


December, 3, 2017

# The Paved PaaS to Microservices


**NETFLIX**

Yunong Xiao,  
Principal Software Engineer, Netflix  
[yunong@netflix.com](mailto:yunong@netflix.com),

 @yunongx,

 yunongx

<http://yunong.io>

 [linkedin.com/in/yunongxiao](https://www.linkedin.com/in/yunongxiao)



架构迎接未来变化  
IAS2017 • NANJING





纸牌屋

NETFLIX

A NETFLIX ORIGINAL SERIES

HOUSE  *of* CARDS

NETFLIX

What is **NETFLIX**?

影音流行，由此起飛。

隨處都能觀賞，隨時都能取消。

免費試用一個月

Great Content  
Incredibly Scale

# Great Content



Watch:

Any time

Any where

Any device

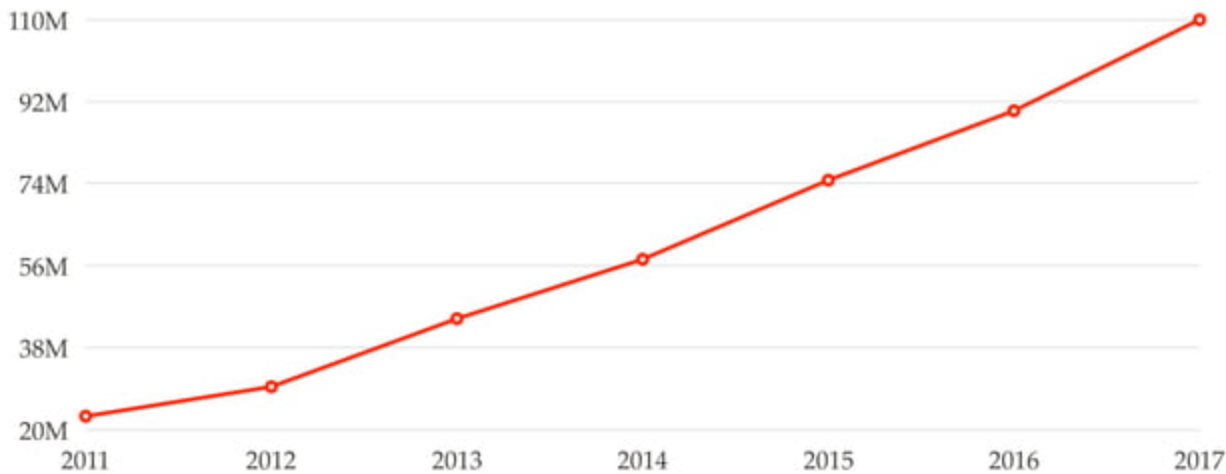
# Incredible Scale: Numbers

**110 000 000** paying customers

in over **190** countries

streaming **125 000 000** hours/day

# Incredible Scale: Subscriber Growth





# Incredible Scale: Bandwidth

Upstream		Downstream		Aggregate	
BitTorrent	18.37%	Netflix	35.15%	Netflix	32.72%
YouTube	13.13%	YouTube	17.53%	YouTube	17.31%
Netflix	10.33%	Amazon Video	4.26%	HTTP - OTHER	4.14%
SSL - OTHER	8.55%	HTTP - OTHER	4.19%	Amazon Video	3.96%
Google Cloud	6.98%	iTunes	2.91%	SSL - OTHER	3.12%
iCloud	5.98%	Hulu	2.68%	BitTorrent	2.85%
HTTP - OTHER	3.70%	SSL - OTHER	2.53%	iTunes	2.67%
Facebook	3.04%	Xbox One Games Download	2.18%	Hulu	2.47%
FaceTime	2.50%	Facebook	1.89%	Xbox One Games Download	2.15%
Skype	1.75%	BitTorrent	1.73%	Facebook	2.01%
	69.32%		74.33%		72.72%




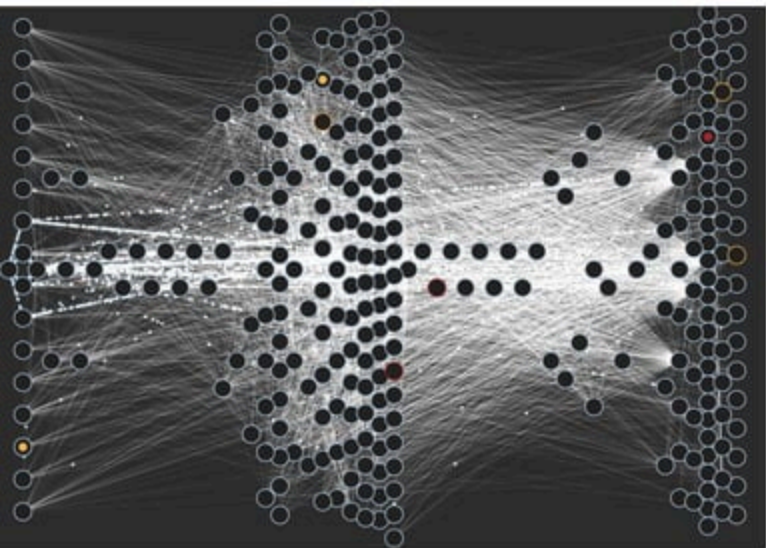
Table 1 - Top 10 Peak Period Applications - North America, Fixed Access

1/3 of **ALL** internet traffic in North America

How?



# Microservices Architecture



Member Experience

Streaming

Playback

Telemetry

Sign up

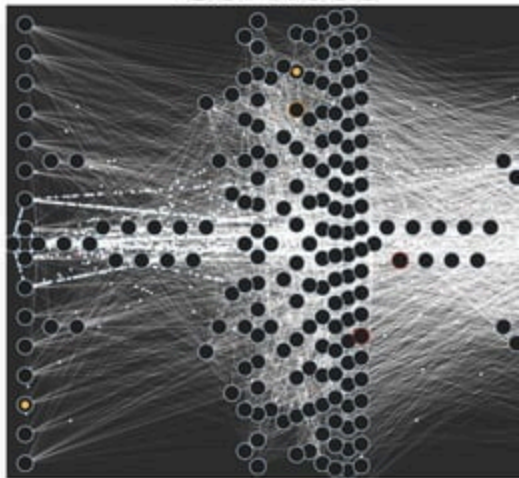
Recommendations

...

## Clients



## Services



**Naive Solution: Every client talks to every service**  
**30 clients \* 800 services = 24000 connections**



**Too Complex!**

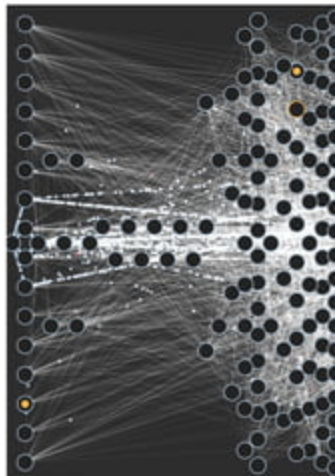
**P & P**  
Great Opt

## Clients



Netflix Edge API  
<http://api.netflix.com>

## Services



# What is the Netflix Edge API?



Bridges the gap between frontend and backend

# What is the Netflix Edge API?



Provides **1** consistent API over **100s** of disparate services



# What is the Netflix Edge API?



Central integration point for frontend and  
backend

**Requirements?**

## A/B Test participation

**Include me in tests and previews**

Turn this off to be shifted to the standard experience now.

