs

➡ Makes sense already for a single application!

# Keycloak provides the login screen for your apps

Sign in to your account

Username or email

Password

Sign In

## Day 2: Become flexible in your setup

- Integrate LDAP and Kerberos
- Brokerage to existing SAML services
- Brokerage to existing OIDC services
- Integrate existing custom stores

➔ Reuse existing user stores!

# Login with LDAP

Sign in to your account

Username or email

Password

Sign In

# Use Brokerage for existing providers





Sign in to your account


Username or email

Password

Sign In

Or sign in with

 GitHub	 OpenShift v4
 StackOverflow	 Google





# Skip the form with Kerberos/SNPEGO!

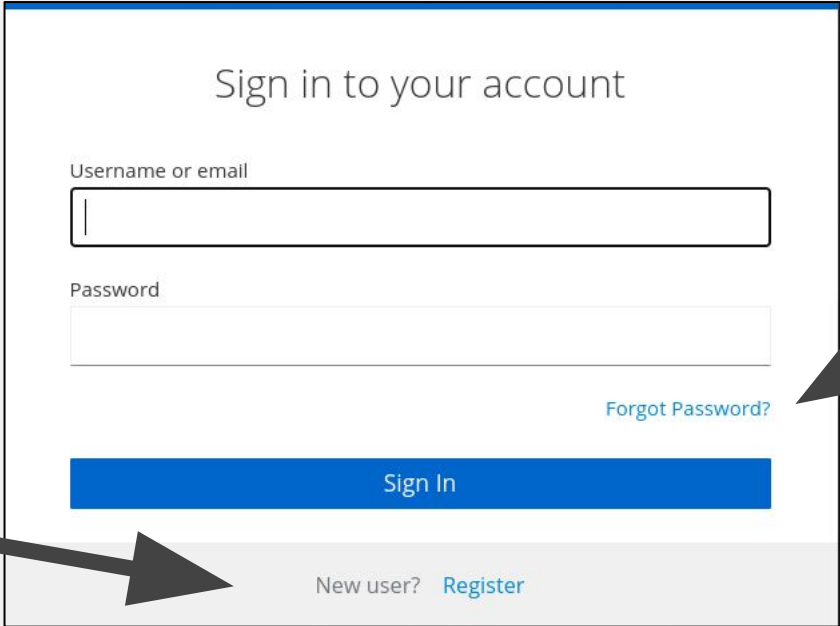
***This page intentionally left blank.***

## Day 3: Eliminate daily churn

- User password recovery (even when using LDAP)
- Self-registration for users
- User data self-management

➡ Resolve the need for calls and tickets!

# Password recovery and self-registration



Sign in to your account

Username or email

Password

[Forgot Password?](#)

[Sign In](#)

New user? [Register](#)

# Declarative User Profile configuration

<

General

Login

Email

Themes

Keys

Events

Lo

AttributesAttributes GroupJSON editor

▼ All groups

Create attribute

Attribute [Name]	Display name
<div>⋮</div> username	\${username}
<div>⋮</div> email	\${email}
<div>⋮</div> firstName	\${firstName}
<div>⋮</div> lastName	\${lastName}

Permission

Who can edit? ⓘ

✓

User

✓

Admin

Who can view? ⓘ

✓

User

✓

Admin

Validations

Validator name	Config
length	{ "min":3, "max":255 }
username-prohibited-characters	{ }
up-username-not-idn-homograph	{ }

# User Profile for admins, registration, and users

**General**

**Username \***

admin

**Email**

admin@keycloak.org

**First name**

**Last name**

**Register**

\* Required fields

**Username \***

**Password \***

**Confirm password \***

**Email \***

**First name \***

**Last name \***

[« Back to Login](#)

