



# Deep Dive into Cloud Native Open Source with NetflixOSS

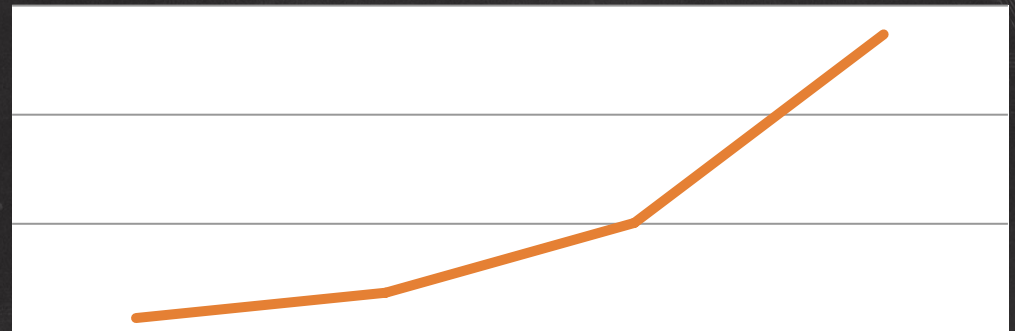
GOTO Copenhagen/Aarhus 2014  
Adrian Cockcroft – Battery Ventures - @adrianco

Thanks to Ruslan Meshenberg – Netflix - @rusmeshenberg

# Congratulations, your startup got funding!

- More developers
- More customers
- Higher availability
- Global distribution
- No time....

Growth



# Your architecture looks like this:



Web UI / Front End API

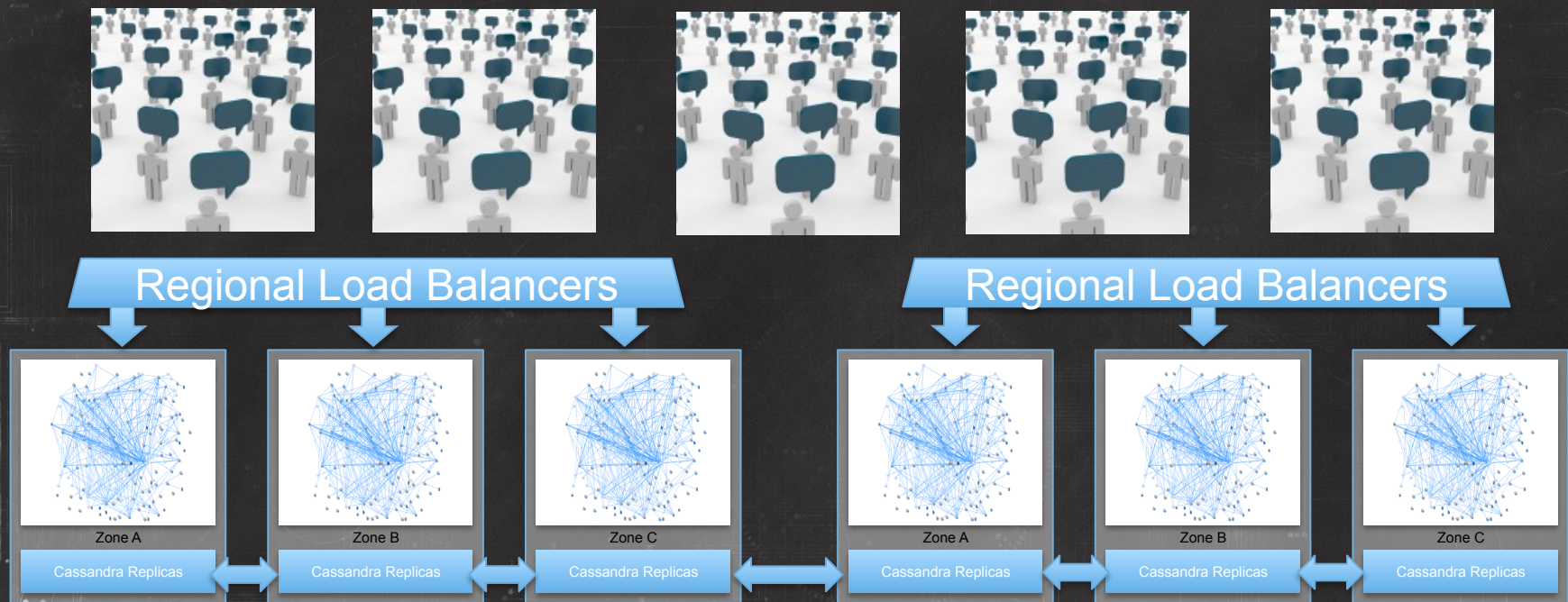
Middle Tier

RDS/MySQL

AWS Zone A

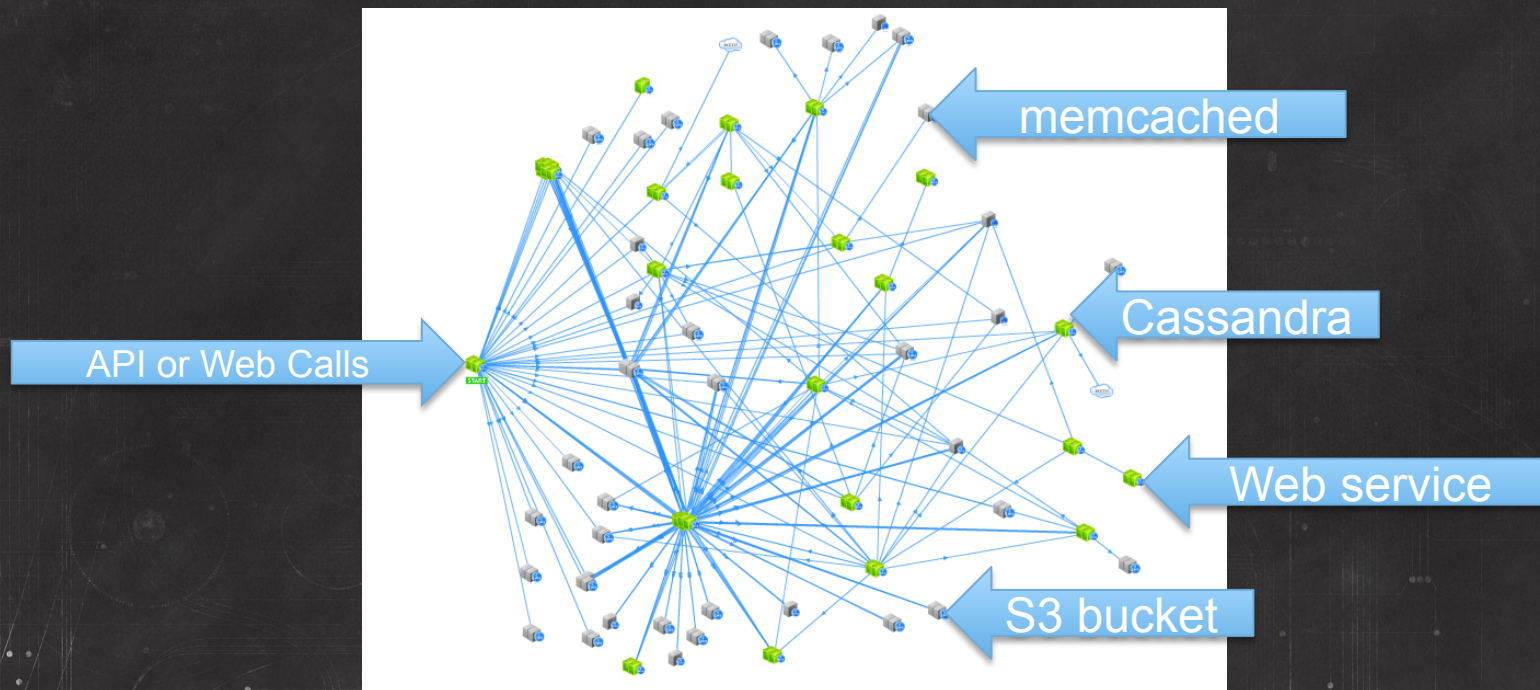


# And it needs to look more like this...





## Inside each AWS zone: Micro-services and de-normalized data stores



The Netflix logo in red, sans-serif capital letters.

NETFLIX

The OSS logo in blue, sans-serif capital letters, with the 'O' and 'S' being stylized.

OSS

We're here to help you get to global scale...  
Apache Licensed Cloud Native OSS Platform  
<http://netflix.github.com>



# Laptop Stickers with the newest logo!

- I have one sheet of 7 with me
- 3 for CPH, 4 for Aarhus
- Whoever uses the most NetflixOSS projects gets first pick of the sticker they want...
- Audience participation time!





# Audience Quiz – How Many Do You Use? (1/6)

## Availability

HYSTRIX



SIMIANARMY



TURBINE



## Cloud Management

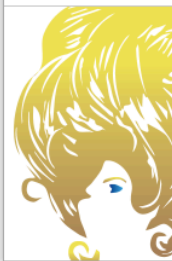
ICE



ASGARD



FRIGGA



GLISTEN

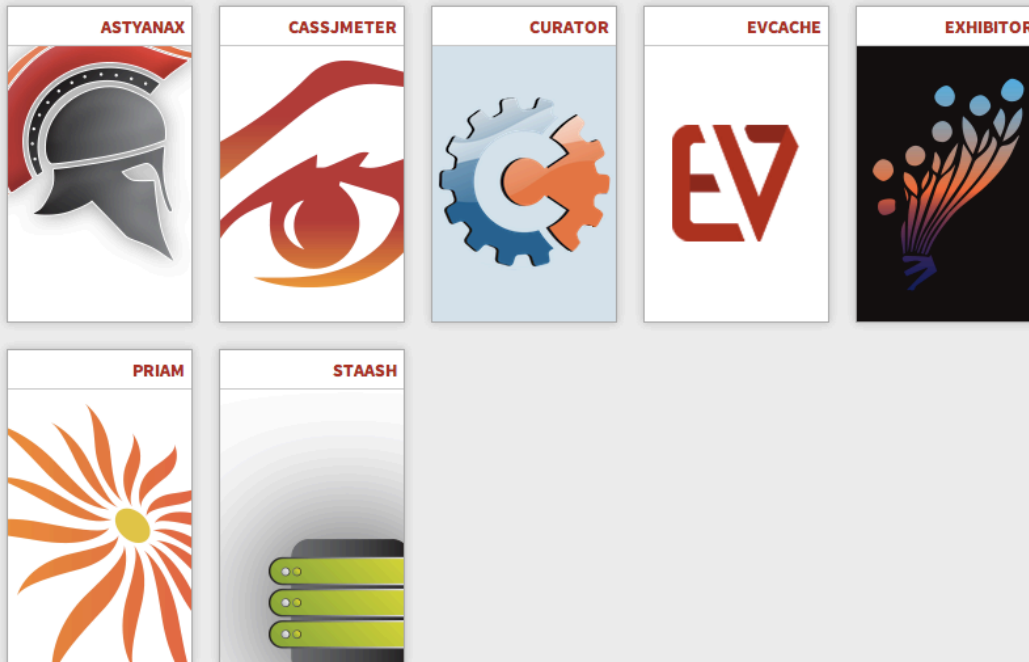


NETFLIX

OSS

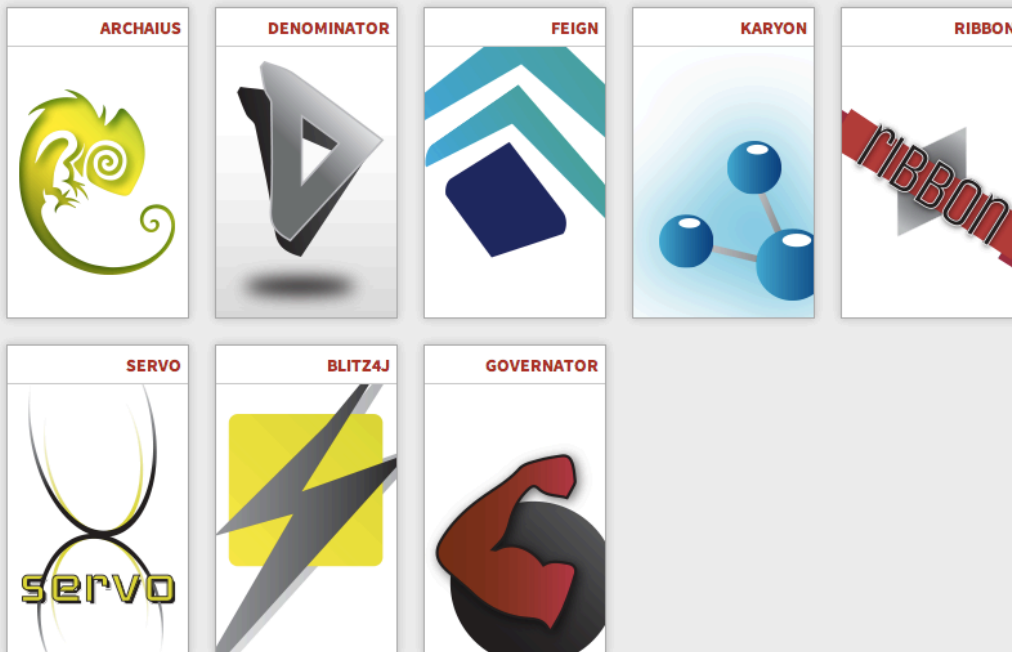
# Audience Quiz – How Many Do You Use? (2/6)

## Persistence Systems



# Audience Quiz – How Many Do You Use? (3/6)

## Platform Libraries



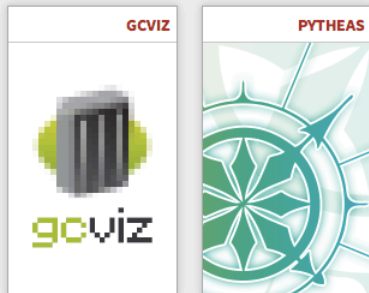


# Audience Quiz – How Many Do You Use? (4/6)

## Infrastructure Services



## Developer Productivity

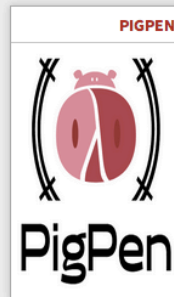
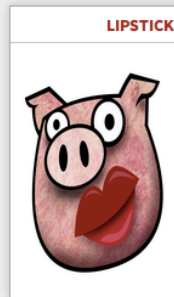
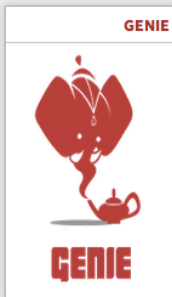