ON



Participate in tests to help improve the Netflix experience and see potential changes before they are available to all members.

[Done](#)

# 1000s of A/B tests per year

Questions? Contact us.

[Audio and Subtitles](#)[Media Center](#)[Privacy Statement](#)[Help Center](#)[Jobs](#)[Gift Cards](#)[Cookie Preferences](#)[Investor Relations](#)[Terms of Use](#)

November 2017

< Today >

Sun

Mon

Tue

Wed

Thu

Fri

Sat

29

30

31

Nov 1

2

3

4

Push!

Push!

Push!

5

6

7

8

9

10

11

Push!

Push!

Push!

# Frequent code pushes

12

13

14

15

16

17

18

Push!

Push!

Push!

Push!

19

20

21

22

23

24

25

Push!

Push!

Push!

26

27

28

29

30

Dec 1

2

Push!

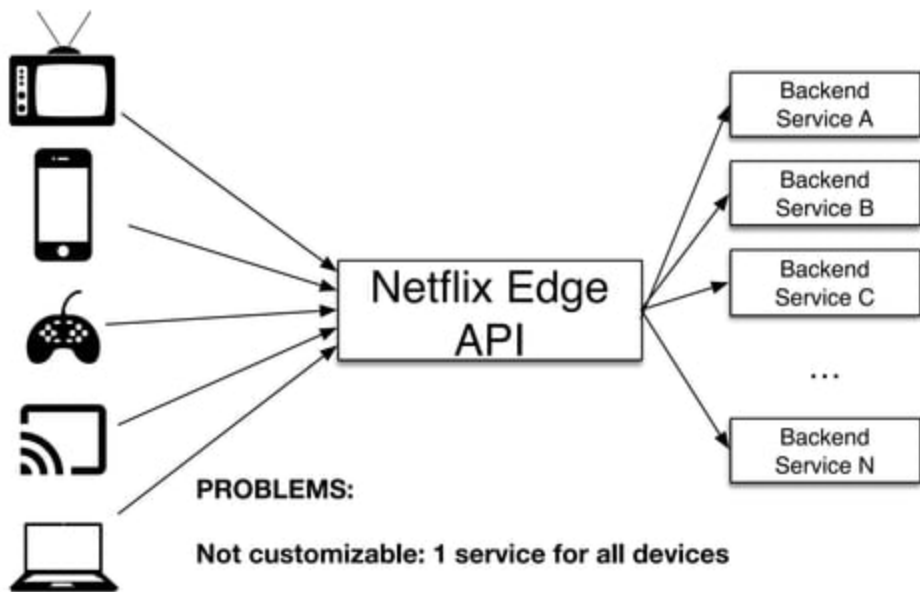
Push!

Push!



# Customization

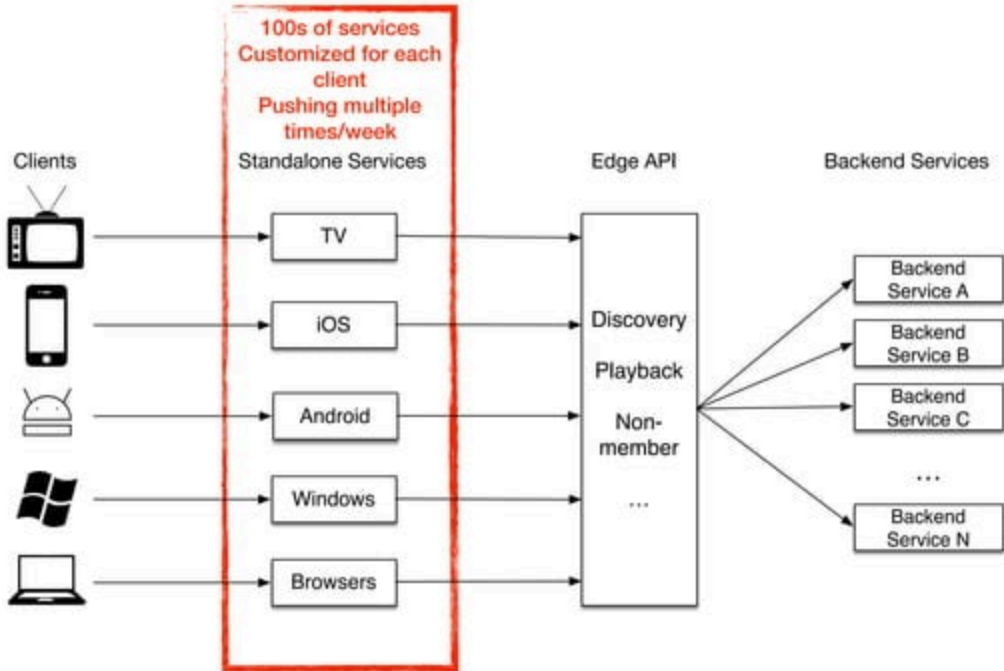




**PROBLEMS:**

**Not customizable: 1 service for all devices**

**Slow rate of change: centralized service, 1 code push a month**



# Goals

**Velocity**

**Reliability**

100s of services  
1000s of A/B tests  
Pushing multiple times/week/service

99.99% Availability





# Who's the owner?

**Client Engineers**

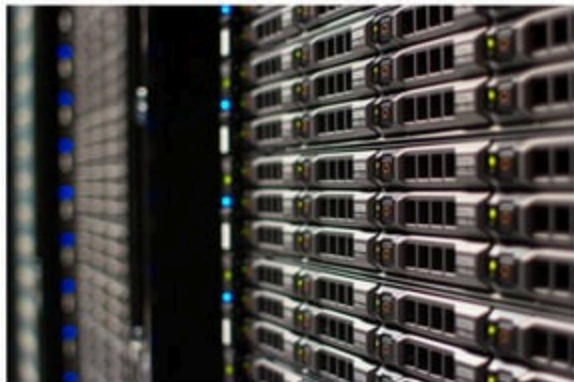


**ROKU**



PlayStation

**Server Engineers**



**Q: Can we achieve velocity & reliability with client engineers?**

**A: Try Platform as a Service**

# What is a Platform as a Service (Paas), Anyway?

“Platform as a service (PaaS)... allows customers to develop, run, and manage applications without the complexity of building and maintaining the infrastructure and platform...”

*-Wikipedia*

## Our PaaS to Achieving Reliability & Velocity

1. Standardized components
2. Preassembled platform
3. Automation and tooling



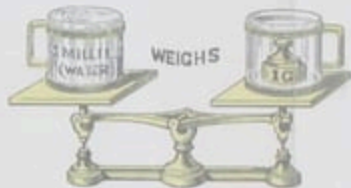
# 1. Standardized Components

1 CUBIC CENTIMETER OR 1000 CUBIC

MILLIMETERS

=

1 MILLILITER.



10 CUBIC CENTIMETERS = 10 MILLILITERS AND OF WATER WEIGHS 1 DEKAGRAM



# What's Inside a Microservice?



# Netflix Open Source Libraries

NETFLIX

OSS

<http://netflix.github.io>



NETFLIX

## Common Runtime Services & Libraries

Runtime containers, libraries and services that power microservices

The cloud platform is the foundation and technology stack for the majority of the services within Netflix. The cloud platform consists of cloud services, application libraries and application containers. Specifically, the platform provides service discovery through [Eureka](#), distributed configuration through [Archaius](#), resilient and intelligent inter-process and service communication through [Ribbon](#). To provide reliability beyond single service calls, [Hystrix](#) is provided to isolate latency and fault tolerance at runtime. The previous libraries and services can be used with any JVM based container.

