DevOps.com WEBINAR

5C's Series

Cloud Migration: Solving for Growth with Containers & DevOps on AWS

How LendingHome Built Scale, Control & Agility

LendingHome aws nclouds Your DevOps Partner

#5CsDevOps

DevOps.com WEBINAR

5 C's That Changed the Future

CI, CD, Containers, Cloud & Culture

Case studies in creating sustainable systems that empower the business to deliver innovation faster, with predictability.

prodea

CI/CD, Containers & Culture

▶ WATCH REPLAY

nDimensional

Cloud Infrastructure & Culture

▶ WATCH REPLAY

LendingHome

Cloud Migration & Containers

▶ WATCH REPLAY

Join us for the series and bring your team!

Cloud Migration: Solving for Growth with Containers & DevOps on AWS

Donovan Bray

Senior DevOps Manager

LendingHome

Juan Villa

Solutions Architect

aws

JT Giri

CEO & Co-founder

nclouds

LendingHome

Using the power of technology and human ingenuity, we're reimagining the mortgage process from the ground-up to create a seamless and transparent process for homebuyers, real estate professionals, and investors.

Accomplishments

- Mortgage loans funded: $2B+
- Homes financed: 10,000+
- Principal returned: $1.3B+
- Equity financing raised: $165MM+

Forbes FINTECH 50 2018

LendIt Industry Awards 2017 WINNER

FINTECH BREAKTHROUGH 2018

TECH100 2018

LendingHome

Past State 2016-2017

- **Replacing CloudFlare - 2016**
  - Biggest contributor to downtime (2016)
  - DNS, CDN, WAF
  - Leap Second Bug

- **Moving out of Heroku - 2017**
  - Biggest contributor to downtime was Heroku routing mesh - H10 errors (2017)
  - All data on Heroku stored on US East 1 (single region)

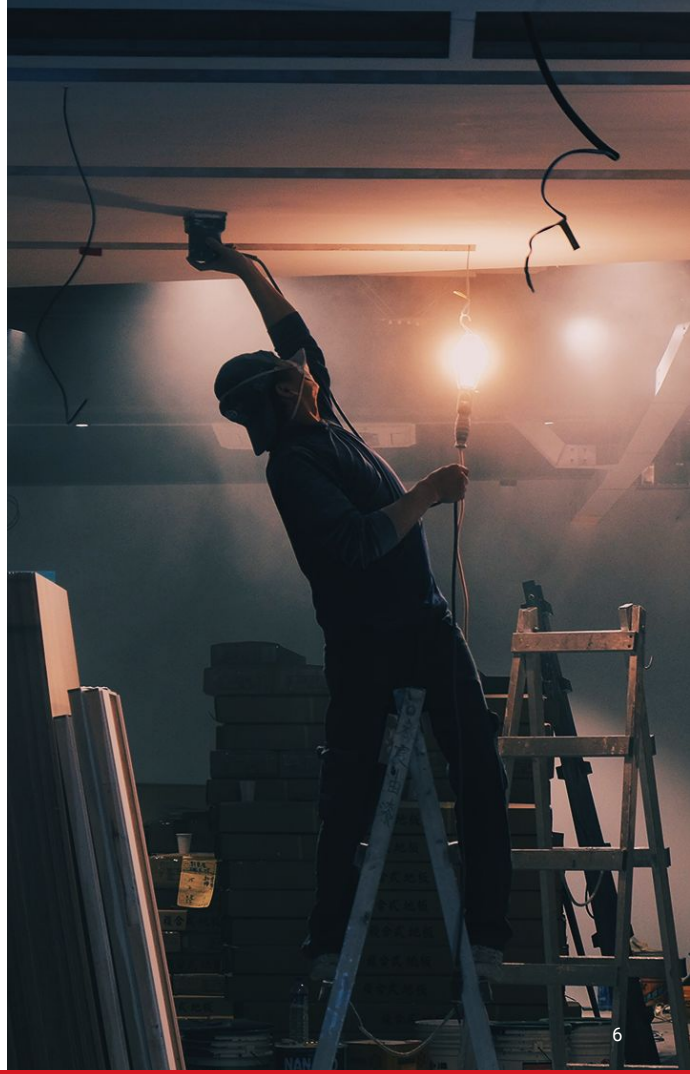
Challenges

Downtime

- Impacting user experience, internal SLAs (loan originators, analysts), operations