# Audience Quiz – How Many Do You Use? (5/6)

## Build and Deploy Tools



## Big Data Tools

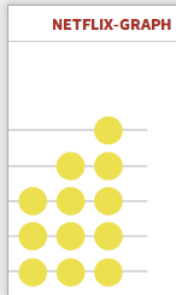


NETFLIX

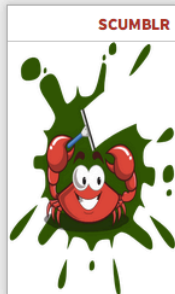
OSS

# Audience Quiz – How Many Do You Use? (6/6)

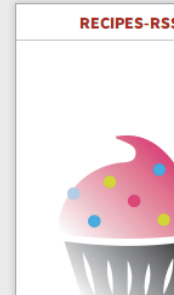
## In-Memory Data Management



## Security



## Sample Applications and Recipes



## Uncategorized



NETFLIX

OSS



# Getting started with NetflixOSS Step by Step

1. Set up AWS Accounts to get the foundation in place
2. Account Management Tools: Asgard for deploy & Ice for cost monitoring
3. Security and access management: Security Monkey, Scumblr, Sketchy
4. Build Tools: Aminator to automate baking AMIs
5. Service Registry and Searchable Account History: Eureka & Edda
6. Configuration Management: Archaius dynamic property system
7. Data storage: Cassandra, Astyanax, Priam, EVCache
8. Dynamic traffic routing: Denominator, Zuul, Ribbon, Karyon
9. Availability: Simian Army (Chaos Monkey), Hystrix, Turbine
10. Developer productivity: Blitz4J, GCViz, Pytheas, RxJava
11. Big Data: Genie for Hadoop PaaS, S3mper, Lipstick and Pigpen for Pig
12. Sample Apps to get started: RSS Reader, ACME Air, FluxCapacitor

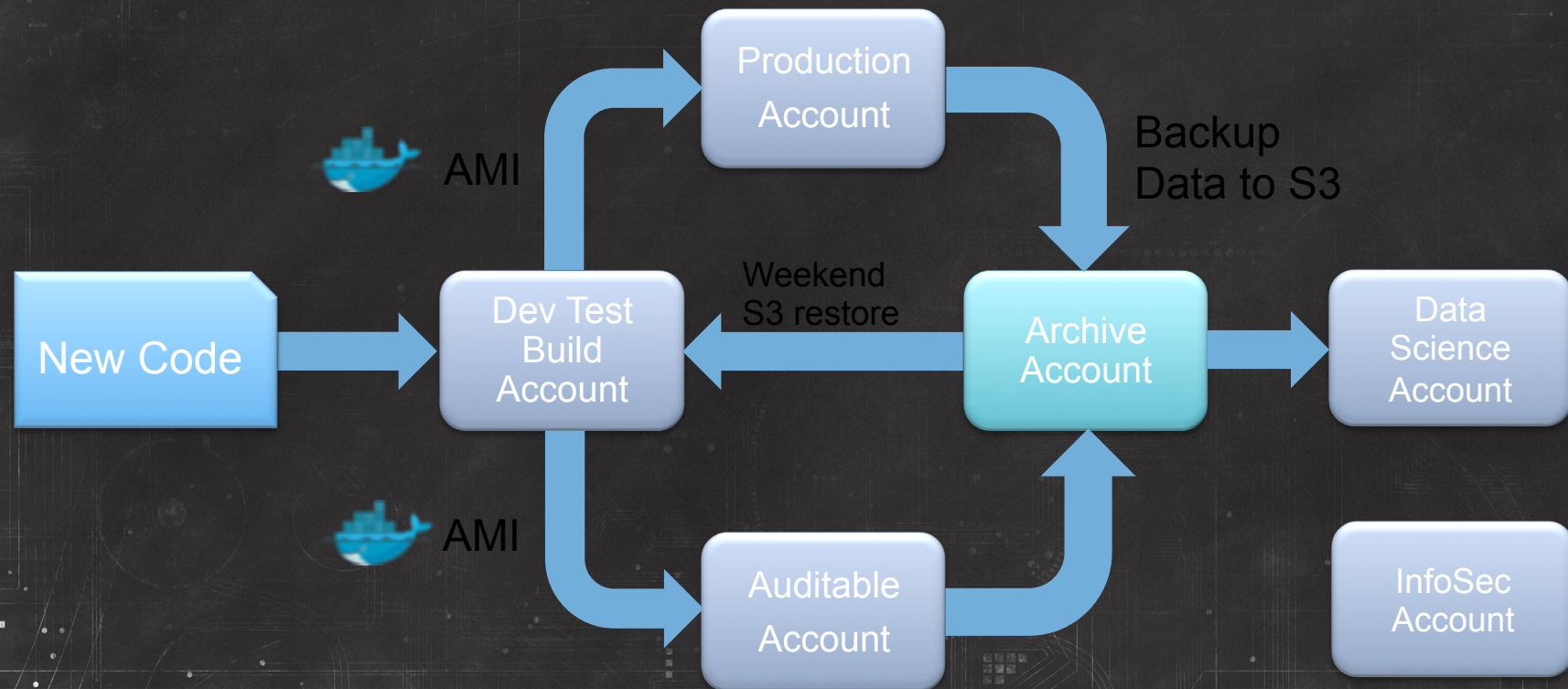




# **AWS Account Setup**



# Flow of Code and Data Between AWS Accounts



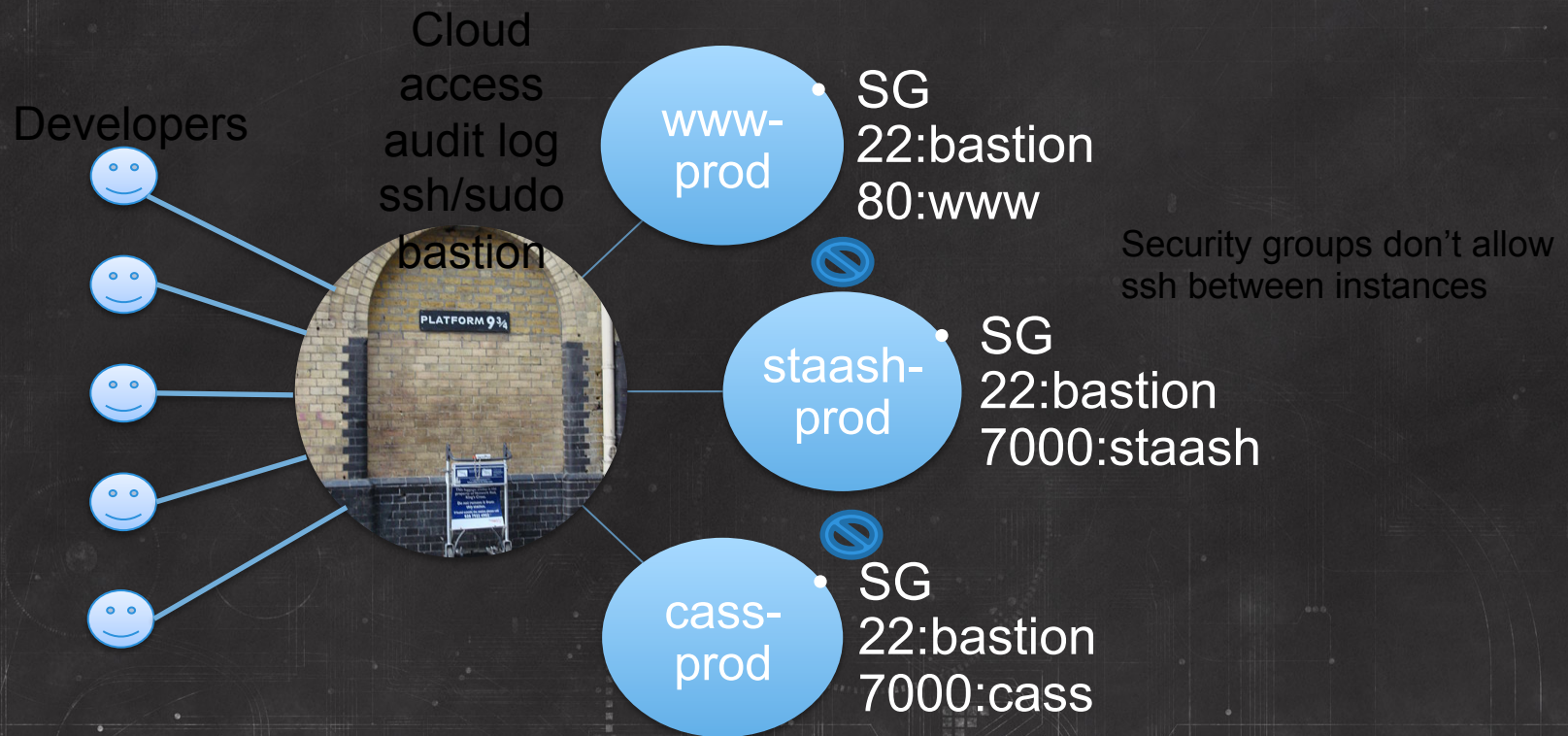


# Account Security (don't be like Codespaces!)

- Protect Accounts
  - Two factor authentication for primary login
- Delegated Minimum Privilege
  - Create IAM roles for everything
- Security Groups
  - Control who can call your services



# Cloud Access Control



# Use Security Monkey to Watch Developers



Security Monkey interface showing a list of items and a detailed view of an issue.

**Search**

Region:

Tech:

Account:

Name:

Search Config:

Status:

Type:

**Items**

Active	Technology	Account	Region	Name
✓	keypair	pat_enterprises	eu-west-1	proxyk
✓	keypair	pat_enterprises	us-east-1	Security
✓	keypair	pat_enterprises	us-west-2	newGP
✓	keypair	pat_enterprises	us-west-2	optiE2
✓	iamrole	pat_enterprises	universal	Security
✓	iamrole	pat_enterprises	universal	Security
✓	iamuser	pat_enterprises	universal	dropbo
✓	s3	pat_enterprises	us-west-2	www.s
✓	s3	pat_enterprises	us-west-1	www.z
✓	s3	pat_enterprises	us-west-2	saythat
✓	s3	pat_enterprises	us-west-1	zsh.sh
✓	securitygroup	pat_enterprises	us-east-1	launch
✓	securitygroup	pat_enterprises	ap-northeast-1	default
✓	securitygroup	pat_enterprises	us-west-2	SSH_H
✓	securitygroup	pat_enterprises	ap-southeast-1	default
✓	securitygroup	pat_enterprises	ap-southeast-2	default

**SSH\_HTTP**

Technology: securitygroup

Region: us-west-2

Account: pk\_enterprises

**Discovery Timeline**

Jun 29, 2014 5:52:14 AM

Jun 29, 2014 5:52:14 AM

**Issues**

Attention! The following issues have been raise and need to be fixed or justified.

Issue	Score	Notes
<input type="checkbox"/> Security Group contains 0.0.0.0/0	5	0.0.0.0/0

**Diff**

Current

```
{
  "description": "SSH_HTTP",
  "rules": [
    {
      "from_port": "22",
      "ip_protocol": "tcp",
      "to_port": "22",
      "owner_id": null,
      "name": null,
      "group_id": null,
      "cidr_ip": "0.0.0.0/0"
    },
    {
      "from_port": "80",
```

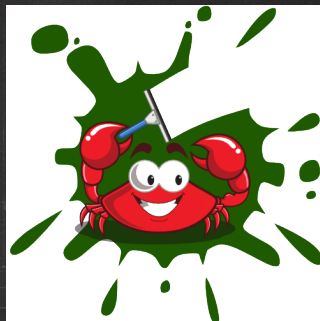
NETFLIX

OSS



# Use Scumblr to Watch for External Attackers

- Search for mentions of your brand online
  - Google, Bing, eBay, Pastebin, Twitter etc.
  - Look for discussions of dos attacks, fraud and user accounts
- Build workflows to notify and respond
  - Capture safely and anonymously using Sketchy



NETFLIX

OSS




# Tooling and Infrastructure



# Fast Start Amazon Machine Images

<https://github.com/Answers4AWS/netflixoss-ansible/wiki/AMIs-for-NetflixOSS>

- Pre-built AMIs for 
  - Asgard – developer self service deployment console
  - Aminator – build system to bake code onto AMIs
  - Edda – historical configuration database
  - Eureka – service registry
  - Simian Army – Janitor Monkey, Chaos Monkey, Conformity Monkey
- NetflixOSS Cloud Prize Contribution
  - Produced by Answers4aws – Peter Sankauskas





# Fast Setup CloudFormation Templates

<http://answersforaws.com/resources/netflixoss/cloudformation/>

**AnsWerS**

- CloudFormation templates for
  - Asgard – developer self service deployment console
  - Aminator – build system to bake code onto AMIs
  - Edda – historical configuration database
  - Eureka – service registry
  - Simian Army – Janitor Monkey for cleanup,

**NETFLIX** | **OSS**



# **CloudFormation Walk-Through for Asgard**

**(Repeat for Prod, Test and Audit Accounts)**

# NetflixOSS CloudFormation Templates

By using our CloudFormation templates, you can bring up many of the NetflixOSS services in just a few clicks. We have templates for:

- Aminator
- Asgard
- Edda
- Eureka
- Simian Army (Chaos Monkey, etc)

To use them, just:

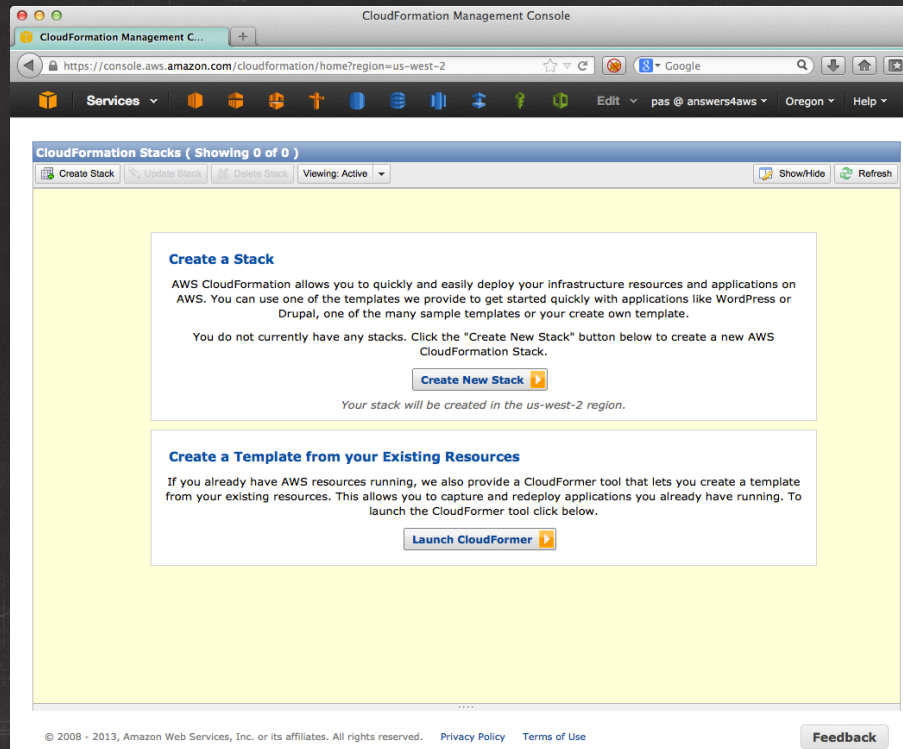
1. Clone the [repository](#)

```
git clone https://github.com/Answers4AWS/netflixoss-ansible.git
```

