

Going From 3 Week To Daily Releases

Daniel Zapata, Edge Delivery Team, Netflix

Streaming Video for TV Shows & Movies

NETFLIX Browse Taste Profile **KIDS** DVDs

Titles, People, Genres Daniel

A NETFLIX ORIGINAL SERIES

HOUSE of CARDS

PLAY

House of Cards 2013-2014 TV-MA 2 Seasons

NETFLIX ORIGINAL
HOUSE of CARDS

Bad, for a greater good.

Three Emmy Award wins including David Fincher ("The Social Network") for Outstanding Directing for a Drama Series and nine nominations including Outstanding Drama Series went to this acclaimed political thriller. Lead Actor nominee Kevin Spacey stars as ruthless, cunning Congressman Francis Underwood, who will stop at nothing to conquer the halls of power in Washington D.C. His secret weapon: his gorgeous, ambitious, and equally conniving wife Claire (Lead Actress nominee Robin Wright).

Directors' Commentary Available

Watch Season 1 of this Emmy-winning series with exclusive scene-by-scene audio commentary from directors including David Fincher and Joel Schumacher.

Genres: [TV Shows](#), [TV Dramas](#)

This show is: [Witty](#), [Cerebral](#), [Dark](#)

More than 48 Million Subscribers

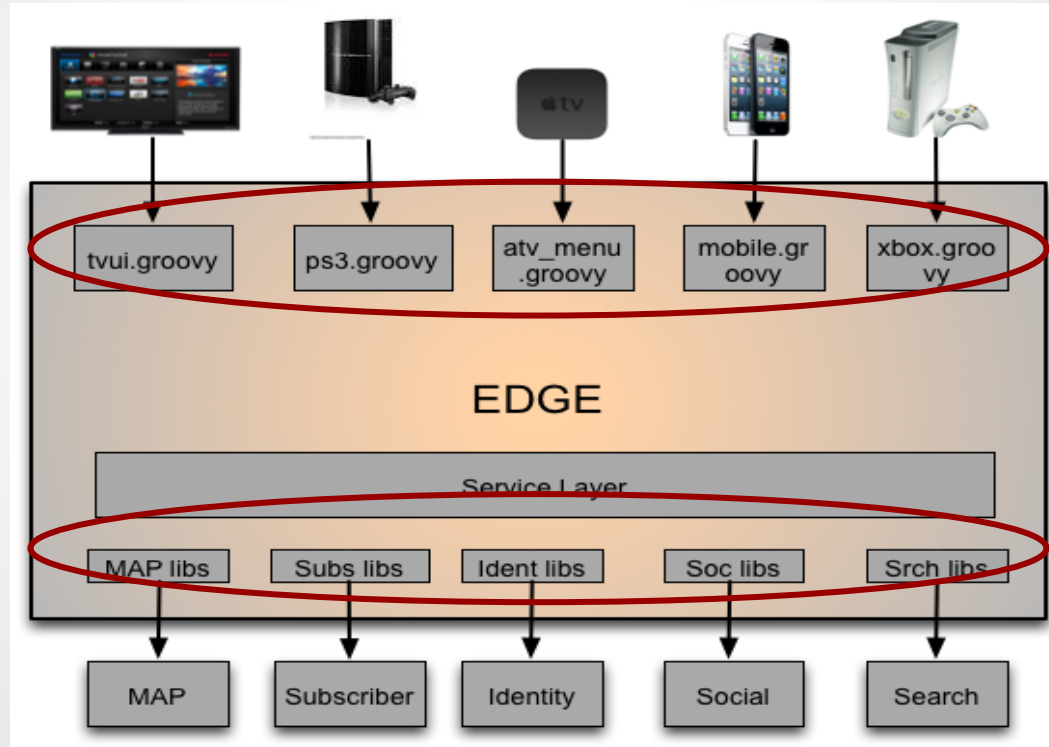


More than 40 Countries

US Internet Traffic During Peak

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	BitTorrent	24.53%	Netflix	34.21%	Netflix	31.09%
2	HTTP	14.27%	YouTube	13.19%	YouTube	12.28%
3	SSL	6.54%	HTTP	11.65%	HTTP	11.84%
4	Netflix	6.44%	iTunes	3.64%	BitTorrent	5.96%
5	YouTube	5.52%	SSL	3.42%	SSL	3.80%
6	Skype	2.23%	BitTorrent	3.40%	iTunes	3.33%
7	Facebook	2.17%	MPEG	2.85%	MPEG	2.62%
8	FaceTime	1.50%	Facebook	1.99%	Facebook	1.83%
9	Dropbox	1.20%	Amazon Video	1.90%	Amazon Video	1.82%
10	iTunes	1.15%	Hulu	1.74%	Hulu	1.58%
		64.40%		76.24%		74.58%

What Do We Do?