Register

**Personal info**

Manage your basic information

**General**

**Username \***

admin

**Email \***

admin@keycloak.org

**First name \***

**Last name \***

Save

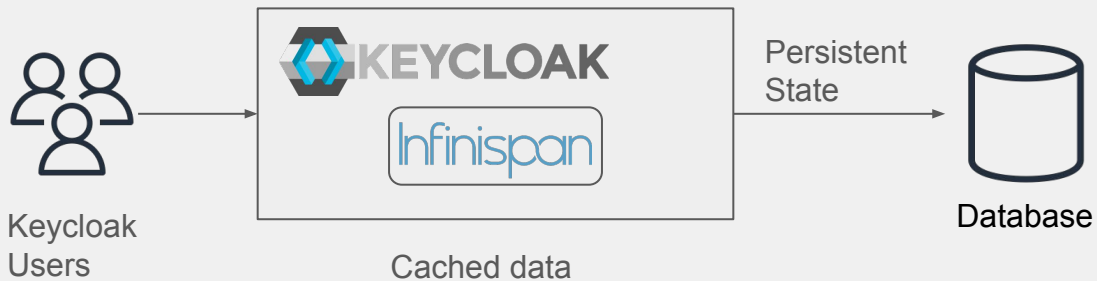
Cancel

**Keycloak is now a critical component in your infrastructure.**

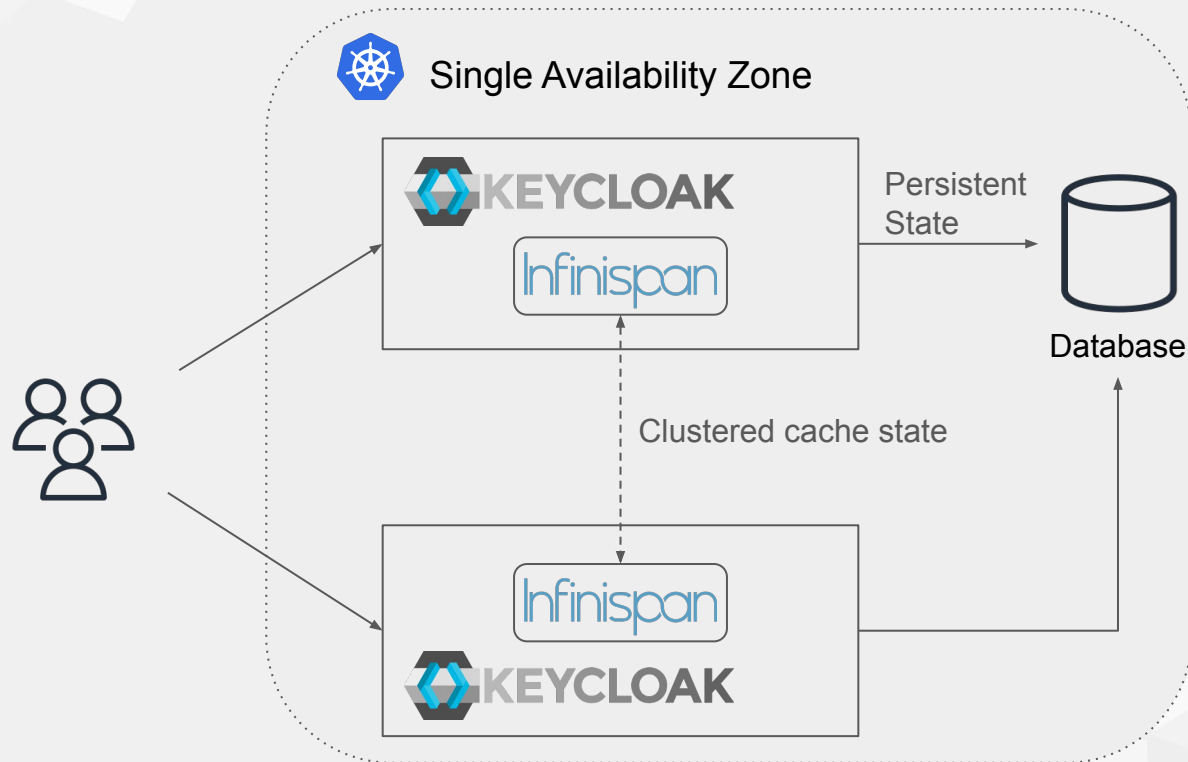
**You want it to be available 24/7.**

# Single Instance

- In-memory cache for reduced latency



# Multiple Keycloak Instances

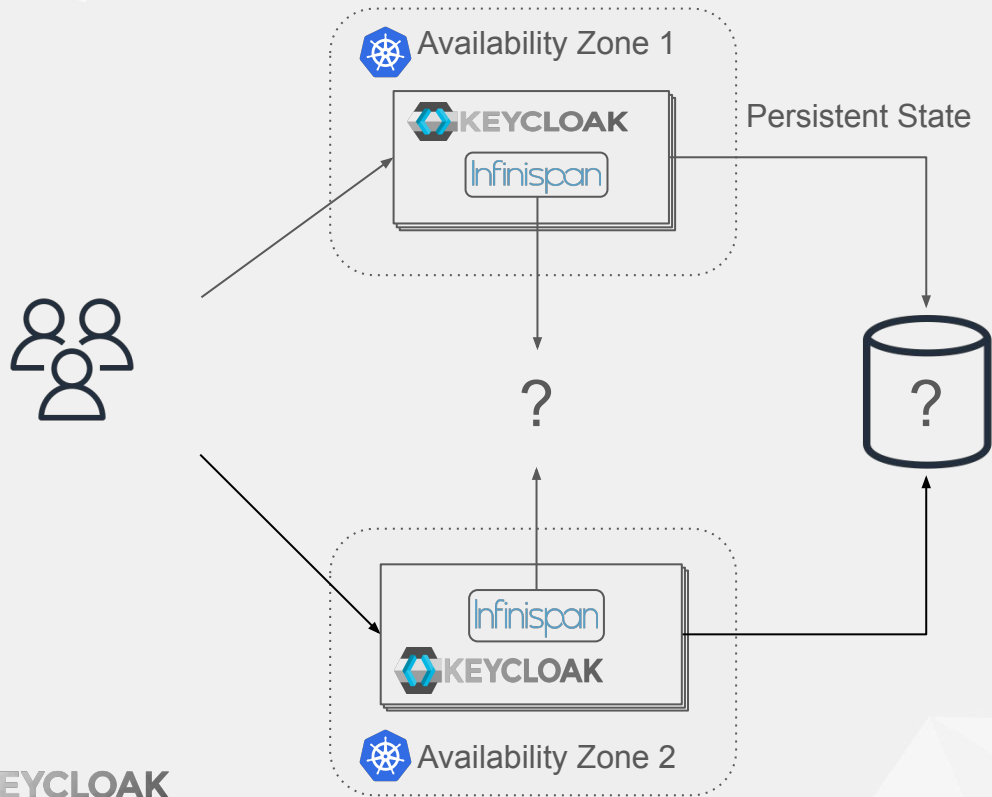


- Increased performance and resilience
- Cache state replicated and invalidated between Keycloak processes
- Deploy with Kubernetes
- Can tolerate K8s node/pod failures