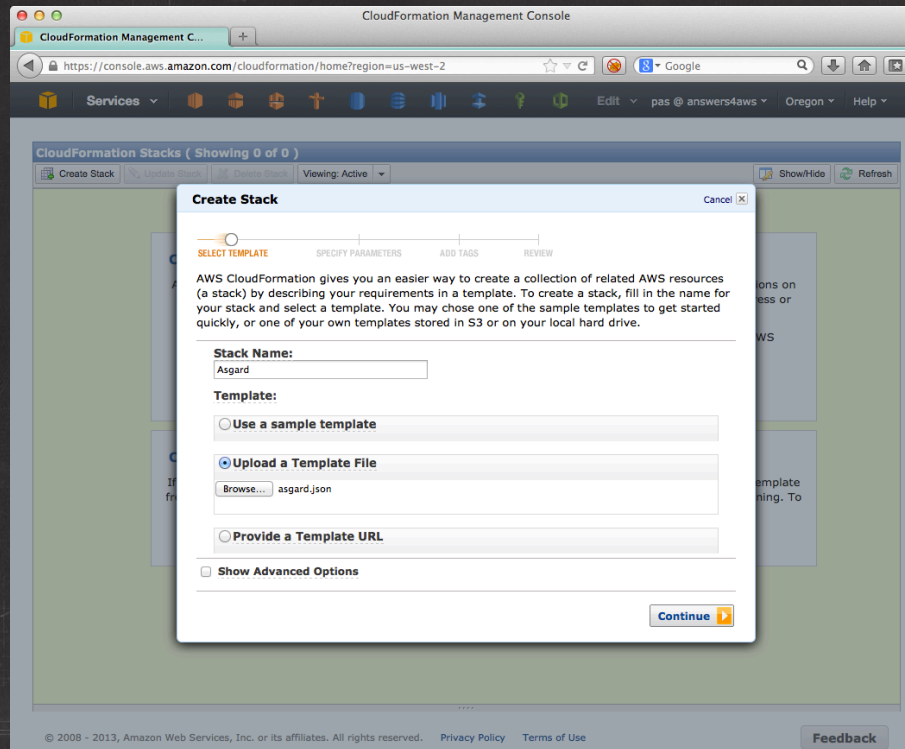
2. Log in to the AWS Web Console
3. Go to the CloudFormation page
4. Create a New Stack, and select the desired CloudFormation template.



# Setting up Asgard – Step 1 Create New Stack



# Setting up Asgard – Step 2 Select Template



# Setting up Asgard – Step 3 Enter IP and Keys

CloudFormation Management Console

CloudFormation Management C... | https://console.aws.amazon.com/cloudformation/home?region=us-west-2

Services | Edit | pas @ answers4aws | Oregon | Help

CloudFormation Stacks ( Showing 0 of 0 )

Create Stack | Update Stack | Delete Stack | Viewing: Active | Show/Hide | Refresh

**Create Stack** [Cancel]

SELECT TEMPLATE | **SPECIFY PARAMETERS** | ADD TAGS | REVIEW

**Stack Description:**NetflixOSS Asgard 1.3.1 - Template by Answers for AWS

**Specify Parameters**  
Below are the parameters associated with your CloudFormation template. You may review and proceed with the default parameters or make customizations as needed below.

**YourIpAddress**   
Your IP address

**KeyPairName**   
Name of an existing EC2 KeyPair to enable SSH access to the instance

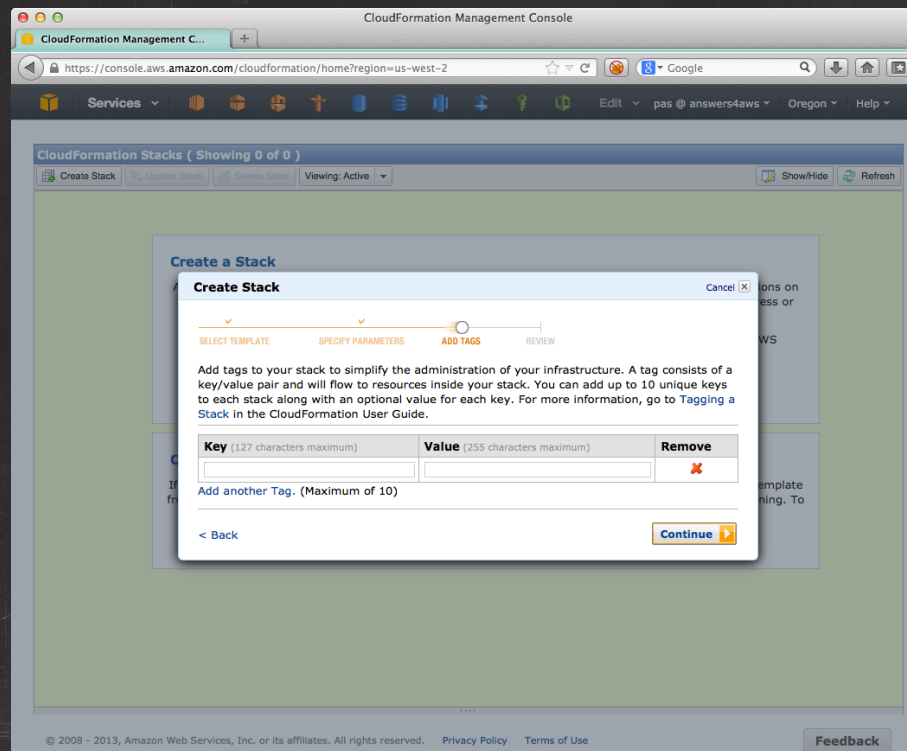
**InstanceType**   
EC2 instance type to launch for Application servers

< Back | Continue >

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. | Privacy Policy | Terms of Use | Feedback



# Setting up Asgard – Step 4 Skip Tags



The screenshot shows the AWS CloudFormation Management Console in a web browser. The main heading is "CloudFormation Stacks ( Showing 0 of 0 )". Below this, there are buttons for "Create Stack", "Update Stack", and "Delete Stack", along with a "Viewing: Active" dropdown. A "Create a Stack" modal window is open, showing a progress bar with four steps: "SELECT TEMPLATE", "SPECIFY PARAMETERS", "ADD TAGS" (which is the current step), and "REVIEW". The "ADD TAGS" step includes a text area for adding tags, a table with columns "Key (127 characters maximum)", "Value (255 characters maximum)", and "Remove", and a "Continue" button. The background of the console shows a list of stacks.

CloudFormation Management Console

CloudFormation Stacks ( Showing 0 of 0 )

Create Stack Update Stack Delete Stack Viewing: Active Show/Hide Refresh

Create a Stack

Create Stack

SELECT TEMPLATE SPECIFY PARAMETERS ADD TAGS REVIEW

Add tags to your stack to simplify the administration of your infrastructure. A tag consists of a key/value pair and will flow to resources inside your stack. You can add up to 10 unique keys to each stack along with an optional value for each key. For more information, go to [Tagging a Stack](#) in the CloudFormation User Guide.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
		X

Add another Tag. (Maximum of 10)

< Back Continue

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

# Setting up Asgard – Step 5 Confirm

The screenshot shows the AWS CloudFormation Management Console with a 'Create Stack' dialog box open. The dialog is in the 'REVIEW' step, indicated by a progress bar at the top. The background shows the 'CloudFormation Stacks' page with a 'Create Stack' button and a 'Show/Hide' button.

**Create Stack**

SELECT TEMPLATE ✓ SPECIFY PARAMETERS ✓ ADD TAGS REVIEW

Please review the information below, then click Continue to create the stack.

**Stack Information** [Edit Stack](#)

<b>Stack Name:</b>	Asgard
<b>Stack Description:</b>	NetflixOSS Asgard 1.3.1 - Template by Answers for AWS
<b>Template:</b>	https://cf-templates-r0loly2q3cpj-us-west-2.s3.amazonaws.com/2013309Mir-asgard.json
<b>IAM Acknowledgement:</b>	false

**Parameters** [Edit Parameters](#)

<b>YourIpAddress</b>	123.4.5.6
<b>KeyPairName</b>	answersforaws
<b>InstanceType</b>	m1.medium

**Notification** [Edit Notification](#)

<b>Notification:</b>	none
<b>Creation Timeout (minutes):</b>	none
<b>Rollback on Failure:</b>	true

[< Back](#) [Continue](#)

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Feedback](#)

# Setting up Asgard – Step 6 Watch CloudFormation

CloudFormation Management Console

CloudFormation Management C... x my ip - Google Search x +

https://console.aws.amazon.com/cloudformation/home?region=us-west-2#ConsoleState:view my ip

Services Edit pas @ answers4aws Oregon Help

CloudFormation Stacks ( Showing 1 of 1 )

Create Stack Update Stack Delete Stack Viewing: Active Show/Hide Refresh

Name	Created	Status	Description
Asgard	2013-11-05 11:24:44 UTC-8	CREATE_IN_PROGRESS	NetflixOSS Asgard 1.3.1 - Template by Answers for AWS

Stack: Asgard

Description Outputs Resources Events Template Parameters Tags

Stack Events Refresh

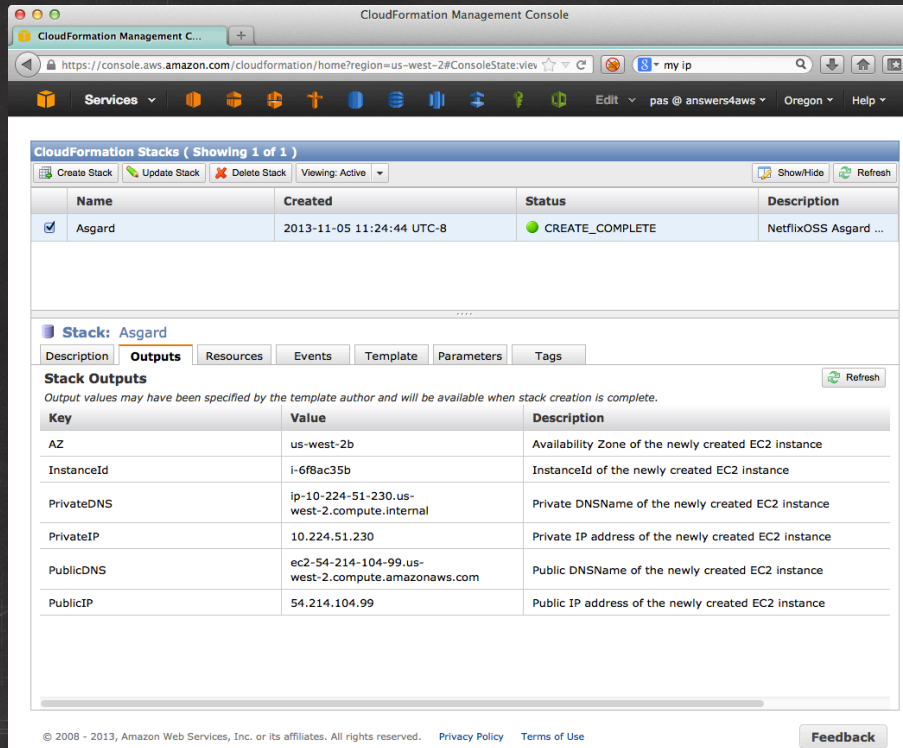
Logical ID	Physical ID	Status	Reason
AsgardInstance	i-6f8ac35b	CREATE_IN_PROGRESS	Resource creation
AsgardInstance		CREATE_IN_PROGRESS	
AsgardSecurityGroup	Asgard-AsgardSecurityGroup-M31XOSUGOWLV	CREATE_COMPLETE	
AsgardSecurityGroup	Asgard-AsgardSecurityGroup-M31XOSUGOWLV	CREATE_IN_PROGRESS	Resource creation
AsgardSecurityGroup		CREATE_IN_PROGRESS	
Asgard	arn:aws:cloudformation:us-west-2:658794617753:stack/Asgard/ece97390-464f-11e3-a042-500160d4da18	CREATE_IN_PROGRESS	User Initiated

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback





# Setting up Asgard – Step 7 Find PublicDNS Name



CloudFormation Management Console

CloudFormation Management C... | my ip

Services | Edit | pas @ answers4aws | Oregon | Help

CloudFormation Stacks ( Showing 1 of 1 )

Create Stack | Update Stack | Delete Stack | Viewing: Active | Show/Hide | Refresh

Name	Created	Status	Description
Asgard	2013-11-05 11:24:44 UTC-8	CREATE_COMPLETE	NetflixOSS Asgard ...

Stack: Asgard

Description | **Outputs** | Resources | Events | Template | Parameters | Tags

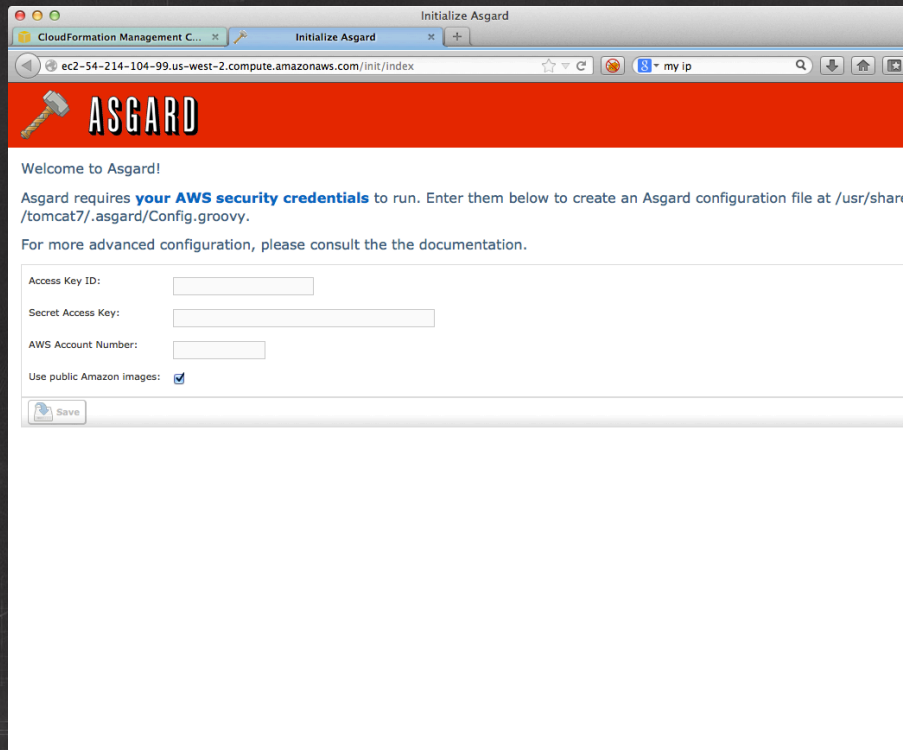
Stack Outputs

Output values may have been specified by the template author and will be available when stack creation is complete. Refresh

Key	Value	Description
AZ	us-west-2b	Availability Zone of the newly created EC2 instance
InstanceId	i-6f8ac35b	InstanceId of the newly created EC2 instance
PrivateDNS	ip-10-224-51-230.us-west-2.compute.internal	Private DNSName of the newly created EC2 instance
PrivateIP	10.224.51.230	Private IP address of the newly created EC2 instance
PublicDNS	ec2-54-214-104-99.us-west-2.compute.amazonaws.com	Public DNSName of the newly created EC2 instance
PublicIP	54.214.104.99	Public IP address of the newly created EC2 instance

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use Feedback

# Open Asgard – Step 8 Enter Credentials




The screenshot shows a web browser window titled "Initialize Asgard". The address bar displays the URL "ec2-54-214-104-99.us-west-2.compute.amazonaws.com/init/index". The page features a red header with the Asgard logo (a hammer icon and the word "ASGARD"). Below the header, the text "Welcome to Asgard!" is followed by instructions: "Asgard requires **your AWS security credentials** to run. Enter them below to create an Asgard configuration file at /usr/share/tomcat7/.asgard/Config.groovy." and a link to documentation. The form contains four input fields: "Access Key ID:", "Secret Access Key:", and "AWS Account Number:". Below these is a checkbox labeled "Use public Amazon images:" which is checked. A "Save" button is at the bottom of the form.