The platform provides JVM container services through [Karyon](#) and [Governator](#) and support for non-JVM runtimes via the [Pana](#) sidecar. While [Pana](#) provides proxy capabilities within an instance, [Zuul](#) (which integrates [Hystrix](#), [Eureka](#), and [Ribbon](#) as part of its IPC capabilities) provides dynamically scriptable proxying at the edge of the cloud deployment.

The platform works well within the EC2 cloud utilizing the Amazon autoscaler. For container applications and batch jobs running on Apache Mesos, [Fenzo](#) is a scheduler that provides advanced scheduling and resource management for cloud native frameworks. [Fenzo](#) provides plugin implementations for bin packing, cluster autoscaling, and custom scheduling optimizations can be implemented through user-defined plugins.

# Why Standards?

BBC ONLINE NETWORK

HOME PAGE | SITE MAP | SCHEDULES | BBC INFORMATION | BBC EDUCATION | BBC WORLD SERVICE

BBC NEWS

News in Audio

News in Video

Newyddion

Hinnochrn

Noticias

اخبار

国际新闻

粵語廣播

Front Page

World

UK

UK Politics

Business

Sci/Tech

Health

Education

Sport

Entertainment

Talking Point

In Depth

On Air

Archive

Thursday, September 30, 1999 Published at 18:53 GMT 19:53 UK

## Sci/Tech

### Confusion leads to Mars failure



The Mars Climate Orbiter: Now in pieces on the planet's surface

The Mars Climate Orbiter Spacecraft was lost because one Nasa team used imperial units while another used metric units for a key spacecraft operation.

## Sci/Tech Contents

### Relevant Stories

24 Sep 99 | Sci/Tech  
[Scientist fights Mars setback](#)

23 Sep 99 | Sci/Tech  
[Mars probe feared destroyed](#)

23 Sep 99 | Sci/Tech  
[What the loss of Mars Climate Orbiter means](#)

17 Jul 99 | Sci/Tech  
[Astronauts call for Mars mission](#)

## Internet Links

[Mars Climate Orbiter](#)

The BBC is not responsible for the content of external internet sites.

Feedback  
Low Graphics  
Help

RAM



# Mélange of RPC

Apache Thrift™



Wangle

C++ networking library



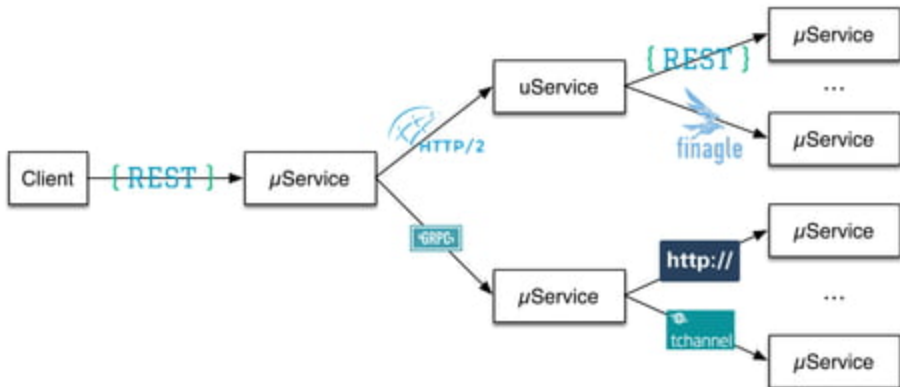
Quick  
UDP  
Internet  
Connections

{ REST }



http://

# Microservice Interactions



# Failure: **When** not If



I don't get to chill. Can I at least Netflix? Fix you're sh!t!

4

RETWEETS

1

FAVORITES



Netflix is down. I have to parent now. **#UNFAIR**  
**Netflix** 不工作了，我得当父母了，**#不公平**

121

RETWEETS

18

FAVORITES



Netflix is down and I feel like my best friend died.

**#becauseitdid**

**Netflix** 不工作了，就像我的好朋友死了

19

RETWEETS

4

FAVORITES



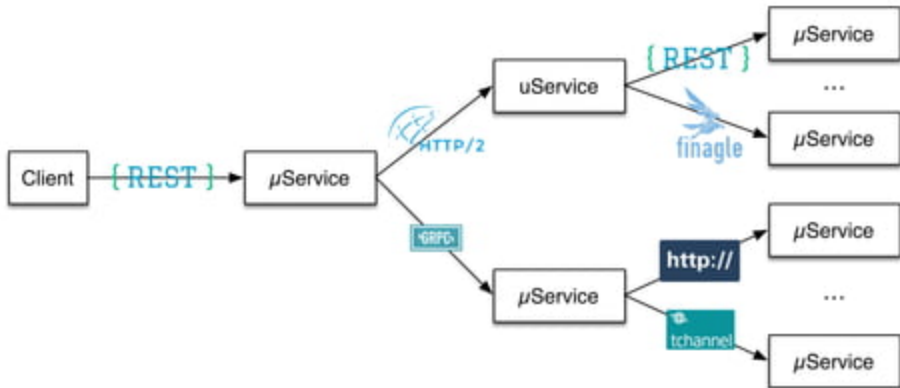
**Mean Time to Detection (MTTD)**

**Mean Time to Repair (MTTR)**

**“Is it fixed yet?”**

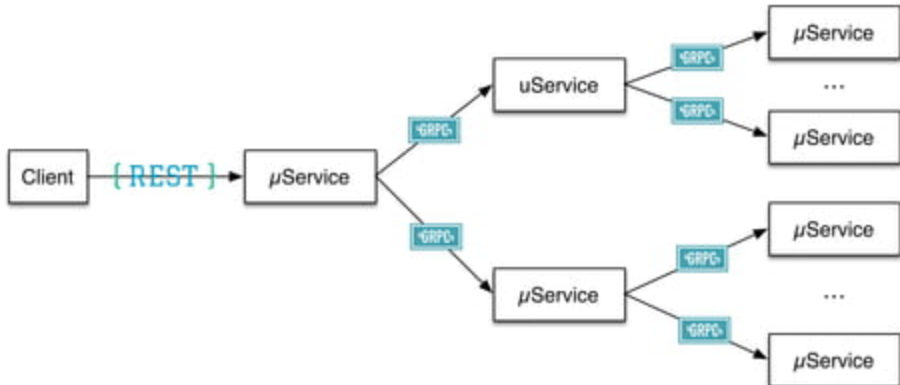
*– Managers everywhere*

# N Flavors of RPC



**How do you debug the system?**

# One Standard RPC



**Reduces MTTR**

# Benefits of Standardizing

RPC

Discovery

Registration

Runtime

OS

Configuration

Metrics

Logging

Tracing

Dashboards

Alerts

Stream  
Processing



Consistency



Leverage



Interoperability



Quality



Support



**You want to be different?**



# Netflix Culture

The dream team model reinforces the idea that your economic security is based on your skills and reputation, not on your seniority at one company. At Netflix, you learn a lot working on hard problems with amazing colleagues and what you learn increases your market value. Knowing that other companies would quickly hire you if you left Netflix is comforting. We see occasional outside interviewing as healthy, and encourage employees to talk with their managers about what they learn in the process.