# Tolerating Availability Zone Failures

# Multi Site

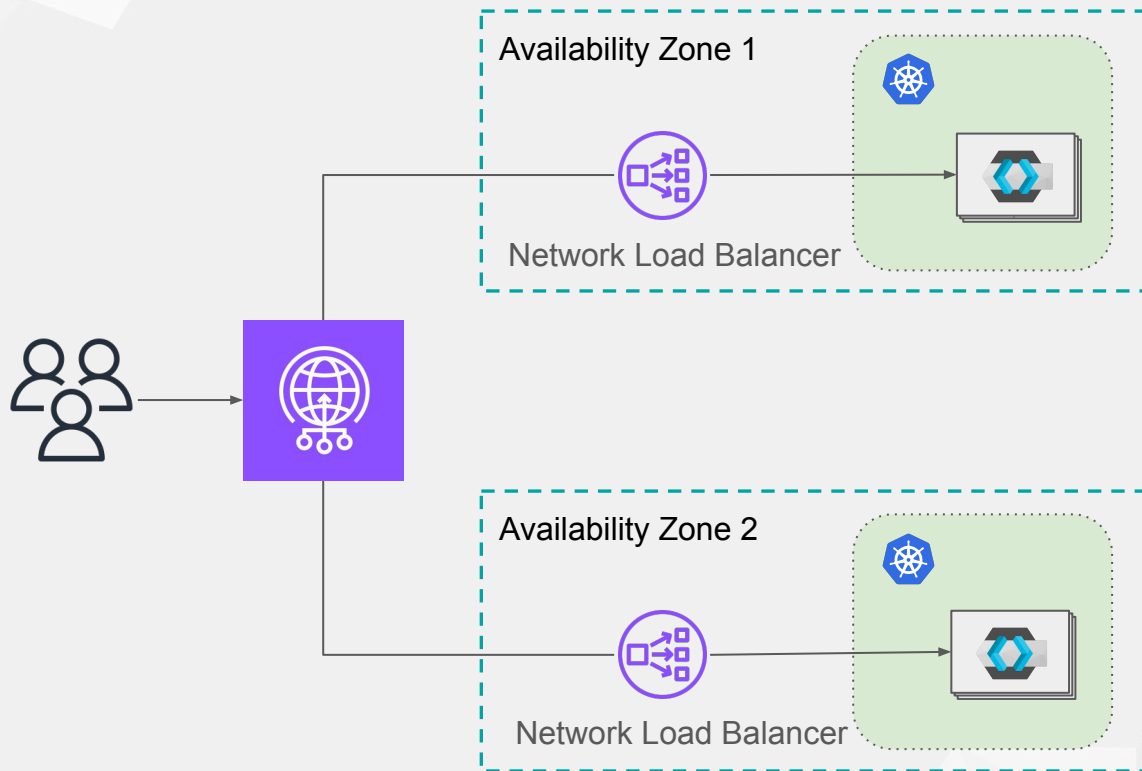


- Deploy Keycloak to multiple availability zones
- *How to manage cache and persistent state?*
- *How to route user requests?*