Initialize Asgard

CloudFormation Management C... Initialize Asgard

ec2-54-214-104-99.us-west-2.compute.amazonaws.com/init/index my ip

 **ASGARD**

Welcome to Asgard!

Asgard requires **your AWS security credentials** to run. Enter them below to create an Asgard configuration file at /usr/share/tomcat7/.asgard/Config.groovy.


For more advanced configuration, please consult the the documentation.

Access Key ID:

Secret Access Key:

AWS Account Number:

Use public Amazon images: ☒

 Save

# Use Asgard – AWS Self Service Portal

The screenshot shows the Asgard prod web interface in a browser window. The address bar displays the URL `ec2-54-214-104-99.us-west-2.compute.amazonaws.com`. The page has a red header with the "ASGARD prod" logo and a "us-east-1" region selector. Below the header is a navigation bar with icons for Home, App, AMI, Cluster, ELB, EC2, SDB, SNS, SQS, RDS, and Task. The main content area is titled "Welcome to Asgard in prod in us-east-1 (Virginia)". It features three columns of links: "Abstractions" (Manage Applications, Push images to one or more Application AutoScaling Groups, Configure outbound security access for Applications), "AWS Objects" (Manage Images, Manage Auto Scaling Groups, Manage Load Balancers, Manage Launch Configurations, Manage Security Groups, Manage Running Instances), and "Asgard Tasks" (Monitor Background Tasks). At the bottom, there are links for Help, Environments, and a "Jump to an instance" section with a text input field and a "Go" button. A "Diagnostics" section at the bottom left provides system information: AWS Account: prod, AWS Region: us-east-1, AWS Accounts: (658794617753=prod), Eureka: There is no Eureka URL for prod in us-east-1, Hostname: ip-10-224-51-230, IP: 10.224.51.230, Version: 1.3.1, and Build: id=2013-10-25\_04-30-29 build#110 @33d7e9bca61730e1f3d7731d2ff3367fa12af2d3.



# Use Asgard - Manage Red/Black Deployments

asgardprod/us-east-1/cluster/show/obiwan This cluster contains two ASGs

ASGARD prod us-east-1 CMC

Home App AMI Cluster ELB EC2 SDB SNS SQS RDS Task

Manage Cluster of Sequential Auto Scaling Groups

Recommended next step: Switch traffic to the preferred group, then delete legacy group

**obiwan-v063**

Launch and Terminate are disabled

Resize to 9 min / 12 max

Delete Disable Enable

9 instances grouped by state

Count	State	Build	ELB	Disc
9	InService	580	OUT_OF_SERVICE	

No traffic on old version

**obiwan-v064**

Resize to 9 min / 12 max

Delete Disable Enable

9 instances grouped by state

Count	State	Build	ELB	Disc
9	InService	583	UP	

Live traffic on new version

Create Next Group: [Advanced Options](#)

**obiwan-v065**

AMI Image ID: 179123456789/obiwan-41.2-1417309

Filter

[Show more AMIs](#)

Instance Type: m1.large \$230.400/mo

Filter

Instance Counts: Min: 9 Desired: 9 Max: 12

After launch: ☒ Wait for Discovery health check pass

[Create Next Group obiwan-v065](#)

NETFLIX

OSS

The background of the slide is a dark, textured surface with a complex, glowing circuit board pattern. The pattern consists of numerous thin, light-colored lines and dots that form a dense, interconnected network, resembling a high-tech or futuristic design. The lines vary in thickness and direction, creating a sense of depth and complexity. The overall effect is a high-tech, digital aesthetic.

# **Track AWS Spend in Detail with ICE**

[illegible]



## Setting up ICE

- Visit github site for instructions
- Currently depends on HiCharts
  - Non-open source package license
  - Free for non-commercial use
  - Download and license your own copy
  - We can't provide a pre-built AML – sorry!
- Long term plan to make ICE fully OSS
  - Anyone want to help?

# Build Pipeline Automation

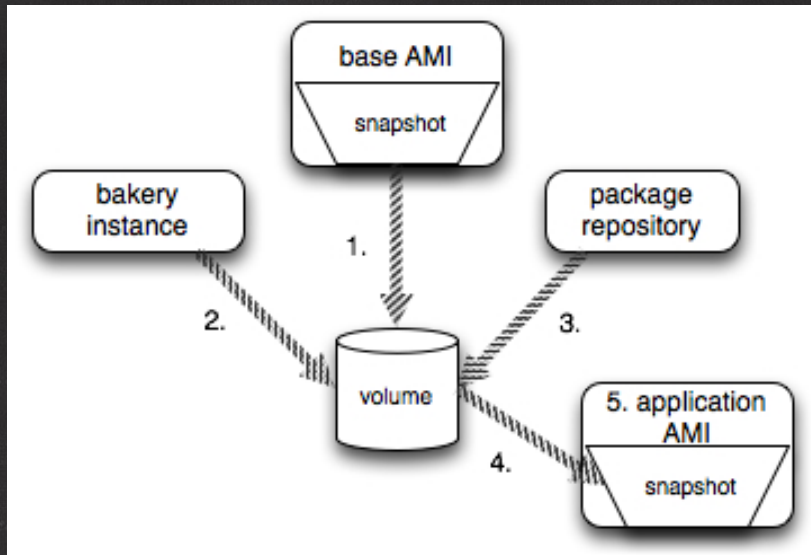
Jenkins in the Cloud auto-builds NetflixOSS Pull Requests

<http://www.cloudbees.com/jenkins>



# Automatically Baking AMIs with Aminator

- AutoScaleGroup instances should be identical
- Base plus code/config
- Immutable instances
- Works for 1 or 1000...
- Aminator Launch
  - Use Asgard to start AMI or
  - CloudFormation Recipe



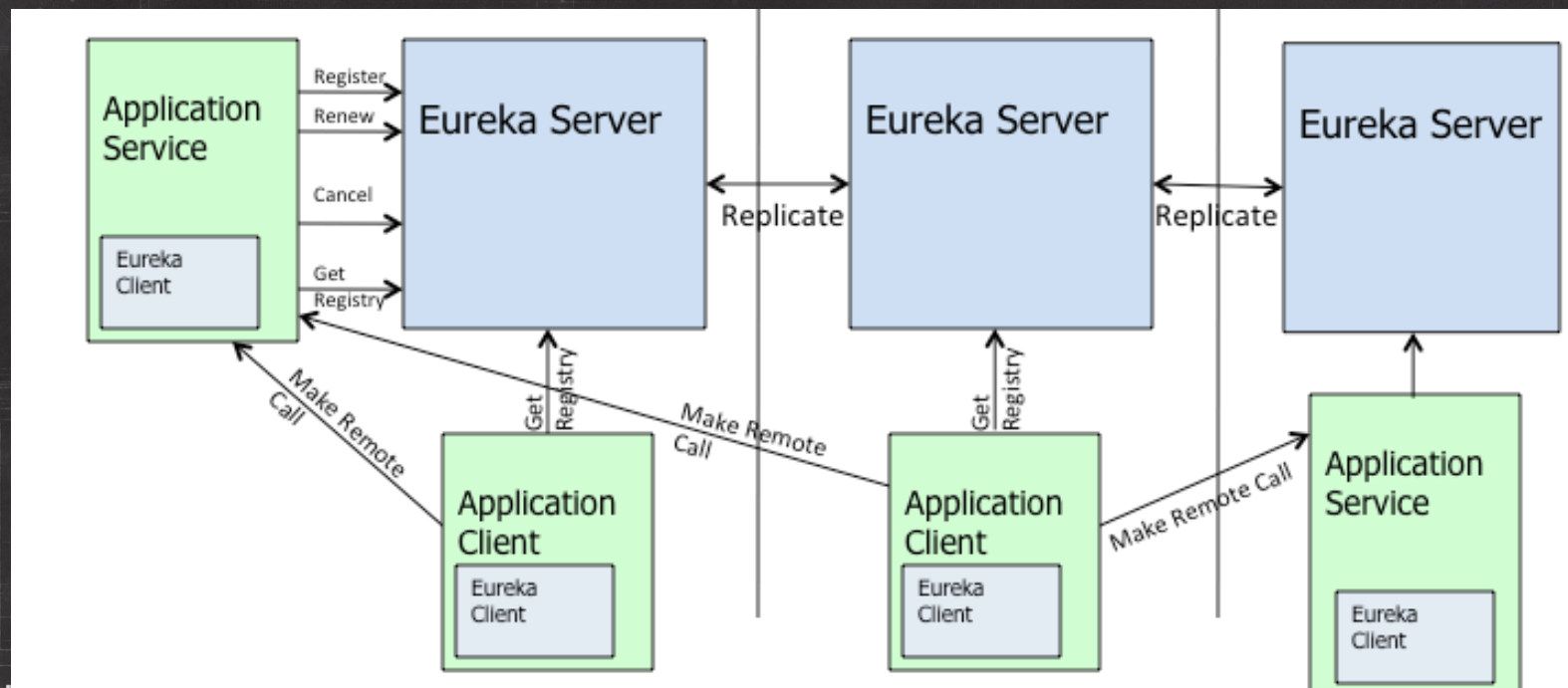


# Discovering your Services - Eureka

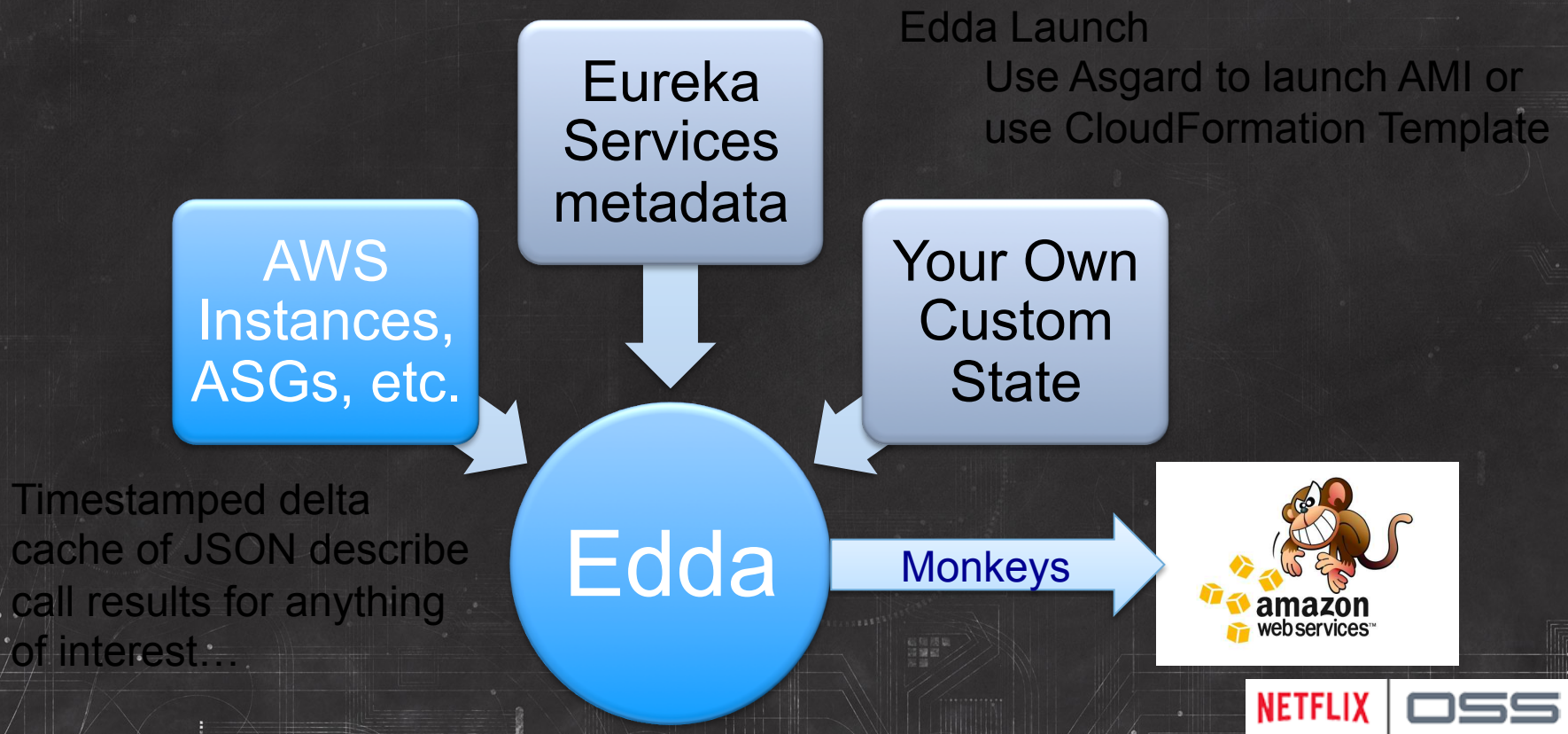
EVCACHE	ami-e794f78e (3), ami-c579ecac (1),	us-east-1c (3), us-east-1d (1),	UP (4) - <a href="#">i-9fdt</a> , <a href="#">i-08c</a> , <a href="#">i-78e</a> , <a href="#">i-45f</a>
---------	-------------------------------------	---------------------------------	---

