The Wild West of High Availability in Open edX

OpenCraft_

Brandon DeRosier
brandon@opencraft.com

Feanil Patel
feanil@edx.org

Agenda

Choosing Your Stack

Examples

Managing Infrastructure

Fork Management

Choosing your stack

Outline

- Infrastructure Choices
  - On Prem
  - AWS
  - OpenStack
  - Other Cloud Providers

On Premise

- Most Control
- Most expertise required
    - You'll be setting up a lot of it yourself
- You might need this based on your organization or local law requirements

Amazon Web Services

- Most managed services Available
- Used by edX Inc.
    - Our tools can become your tools
    - Con: Our tools are not always built for everyone

OpenStack

- Pro: Used by OpenCraft
    - OpenCraft adding support into existing edX tools where possible
- Con: no two openstack clouds are the same

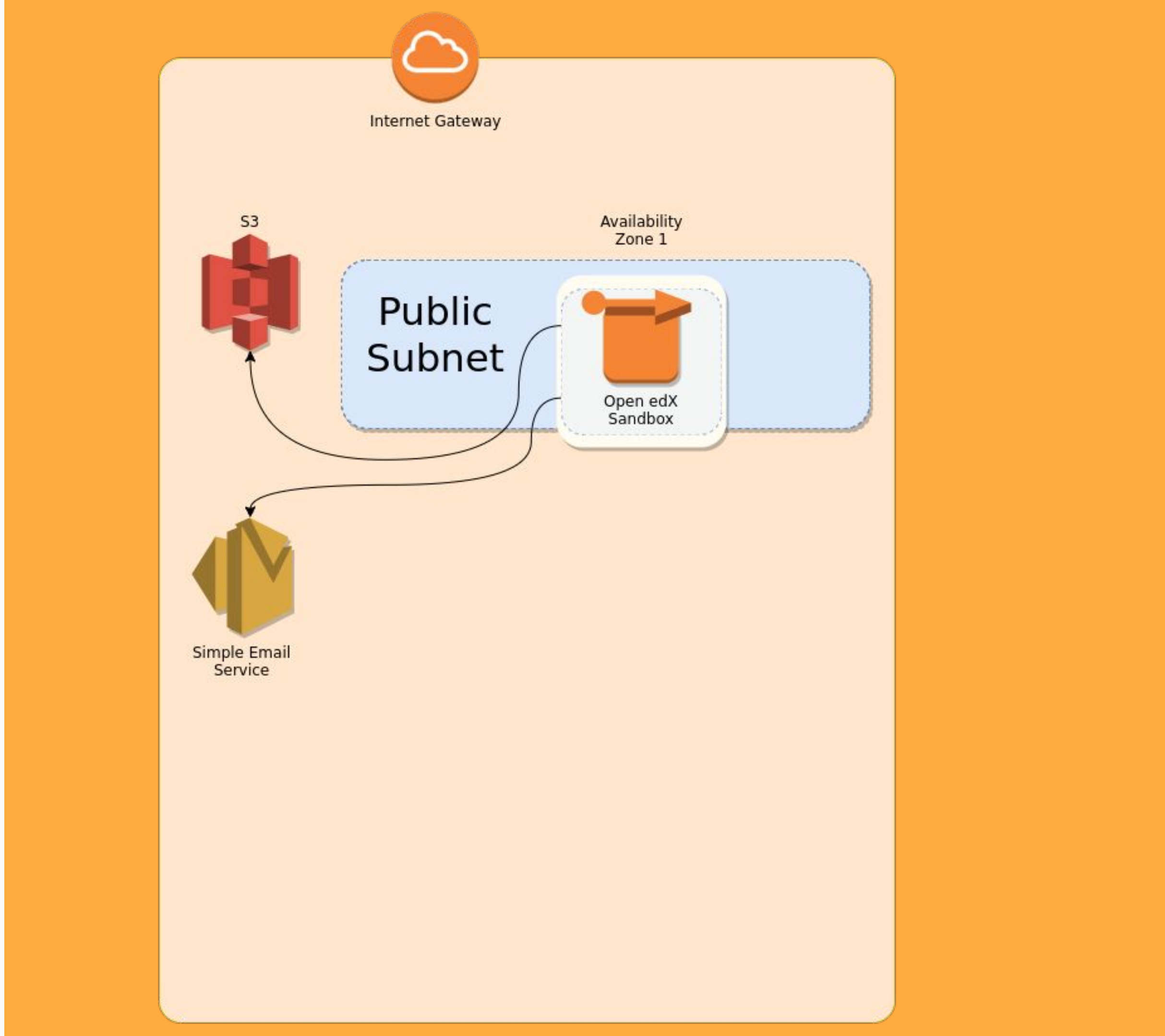
Other Infra Providers

- Usually provide the basic abstractions you need
    - Compute
    - Load Balancers
    - Storage
    - RDBMS as a Services
- Biggest differentiators are usually price, reliability, and tooling availability