Hosts functionality
for Clients

Hosts functionality
for Services

Dependency Relationships



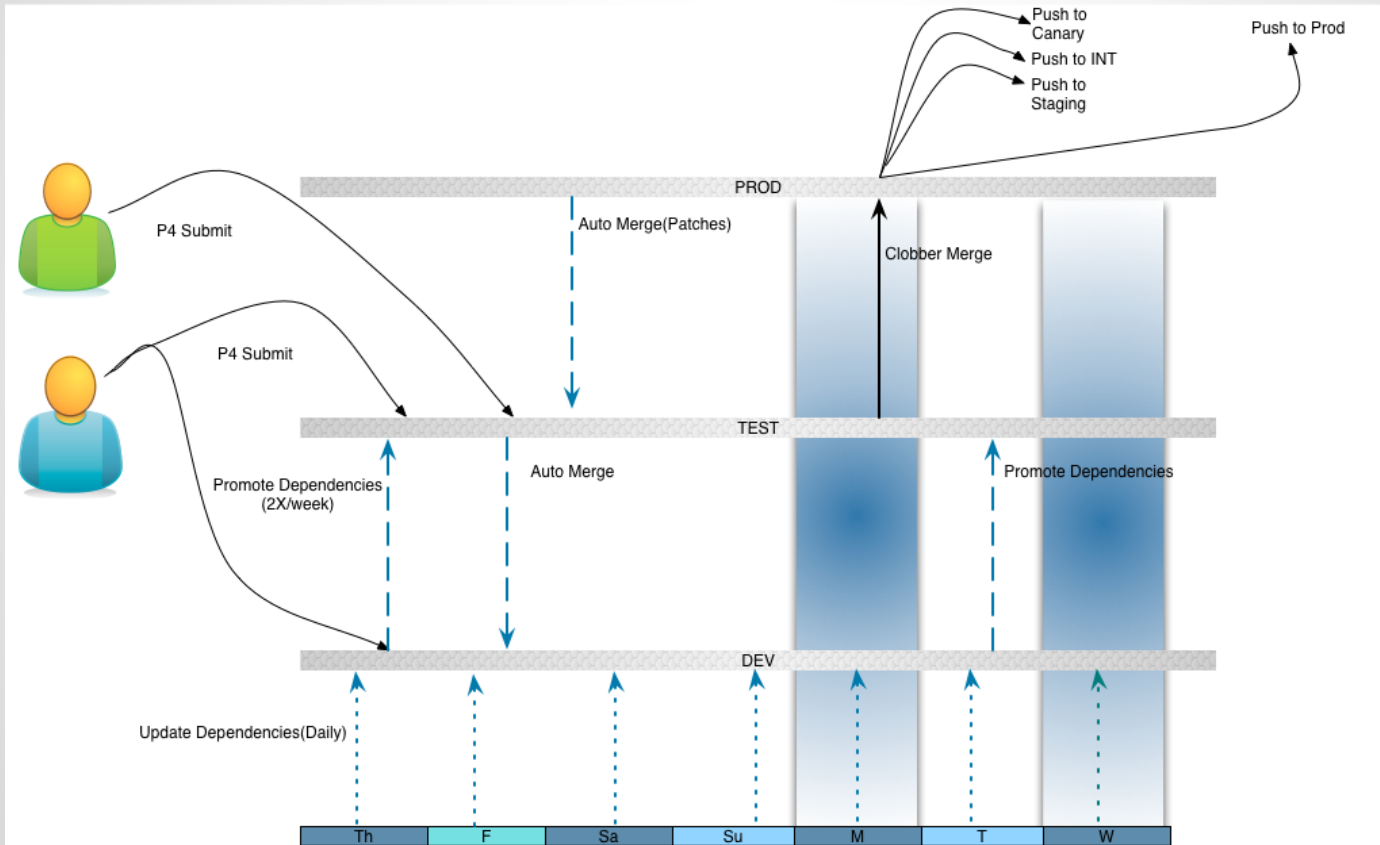
EDGE Server Operational Stats

- 2B+ Incoming Requests/Day to the EDGE Server
- 30+ Distinct Backend Services for the EDGE Server
- 14B+ Outbound Calls to our Backend Services
- Provides Data to 1K+ Device Types
- Deploy Hundreds of Cloud Servers in <60m

