Netflix’s Journey to the Cloud: Lessons Learned from Netflix’s Migration to the Public Cloud

Jason Chan, Cloud Security Architect
In-Depth Seminars Track – D1
Agenda

• Background
• Key decisions
  – Why cloud?
  – Which cloud, and how?
• Cloud security @ Netflix
  – Basic approach
  – Implementation specifics
• Lessons learned
BACKGROUND
Netflix Inc.

With more than 27 million streaming members in the United States, Canada, Latin America, the United Kingdom and Ireland, Netflix, Inc. is the world's leading internet subscription service for enjoying movies and TV programs . . .

Source: http://ir.netflix.com
Me

• Cloud Security Architect @ Netflix
• Responsible for:
  – Cloud app, product, and operational security
• Previously:
  – Led security team at VMware
  – Previously, primarily security consulting at @stake, iSEC Partners
• ISACA
  – CISM, CISA
WHY CLOUD?
Outages and Availability

Netflix Outage Angers Customers
Some going to Blockbuster

By Mike Sachoff · August 14, 2008 · 7 Comments

• Large-scale outage of data center systems
• Roughly 3 days of DVD shipping outage
• During initial stages of streaming service

http://www.webpronews.com/netflix-outage-angers-customers-2008-08
http://www.reuters.com/article/2008/08/15/netflix-outage-idUSN1539639720080815
Streaming Service

• Goal is a global streaming service
  – DVD is US only

• Availability becomes much more critical
  – DVD involves a more predictable pattern

• Capacity and usage

---

Netflix Eats Up 32 Percent of U.S. Bandwidth During Peak Times

By Chloe Albanesius | October 27, 2011 09:40am EST | 7 Comments

http://www.pcmag.com/article2/0,2817,2395372,00.asp
Outgrowing the Data Center

Netflix API: Requests Per Month

Requests in Billions

Datacenter Capacity

Jan-10 to Jul-11
Seven Aspects of Netflix Culture

- Values are what we value
- High Performance
- Freedom & Responsibility
- Context, not Control
- Highly Aligned, Loosely Coupled
- Pay Top of Market
- Promotions & Development

http://www.slideshare.net/netflix
Things We Don’t Do

- Get stuck with wrong config
- Wait
- File tickets
- Wait
- Ask permission
- Wait
- Things We Don’t Do
- Wait
- Run out of space/power
- Wait
- Plan capacity in advance
- Have meetings with IT
- Wait
Why Cloud? A Summary

• Needed
  – Better availability
  – Support a fast-growing, global service
  – Technical agility to match company culture

• Textbook use case for cloud
WHICH CLOUD?
Public cloud – why?

• We want to use clouds, we don’t have time to build them
  – Public cloud for agility and scale
  – Undifferentiated heavy lifting (Bezos, Vogels)

• Netflix choice was AWS with our own platform and tools
  – Unique platform requirements and extreme scale, agility and flexibility
AWS and Alternatives

• Public Cloud Alternatives to AWS
  – Far fewer features, much smaller scale
  – Less mature APIs, many variants of APIs
  – Some have additional features or performance

• Private Cloud Alternatives
  – Often harder to build and run than you think
  – Much higher costs w/o scale and multi-tenancy
  – Often driven by IT-Ops needs rather than developers
What about other PaaS?

• CloudFoundry – Open Source by VMware
  – Developer-friendly, easy to get started
  – Missing scale and some enterprise features

• Rightscale
  – Widely used to abstract away from AWS
  – Creates its own lock-in problem

• AWS is growing into this space
  – We didn’t want a vendor between us and AWS
  – We wanted to build a thin PaaS, that gets thinner
HOW?
Netflix PaaS Principles

• Maximum functionality
  – Developer productivity and agility

• Leverage as much of AWS as possible
  – AWS is making huge investments in features/scale

• Interfaces that isolate apps from AWS
  – Avoid lock-in to specific AWS API details

• Portability is a long term goal
  – Gets easier as other vendors catch up with AWS
Build a global PaaS on AWS IaaS

- Supports all AWS regions and availability zones
- Supports multiple AWS accounts
- One-click deployment and balancing across three data centers
- Cross-region and account data replication and archive
- Dynamic and fine-grained security
- Automatic scaling to thousands of instances
- Monitoring for millions of metrics
- I18n, L10n, geo IP routing
Organization Rearchitecture

• Cloud is run by developer organization
  – Our IT department is the AWS API
  – We have no IT staff working on cloud (they do corp IT)

• Cloud capacity is 10x bigger than Datacenter
  – Datacenter oriented IT staffing is flat
  – We have moved a few people out of IT to write code

• Traditional IT Roles are going away
  – Less need for SA, DBA, Storage, Network admins
  – Developers deploy and run what they wrote in production
Cloud and Platform Engineering

- Build an engineering organization focused on facilitating and optimizing cloud usage

<table>
<thead>
<tr>
<th>Engineering Tools</th>
<th>• Orchestration, build and deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Solutions</td>
<td>• Monitoring, consulting, Simian Army</td>
</tr>
<tr>
<td>CORE</td>
<td>• 24/7 site reliability</td>
</tr>
<tr>
<td>Platform Engineering</td>
<td>• Core shared components and libraries</td>
</tr>
<tr>
<td>Security</td>
<td>• Application, engineering, and operational</td>
</tr>
<tr>
<td>Cloud Persistence Engineering</td>
<td>• Cassandra, SDB, RDS, S3</td>
</tr>
<tr>
<td>Cloud Performance</td>
<td>• Testing, optimization, cost</td>
</tr>
<tr>
<td>Cloud Architecture</td>
<td>• Overall design patterns</td>
</tr>
</tbody>
</table>
Netflix Cloud Camp