- Map applications by name to
  - AMI, instances, Zones
  - IP addresses, URLs, ports
  - Keep track of healthy, unhealthy and initializing instances
- Eureka Launch
  - Use Asgard to launch AMI or use CloudFormation Template

# Deploying Eureka Service – 1 per Zone



# Searchable state history for a Region / Account





# Edda Query Examples

Find any instances that have ever had a specific public IP address

```
$ curl "http://edda/api/v2/view/instances;publicIpAddress=1.2.3.4;_since=0"
["i-0123456789","i-012345678a","i-012345678b"]
```

Show the most recent change to a security group

```
$ curl "http://edda/api/v2/aws/securityGroups/sg-0123456789;_diff;_all;_limit=2"
--- /api/v2/aws.securityGroups/sg-0123456789;_pp;_at=1351040779810
+++ /api/v2/aws.securityGroups/sg-0123456789;_pp;_at=1351044093504
@@ -1,33 +1,33 @@
{
...
  "ipRanges" : [
    "10.10.1.1/32",
    "10.10.1.2/32",
+   "10.10.1.3/32",
-   "10.10.1.4/32"
  ]
}
```

# Archaius – Property Console

Persisted Properties Console : TEST us-east-1

Properties Diff Journal Logs Documentation Reports

Home / FastProperty Explorer /

### Create New Fast Property in test

Name: prop.foo  
Value: 97  
Constraints: int:0-99  
Description: property to control %  
UpdatedBy: stonse@netflix.com  
Notification Email:  
CMC:  
Scope: Not Selected

Select Scope

#### Scope Selection

Region: us-east-1 (Virginia)

Application: helloworld 1

Stack:

Cluster: helloworld-example 12

ASG: helloworld-example-v140 14

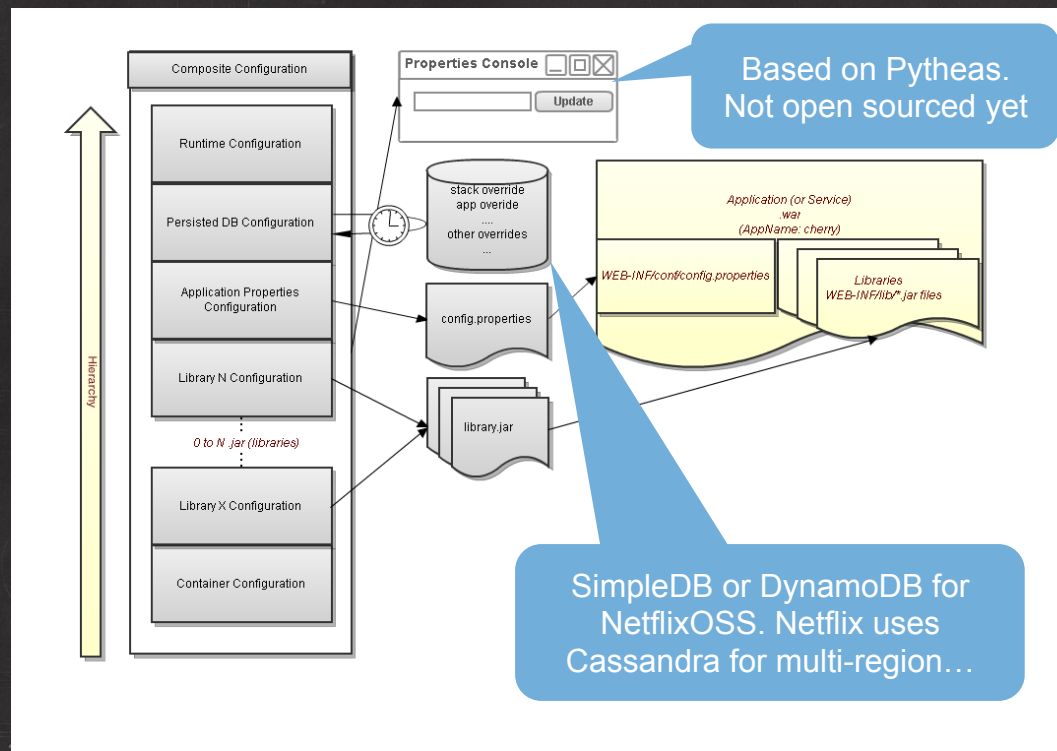
Select ASG Skip Done Repeat Selection

#### Quick Impact View

Total instances: 2  
Current Scope

region	us-east-1
appld	helloworld
cluster	helloworld-example

# Archaius library – configuration management







# Data Storage and Access

# Data Storage Options

- RDS for MySQL
  - Deploy using Asgard
- DynamoDB
  - Fast, easy to setup and scales up from a very low cost base
- Cassandra
  - Provides portability, multi-region support, very large scale
  - Storage model supports incremental/immutable backups
  - Priam: easy deployment automation for Cassandra on AWS

NETFLIX

OSS



## Priam – Cassandra co-process

- Runs alongside Cassandra on each instance
- Fully distributed, no central master coordination
- S3 Based backup and recovery automation
- Bootstrapping and automated token assignment.
- Centralized configuration management
- RESTful monitoring and metrics
- Underlying config in SimpleDB (Cass\_turtle for MR)

NETFLIX

OSS



# Astyanax Cassandra Client for Java

- Features

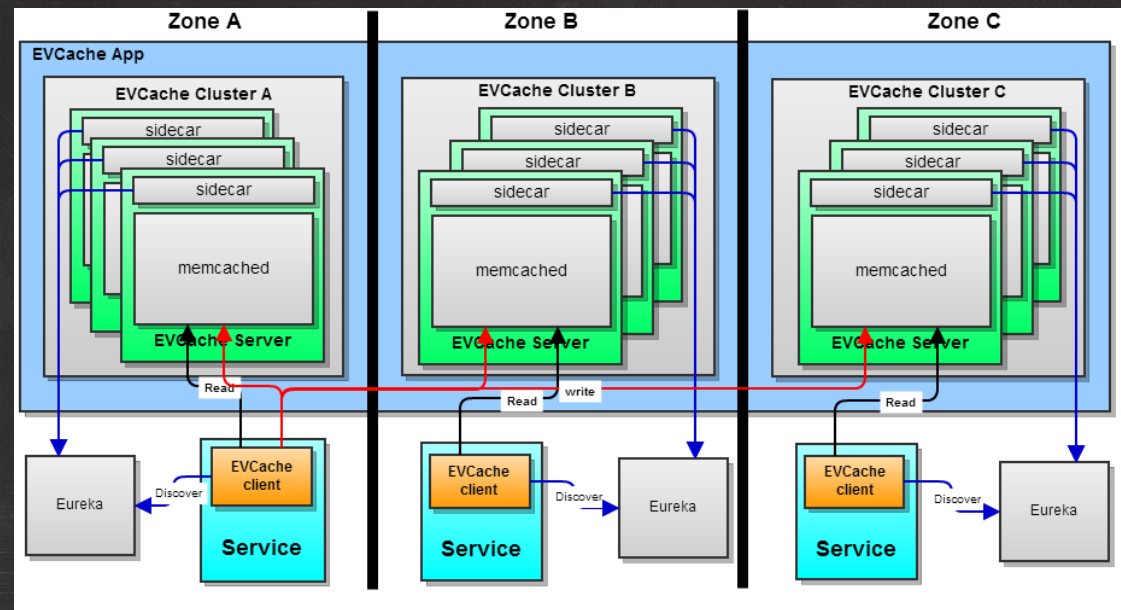
- Abstraction of connection pool from RPC protocol
- Fluent Style API
- Operation retry with backoff
- Token aware
- Batch manager
- Many useful recipes
- Entity Mapper based on JPA annotations

# Cassandra Astyanax Recipes

- Distributed row lock (without needing zookeeper)
- Multi-region row lock
- Uniqueness constraint
- Multi-row uniqueness constraint
- Chunked and multi-threaded large file storage
- Reverse index search
- All rows query
- Durable message queue
- Contributed: High cardinality reverse index

# EVCache - Low latency data access

- multi-AZ and multi-Region replication
- Ephemeral data, session state (sort of)
- Client code
- Memcached

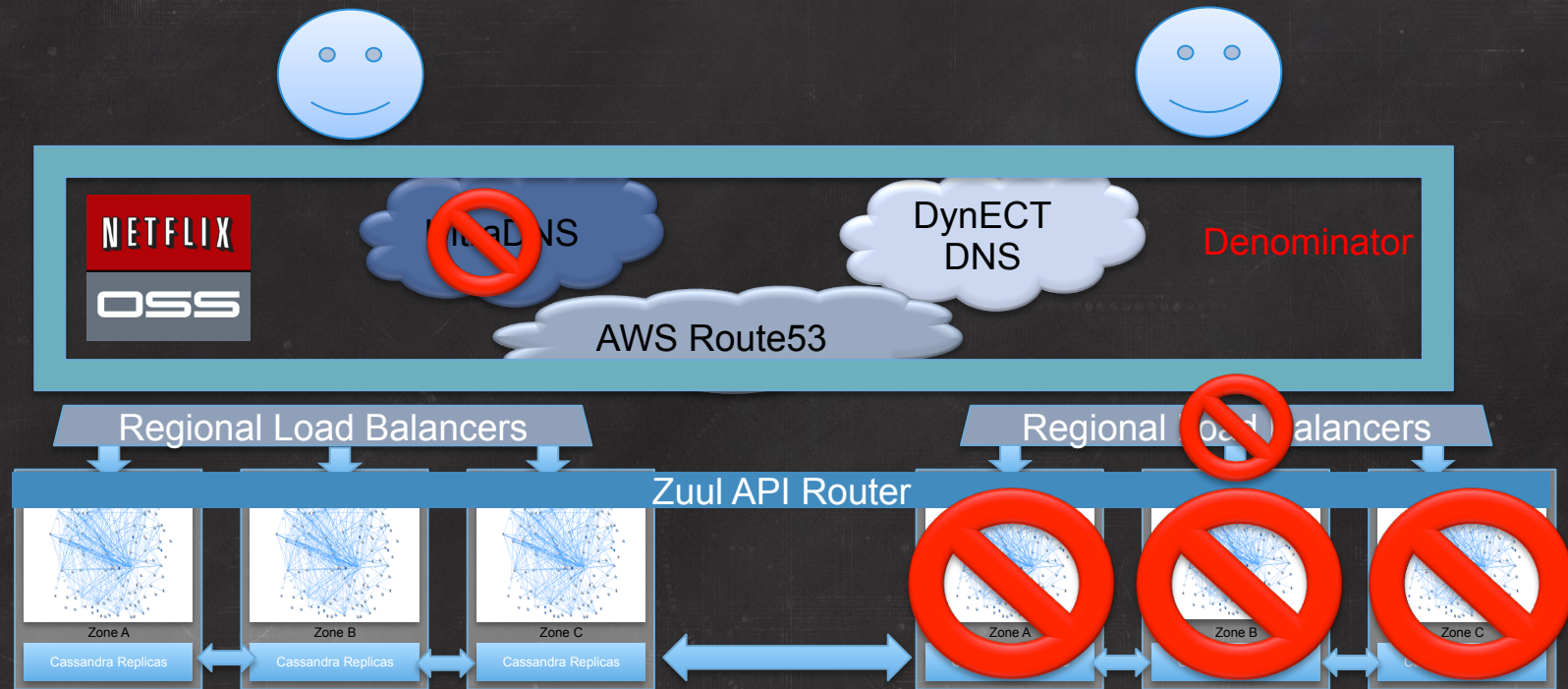






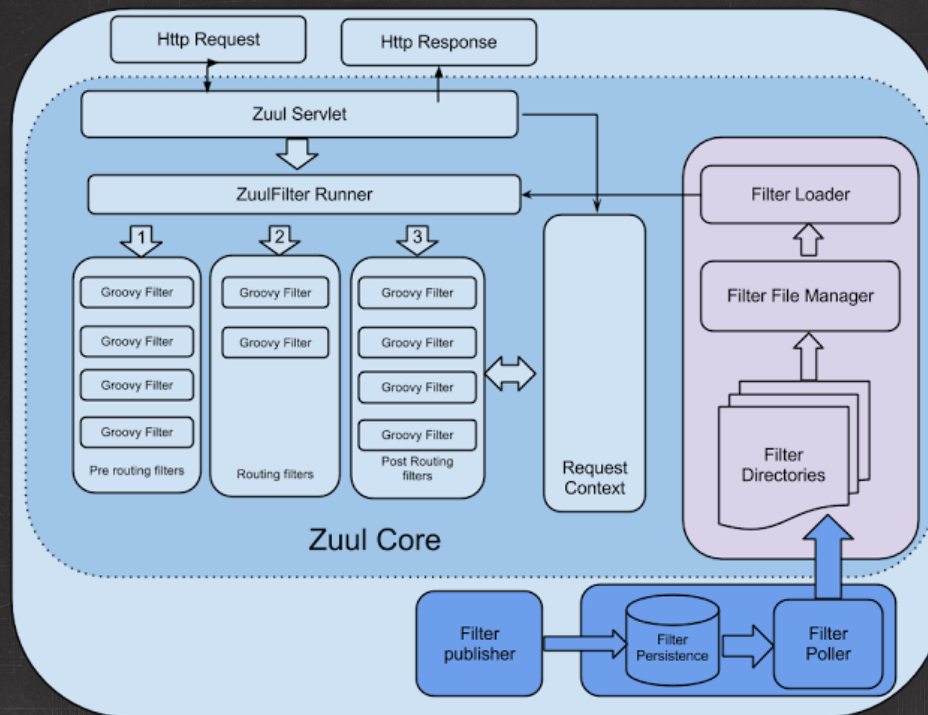
# Routing Customers to Code

# Denominator: DNS for Multi-Region Availability



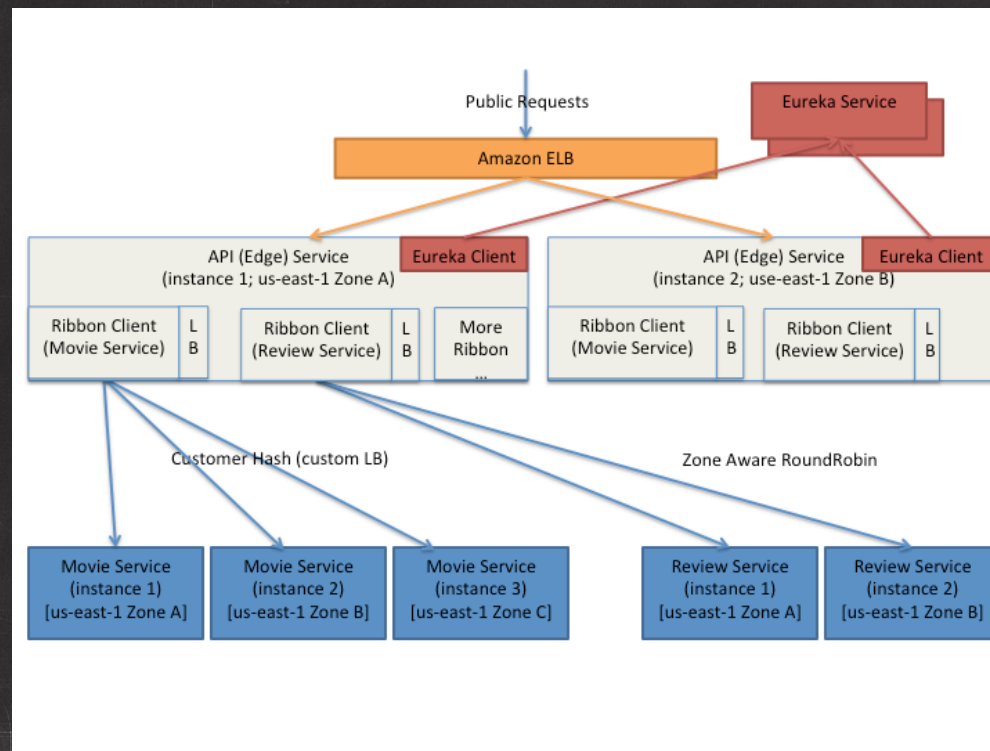
Denominator – manage traffic via multiple DNS providers with Java code