While our teammates are fantastic, and we work together very well, we know we can always do better. We strive to have calm confidence, and yet yearn to improve. We are mediocre compared to how great we want to become.

## **Freedom & Responsibility** <http://jobs.netflix.com/culture>

There are companies where people walk by trash on the floor in the office, waiting for someone else to pick it up, and there are companies where people lean down to pick up the trash they see, as they would at home. We try hard to be the latter, a company where everyone feels a sense of responsibility to do the right thing to help the company at every juncture. Picking up the trash is the metaphor for taking care of problems, small and large, as you see them, and never thinking "that's not my job." We


# Freedom

# &

# Responsibility

- Use non-standard components/architecture
- Fosters innovation
- Reintegrate changes back into platform

- Use new technologies judiciously
- Don't develop alone in a vacuum
- Increases burden to the rest of the company

A collection of various mechanical parts and tools, including gears, springs, and shafts, arranged on a light surface. The parts are scattered across the frame, with some larger components like a gear and a spring in the upper left, and many smaller parts like bolts, washers, and small gears throughout. The text "2. Preassembled Platform" is overlaid in the center.

## 2. Preassembled Platform

# Assembly Required

Discovery

Runtime

OS

Configuration

RPC

Registration

Logging

Dashboards

Metrics

Stream  
Processing

Tracing

Alerts

**Let's use an Analogy**

[Home](#) / [Dining](#) / [Dining chairs](#) / [Dining chairs](#)


**NEW  
LOWER  
PRICE**

## STEFAN

Chair, brown-black ~~\$29.00~~

**\$25.00**

Article Number: 002.110.88

★★★★★ 4.5 (11) [Write a review](#)

Solid wood is a durable natural material.

Quantity:

1

Buy online

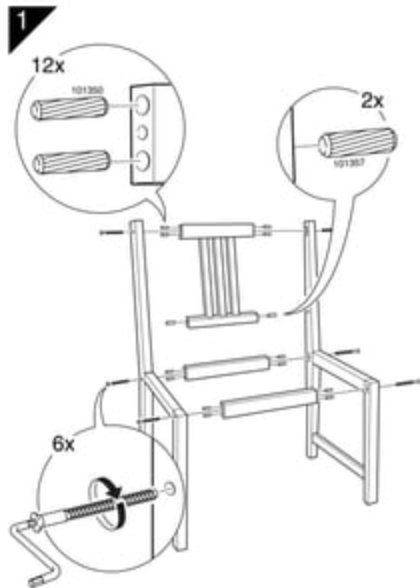
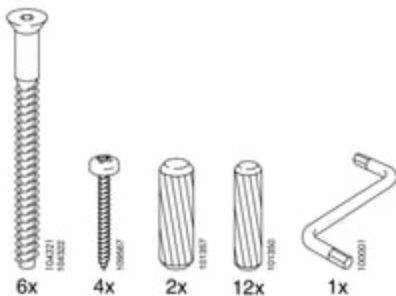
Add to shopping list

Add to Registry

Check stock at your local store

Store: Choose





A photograph of a child with long brown hair, wearing a blue and white striped shirt and dark pants, crawling on a light-colored wooden floor. In the background, a dark wooden chair is broken, with its backrest and seat detached and lying on the floor. The scene is indoors, likely in a dining room or kitchen.

Most engineers aren't experts  
on how a microservice is built,  
they just want a pre-built  
microservice that works



# Microservice Assembly Instructions

Read documentation

Copy/paste (wrong) source code

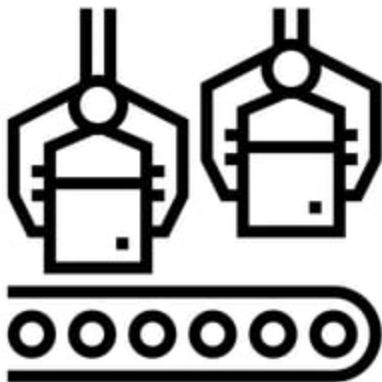
Which versions of dependencies?

How to initialize all components?

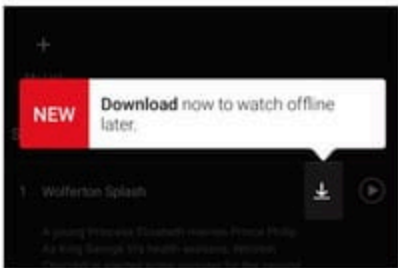
How to configure all components?

Missing components

...



# Slow: Can Take Weeks



**New feature**



**Velocity**



**Reliability**

**Not a single line of business logic!**

# Preassembled Platform



RPC

Discovery

Registration

Runtime

OS

Configuration

Metrics

Logging

Tracing

Dashboards

Alerts

Stream  
Processing

# Easy Access to All Platform Components

```
1  function getHelloWorld(req, res, next) {
2      // logger
3      req.getLog().info('this is a log');
4      // metrics client
5      req.getAtlas().counter('foo')
6      // dynamic configuration
7      req.getProperties.get('foo.bar')
8      // API client
9      req.getReactiveDatasource();
10     // distributed tracing
11     req.getSalpClient();
12     // failure injection client
13     req.getFITClient();
14 }
```

**Out of the Box**



# Component Management





Logs



Metrics



Alerts

**MISSING**



# Insights





# Application, system, and runtime metrics & logs



**Different for each service!**

**Consistent** application, system, and runtime insight



**Reduces MTTD & MTTR**



**Launching a microservice  
safely**

# Configures and Initializes Correctly

## BOEING 747-400 NORMAL PROCEDURES CHECKLIST

### POWER UP / SAFETY CHECK

First Officer

Captain

CIRCUITBREAKERS.....CHECKED  
BATTERY.....ON  
STANDBY POWER.....AUTO  
HYDRAULIC DEMAND PUMPS.....OFF  
WINDSHIELD WIPERS.....OFF  
ALTERNATE FLAPS AND GEAR.....OFF  
GEAR LEVER.....DOWN  
FLAPS.....CHECKED  
APU.....RUNNING  
ELECTRICAL SYSTEM.....SET/APU AVAIL ON  
APU BLEED AIR.....ON  
ISOLATION VALVES.....OPEN  
PACKS.....NORMAL

### PREFLIGHT

First Officer

Captain

EMERGENCY EQUIPMENT.....CHECKED  
FIRE PROTECTION.....CHECKED  
INTERRUPT SWITCHES.....ON  
PASSENGER OXYGEN.....NORMAL  
STAB TRIM CUTOUT SWITCHES.....AUTO

TRANSPONDER.....SET  
SOURCE SELECTORS.....SET  
CLOCKS.....SET  
CRT SELECTORS.....NORMAL  
PFD.....CHECKED  
ND.....CHECKED  
AUTOBRAKES.....RTO  
EIU SELECTOR.....AUTO  
HDG REFERENCE SWITCH.....NORMAL  
FMC MASTER SELECTOR.....LEFT  
GROUND PROX SYSTEM.....CHECKED