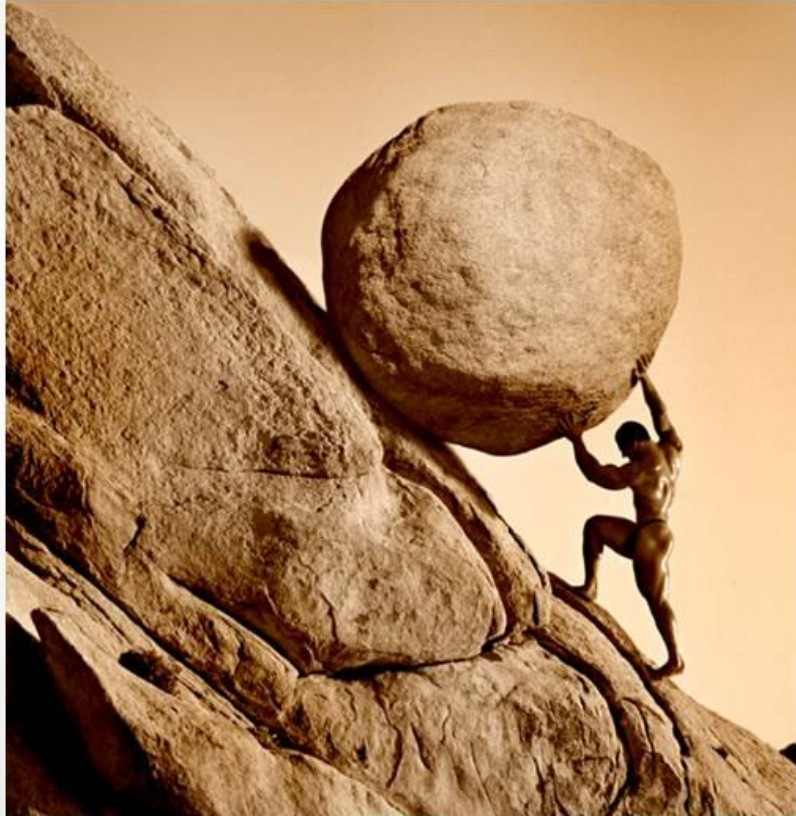
The Way Things Were



How Code Moved



Old Production Push Process



Manual Deployments

api-staging-v898

Launch and Terminate are disabled

 **Resize** to


 **Delete**  **Disable**  **Enable**




3 instances grouped by state

Count	State	Build	ELB	Eureka
-------	-------	-------	-----	--------

 3	InService	 1315		OUT_OF_SERVICE
--	-----------	--	--	----------------

api-staging-v899

 **Resize** to

 **Delete**  **Disable**  **Enable**

3 instances grouped by state

Count	State	Build	ELB	Eureka
-------	-------	-------	-----	--------

 3	InService	 1315		UP
---	-----------	--	---	----

Create Next Group:

 **Advanced Options**

api-staging-v900


Instance Bounds: Min: Max:

Desired Capacity: 3 instances 

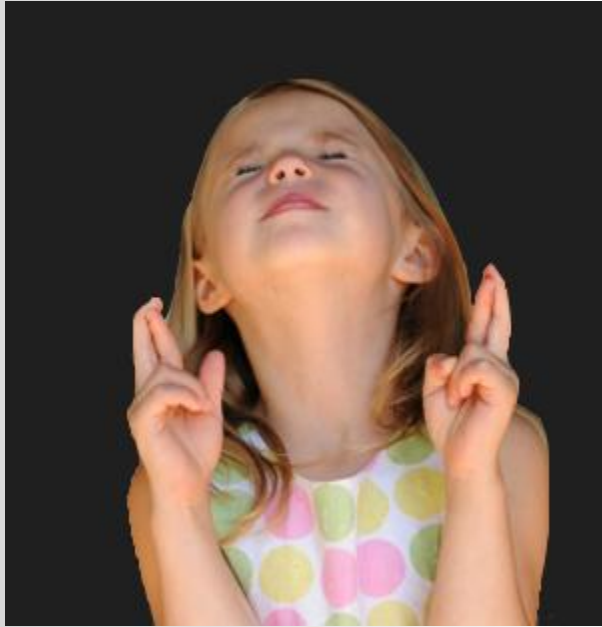
AMI Image ID:

 **Show more AMIs**

After launch: ☒ Wait for Eureka health check pass

 **Create Next Group api-staging-v900**

Life Of A Push Lead

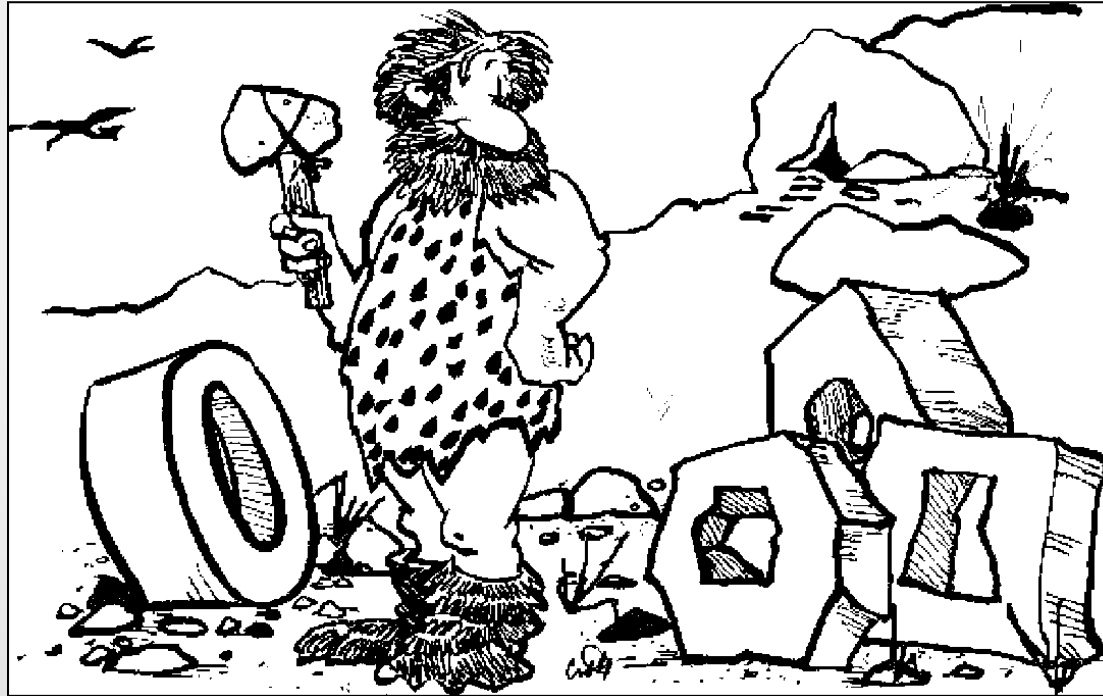


How Can We Do Better?