Example:
The bare minimum

Sandbox

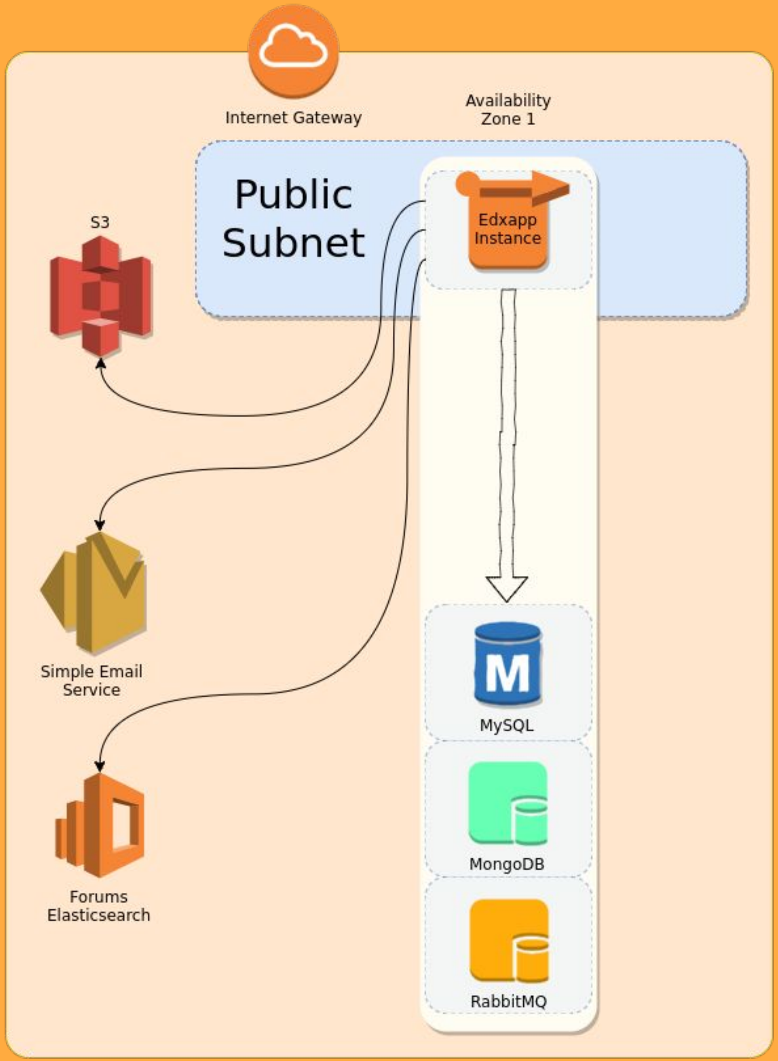
- ❖ Not Stateless
- ❖ Not Scalable
- ❖ Not Highly Available



Sandbox?
… sort of

- ~~Not Stateless~~
- Not Scalable… kind of
- Not Highly Available

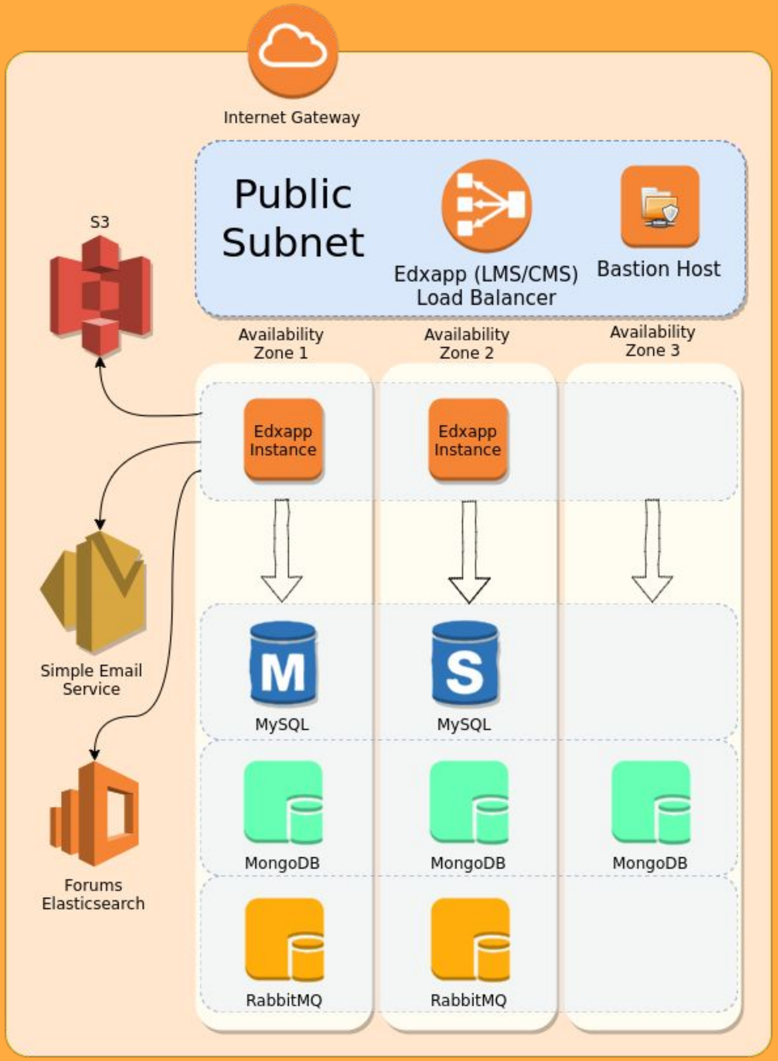
- Can fully monitor services separately
- But, more **single points of failure**