### BEFORE STARTING

First Officer

Captain

HYDRAULIC DEMAND PUMPS.....  
.....AUTO, AUX (1 AND 4)  
BRAKE PRESSURE.....NORMAL  
FUEL QUANTITY.....KG  
FUEL SYSTEM.....SET  
X-FEEDS.....OPEN (1 & 4 CLOSED)  
SEAT BELTS SIGN.....ON  
NOTOC.....CHECKED  
SHIPS PAPERS.....ON BOARD  
PERFORMANCE DATA.....CHECKED AND SET



```
[2017-12-02T12:27:32.369Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: installing routes... done! (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.379Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: starting server on port 7001... (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.382Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: starting server on port 7001... done! (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.384Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: starting debug info server on port 8077... (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.385Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: starting debug info server on port 8077... done! (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.385Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: initializing Eureka registration... (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.401Z] INFO: nodequark-bootstrap/eureka-registration/8348 on titus-2371338-worker-0-352: Starting
Eureka registration for nq_uiplatform_yunongnqtest with vipAddress: nq_uiplatform_yunongnqtest-prod-^1.0.0 and id: 4c73
814e-7472-4927-a328-5ee8d8035db4 (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.403Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: initializing Eureka registration... done! (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.404Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: starting interval tasks... (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.405Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: starting interval tasks... done! (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
[2017-12-02T12:27:32.405Z] INFO: nodequark-bootstrap/uiplatform_yunongnqtest_1.0.0/8348 on titus-2371338-worker-0-352:
init: NodeQuark startup success (nqOrg=uiplatform, nqVersion=1.0.0, nqName=yunongnqtest, nqPlatformVersion=5.5.1)
```

Starts up all components in the correct order

Starts up safely every time

3 days ago

**v1.4.0**

869e373 zip tar.gz

18 days ago

**v1.4.0-pre1**

52bcda9 zip tar.gz

19 days ago

**v1.3.9**

28aee69 zip tar.gz

25 days ago

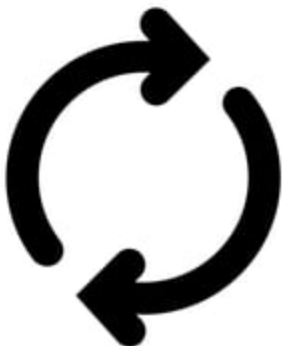
**v1.3.7**

e680329 zip tar.gz

on May 25

**v1.3.6**

ef2c853 zip tar.gz




**Updates**



**Compatibility**



**Versions**

A photograph of a curved path made of light-colored, irregular cobblestones. The path starts from the bottom right and curves towards the top left, disappearing into the distance. It is bordered by a lush green lawn on both sides. The lighting is soft, suggesting an overcast day or early morning/late afternoon. The overall tone is peaceful and natural.

**Stay on paved path!**



# Adjust as Needed



Config overrides

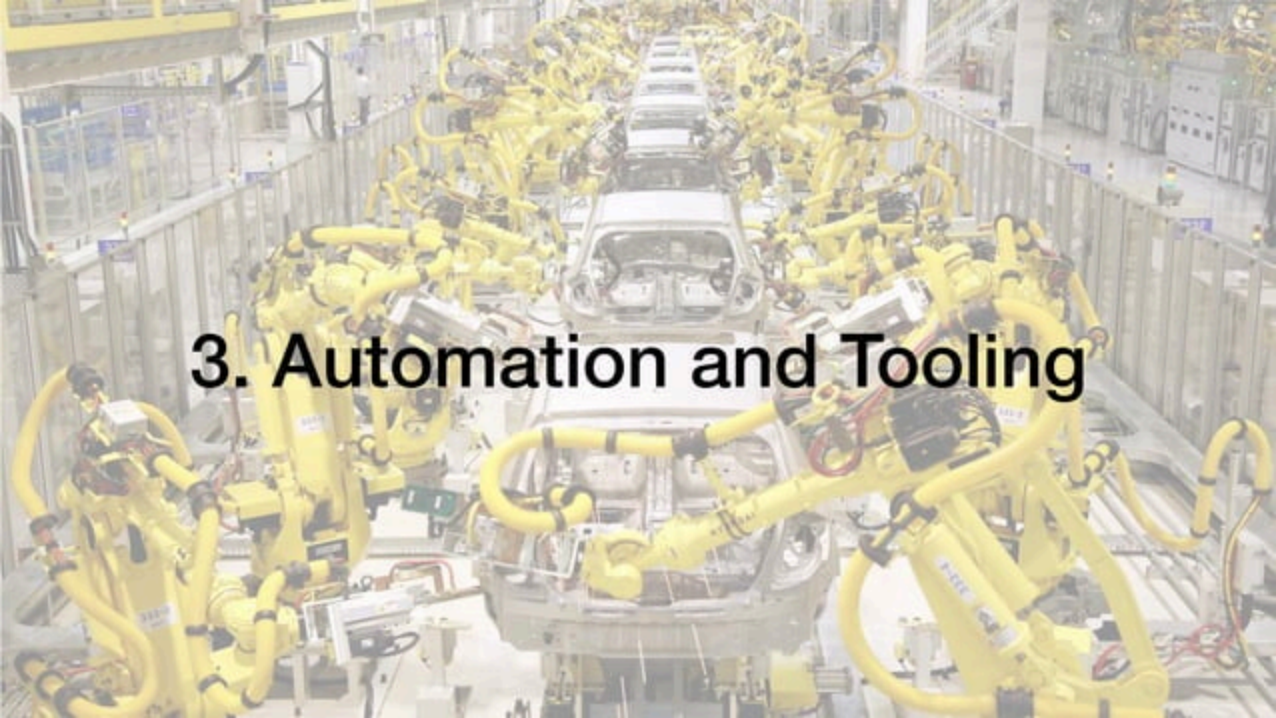
Startup & shutdown hooks

Access to 3rd party libs

Swap, disable, or configure components

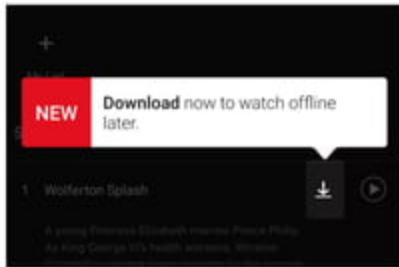
Raw component access



A high-angle, wide shot of a modern automotive assembly line. Numerous yellow robotic arms are positioned along both sides of the production line, working on silver car chassis. The cars are arranged in a single file, receding into the distance. The factory floor is clean and industrial, with various mechanical components and wiring visible on the robots. The lighting is bright and even, highlighting the repetitive nature of the automated process.

### **3. Automation and Tooling**

# Ship a Feature



# Steps



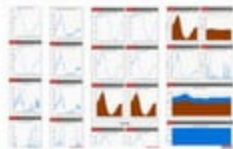
Development



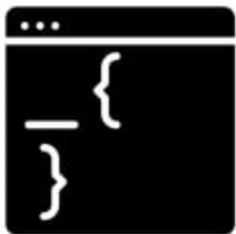
Testing



Deployment



Operations



**Development**

A professional motorcycle racer is shown leaning into a turn on a racetrack. The rider is wearing a blue and yellow racing suit and helmet, and is riding a blue Yamaha motorcycle with yellow and green accents. The motorcycle has various sponsor logos, including Yamaha, Monster, and Energy. The rider is leaning forward and to the left, with the motorcycle tilted at a steep angle. The background is a blurred racetrack and greenery.