# Managing User Connections

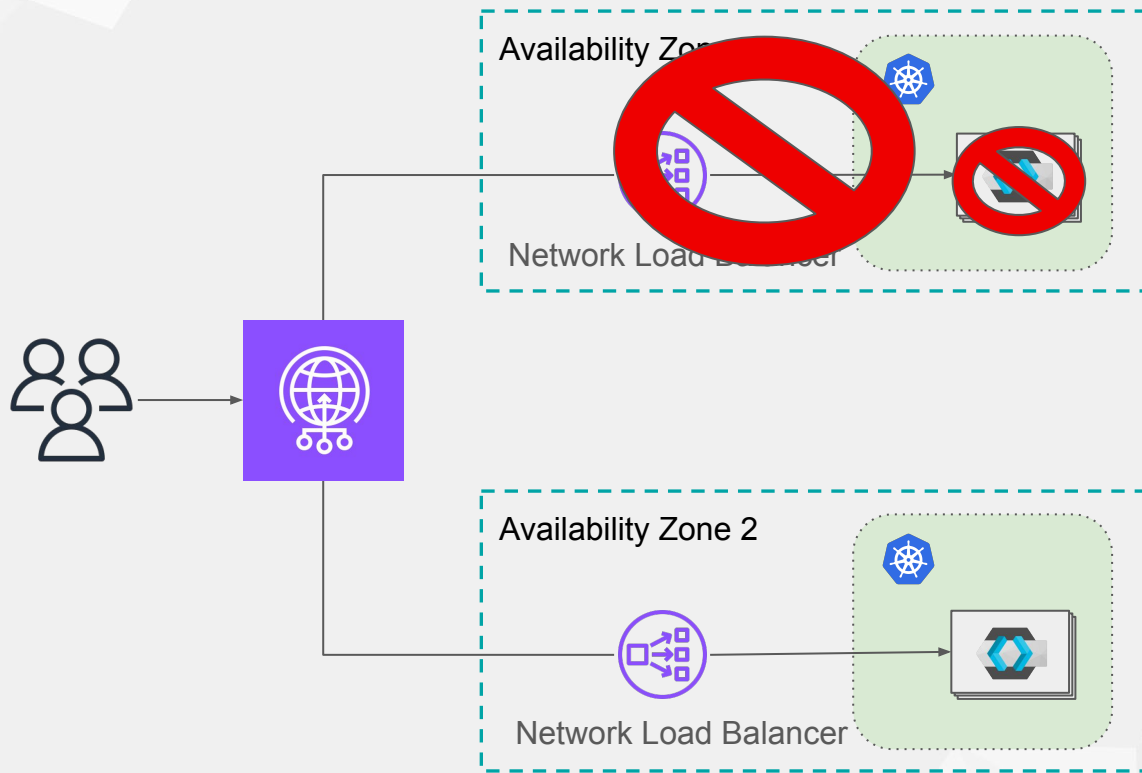


# User Connections



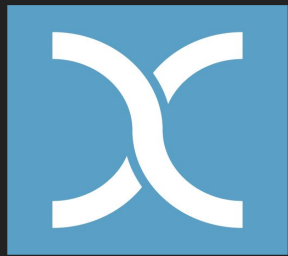
- All user requests via ***AWS Global Accelerator***
- NLB routes traffic to Keycloak pods
- Periodic Health checks determine NLB health

# User Connections



- Traffic only routed to healthy sites

# Cache Invalidation



Infinispan

# Infinispan

- In-memory Key/Value Cache
- Advanced clustering capabilities
- Independent Project
- Apache 2.0 License
- Redis Compatible