There has to be a better way

Figure 1 In search of continuous improvement

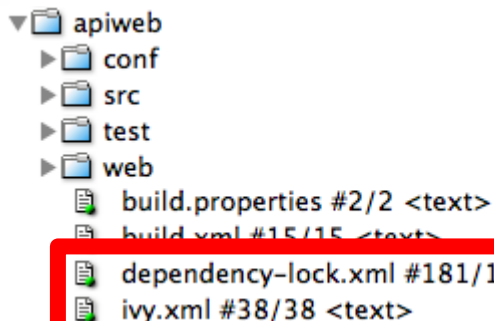


What Wasn't Working

1. Dependency jar updates constantly breaking builds
2. Manual and painful release process
3. Flakey tests
4. 'Black Box' production pushes
5. Every commit affects everyone
6. Long build times

Dependency Locking and Auto-Updating

```
<dependency org="netflix" name="p13nalgorecs-protocol" rev="${netflix.p13nalgorecs-protocol.rev}" conf="compile" force="true" />
<dependency org="netflix" name="metadataManager" rev="${netflix.metadataManager.rev}" conf="compile" force="true" />
<dependency org="netflix" name="server-context" rev="${netflix.server-context.rev}" conf="compile" force="true" />
<dependency org="netflix" name="base-server" rev="${netflix.base-server.rev}" conf="compile" force="true" />
<dependency org="netflix" name="basicTypes" rev="${netflix.basicTypes.rev}" conf="compile" force="true" />
```



apiweb

- conf
- src
- test
- web
 - build.properties #2/2 <text>
 - build.xml #15/15 <text>
 - dependency-lock.xml #181/181 <text>
 - ivy.xml #38/38 <text>

```
<dependency org="netflix" name="p13nalgorecs-protocol" revConstraint="latest.release" conf="compile" force="true" transitive="false" />
<dependency org="netflix" name="metadataManager" revConstraint="latest.release" conf="compile" force="true" transitive="false" />
<dependency org="netflix" name="server-context" revConstraint="latest.release" conf="compile" force="true" transitive="false" />
<dependency org="netflix" name="base-server" revConstraint="latest.release" conf="compile" force="true" transitive="false" />
<dependency org="netflix" name="basicTypes" revConstraint="latest.release" conf="compile" force="true" transitive="false" />
```

Better Tests And Test Reporting

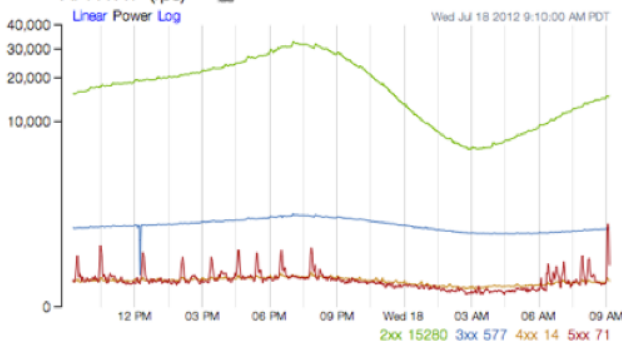
Test :	com.netflix.atf.api.video.json.CatalogTitlesTests.testEpisodeDetails.1
Parameters :	id=http://api.netflix.com. v=2.0 esn=HFXZ1CKEX9FFMAR params={v=2.0, esn=HFXZ1CKEX9FFMAR, debug=true, output=json, country=US} output=json country=US
Duration :	11:53:07 - 11:53:07

Assertions

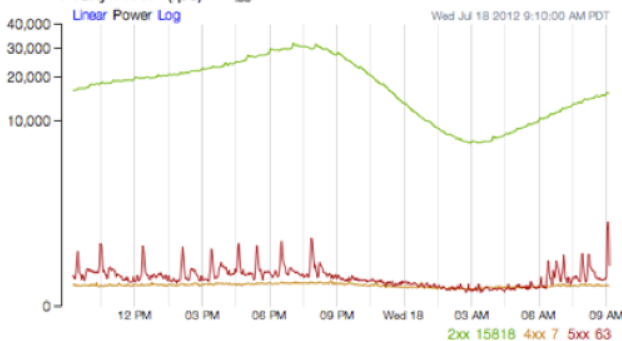
Num	Test	Result	Message
1	assertEquals	PASSED	Expected: 'Episode 2' Actual: 'Episode 2'
2	assertEquals	PASSED	Expected: 'Example Show: Ssn 1: Ep 2' Actual: 'Example Show: Ssn 1: Ep 2'
3	assertEquals	PASSED	Expected: "" Actual: ""
4	assertEquals	PASSED	Expected: "" Actual: ""
5	assertEquals	PASSED	Expected: 'Title[titleShort=Example Show: Ssn 1: Ep 2,regular=Episode 2,episodeShort=,episodeShort=]' Actual: 'Title[titleShort=Example Show: Ssn 1: Ep 2,regular=Episode 2,episodeShort=,episodeShort=]'
6	assertEquals	PASSED	Expected: 'Episode 2' Actual: 'Episode 2'
7	assertNotNull	FAILED	Assertion Error: sequence is not as expected Expected: Object to be 'NOT NULL' Actual: 'null'