Black box (with Heroku)

- H10 errors (in routing mesh), issues inside the black box
- Trading abstraction for control & flexibility



Challenges

Regulatory & Audit compliance

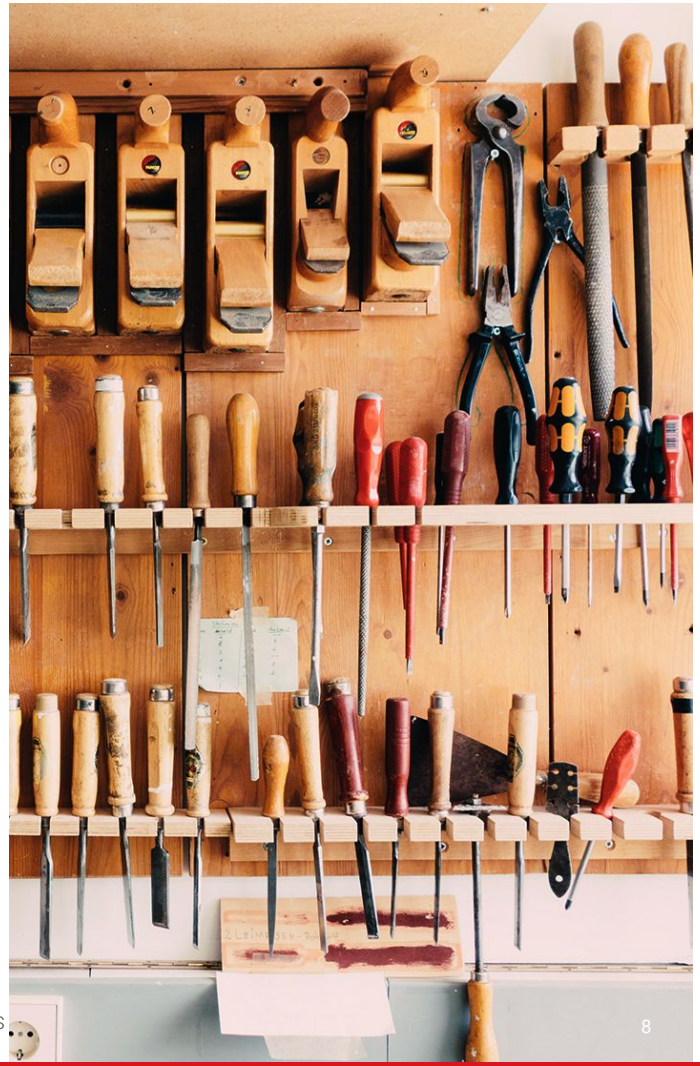
- How to answer, “Where is the second data center?”
- **Heroku could only be in US East (using public spaces)**

Security

- Inability to prevent databases or applications from being exposed to the Internet
- No granular access to third-party service add-ons (e.g., New Relic)