# Zuul – Smart and Scalable Routing Layer



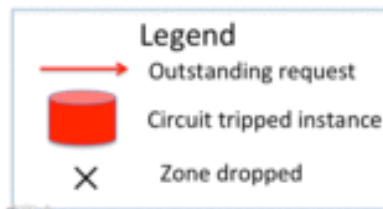


# Ribbon library for internal request routing



# Ribbon – Zone Aware LB

Average Active Requests = outstanding requests / (total instances - tripping instances)



NETFLIX

OSS

# Karyon - Common server container



- Bootstrapping
  - Dependency & Lifecycle management via Governor.
  - Service registry via Eureka.
  - Property management via Archaius
  - Hooks for Latency Monkey testing
  - Preconfigured status page and healthcheck servlets



# Karyon

- Embedded Status Page Console

- Environment
- Eureka
- JMX



PLATFORMSERVICE v1.0 us-east-1 prod

Asynux Counters Debug Data Diagnostics Discovery Environment Info Jars JMX NWS Properties SLA Tracers Refresh Machine Readable Filter

Filter:

- Catalina
- JMXImplementation
- Library-Atlas\_plugin-com.netflix.library.management
- Library-epicplugin-com.netflix.library.management
- Library-nfsasynux-com.netflix.library.management
- Library-platform-com.netflix.library.management
- com.netflix.MonitoringResources
  - APPLICATION
    - ASTYANAX
      - cassturtle.platformservice.asynux
        - cassturtle.platformservice
- PLATFORM
- com.netflix.messaging
- com.netflix.servo
  - com.netflix.servo
    - ActiveChukwaConsumers
    - ActiveChukwaSenders
    - ChunkQueueSize
    - DiscoveryClient\_Cancel
    - DiscoveryClient\_Failed
    - DiscoveryClient\_FetchRegistry
    - DiscoveryClient\_GetServiceUrlsFromDNS
    - DiscoveryClient\_Refresh
    - DiscoveryClient\_RefreshDelta
    - DiscoveryClient\_Register
    - DiscoveryClient\_Renew
    - DiscoveryClient\_Reregister
    - DiscoveryClient\_Retry
    - LBStats\_AvailableZones
    - LBStats\_CircuitBreakerTrippedCount
    - LoadBalancer\_AllServerList
    - LoadBalancer\_ChooseServer
    - LoadBalancer\_UpServerList
    - PoolSize
    - SQS-changeMessageVisibilityTimeoutBatchCount

com.netflix.servo.name=LBStats\_AvailableZones,\*

Attributes

Refresh Search:

Key	Name	Value
[{"class":"LoadBalancerStats","id":"akmsclient","level":"INFO","type":"INFORMATIONAL"}]	value	[us-east-1e, us-east-1d, us-east-1c]
[{"class":"LoadBalancerStats","id":"atlas_publish","level":"INFO","type":"INFORMATIONAL"}]	value	[us-east-1e, us-east-1d, us-east-1c]
[{"class":"LoadBalancerStats","id":"chukwaclient","level":"INFO","type":"INFORMATIONAL"}]	value	[us-east-1e, us-east-1d, us-east-1c]
[{"class":"LoadBalancerStats","id":"defaultRestClient","level":"INFO","type":"INFORMATIONAL"}]	value	[us-east-1e, us-east-1d, us-east-1c]
[{"class":"LoadBalancerStats","id":"epic_publish","level":"INFO","type":"INFORMATIONAL"}]	value	[us-east-1e, us-east-1d, us-east-1c]
[{"class":"LoadBalancerStats","id":"epicplugin","level":"INFO","type":"INFORMATIONAL"}]	value	[us-east-1e, us-east-1d, us-east-1c]

Showing 1 to 6 of 6 entries

Error: Visible: 698 Total: 698 Last updated: Mon Mar 04 2013 23:19:52

# Karyon



- Sample Service using Karyon available as "Hello-netflix-oss" on github

```
localhost:8989/hello-netflix-oss/rest/v1/hello/to/newbee
{
  Message: "Hello newbie from Netflix OSS"
}
```



# Availability



Either you break it, or users will

NETFLIX



NETFLIX | OSS

# Add some Chaos to your system



# Clean up your room! – Janitor Monkey

Works with Edda history to clean up after Asgard



NETFLIX

OSS



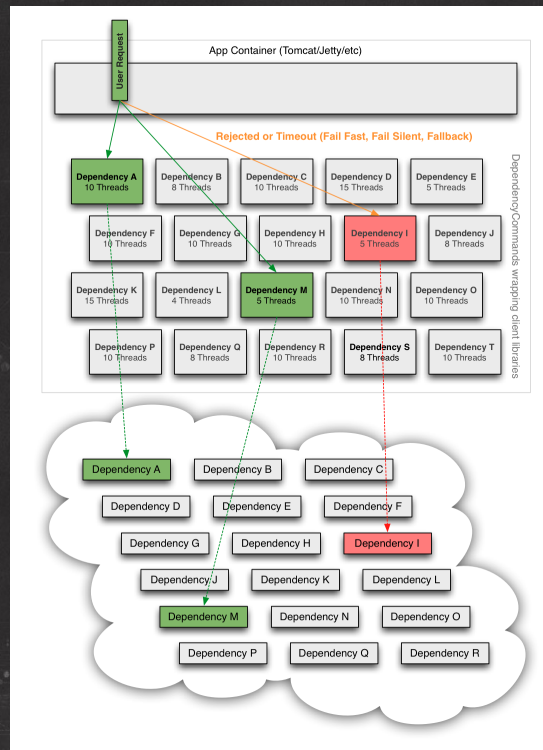
# Conformity Monkey

Track and alert for old code versions and known issues

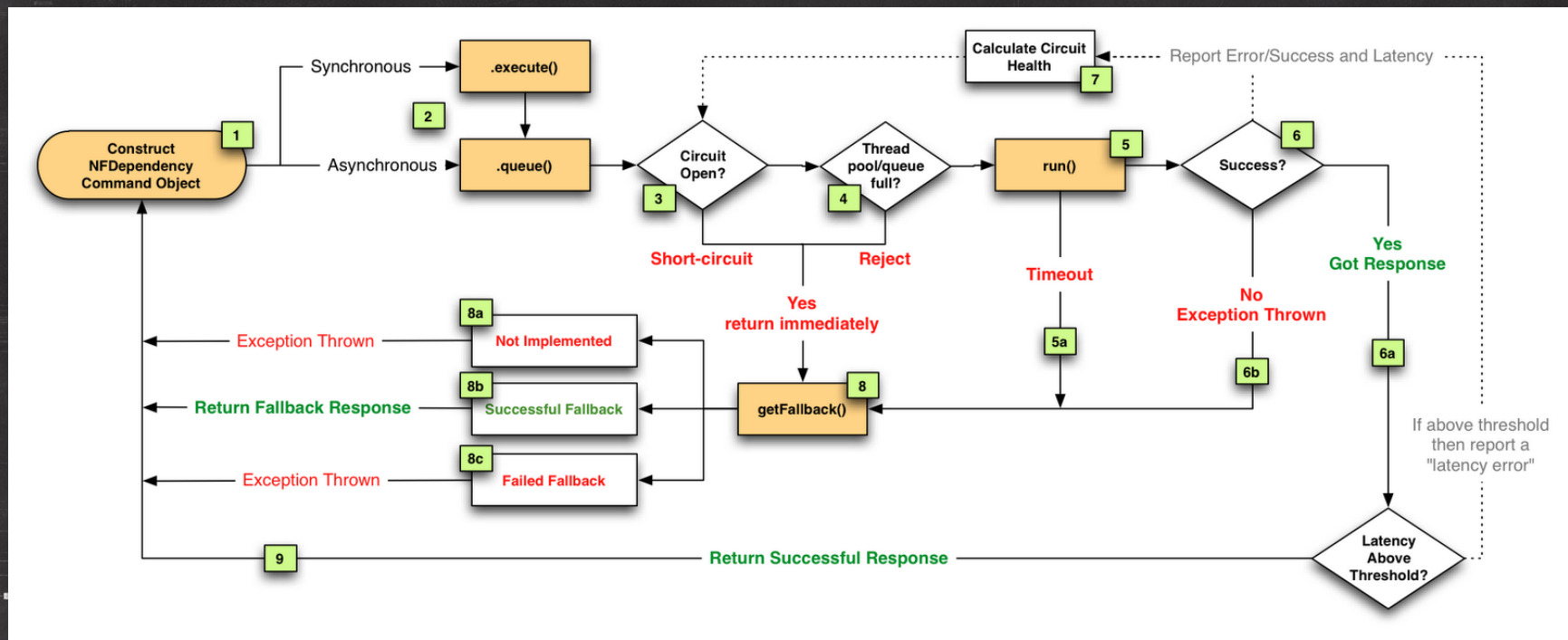
Walks Karyon status pages found via Edda



# Hystrix Circuit Breaker: Fail Fast -> recover fast



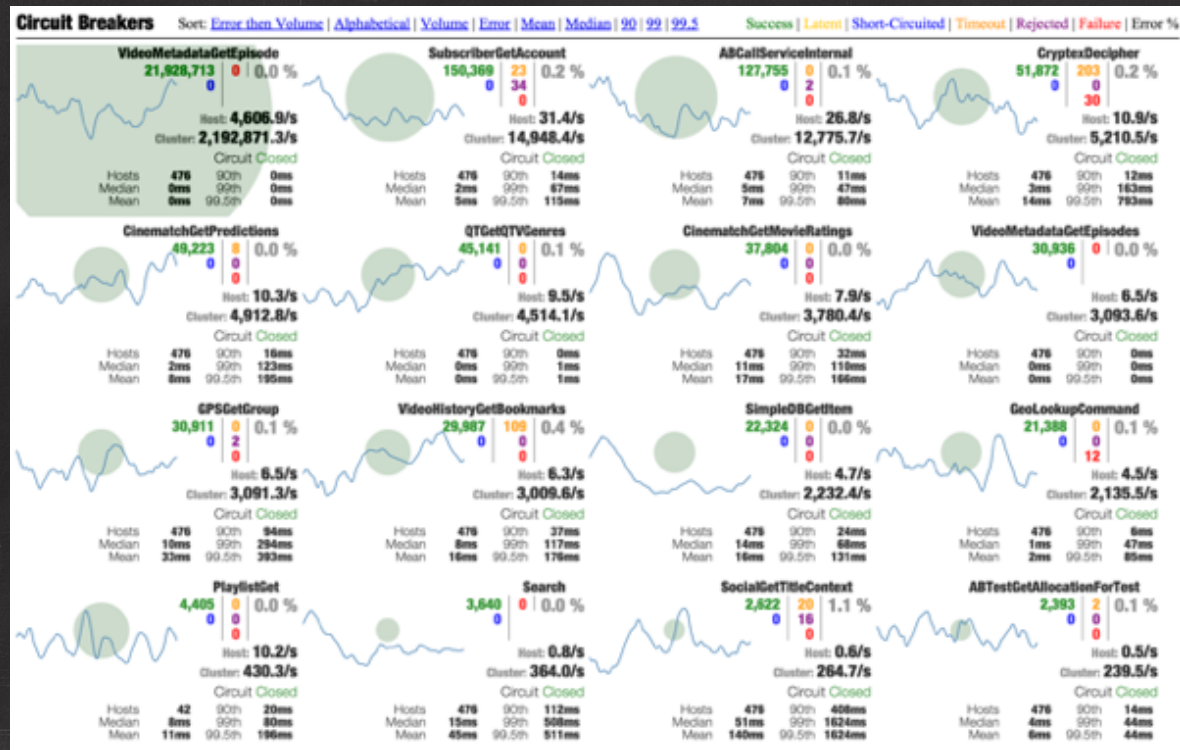
# Hystrix Circuit Breaker State Flow





# Turbine Dashboard

## Per Second Update Circuit Breakers in a Web Browser



NETFLIX

OSS



# Developer Productivity

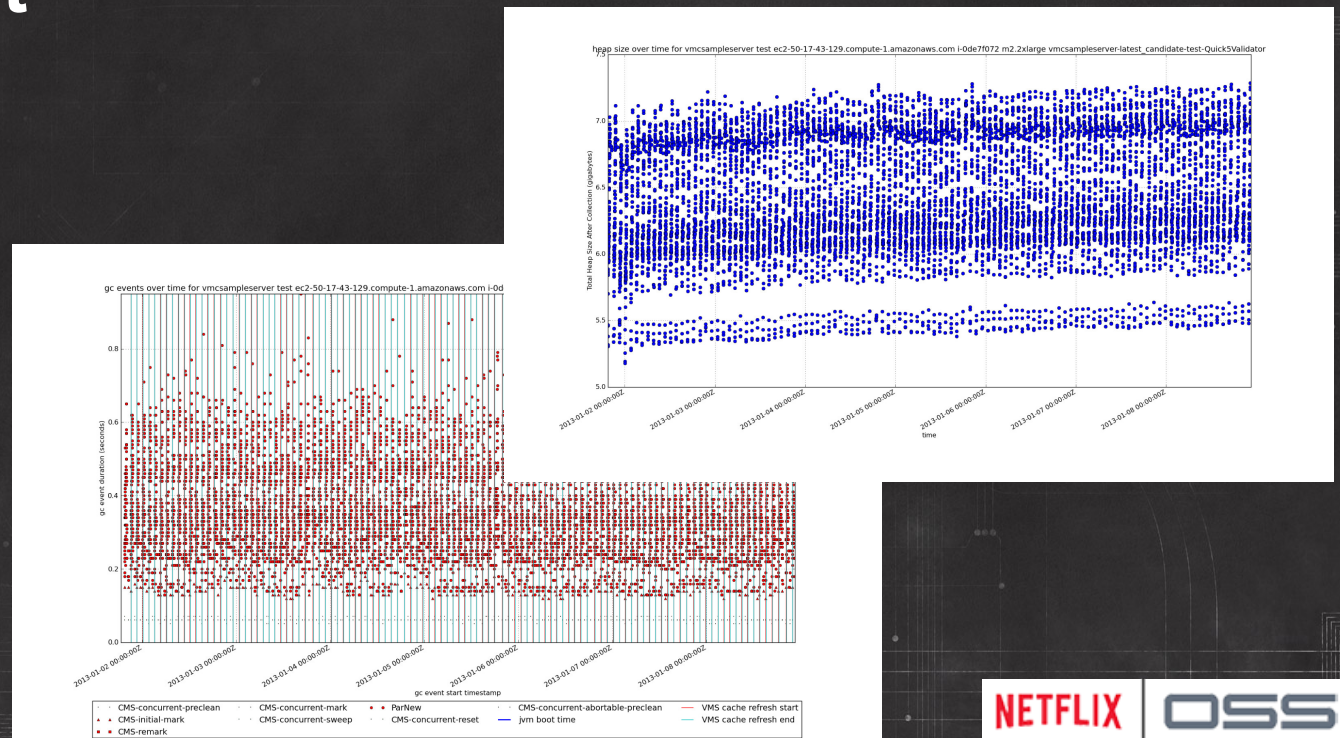
## Blitz4J – Non-blocking Logging

- Better handling of log messages during storms
- Replace sync with concurrent data structures.
- Extreme configurability
- Isolation of app threads from logging threads