- For developers – one day orientation
- ½ presentations, ½ hands-on
- Build “Hello World” using NFLX PaaS
- Build and security integration, monitoring
- Cassandra read/writes
Service Rearchitecture

- Data Center
- Central SQL Database
- Sticky In-Memory Session
- Chatty Protocols
- Tangled Service Interfaces
- Instrumented Code
- Fat Complex Objects
- Components as Jar Files

- Cloud Architecture
- Distributed Key/Value NoSQL
- Shared Memcached Session
- Latency Tolerant Protocols
- Layered Service Interfaces
- Instrumented Service Patterns
- Lightweight Serializable Objects
- Components as Services
Progression: Netflix Deployed on AWS

2009
- Content
  - Video Masters
  - EC2
  - S3
  - CDNs

2009
- Logs
  - S3
  - EMR Hadoop
  - Hive
  - Business Intelligence

2010
- Play
  - DRM
  - CDN routing
  - Bookmarks
  - Logging

2010
- WWW
  - Sign-Up
  - Search
  - Movie Choosing
  - Ratings

2010
- API
  - Metadata
  - Device Config
  - TV Movie Choosing
  - Social

2010
- CS
  - International CS lookup
  - Diagnostics & Actions
  - Customer Call Log
  - CS Analytics
Netflix OSS

• Open source components to drive innovation
CLOUD SECURITY @ NETFLIX: BASIC APPROACH
First, some notes on scale

• Thousands of:
  – Instances
• Hundreds of:
  – Developers
  – Applications
• Dozens of:
  – Engineering teams
  – Deployments per day
• Zero of:
  – Architectural review committees
  – Change review boards
Word Association

Cloud
• Freedom
• Agility
• Self-service
• Scale
• Automation

Security
• Pain
• Gatekeeper
• Standards
• Control
• Centralized
Risk-Based Approach

• Understand organization’s risk appetite
• Not everything is equal value
• Understand what’s important and prioritize appropriately
Integrate with and Leverage Tooling

• Build and deployment pipeline is a key point for security integration
• Security uses the same tools as developers
• Think integration vs. separation
Make Doing the Right Thing Easy

- Developers are lazy
- Operational model incentivizes robust code
- Sensible defaults
- Libraries for common, but difficult, security tasks
- Publish and evangelize patterns
Embrace Self-Service, with Exceptions

• IMHO, self-service is the breakthrough characteristic of the cloud

• Put security configuration in the hands of end-users, with some exceptions:
  – SSL certificate management
  – Some firewall rules
  – User and permissions management
CLOUD SECURITY @ NETFLIX: PROGRAMMABLE INFRASTRUCTURE AND THE SECURITY MONKEY
Common Challenges for Security Engineers

• Lots of data from different sources, in different formats
• Too many administrative interfaces and disconnected systems
• Too few options for scalable automation
How do you . . .

- Add a user account?
- Inventory systems?
- Change a firewall config?
- Snapshot a drive for forensic analysis?
- Disable a multi-factor authentication token?

- CreateUser()
- DescribeInstances()
- AuthorizeSecurityGroupIngress()
- CreateSnapshot()
- DeactivateMFADevice()
Security Monkey

• Designed to support culture of freedom and responsibility
• Centralized framework for cloud security monitoring and analysis
• Certificate and cipher monitoring
• Firewall configuration checks and cleanup (with Janitor Monkey)
• User/group/policy monitoring
CLOUD SECURITY @ NETFLIX: MODEL-DRIVEN ARCHITECTURE
Data Center Patterns

• Long-lived, non-elastic systems
• Push code and config to running systems
• Tech-specific deployment processes
• ‘Snowflake phenomenon’
• Difficult to sync or reproduce environments (e.g. test and prod)
Cloud Patterns

- Ephemeral nodes
- Dynamic scaling
- Hardware is abstracted
- Programmable infrastructure
- Cloud primitives support common deployment patterns
Netflix Build and Deploy

Autoscaling Deployments

Baked AMI
- Base Linux
- App code
- App dependencies
- App-specific config

Launch Config
- Instance type
- Security group config

Autoscaling Group
- Target data centers
- Cluster min/max

Netflix Web App X
Autoscaling Results and Ramifications

• Goals:
  – # of systems matches load requirements
  – Load per server remains constant

• Continuously adding and removing nodes
  – Based on demand, system health

• New nodes must mirror existing

Every change is a new push
Operational Impact

- No changes to running systems
- No CMDB
- No systems management infrastructure
- No snowflakes
- Fewer logins to prod systems
- Trivial “rollback”
- No room for dev vs. ops argument!
Security Impact

• File integrity monitoring
• User activity monitoring
• Vulnerability management
• Patch management
CLOUD SECURITY @ NETFLIX: LESSONS LEARNED
Tools and Vendors

• Many data center oriented tools don’t travel to the cloud well

• Drive security vendors/tool makers to:
  – Scale
  – Handle dynamic environments
  – Make everything API-accessible/driven
Organizational and Operational

• Understand the personnel you need for this kind of environment
  – Security staff must be able to write code
  – Need familiarity with engineering processes to efficiently integrate

• Monitor and instrument the events and elements you care about
  – Automated alerting and escalation vs. NOC/SOC staring at multi-displays
Regulatory Compliance

• **Pathfinders beware!**

• Security, auditors, and regulators are still in early stages of defining adequate, secure, and compliant cloud operations

• Be prepared for a knowledge/experience/comfort gap:
  – N-tier vs. distributed systems
  – SOD vs. DevOps
  – QA/UAT vs. CI/CD
Questions?

• chan@netflix.com
• http://techblog.netflix.com