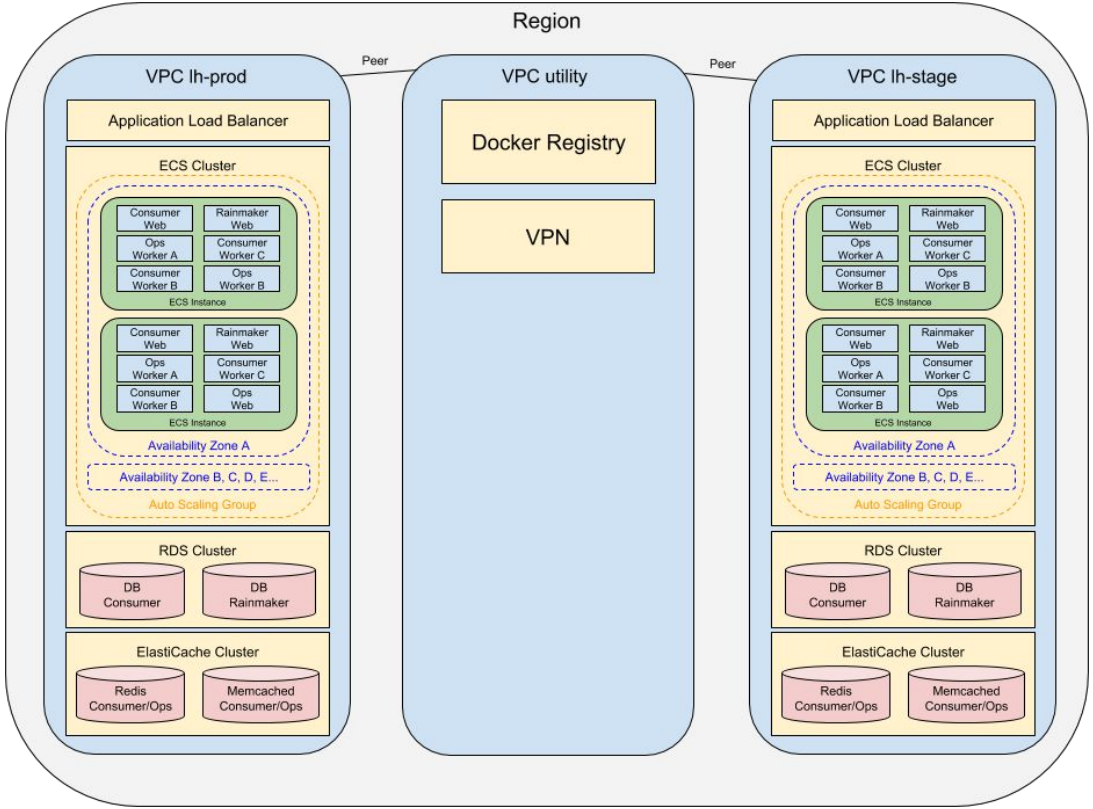
Solution Functional Building Blocks

- VPN
- VPC peering
- Containerization (ECS) - building the DevOps layer
- Automated CI/CD pipeline (Buildkite)
- DNS Made Easy (primary), easyDNS (secondary)
- Amazon CloudFront CDN
- AWS Web Application Firewall (WAF)
- AWS Lambda
- AWS CloudFormation, with our own CLI
- Amazon ElastiCache
- AWS Application Load Balancer
- Amazon Aurora