Goal: Achieve **higher velocity**  
through better developer  
**ergonomics**

**Q: What's the first thing you have to do when developing a microservice?**

```
$ gcc
xcode-select: note: no developer tools were found at '/Applications/Xcode.app',
requesting install. Choose an option in the dialog to download the command line
developer tools.
$
```



**The "gcc" command requires the command line developer tools. Would you like to install the tools now?**

Choose Install to continue. Choose Get Xcode to install Xcode and the command line developer tools from the App Store.

Get Xcode

Not Now

Install



## CLI for common dev experience



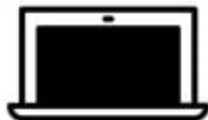
npm



node



GitHub



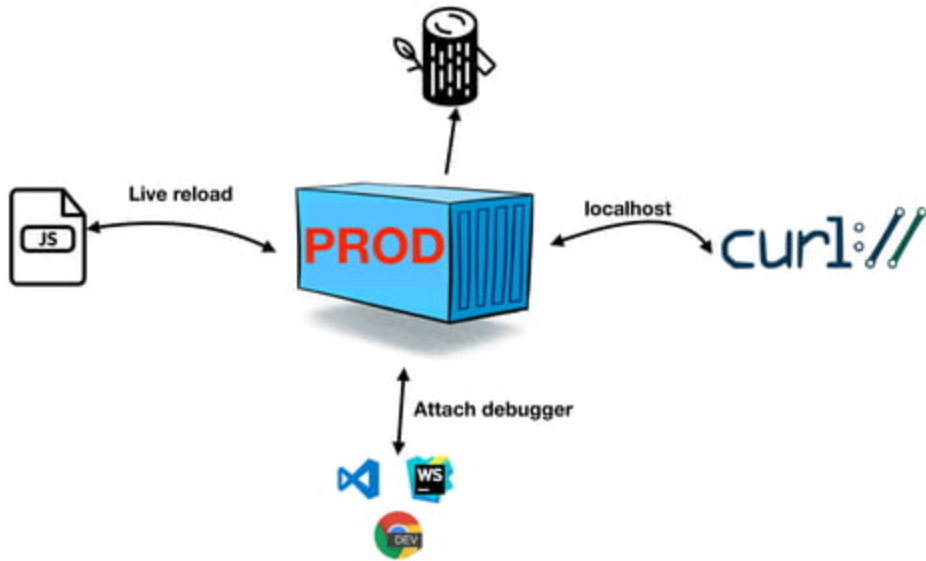
Environment bootstrap

Integrate tooling & services

Run local & cloud



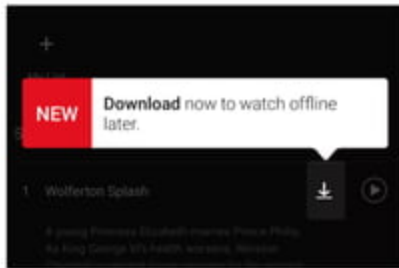
# Local Development





# Testing

# Catch Bugs in Test, not Prod



# Testing



## Preassembled Platform

Owned by  
service team



RPC

Discovery

Registration

Runtime

OS

Configuration

Metrics

Logging

Tracing

Dashboards

Alerts

Stream  
Processing



# Preassembled Platform



RPC

Discovery

Registration

Runtime

OS

Configuration

Metrics

Logging

Tracing

Dashboards

Alerts

Stream  
Processing

**Unstable  
Test Environment  
Availability:  
<<<<<99.99%**

# Solution: Provide First Class Mocks



## Preassembled Platform



RPC

Discovery

Registration

Runtime

OS

Configuration

Metrics

Logging

Tracing

Dashboards

Alerts

Stream  
Processing

# Platform Testing API

You should find this `require` statement in all test files that are auto-generated by `newt init`.

More information about getting started writing and running tests can be found in the [testing user guide](#).

## Mocks

`nqt.mocks`

- Just like a runtime API, need a testing API

The testing framework provides a number of factory methods for generating mocks. The mocks that these methods generate make liberal use of [sinon](#), a stubbing, spying, and introspection library that makes it easy to test that you called APIs correctly. You will find these most useful for testing the NodeQuark runtime APIs. This allows you to write tests that can work independently of the NodeQuark platform itself.

- Provide mocks interface for components

- Gets platform out of the loop for providing mocks

`nqt.mocks.req([options], callback)`

This method creates a mocked request object. This object has all of the methods available to [Restify's](#) request object.

parameter	type	description
<code>[options]</code>	Object	an optional options object
<code>[options.atlas]</code>	Object	an atlas instance to use for this request



## Table of contents

### Mocks

- `nqt.mocks`
- `nqt.mocks.req([options], callback)`
- `nqt.mocks.res([options])`
- `nqt.mocks.properties([props], callback)`
- `nqt.mocks.log([options])`
- `nqt.mocks.atlas([commonTa...])`
- `nqt.mocks.create([options], callback)`

### Integration Testing

- `nqt.initPlatform([options], callback)`
- `platform.client`
- `platform.client.[verb]`
- `platform.client.auth([opts], callback)`
- `platform.dnaDataSource`
- `platform.dnaDataSource.pr...`
- `platform.properties`
- `platform.properties.set(key, val)`
- `platform.properties.unset(k...`
- `platform.properties.reset()`
- `platform.shutdown()`

### Integration Testing with DNA



Deployment





**“Production is war!”**



# Experience Differences



**Veteran**



**Recruit**

# Deploy and Manage Services

- Pre-configured pipelines for publish, canary, deployment and cleanup
- Single command deploy to any stack
- Integration for automated canary analysis
- Pre-configured auto scaling

# Deploy and Manage Services

- **Pre-configured pipelines for publish, canary, deployment and cleanup**
- Single command deploy to any stack
- Integration for automated canary analysis
- Pre-configured auto scaling

**Netflix runs 100% in the cloud**



**amazon**  
**web services**

# How does Netflix Manage Cloud Infrastructure?



# Continuous Delivery for Enterprise

Fast, safe, repeatable deployments

[GET STARTED](#) [INSTALL LATEST](#)

Spinnaker is an open source, multi-cloud continuous delivery platform for releasing software changes with **high velocity** and **confidence**.

Created at Netflix, it has been battle-tested in production by hundreds of teams over millions of deployments. It combines a powerful and flexible pipeline management system with integrations to the major cloud providers.



## Multi-Cloud

Deploy across multiple cloud providers including AWS EC2, Kubernetes, Google Compute Engine, Google Kubernetes Engine, Google App Engine, Microsoft Azure, and Openstack, with Oracle Bare Metal and DC/OS coming soon.

<http://www.spinnaker.io>

NETFLIX | OSS



# Pipelines [🔗](#) What is a “Pipeline”?

[Suggest an Edit](#)

Pipelines are your way of managing deployments in a consistent, repeatable and safe way. A pipeline is a sequence of stages provided by Spinnaker, ranging from functions that manipulate infrastructure (deploy, resize, disable) as well as utility scaffolding functions (manual judgment, wait, run Jenkins job) that together precisely define your runbook for managing your deployments.