# JVM Garbage Collection issues? GCViz!

- Convenient
- Visual
- Causation
- Clarity
- Iterative

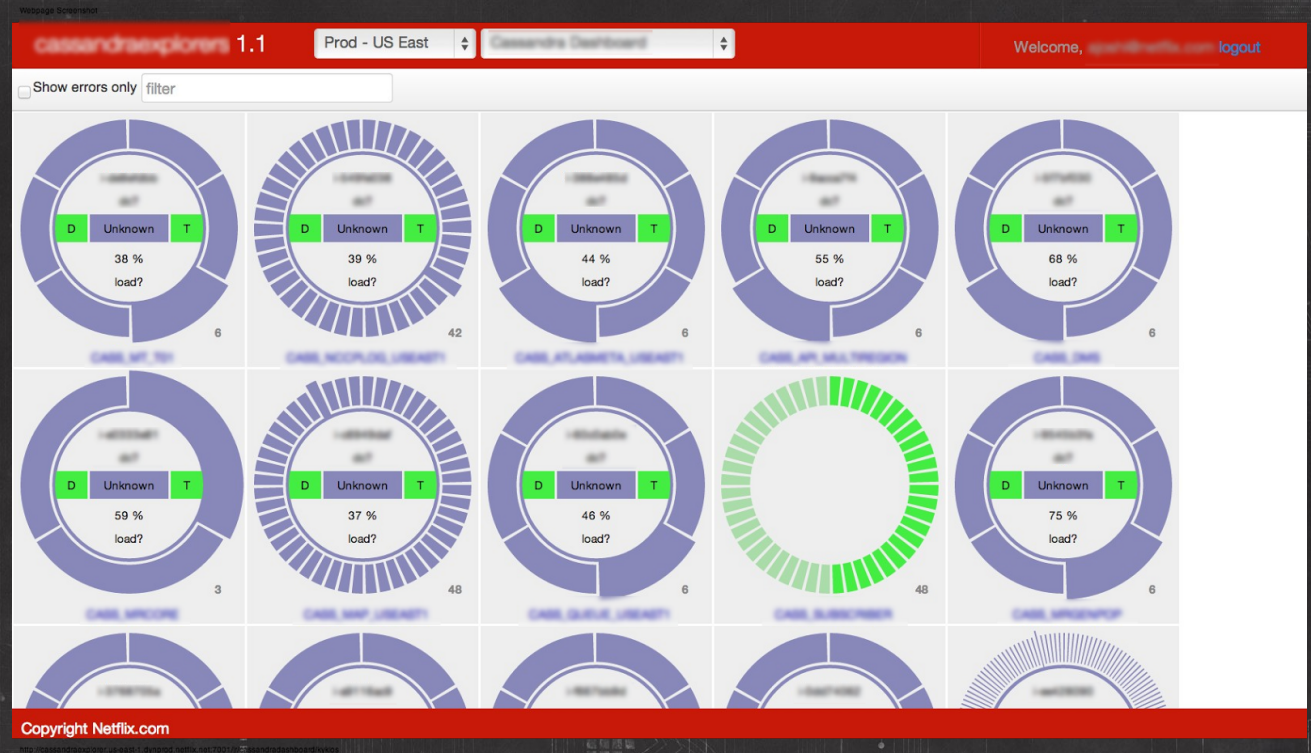


NETFLIX

OSS

# Pytheas – OSS based tooling framework

- Guice
- Jersey
- FreeMarker
- JQuery
- DataTables
- D3
- JQuery-UI
- Bootstrap





# RxJava - Functional Reactive Programming

- A Simpler Approach to Concurrency
  - Use Observable as a simple stable composable abstraction
- Observable Service Layer enables any of
  - conditionally return immediately from a cache
  - block instead of using threads if resources are constrained
  - use multiple threads
  - use non-blocking IO
  - migrate an underlying implementation from network based to in-memory cache

NETFLIX

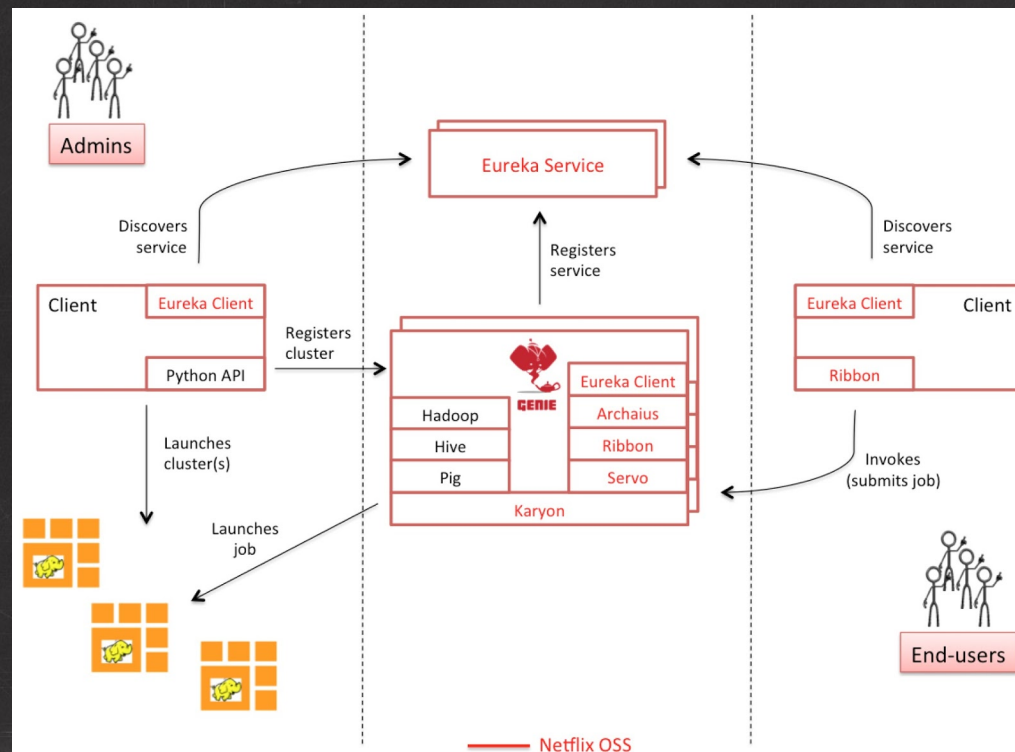
OSS



The image features a dark, textured background with intricate, glowing circuit-like patterns and geometric shapes. A solid blue horizontal bar is positioned at the top. The text "Big Data and Analytics" is centered in a white, bold, sans-serif font.

# Big Data and Analytics

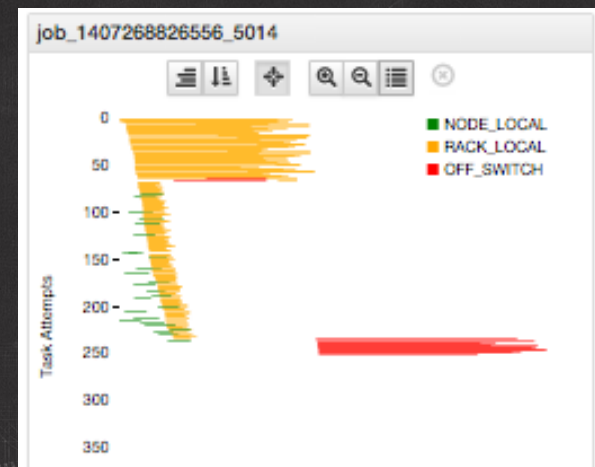
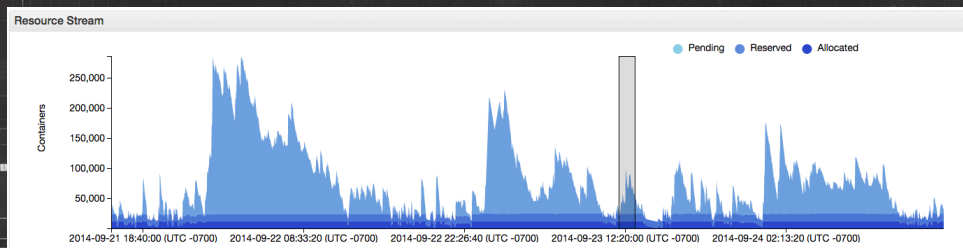
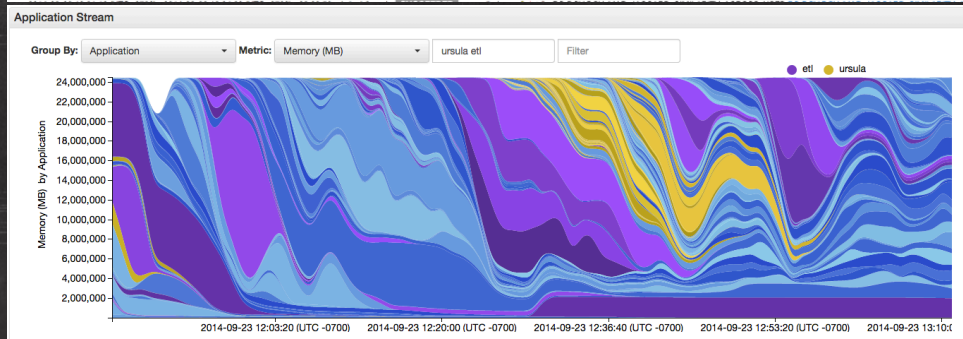
# Hadoop jobs - Genie



# Hadoop Monitoring and Visualization



Start	Stop	Duration	Cluster	Job Status	Username	Links	Genie Name	Genie ID	MR Job ID	MR Job Name	Dataoven ID	Map Tasks	Reduce Tasks
2014-08-08 10:54:24 (UTC -0700)	2014-08-08 11:19:07 (UTC -0700)	00:24:42	bdp_h2prod	SUCCEEDED	smedhekar		PG.DCV.DCV_MAP_JACCARD_SIMILARITY_ACROSS_USEF	PG.DCV.DCV_MAP_JACCARD_SIMILARITY_ACROSS_USEF	job_1406918339206_88197	PigLatin:"dov_map_jacc,similarity_across_users_20140805_"; pig_1407508491.PG.DCV.DCV_MAP_JACCARD_SIMILARIT	hive_1407508491.PG.DCV.DCV_MAP_JACCARD_SIMILARIT	44	1000
2014-08-08 11:01:05 (UTC -0700)	2014-08-08 11:18:10 (UTC -0700)	00:17:04	bdp_h2prod	SUCCEEDED	jaibcock		HV.DCV.PROCESS_SEARCH_LOGS_ALL_DATA_AUTOCOM	HV.DCV.PROCESS_SEARCH_LOGS_ALL_DATA_AUTOCOM	job_1406918339206_88273	insert overwrite table etl.search_logs_c_all(Stage-4)	hive_1407509303.HV.DCV.PROCESS_SEARCH_LOGS_ALL	40	0
2014-08-08 10:54:11 (UTC -0700)	2014-08-08 11:00:08 (UTC -0700)	00:05:56	bdp_h2prod	SUCCEEDED	jaibcock		HV.DCV.PROCESS_SEARCH_LOGS_ALL_DATA_AUTOCOM	HV.DCV.PROCESS_SEARCH_LOGS_ALL_DATA_AUTOCOM	job_1406918339206_88174	insert overwrite table etl.search_logs_c_all(Stage-2)	hive_1407509303.HV.DCV.PROCESS_SEARCH_LOGS_ALL	503	0
2014-08-08 10:43:22 (UTC -0700)	2014-08-08 10:53:25 (UTC -0700)	00:10:03	bdp_h2prod	SUCCEEDED	smedhekar		PG.DCV.DCV_MAP_JACCARD_SIMILARITY_ACROSS_USEF	PG.DCV.DCV_MAP_JACCARD_SIMILARITY_ACROSS_USEF	job_1406918339206_88103	PigLatin:"dov_map_jacc,similarity_across_users_20140805_"; pig_1407508491.PG.DCV.DCV_MAP_JACCARD_SIMILARIT	hive_1407508491.PG.DCV.DCV_MAP_JACCARD_SIMILARIT	2	400
2014-08-08 07:50:19 (UTC -0700)	2014-08-08 10:45:16 (UTC -0700)	02:54:56	bdp_h2prod	SUCCEEDED	jaibcock		HV.DCV.PROCESS_SEARCH_LOGS_ALL_DATA_AUTOCOM	HV.DCV.PROCESS_SEARCH_LOGS_ALL_DATA_AUTOCOM	job_1406918339206_86549	insert overwrite table etl.search_logs_c_all(Stage-11)	hive_1407509303.HV.DCV.PROCESS_SEARCH_LOGS_ALL	15	300
2014-08-08 10:31:28 (UTC -0700)	2014-08-08 10:42:21 (UTC -0700)	00:10:53	bdp_h2prod	SUCCEEDED	smedhekar		PG.DCV.DCV_MAP_JACCARD_SIMILARITY_ACROSS_USEF	PG.DCV.DCV_MAP_JACCARD_SIMILARITY_ACROSS_USEF	job_1406918339206_87977	PigLatin:"dov_map_jacc,similarity_across_users_20140805_"; pig_1407508491.PG.DCV.DCV_MAP_JACCARD_SIMILARIT	hive_1407508491.PG.DCV.DCV_MAP_JACCARD_SIMILARIT	800	400
2014-08-08 10:31:13 (UTC -0700)	2014-08-08 10:34:23 (UTC -0700)	00:03:12	bdp_h2prod	SUCCEEDED	smedhekar		PG.DCV.DCV_MAP_JACCARD_SIMILARITY_ACROSS_USEF	PG.DCV.DCV_MAP_JACCARD_SIMILARITY_ACROSS_USEF	job_1406918339206_87975	PigLatin:"dov_map_jacc,similarity_across_users_20140805_"; pig_1407508491.PG.DCV.DCV_MAP_JACCARD_SIMILARIT	hive_1407508491.PG.DCV.DCV_MAP_JACCARD_SIMILARIT	2	400