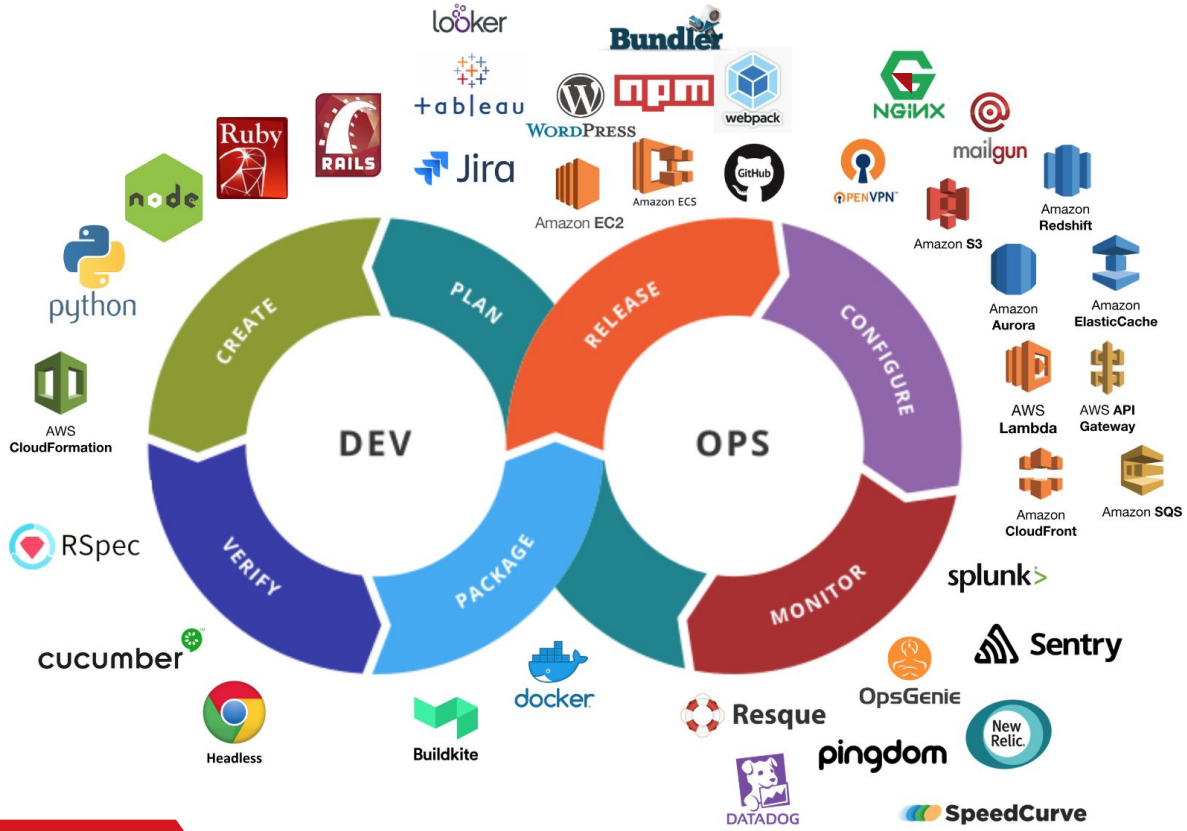


Solution Architecture

- VPC is environment boundary
- Single ECS cluster per VPC
- Shared services VPC topology
- Each cluster has at least a root variant
- Each variant:
  - Has 6 applications currently
  - Comprised of 35 services
- Largest cluster:
  - 25 variants
  - 847 services (not all scaled up)
  - 470 tasks
  - 60 ECS hosts - m4.xlarge



Solution Tools & Services



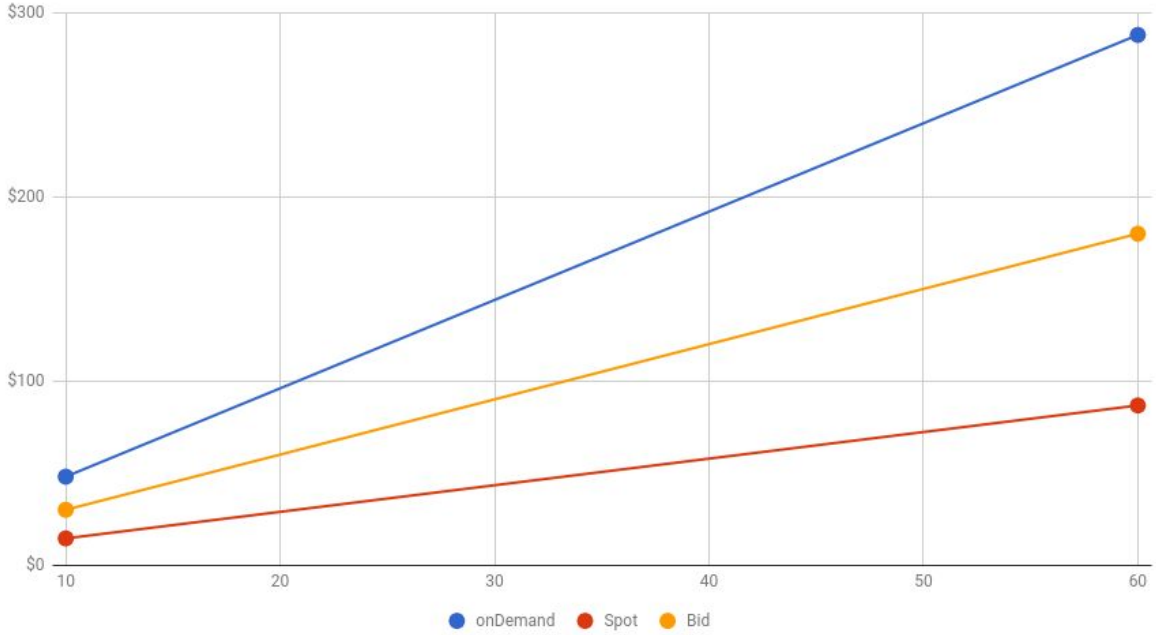
AWS Services

- Amazon Aurora
- Amazon API Gateway
- Amazon CloudFront
- AWS CloudFormation
- Amazon ECS
- Amazon EC2
- Amazon ElastiCache
- Amazon Redshift
- Amazon S3
- Amazon SQS

Spot Instances

- **Works well**
  - Testing environment async workers to reduce cost
  - CI/CD - 100’s of instances - saving $$
- **Less well for web app servers (where ECS set at 0)**