Operator



QUARKUS



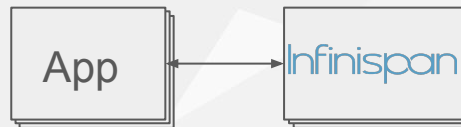
spring boot

# Infinispan

- Embedded Mode

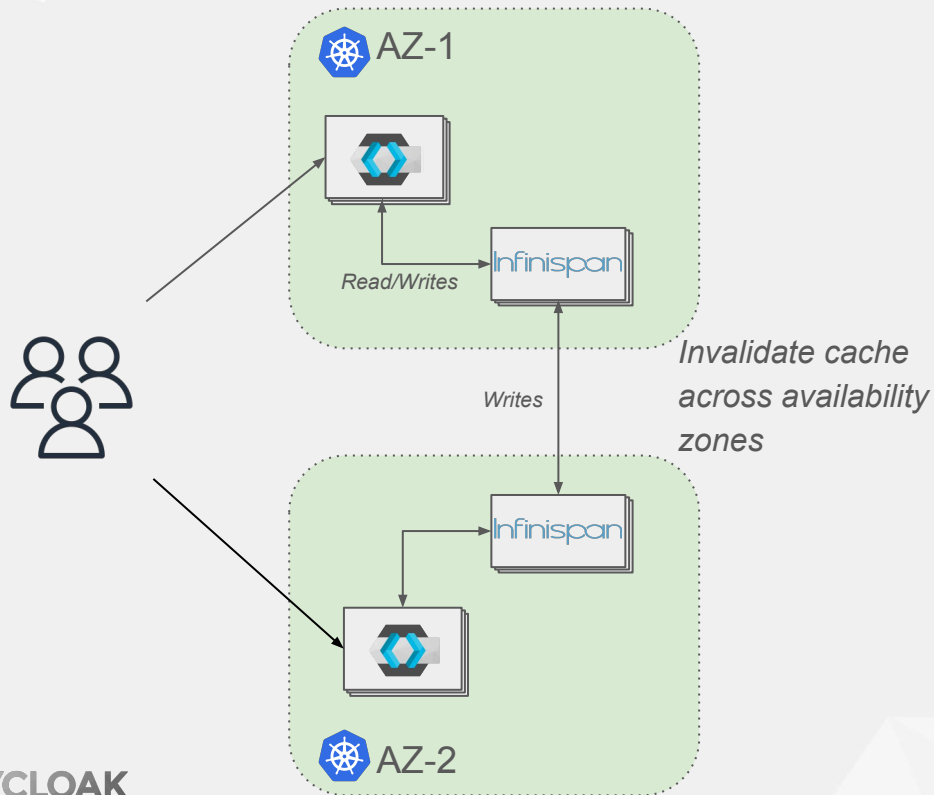


- Client/Server





# Multi-site Cache Invalidation

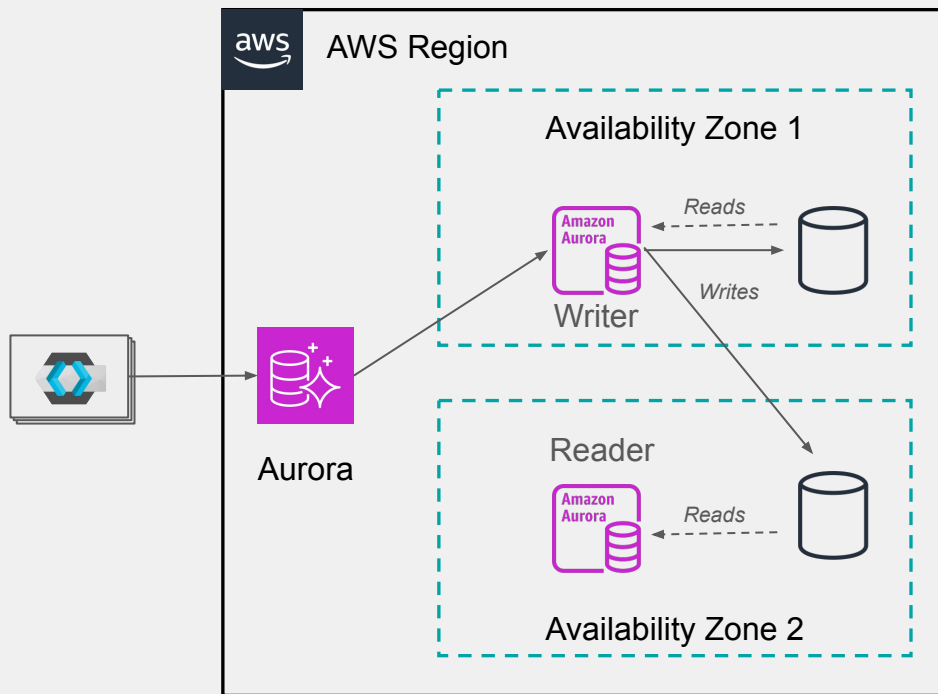


- Utilise Infinispan Server
- Infinispan Cross-Site replication used to update cache state
- Infinispan server supports advanced admin operations for Cross-Site management

# Database Failover

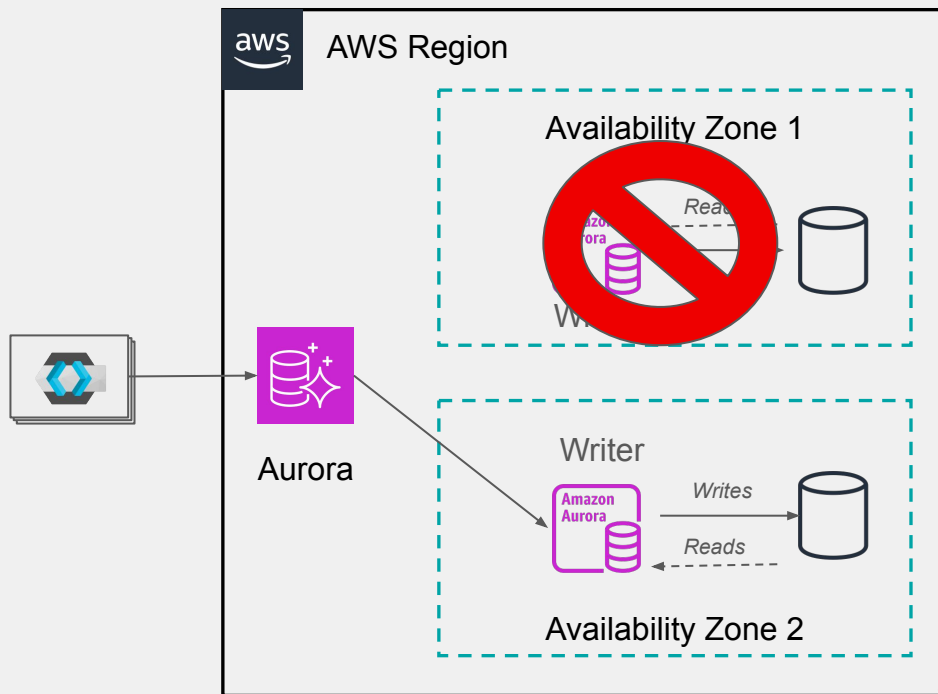


# Aurora DB



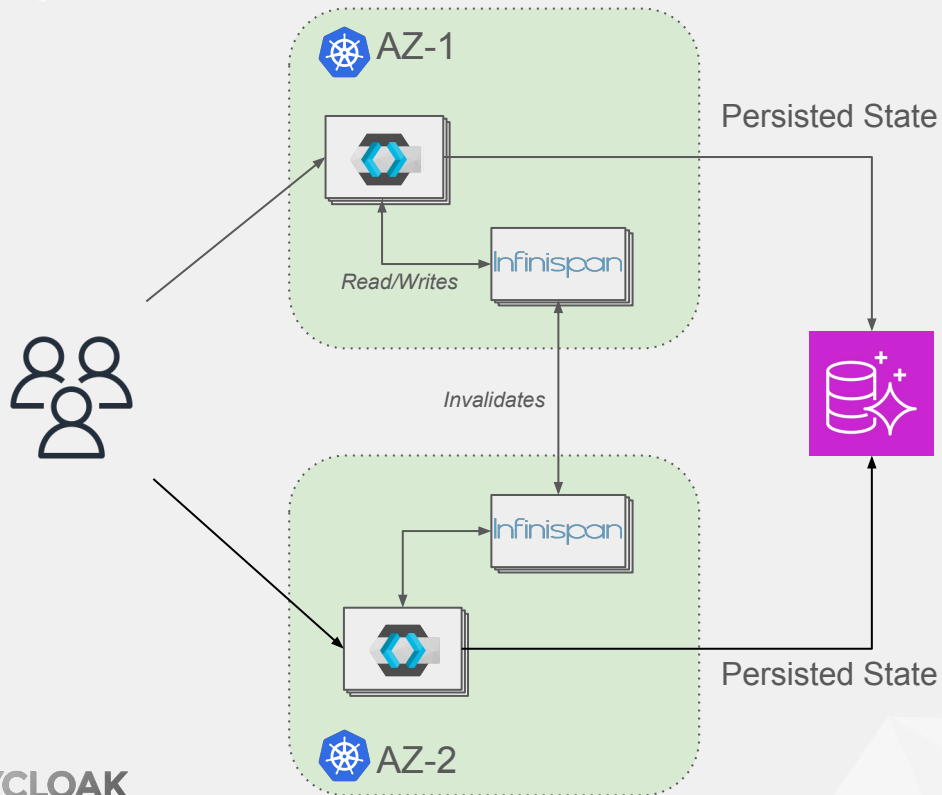
- All read/writes handled by a single “Writer” instance
- Data written to both Availability Zones to allow failover