NETFLIX

OSS

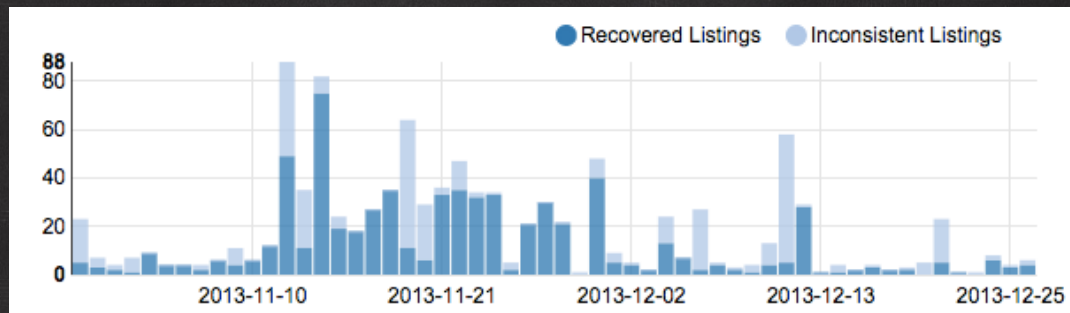
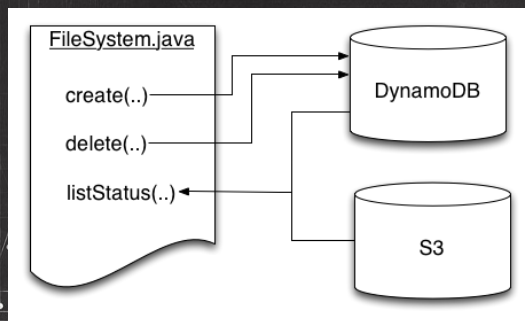


# S3mper – Tracking Eventual Consistency

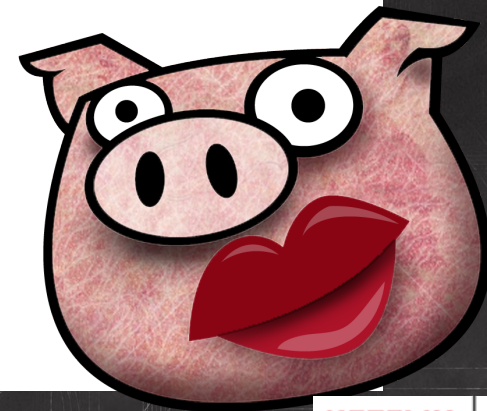
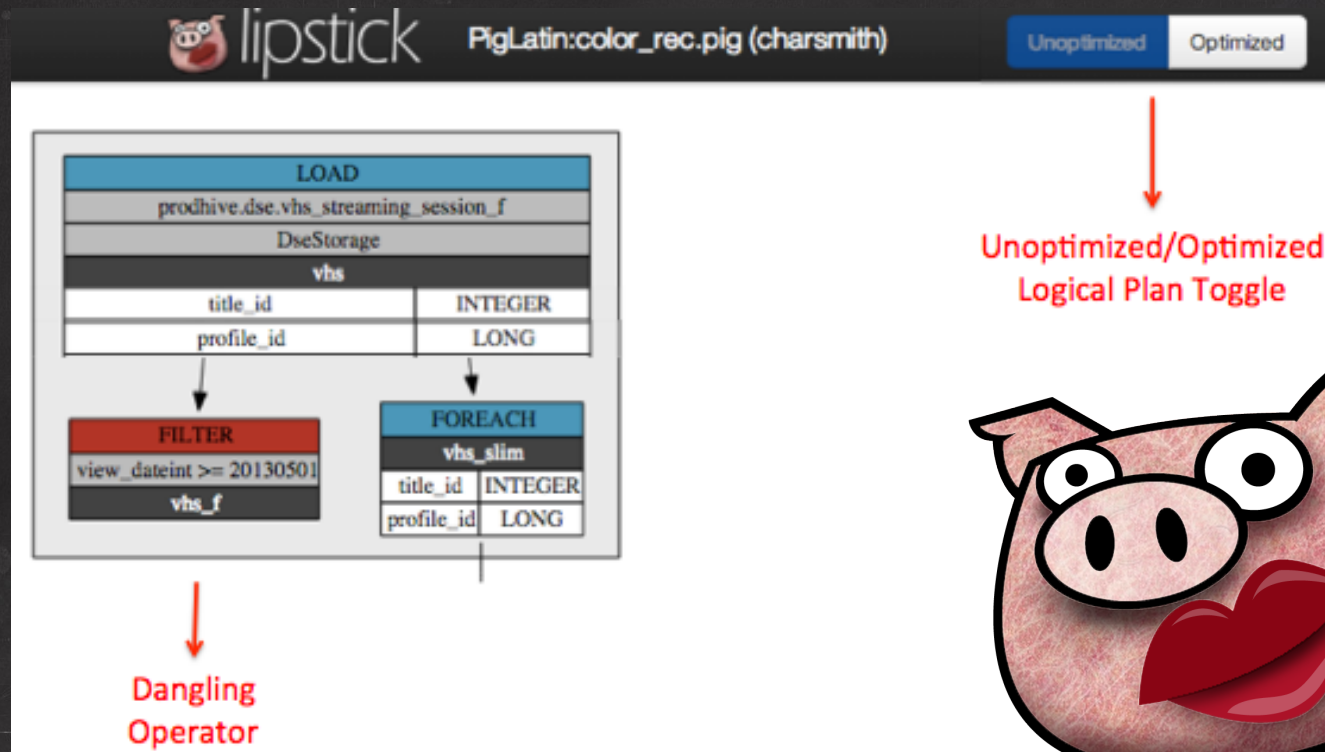
<http://techblog.netflix.com/2014/01/s3mper-consistency-in-cloud.html>

- S3 updates may take time to propagate for EMR
- S3mper uses a DynamoDB index to track state

## S3MPER



# Lipstick - Visualization for Pig queries

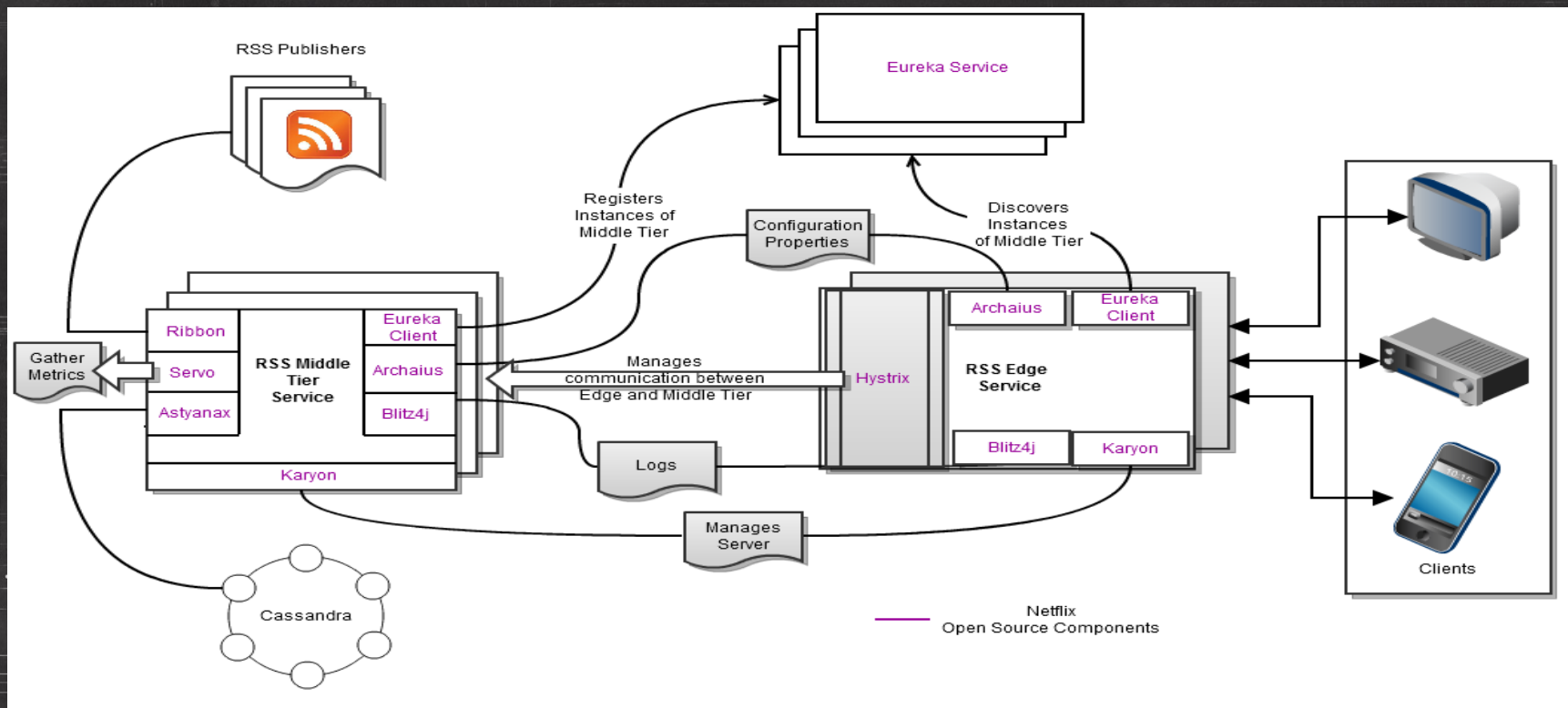




**Putting it all together...**



# Sample Application – RSS Reader



## 3<sup>rd</sup> Party Sample App by Chris Fregly

### fluxcapacitor.com

Flux Capacitor is a Java-based reference application demonstrating the following:

[archaius \(zookeeper-based dynamic configuration\)](#)

[astyanax \(cassandra client\)](#)

[blitz4j \(asynchronous logging\)](#)

[curator \(zookeeper client\)](#)

[eureka \(discovery service\)](#)

[exhibitor \(zookeeper administration\)](#)

[governator \(guice-based DI extensions\)](#)

[hystrix \(circuit breaker\)](#)

[karyon \(common base web service\)](#)

[ribbon \(eureka-based REST client\)](#)

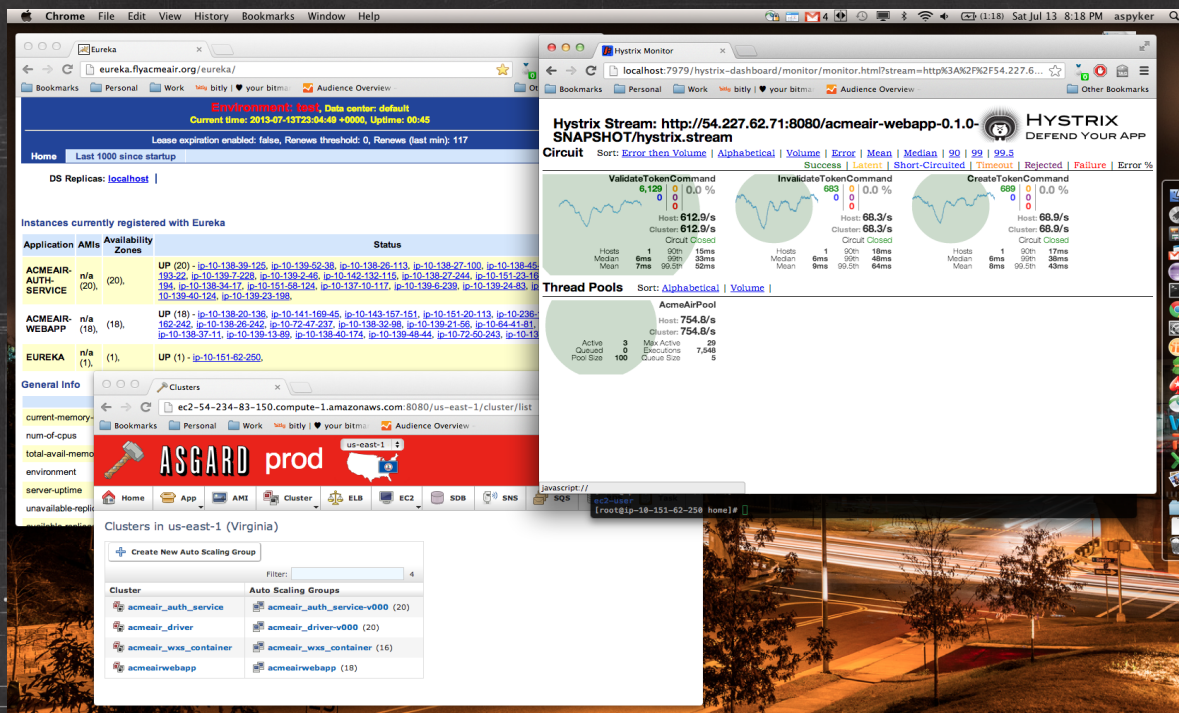
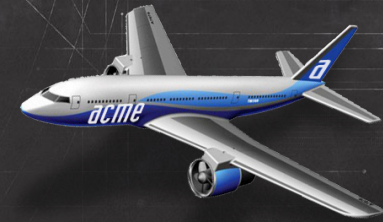
[servo \(metrics client\)](#)

▪ [turbine \(metrics aggregation\)](#)

Flux uses many popular open source tools such as Graphite, Jersey, Jetty, Netty, and Tomcat.

# 3<sup>rd</sup> party Sample App by IBM

<https://github.com/aspyker/acmeair-netflix/>





# Some of the companies using NetflixOSS

(There are many more, please send in your logo!)

**NETFLIX** | **OSS** Netflix Open Source Software Center

Repositories

Powered By NetflixOSS

vennetics



yammer



coursera



MORTAR

AnsWerS

YAHOO!

EUCALYPTUS



bazaarvoice:

OpenSCG™





# Takeaway



**Use NetflixOSS to scale your startup or re:Invent your Enterprise**

**Contribute to existing github projects and add your own**



# Thank you!