Daily Price Comparison m4.xlarge



- Scales: 10 - 60 instances
- Max. willing to pay price: $0.125/h
- Daily savings: $33 - $201, depending on scaling events

Results

Uptime

- Production - virtually no downtime attributed to infrastructure since migration

Modular infrastructure

- Flexibility to mix & match best-of-breed

Reduced cost

- Leveraging Spot Instances

Results *(cont.)*

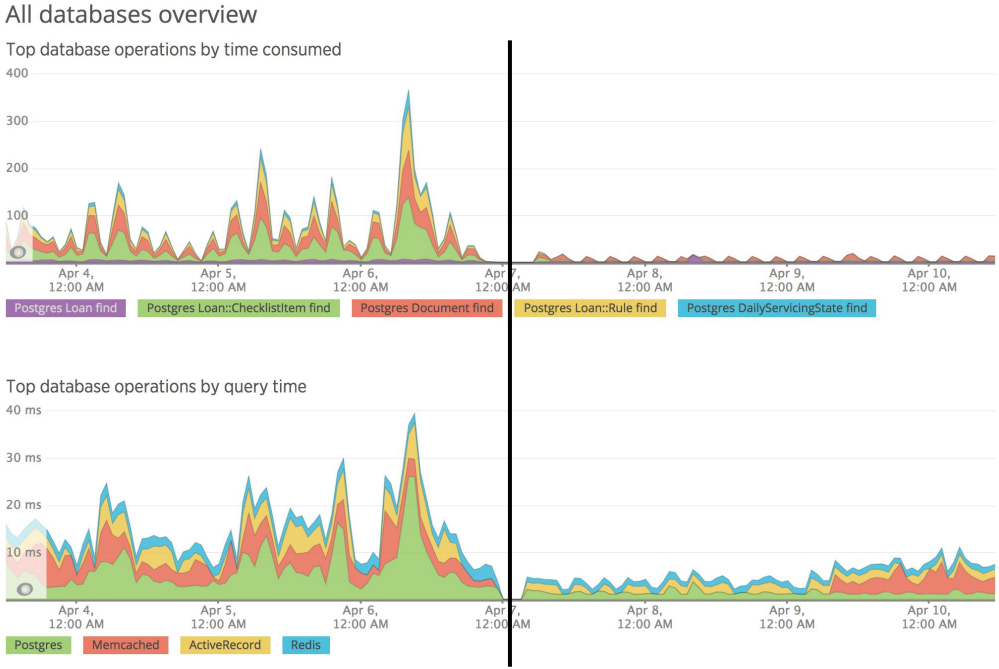
Performance

- **The Power of Aurora**
  - Aurora versions of PSQL & MySQL up to 3.5x faster performance vs. standard versions
- **SpeedIndex**
  - Before: 3.11s
  - After: 2.92s

Improved security

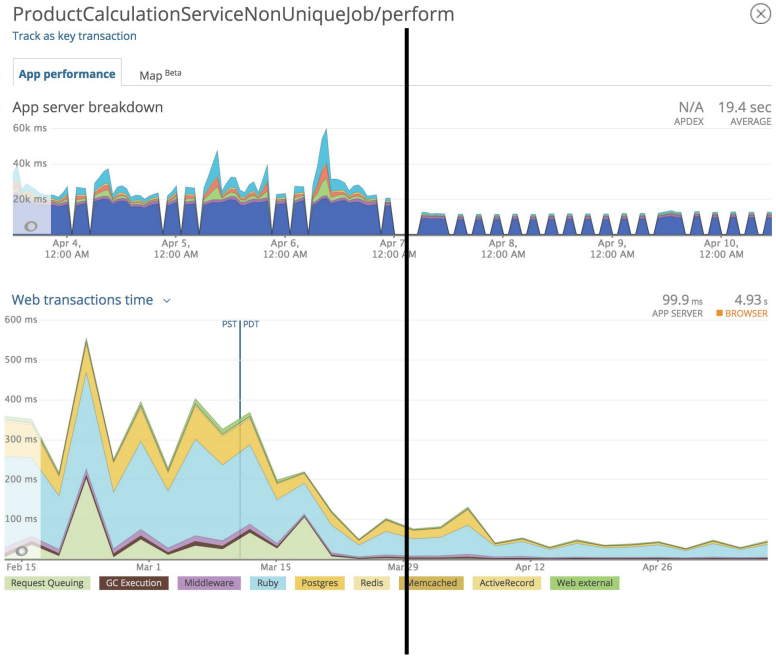
- Infrastructure set up following industry best practices with security built into the automation
- Moved private apps & DB servers behind VPN