More Insights and Data

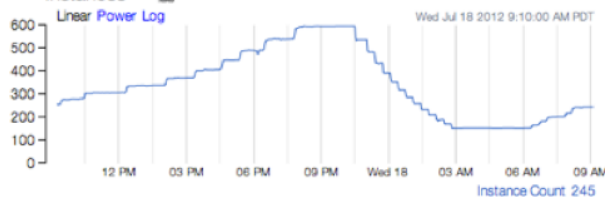
API HTTP (rps)



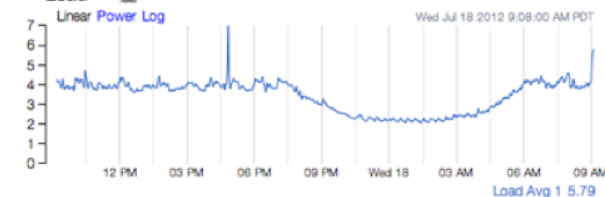
Proxy HTTP (rps)



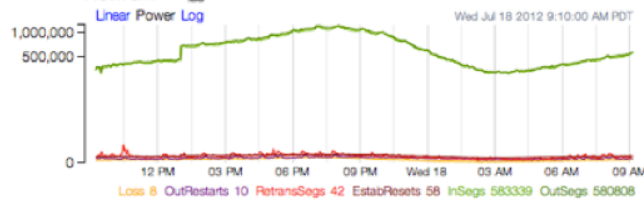
Instances



Load



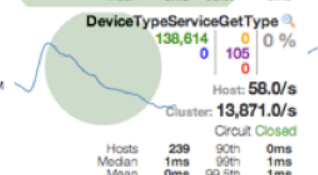
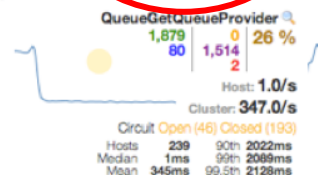
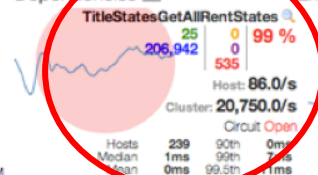
Network



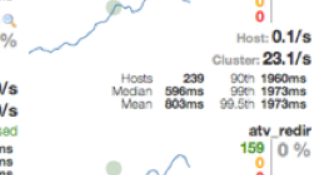
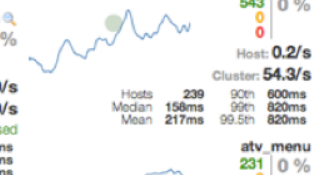
Garbage Collection (per minute for cluster)