The screenshot shows the Spinnaker web interface. The top navigation bar includes 'SPINAKER', 'Search', 'Projects', and 'Applications'. The user is logged in as 'sdverdik@spinnaker-led.net'. The main navigation bar has 'mdservice' and tabs for 'PIPELINES', 'CLUSTERS', and 'TASKS'. On the right, there are links for 'LOAD BALANCERS', 'SECURITY GROUPS', and 'CONFIG'. The main content area displays a pipeline named 'Promote to Prod'. The pipeline consists of several stages: Configuration, Find image from Stage, Deploy Canary, Cutover Manual Approval, Deploy Prod (Red/Black), Tear Down Canary, Wait 2 hrs, and Destroy Old Prod. The 'Deploy Prod (Red/Black)' stage is currently selected. Below the pipeline diagram, there are buttons for 'Add stage' and 'Copy an existing stage'. The details for the 'Deploy Prod (Red/Black)' stage are shown below, including its stage type, description, stage name, and dependencies. At the bottom, there are sections for 'DEPLOY CONFIGURATION', 'EXECUTION OPTIONS', 'NOTIFICATIONS', and 'COMMENTS'. The 'DEPLOY CONFIGURATION' section shows a table with columns for Provider, Account, Cluster, Region, Strategy, Capacity, and Actions.

Provider	Account	Cluster	Region	Strategy	Capacity	Actions
	SP-BURBANK-ACCOUNT	mdservice-prod	default	redblack	Min: 8, Max: 32, Desired: 1	<a href="#">🔗</a> <a href="#">🔗</a> <a href="#">🔗</a>

SPINNAKER Applications Search Properties Analytics Search What's New Help Availability

q\_tvui\_darwin PIPELINES CLUSTERS TASKS LOAD BALANCERS SECURITY GROUPS PROPERTIES CONFIG

### Pipelines

SEARCH

PIPELINES

- CANARY-ALLOCATION
- CANARY-ANALYSIS
- CANARY-SCORE-REVIEW
- DEPLOY
- PUBLISH
- SAFE-CLUSTER-CLEANUP
- SHRINK-SERVER-GROUPS

Reorder Pipelines

STATUS

- Running

Group by Pipeline Show 40 executions per pipeline stage durations

>	CANARY-ALLOCATION	Configure	Start Manual Execution
>	TITUSPRODVPC CANARY-ANALYSIS	Configure	Start Manual Execution
>	CANARY-SCORE-REVIEW	Configure	Start Manual Execution
>	TITUSPRODVPC TITUSTESTVPC DEPLOY	Configure	Start Manual Execution
>	PUBLISH	Configure	Start Manual Execution
>	TITUSPRODVPC SAFE-CLUSTER-CLEANUP	Configure	Start Manual Execution
>	TITUSPRODVPC TITUSTESTVPC SHRINK-SERVER-GROUPS	Configure	Start Manual Execution

# Safe reproducible pipelines

# Deploy and Manage Services

- Pre-configured pipelines for publish, canary, deployment and cleanup
- **Single command deploy to any stack**
- Integration for automated canary analysis
- Pre-configured auto scaling

# Single Command

```
[yunong@gud-yunong] - [~] - [2017-12-01 10:12:14]  
[1] ➤ newt deploy --version=2.0.5 --regions=us-east-1,eu-west-1 \--strategy=instant --memory=8196 --cpu=2 --  
disk=61440 --instances=210 --stack=prod
```

Reduces user error

Abstraction over tooling

Increases developer velocity

# Deploy and Manage Services

- Pre-configured pipelines for publish, canary, deployment and cleanup
- Single command deploy to any stack
- **Integration for automated canary analysis**
- Pre-configured auto scaling

# What's a Canary?

Clients



99.9%

0.1%



Canary: New Build

Relatively “safe” way to test new builds in production

## edge-paas

Canary Score: 100% **PASS**

Completed at: Fri Nov 17 2017 00:29:00 GMT+0800 (CST)  
Region: us-west-2-prod  
Canary: nq\_trui\_darwin-prod-\*105.0.0-105.0.1-canary  
Baseline: nq\_trui\_darwin-prod-\*105.0.0-105.0.1-baseline  
Duration: PT360M  
Version: N/A

## Metric Analysis

Show ☐ pass ☐ low ☐ high ☐ nodataFilter 

Group Name	metrics	Deviation	Score
http	7	100%	<b>PASS</b>
nq.requests.4xx.percentage		0%	<b>PASS</b>
nq.requests.2xx.percentage		-0.0%	<b>PASS</b>
nq.requestLatency.90th.percentile		+0.1%	<b>PASS</b>
nq.requestLatency.99.5th.percentile		+0.4%	<b>PASS</b>
nq.requestLatency.99th.percentile		-0.3%	<b>PASS</b>
nq.requestLatency.95th.percentile		-0.1%	<b>PASS</b>
nq.requests.5xx.percentage		+1.5%	<b>PASS</b>
system	5	100%	<b>PASS</b>

Execute

Config

Dashboard

## Tag: http

Score: 100% **PASS**

## 7 Metrics

Metric Name	Deviation	
nq.requests.4xx.percentage	0%	<b>PASS</b>
nq.requests.2xx.percentage	-0.0%	<b>PASS</b>
nq.requestLatency.90th.percentile	+0.1%	<b>PASS</b>
nq.requestLatency.99.5th.percentile	+0.4%	<b>PASS</b>
nq.requestLatency.99th.percentile	-0.3%	<b>PASS</b>
nq.requestLatency.95th.percentile	-0.1%	<b>PASS</b>
nq.requests.5xx.percentage	+1.5%	<b>PASS</b>

## MANUAL START

david@netflix.com

4 months ago

```
baselineVersion: "2.0.0"
canaryAnalysisInterval: "10"
canaryConfig: "edge-pass"
canaryRolloutStrategy: "24"
canarySuccessThreshold: "90"
canaryUnhealthyThreshold: "10"
canaryVersion: "2.0.0"
cleanupCanaryCluster: "true"
cpus: "2"
disk: "8192"
envConfig: "latest_release"
fullRolloutOnSuccess: "true"
instances: "3"
memory: "8192"
pathForInspect: "/node"
percentage: "3"
regions: "us-east-1,us-west-2,us-west-1"
restartExecutionDuringTimeout: "false"
stack: "prod"
```



## Provide a pre-configured pipeline for canaries





# Deploy and Manage Services

- Pre-configured pipelines for publish, canary, deployment and cleanup
- Single command deploy to any stack
- Integration for automated canary analysis
- **Pre-configured auto scaling**

## Requests per Second (RPS)



■ us-west-2

Max :

Min :

Avg :

Last :

Tot :

Cnt :

Frame: 2d, End: 2017-12-02T00:05-08:00[US/Pacific], Step: 5m

Fetch: 1895ms (L: 60.4k, 2.2k, 1.0; D: 3.6M, 1.3M, 576.0k)



Services ▾



EC2



S3



CloudWatch



IAM



SNS

Edit ▾

[1. Configure Auto Scaling group details](#)[2. Configure scaling policies](#)[3. Configure Notifications](#)[4. Review](#)

## Create Auto Scaling Group

Launch Configuration ⓘ

web-server-scale

Group name ⓘ

1

Group size ⓘ

Start with 1 instances

Network ⓘ

Do not launch into a VPC ▾

[Create new VPC](#)

Availability Zone(s) ⓘ

us-east-1b ✕

us-east-1c ✕

us-east-1d ✕

### ▼ Advanced Details

Load Balancing ⓘ

☒ Receive traffic from Elastic Load Balancer(s)

Health Check Grace Period ⓘ

300

seconds

Monitoring ⓘ

Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration web-server-scale. Instances launched from it will use Basic Monitoring metrics, provided at 5 minute frequency.