Performance



Before

After



Before

After

Lessons Learned

- **Limits, limits, limits**
- **Black box vs. Control**
  - Not like coming out of any other data center … Heroku does so much for you
  - So we needed to build a DevOps layer on AWS
- **Visibility/monitoring**
  - Loss of Logentries and Librato
  - Added Datadog, Splunk
- **Limitations of Database Migration Service (DMS)**
  - Moving data is a big part of the migration from Heroku to AWS
  - RDS Postgres jsonb and hstore limits
  - Needed cooperation from Heroku for administrative changes

Lessons Learned *(cont.)*

- **Create support infrastructure**
- **Write your own scripts - bridging the gap for all tools**
- **CloudFormation-Lambda framework - used API to call Lambdas (DNS, IP plan)**
- **Segment/segregate CloudFormation templates - no monoliths *(see slide)***
- **Comprehensive IP plan - ability to summarize *(see slide)***
- **Naming Convention Policy including abbreviation rules - stick to it religiously *(see slide)***

Segmenting AWS CloudFormation Scripts

1. **Stack**
   - Network
     - VPC, Subnets, Gateways, NAT's, Peering, CDN
   - ECS
   - EFS
   - WAF
2. **Application**
   - ALB
   - DNS
   - Databases
     - RDS
     - ElastiCache
3. **Variant**
   - Environment variables
   - DNS
   - Listeners
   - Roles
4. **Deployment**
   - Service
   - Tasks
   - Outputs are collected at each level and fed into the next level as CloudFormation parameters via our CLI

Comprehensive IP Plan

- Summarizable routes by blocks based on type and location
- Allow space for merger and acquisitions
- Using the Corporate VPN only gets you into Corporate resources
- Using the Production VPN only gets you into Production resources
- Nothing from our infrastructure shall prevent a capable VPN from connecting to both the corporate and production VPNs at the same time
- There are expected to be primary and secondary allocations of each type for a location

- Design VPC's for subnets for at least 4 AZ's
- Allow space for non-AWS use
- Prefer to avoid 192.168 because it is used in consumer routers
- Prefer to avoid 172.16 because some ISP's are now using it

A	B	C	D
Company (10)	Usage	Silo (11)	Usage
10.0.0.0/10	Lendinghome	10.0.0.0/11	Corp
		10.32.0.0/11	Production
10.64.0.0/10	Unused	10.64.0.0/11	Unused
		10.96.0.0/11	Unused
10.128.0.0/10	Unused	10.128.0.0/11	Unused
		10.160.0.0/11	Unused
10.192.0.0/10	Unused	10.192.0.0/11	Unused
		10.224.0.0/11	Unused

C	D	E	F	G	H	I	J	L	M	N	O	P	Q
Silo (11)	**Usage**	**Location (14)**	**Usage**	**Region (16)**	**Usage**	**Project (22)**	**Usage**	**AZ (25)**	**Usage**	**Visibility (26)**	**Usage**	**Subnet Name**	**Hosts**
10.32.0.0/11	Production	10.32.0.0/14											
		10.36.0.0/14	AWS-US	10.36.0.0/16	us-east-1	10.36.0.0/22							
						10.36.4.0/22	utility	10.36.4.0/25	us-east-1a	10.36.4.0/26	Public	utility-us-east-1a-public	62
							utility		us-east-1a	10.36.4.64/26	Private	utility-us-east-1a-private	62
							utility	10.36.4.128/25	us-east-1b	10.36.4.128/26	Public	utility-us-east-1b-public	62
							utility		us-east-1b	10.36.4.192/26	Private	utility-us-east-1b-private	62
							utility	10.36.5.0/25	us-east-1c	10.36.5.0/26	Public	utility-us-east-1c-public	62
							utility		us-east-1c	10.36.5.64/26	Private	utility-us-east-1c-private	62
							utility	10.36.5.128/25	us-east-1d	10.36.5.128/26	Public	utility-us-east-1d-public	62
							utility		us-east-1d	10.36.5.192/26	Private	utility-us-east-1d-private	62
							utility	10.36.6.0/25	us-east-1e	10.36.6.0/26	Public	utility-us-east-1e-public	62
							utility		us-east-1e	10.36.6.64/26	Private	utility-us-east-1e-private	62
							utility	10.36.6.128/25					
							utility	10.36.7.0/25					
							utility	10.36.7.128/25					
						10.36.8.0/22							
						10.36.12.0/22	lh-prod	10.36.12.0/25	us-east-1a	10.36.12.0/26	Public	lh-prod-us-east-1a-public	62