Dependencies



Endpoints



Send In The Canary



**"This means something
but I can't remember what!"**

Automated Production Pushes



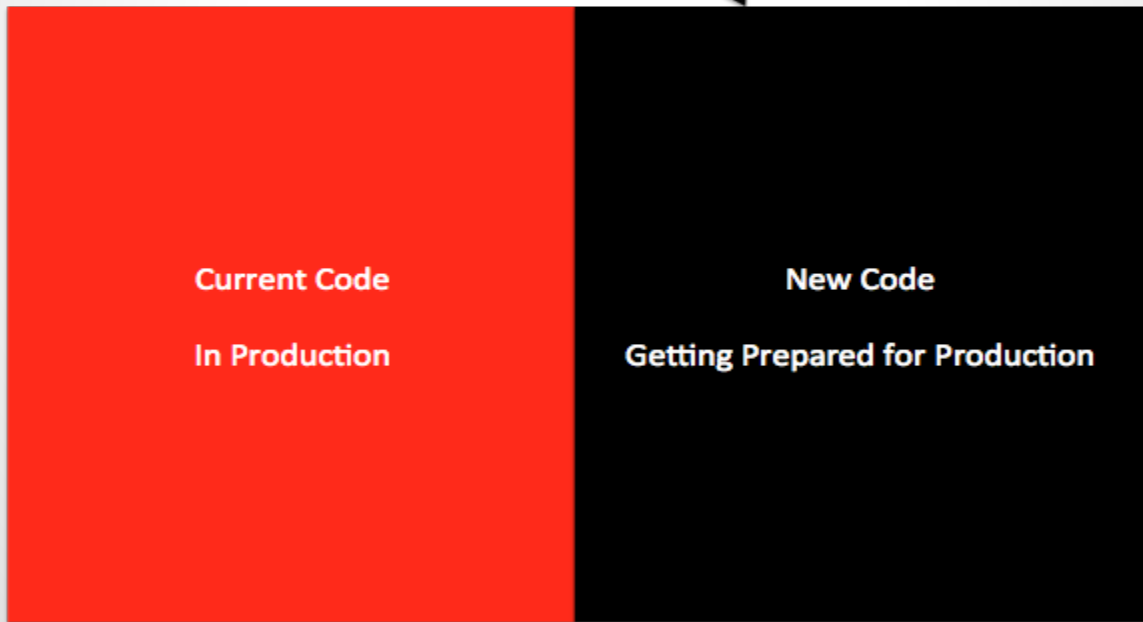
Current Code

In Production

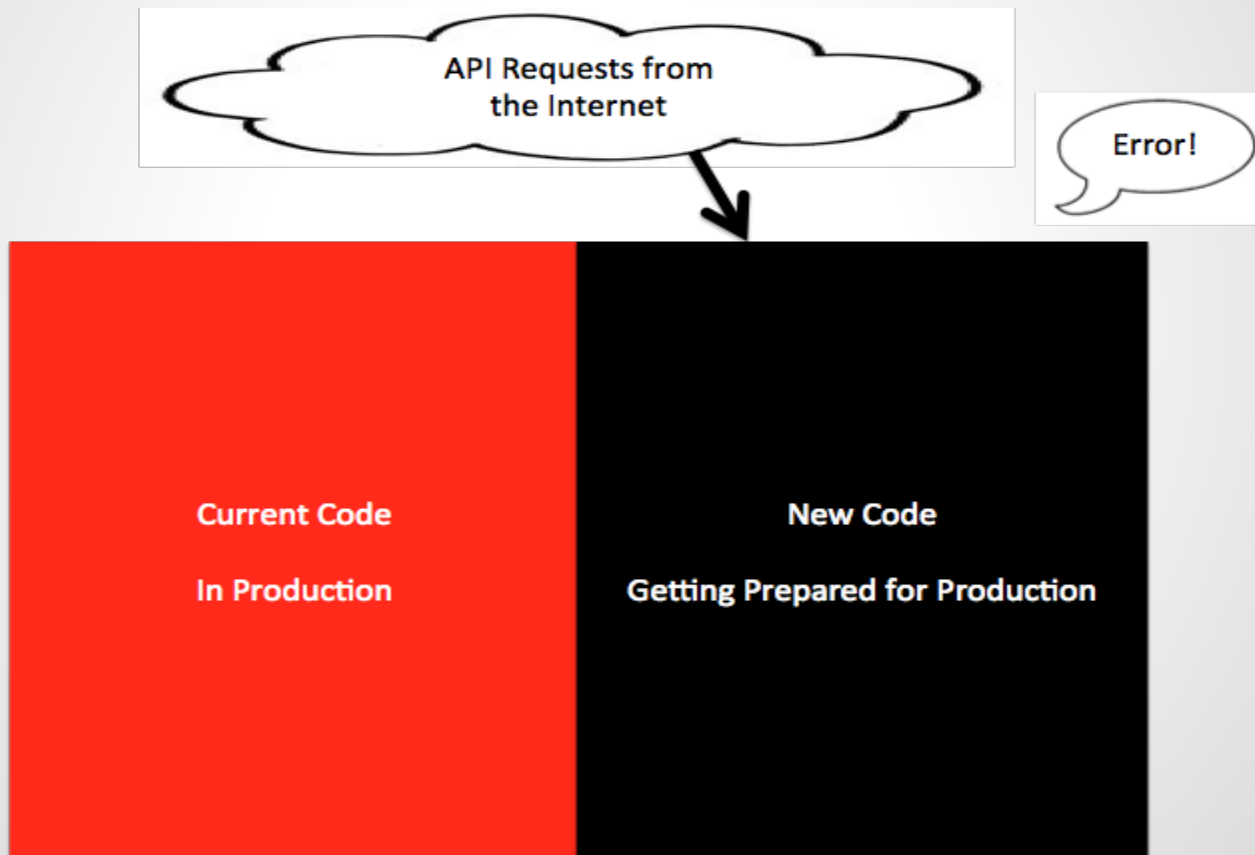
New Code

Getting Prepared for Production

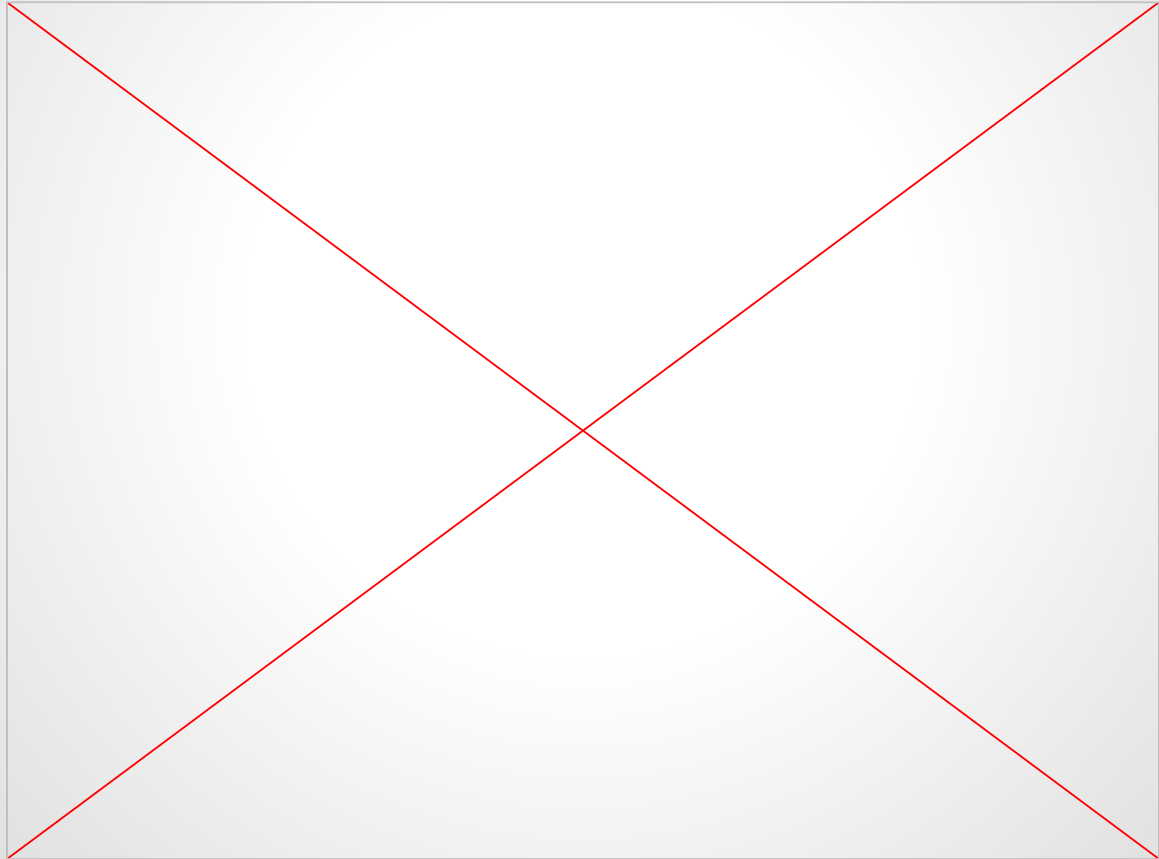
Automated Production Pushes



Automated Production Pushes



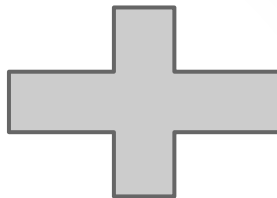
Automated Production Pushes
















Automated Production Pushes

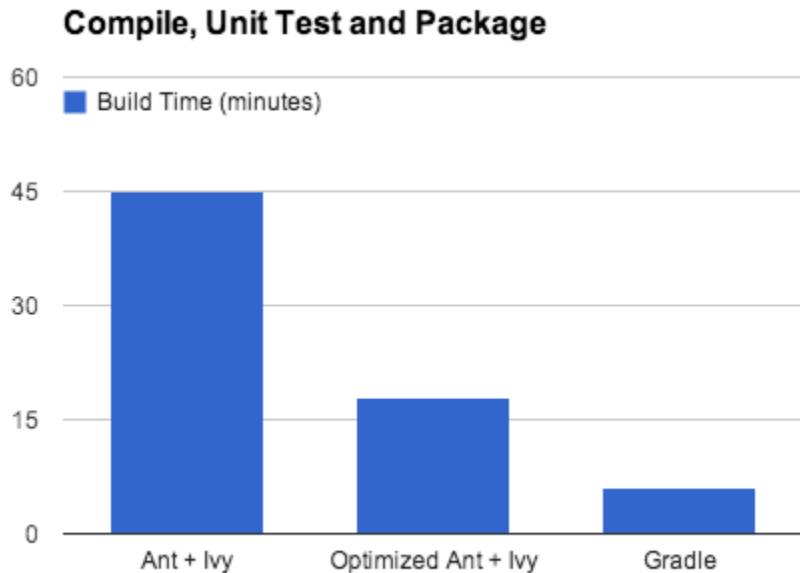


Goodbye Perforce, Hello Stash



Overview Files Commits Branches Pull requests Settings						
master ...						
Author	Commit	Message	Commit Date	Issues	Builds	
 Aaron Tull	0dc473d	Manually updating dependencies. Switching Streaming Content jars to the standard latest.	2 days ago			
 Bob T Builder	e5ada91 M	Merge remote-tracking branch 'origin/master' into HEAD	2 days ago			
 Jason Cacciatore	d301f8d	Need to pass requestToken for updateGame instead of gameld	2 days ago			
 Josh Gordineer	4f769d0	add isHidden to APIViewing so devices can filter title information from the in-player upgrade feature	2 days ago			
 Bob T Builder	6937b99 M	Merge branch 'prod' of ssh://stash.corp.netflix.com:7999/edge/server into automerge-prod-to-master	2 days ago			
 Aaron Tull	266eb17	Rebuilding dependencies.lock manually	2 days ago			
 Josh Gordineer	816ff8d	protect against null packages in class definitions	2 days ago			