# Aurora DB



- Election of new writer instance managed by Aurora
- Failover takes ~ 1m
- AWS JDBC wrapper ensures Keycloak pods drop old connections on failover
- No additional Keycloak semantics required

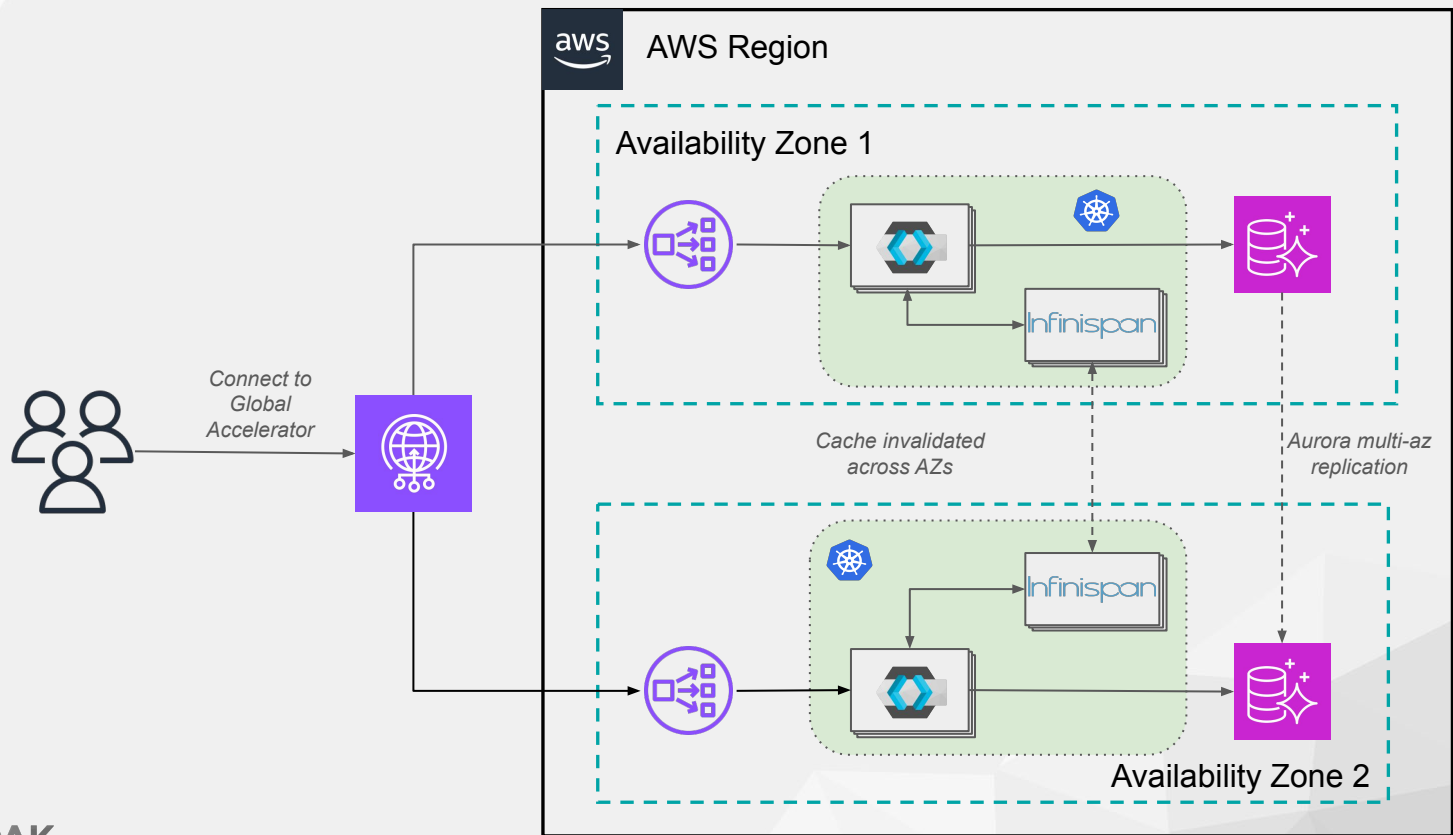
# Aurora DB



- Increased latency for Keycloak writes in AZ hosting Aurora *Reader* instance

# Architecture Overview

# HA Architecture

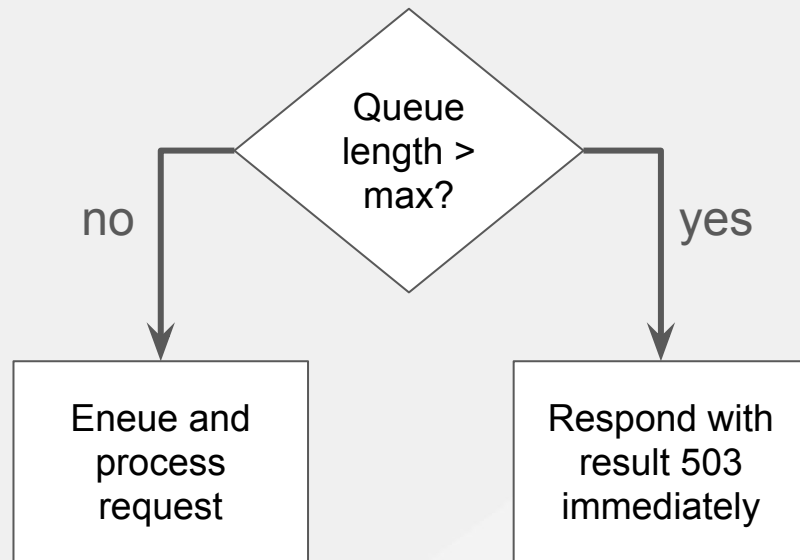


# Surprising system behaviors under load



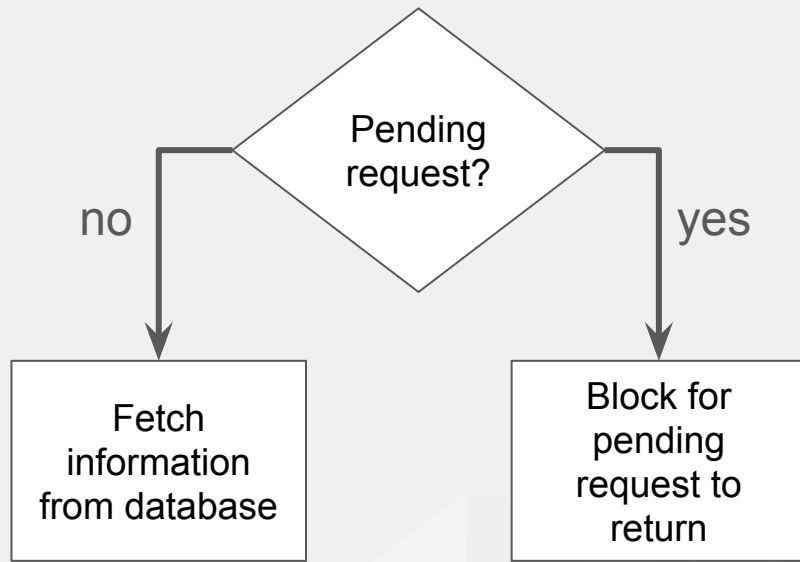
# Overload situations requiring load shedding

- **Overload:** More requests incoming than can be handled
- **Observation:** Requests queue up, memory usage increases, client requests time out.
- **Remedy:** Drop requests by replying with 503



# Cache stampede protection when restarting pods

- **Cache stampede:** When restarting under high load, parallel requests access the database while the cache is empty
- **Observation:** Timeouts, exhaustion of DB connections.
- **Remedy:** JVM locking if the same resource is about to be fetched from the database.



# Tackling blocking probes and metrics

- **Overload:** Too many requests make responses slow or lead to load shedding.
- **Observation:** Blocking probes stop working and Pod restart. Metrics become unavailable.