Naming Convention Policy

Limits

- Security group: up to 255 characters in length
- ELB: cannot be longer than 32 characters
- Role: cannot exceed 64 characters
- ECS service: cannot exceed 32 ASCII characters
- CloudFormation:
  - Maximum size of each mapping name. 255 characters
  - Maximum size of an output name 255 characters
  - Maximum size of a parameter name 255 characters
  - Maximum size of a resource name 255 characters
- RDS
  - Postgres
    - You can't use "admin" for the DB user

Small Excerpt:

- The focus of the rules are of and about the "application"
- The service name and any modifiers (stage | prod) should be abbreviated if it is over 15 combined characters
- The VPC name must be omitted from underlying services
- The name style for multi-words must be applied in this order and only move onto the next style if the context demands it
  - dasherized
    - lh-stage-consumer
  - underscored
    - lh_stage_consumer
  - CamelCased
    - LHStageConsumer
- Unless it is REQUIRED for disambiguation omit the object type from names
  - good for a security group
    - vpn-ldap
  - bad for a security group
    - vpn-ldap-sg

What's Next?

- Regional replication (Amazon Aurora Postgres)
- Passive second region
- Replace visibility with Splunk, Datadog
- Right-sizing count & size of containers for cost optimization
    - If you can keep container-to-ECS-instance cost the same then going with larger ECS instances is advantageous with vendors like New Relic who license by host and ignore containers
- Reserved Instances

Resources

- Route Summarization
  - http://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise/design-zone-smart-business-architecture/sba_ipAddr_dg.pdf
- Jodies IP Subnet Calculator
  - http://jodies.de/ipcalc
- Google IP Functions
  - https://docs.google.com/document/d/18uB0Cbs37WOe1C-em5Rae6Hu8y5rUOvrMo2MoOyVlG4/edit
- Top 13 Amazon Virtual Private Cloud (VPC) Best Practices
  - http://cloudacademy.com/blog/top-13-amazon-virtual-private-cloud-best-practices/
- AWS VPC configuration: 5 kick-yourself mistakes
  - http://cloudacademy.com/blog/aws-vpc-configuration-five-kick-yourself-mistakes/
- Invalid VPC Peering Connection Configurations
  - http://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/invalid-peering-configurations.html
- Tail Stack Events
  - https://www.npmjs.com/package/tail-stack-events


nclouds
Your DevOps Partner
Create sustainable modern infrastructure that empowers the business to deliver innovation faster, with predictability.

DevOps.com WEBINAR

5 C's That Changed the Future

CI, CD, Containers, Cloud & Culture

Case studies in creating sustainable systems that empower the business to deliver innovation faster, with predictability.

prodea

CI/CD, Containers & Culture

▶ WATCH REPLAY

nDimensional

Cloud Infrastructure & Culture

▶ WATCH REPLAY

LendingHome

Cloud Migration & Containers

▶ WATCH REPLAY

Join us for the series and bring your team!

Cloud Migration: Solving for Growth with Containers & DevOps on AWS

Donovan Bray

Senior DevOps Manager

LendingHome

Juan Villa

Solutions Architect

aws

JT Giri

CEO & Co-founder

nclouds

DevOps.com WEBINAR

5 C's That Changed the Future

CI, CD, Containers, Cloud & Culture

Case studies in creating sustainable systems that empower the business to deliver innovation faster, with predictability.

prodea

CI/CD, Containers & Culture

▶ WATCH REPLAY

nDimensional

Cloud Infrastructure & Culture

▶ WATCH REPLAY

LendingHome

Cloud Migration & Containers

▶ WATCH REPLAY

Join us for the series and bring your team!