Goodbye Ant, Hello Gradle



Automated Build Pipeline



Create Dev
Branch

Commit
Changes

Open Pull
Request

Compile, Unit
Test, +
Package

Patch CI
Server

Run Smoke
Tests

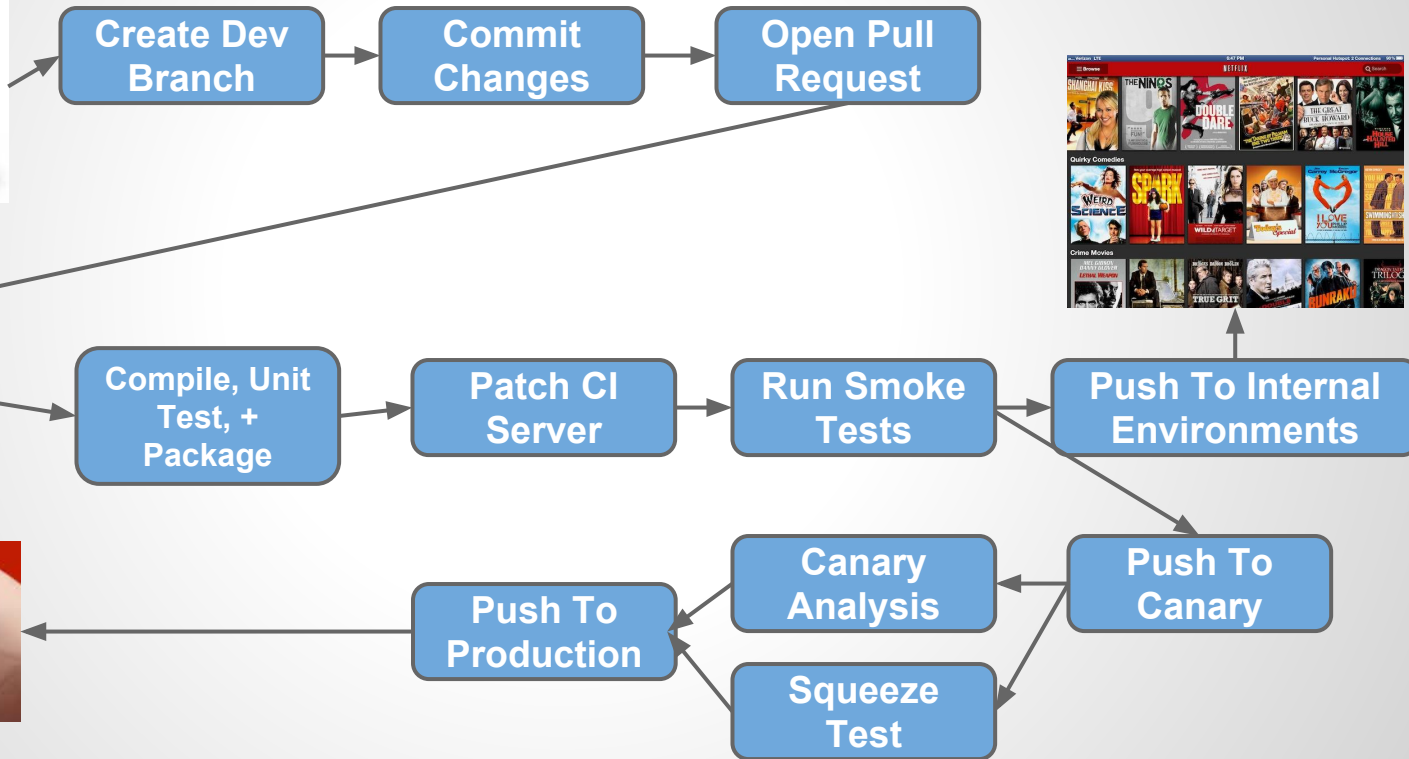
Push To Internal
Environments

Push To
Production

Canary
Analysis

Push To
Canary

Squeeze
Test



Canaries For All



Mission Control

CANARIES

release

gburrell@netflix.com

LAUNCHED

NA

Hour(s) : 1

EDGE-Master-Family-Build-1295

Launched : 06/11 16:48 PDT

us-east-1

NA

[Diff Report](#)

[api-prod-release-canary](#) (6)

[api-prod-release-baseline](#) (6)

EDGE-Master-Family-Build-1295

EDGE-Master-Family-Build-1255

ami-68f10c00 **UP** (6)

eu-west-1

[api-prod-release-canary](#) (6)

EDGE-Master-Family-Build-1295

ami-7f26e808 **UP** (6)

  [Actions](#)

patch

edge-support@netflix.com

LAUNCHED

96

Hour(s) : 7

EDGE-Prod-Family-Build-164

Launched : 06/11 11:38 PDT

us-east-1

96

[Canary Report](#) [Diff Report](#)

[api-prod-patch-canary](#) (6)












[api-prod-patch-baseline](#) (6)

EDGE-Prod-Family-Build-164

EDGE-Master-Family-Build-1255

master

EDGE-Master-Family-Build

#		Date	Committer(s)	Build	Bake	Patch Server	Smoke Tests
1296	 	Jun 11 17:07	Josh Gordineer				
1295	 	Jun 11 15:27	Josh Gordineer				
1294		Jun 11 15:17	Noel Yap 				

More Frequent Release Cycles

“In software, when something is painful, the way to reduce the pain is to do it more frequently , not less.”

Humble, Jez; Farley, David (2010-07-27). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series (Fowler))

Our Keys To Success

1. Lock those dependencies
2. Reliable and representative test signal
3. Have insights into your code
4. Canary your code before a full production push
5. Automate pushes
6. Rollback quickly
7. Optimize your SCM and build tooling
8. Implement an automated pipeline
9. Use a dashboard

Lessons Learned



Looking Ahead



Links

1. Netflix OSS: <http://netflix.github.io>
2. Gradle: <http://www.gradle.org/overview>
3. Netflix Gradle Goodness: <http://nebula-plugins.github.io>
4. We're hiring! <http://jobs.netflix.com/jobs.php>