- **Remedy:** Use non-blocking probes that don't enqueue in an overload situation. Disable load-shedding for metrics.

Symptoms: Kubernetes events similar to:

Liveness probe failed / Timeout / n times in the last x seconds

Container xxx failed liveness probe, will be restarted

# Good habits

# Up-to-date documentation

Use with static site publishing as you, including onboarding.



[antora.org](https://antora.org)



## Keycloak Benchmark

Search the docs

In this guide



# Keycloak Benchmark

### Contents

- Next steps

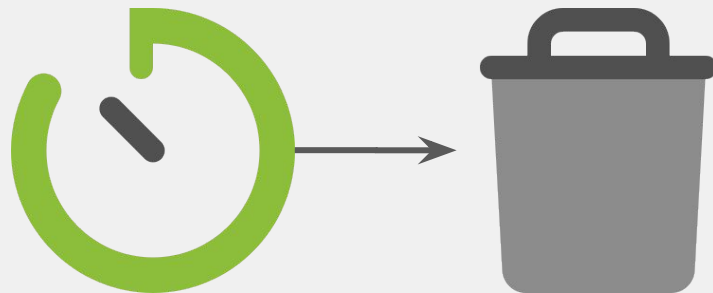
The [Keycloak Benchmark repository](#) contains the necessary tools to run performance tests of a [Keycloak server](#).

It has the following goals:

1. Setup Keycloak for reproducible results.
2. Run load tests against any Keycloak instance

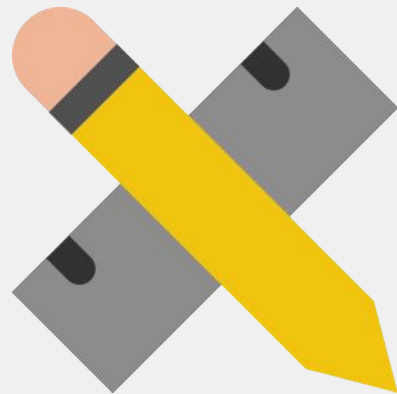
# Ephemeral environments

You can have as many environments as you want, but they will be deleted automatically at the end of the day.



# Measure, record and repeat

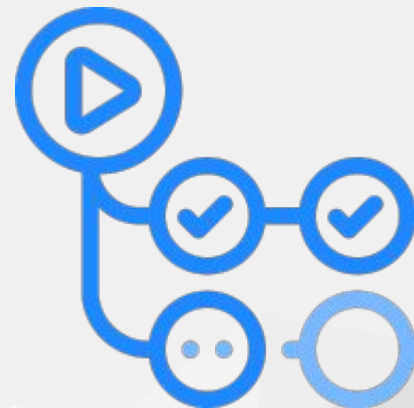
- Add Metrics collection and searchable logs as early as possible.
- Capture all insights after a run automatically for an ephemeral environment is hard but saves time in the long run.
- Nightly runs for performance and functional tests show if there are regressions.