[Learn more](#)

Easy to get wrong!

## Clusters

[Edit multiple](#)

Show

☒ Instances☐ with details[Create Server Group](#)

Filtered by: SEARCH: ACCOUNT: prod STACK: prod Clear All

PROD

428 ▲ : 100%

EU-WEST-1

V462: Build: #4969

183 ▲ : 100%

US-EAST-1

V470: Build: #4969

100 ▲ : 100%

US-WEST-2

V396: Build: #4969

145 ▲ : 100%



## Preconfigured Auto Scaling

[Server Group Actions](#)[Insight](#)

- ✓ AZRebalance
- ✓ HealthCheck
- ✓ ReplaceUnhealthy
- ✓ ScheduledActions

[Edit Scaling Processes](#)

### SCALING POLICIES

#### SIMPLE SCALING

Whenever Average of CPUUtilization is  $\leq 30$   
for at least 7 consecutive periods of 60  
seconds



#### SIMPLE SCALING

Whenever Average of CPUUtilization is  $\geq 40$   
for at least 4 consecutive periods of 60  
seconds

[Create new scaling policy](#)



# Operations

# Consolidated View



SKIPPER

SERVICES

API

DOCUMENTATION

SUPPORT

yungh

SERVICES > CHROMECAST



Stack     Alerts

DEPLOYMENTS

PUBLICATIONS

CANARY EXECUTIONS

RECENT ACTIVITY











































STACK

PROD

STAGING

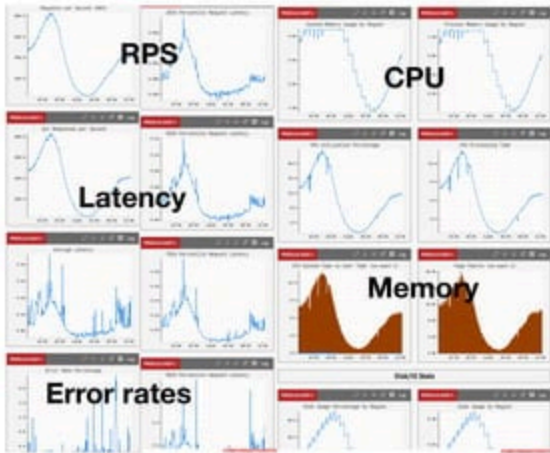
INT

TEST

PROD	Version Constraint	Image Version	Region	Instance Counts	Created	Traffic (e-2w)	Dashboard	Stream Logs	Actions
	*3.0.0	3.0.0	us-east-1	1     	2017-06-20 08:58:54 PDT				 
		3.0.0	us-west-2	1     	2017-06-20 08:58:54 PDT				 
		3.0.0	eu-west-1	1     	2017-06-20 08:58:55 PDT				 
	*1.0.0	1.0.32	us-east-1	30     	2017-06-07 10:28:12 PDT				 
		1.0.32	us-west-2	30     	2017-06-07 11:09:15 PDT				 
		1.0.32	eu-west-1	30     	2017-06-07 13:00:07 PDT				 

 Up (Taking traffic)  Down  Starting  Disabled/Out of Service

# Generated Dashboards & Alerts



- alert-do-not-reply** Jun 22
- ☐ Average latency is more than 1s for `rs` in `eu-west-1`
- Prod eu-west-1 High\_Latency Summary: Average latency is more than 1s for `rs` in `eu-west-1`
- alert-do-not-reply (4)** Jun 22
- ☐ High Process Starts for `rs` in `us-west-2`
- Prod us-west-2 High\_Process\_Starts Summary: High Process Starts for `rs` in `us-west-2`
- alert-do-not-reply** Jun 22
- ☐ Average latency is more than 1s for `rs` in `us-west-2`
- Prod us-west-2 High\_Latency Summary: Average latency is more than 1s for `rs` in `us-west-2`
- alert-do-not-reply (4)** Jun 22
- ☐ High Process Starts for `rs` in `us-east-1`
- Prod us-east-1 High\_Process\_Starts Summary: High Process Starts for `rs` in `us-east-1`
- alert-do-not-reply (4)** Jun 22
- ☐ High Process Starts for `rs` in `eu-west-1`
- Prod eu-west-1 High\_Process\_Starts Summary: High Process Starts for `rs` in `eu-west-1`
- alert-do-not-reply (2)** Jun 22
- ☐ Disk Usage is over 80% for `rs` in `us-east-1`
- Prod us-east-1 High\_Disk\_Usage Summary: Disk Usage is over 80% for `rs` in `us-east-1`
- alert-do-not-reply** Jun 22
- ☐ Disk Usage is over 80% for `rs` in `us-west-2`
- Prod us-west-2 High\_Disk\_Usage Summary: Disk Usage is over 80% for `rs` in `us-west-2`
- alert-do-not-reply (2)** Jun 21
- ☐ Availability dropped - Action Required
- Prod us-west-2 P90\_Availability Summary: Availability dropped - Action Required Check time

# Automated Analytics & Tooling



**CPU profiling**

3c11e0d539	18134	3 <anonymous> (as jspb.Map.Entry_): key, ...
3c11ee5721	2345	13 HTTPParser: 0, 1, 2, 3, 4, _headers, ...
3c11e12559	18699	3 Object: onReceiveMetadata, ...
3c11e43471	18176	2 InterceptingListener: delegate, ...
3c11e42d99	4225	9 Object: context, setupRuntimeApi, ...
3c11e0d0e9	19884	2 InterceptingCall: next_call, requester
3c11e9b599	3956	10 logger: domain, _events, _eventsCount, ...
3c11e0c859	13375	3 Object: server, content-type, ...
333a0f7c6011	2	23940 Array
3c11e34f99	51062	1 Array
3c11e682e9	2364	23 WritableState: objectMode, ...
9366386b51	2514	23 Array
3c11e42dd1	4985	12 Url: protocol, slashes, auth, host, ...
3c11e68581	13795	6 <anonymous> (as proto.google.protobuf.StringV
9366386d39	3240	27 ServerResponse: domain, _events, ...
3c11edcf89	2976	30 IncomingMessage: _readableState, ...
3c11e3de69	49756	2 Object: name, time
3c11e85321	4815	21 Object: start, stop, getDebugInfo, ...
3c11e0d459	50789	2 Array
3c11e9c329	3104	45 Object: x-netflix.client.asg.name, ...
3c11e68209	6794	21 ReadableState: objectMode, ...
3c11e2e3e1	3738	90 Array

**Core dump analysis**



# Our PaaS to Velocity & Reliability

1. Standardized components
2. Preassembled platform
3. Automation and tooling


December, 3, 2017

Questions? 问题?

We're hiring! 欢迎加入我们的团队!


NETFLIX

Yunong Xiao,  
Principal Software Engineer, Netflix  
[yunong@netflix.com](mailto:yunong@netflix.com),

 @yunongx,

 yunongx

<http://yunong.io>

 [linkedin.com/in/yunongxiao](https://www.linkedin.com/in/yunongxiao)



架构迎接未来变化  
IAS2017 • NANJING