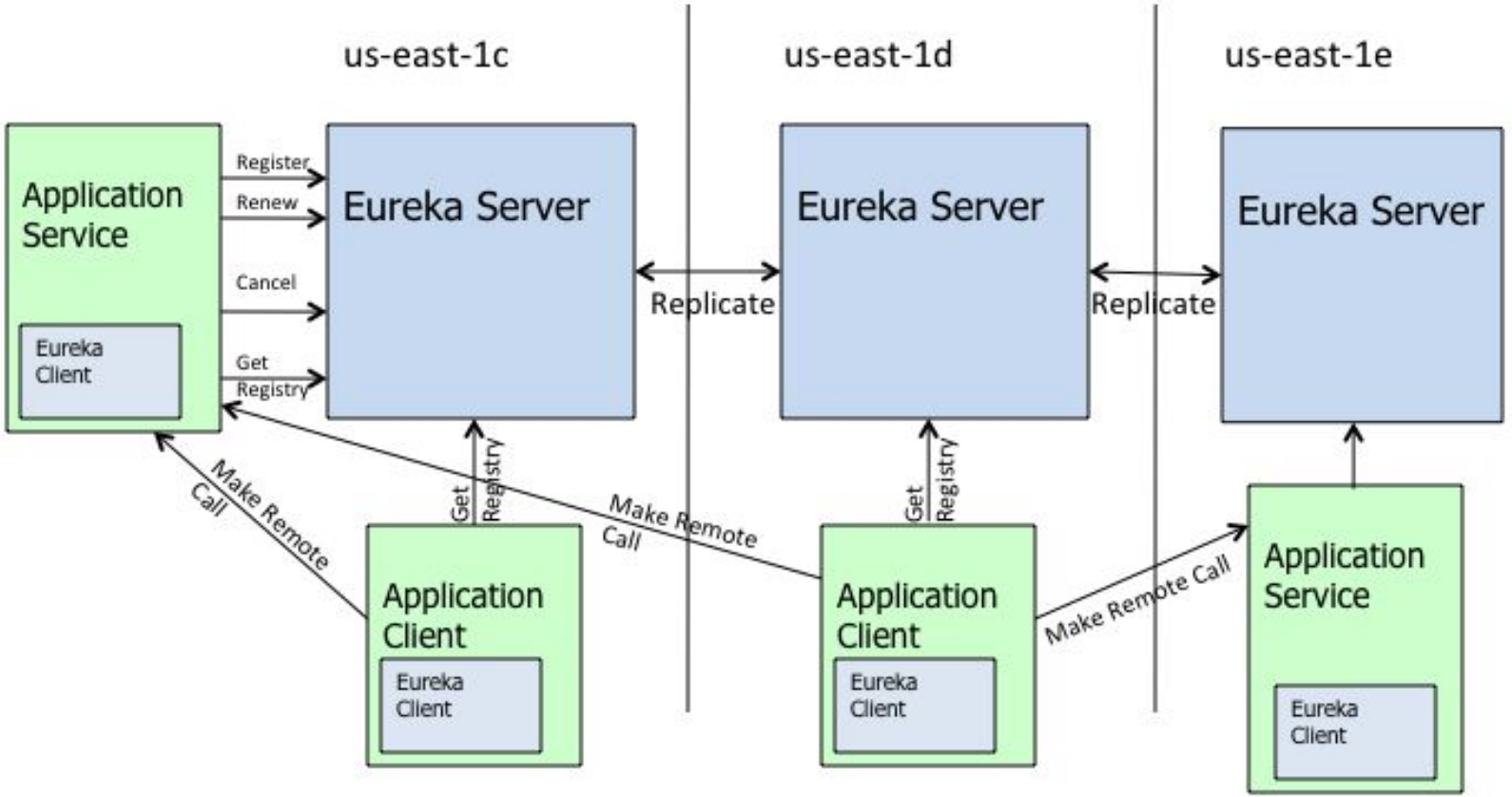
Pivotal

Spring Cloud Netflix Eureka Server

© Copyright 2017 Pivotal Software, Inc. All rights Reserved. Version 1.0

Eureka Architecture



Endpoints

- Register (and de-register)
- Renew Registration
- Fetch Registry

See https://github.com/Netflix/eureka/wiki/Eureka-REST-operations

Register

- Service registers with eureka on startup
- With Spring Cloud, the `spring.application.name` property is used as the registration key (or virtual hostname)
- Registration can be turned off by setting configuration property `eureka.client.registerWithEureka` to false
- `eureka.client.serviceUrl.defaultZone` can be used to specify default url for contacting eureka