# Continuous Integration



- Nightly CI testing with GitHub Workflows and Actions
  - Provisioning the environment
  - Reporting on the results
  - Workflow dispatch via CLI and UI





# CNCF Projects that were most helpful



kubernetes



Prometheus



# Kubernetes for scheduling deployments



# kubernetes

- Utilize status information in resources and from Operators:

```
kubectl wait  
--for=condition=...  
--timeout=1200s <resource>
```
- Red Hat OpenShift Service on AWS (ROSA) for ephemeral environments (logging, metrics, etc. as bundled add-ons)

# Helm for fast turnaround in deployment

- Simple templating language
- Try out changes to your charts in seconds
- Our charts grown in variants for development and performance testing, no longer suitable for a production deployment



# OpenTelemetry Java agent for metrics and traces

Adds instrumentation to a lot of well-known libraries, even if your Java application doesn't support tracing out-of-the box.

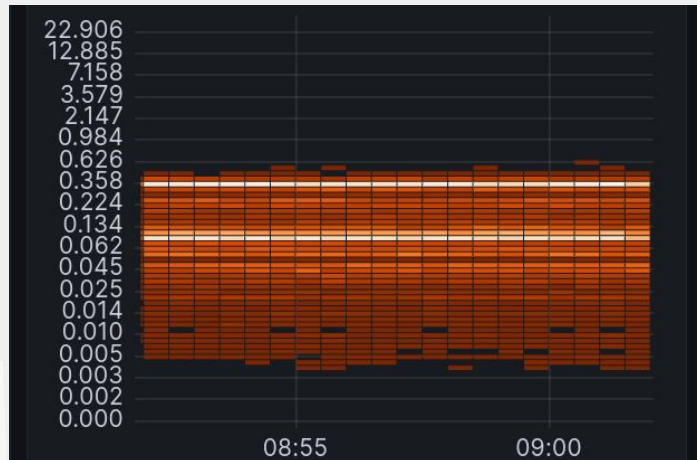


# Grafana for interactive dashboards

- create dashboards and store them as JSON to share them with the team.
- Plot histograms as heat maps to visualize SLOs.
- Jump from metrics to traces to logs.



# Grafana



# Running Gatling on multiple nodes

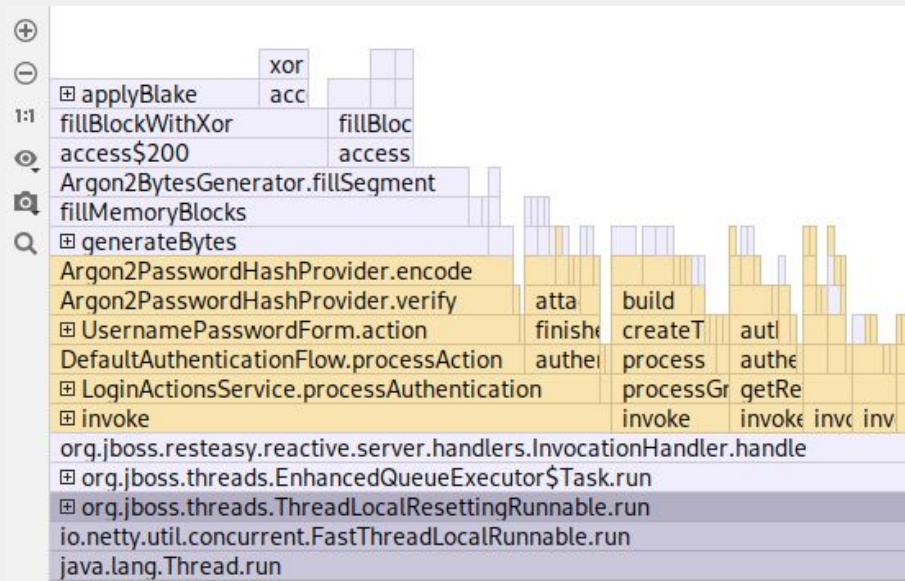
- Gatling running on multiple ephemeral nodes to overcome OS network stack limitations under high load.  
(~ 250 connections per second per load driving host)



<https://github.com/keycloak/keycloak-benchmark/tree/main/ansible>

# Java profiler to analyze performance

- Flame graphs help to capture activity.
- Wall-clock async profiling works even in containers with reduced profiling permissions.
- Cryostat.io for simplified integration in container environments.





# The Future

# Future roadmap for Keycloak

Ideas for enhancements (but no planned releases):

- Zero-downtime upgrades
- Support more cloud platform and databases
- ARM Support
- Provide security-hardened blueprints
- Leveraging more tools from CNCF eco-system

# Links

- **Keycloak**  
<https://www.keycloak.org/>
- **Keycloak Benchmark Project**  
<https://www.keycloak.org/keycloak-benchmark/>
- **Keycloak High Availability Guide**  
<https://www.keycloak.org/high-availability/introduction>
- **Keycloak Book 2nd Edition**  
<https://www.packtpub.com/product/kc/9781804616444>
- **Infinispan**  
<https://infinispan.org/>

# Contact



Kamesh Akella  
Principal Software Quality Engineer

[kakella@redhat.com](mailto:kakella@redhat.com)  
[github.com/kami619](https://github.com/kami619)

Ryan Emerson  
Principal Software Engineer

[remerson@redhat.com](mailto:remerson@redhat.com)  
[github.com/ryanemerson](https://github.com/ryanemerson)

**Questions?**