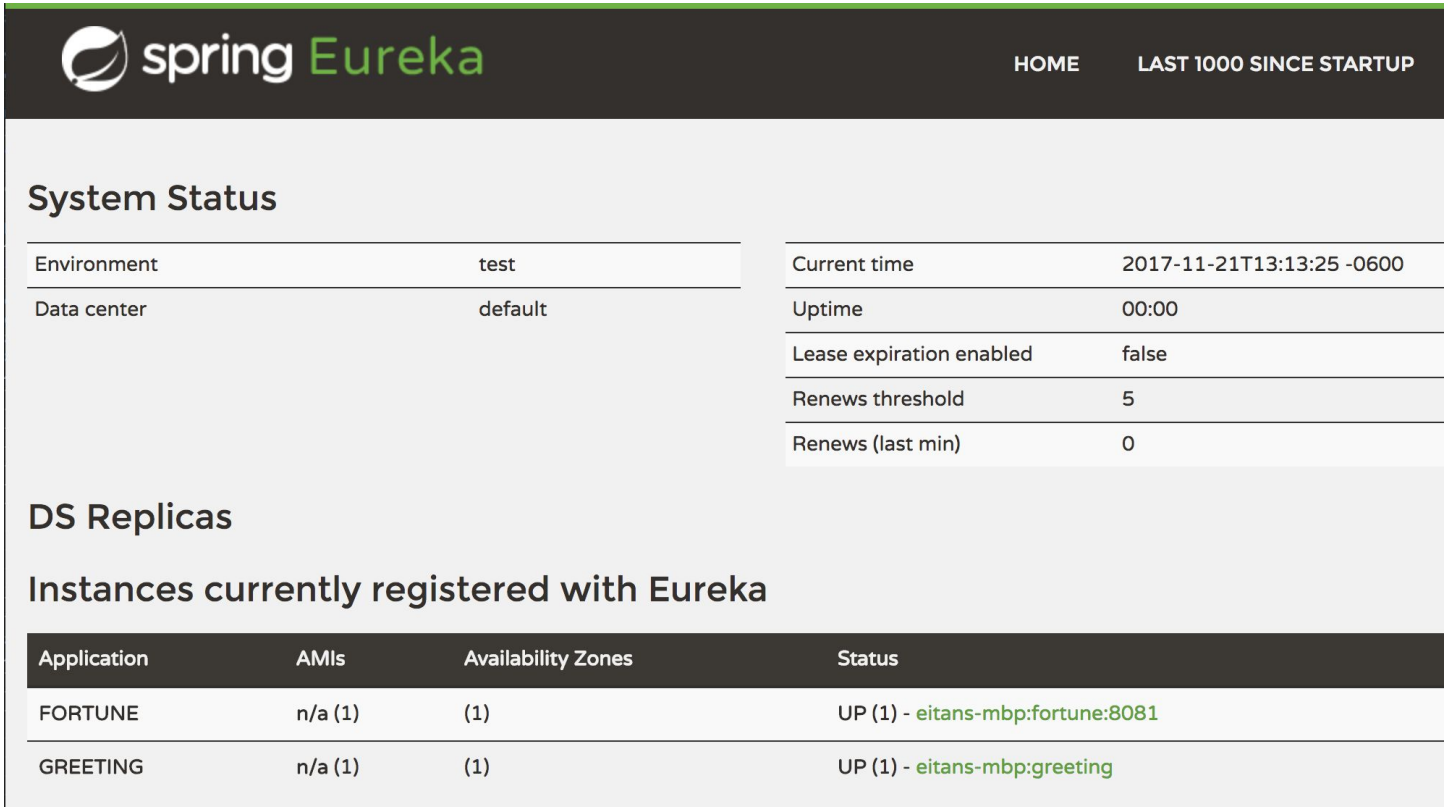
Renew Registration

- Services must periodically renew their registration, which would otherwise expire
- aka “Heartbeats”
- The configuration property `eureka.instance.leaseRenewalIntervalInSeconds` governs how often a service renews their registration

Fetch Registry

- Clients fetch a copy of the registry periodically
- An optimization, allows lookups to be performed directly against a cached copy
- `eureka.client.fetchRegistry` can be used to control whether to fetch the registry
- `eureka.client.registryFetchIntervalSeconds` controls how frequently to fetch a new copy

The Eureka Dashboard



Availability through Peer Replication

- Eureka servers are stateful
- Eureka store registration information in-memory,
  Not in a backing persistent database
- Peer replication can be configured between Eureka Server nodes to replicate between instances
- Eureka servers are also eureka clients configured to discover registration information from their peers.
- Discovery clients can be configured to consume from multiple eureka server peers.

Self Preservation Mode

- Feature designed to detect patterns of eureka client connection failures, and prevent wiping of registry information during network partition or transient network failures.
- Suppresses eviction of registered clients when > 15% of the population does not renew registrations
- You may see the following warning in the Eureka Dashboard when Self Preservation Mode is active:

EMERGENCY! EUREKA MAY BE INCORRECTLY CLAIMING INSTANCES ARE UP WHEN THEY'RE NOT. RENEWALS ARE LESSER THAN THRESHOLD AND HENCE THE INSTANCES ARE NOT BEING EXPIRED JUST TO BE SAFE.

Self Preservation Mode - Tradeoffs

- In modern Cloud architectures where co-locating microservices and service registries, ideally networks should not suffer partitions, may not be necessary.

- For development or small scale eureka deployment with small number of registered clients, this can cause false positives, and recommended to disable it.

- See the following:
  - https://github.com/Netflix/eureka/wiki/Understanding-Eureka-Peer-to-Peer-Communication
  - https://github.com/Netflix/eureka/wiki/Server-Self-Preservation-Mode
  - https://stackoverflow.com/questions/39032741/what-does-renews-and-renews-threshold-mean-in-eureka