Definitely not a sandbox!

- ~~Not Stateless~~
- ~~Not Scalable~~
- ~~Not Highly Available~~

- Can fully monitor services separately
- ~~But, more single points of failure~~
- No **single points of failure**

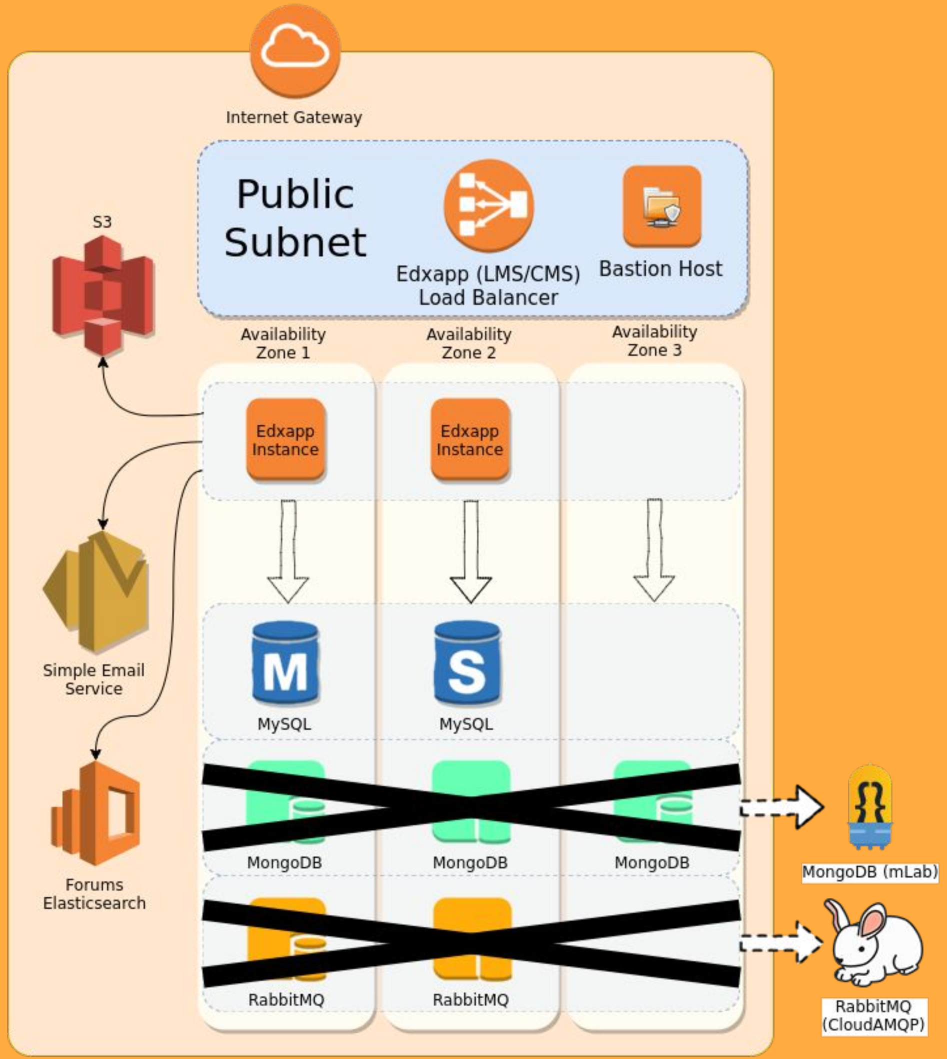


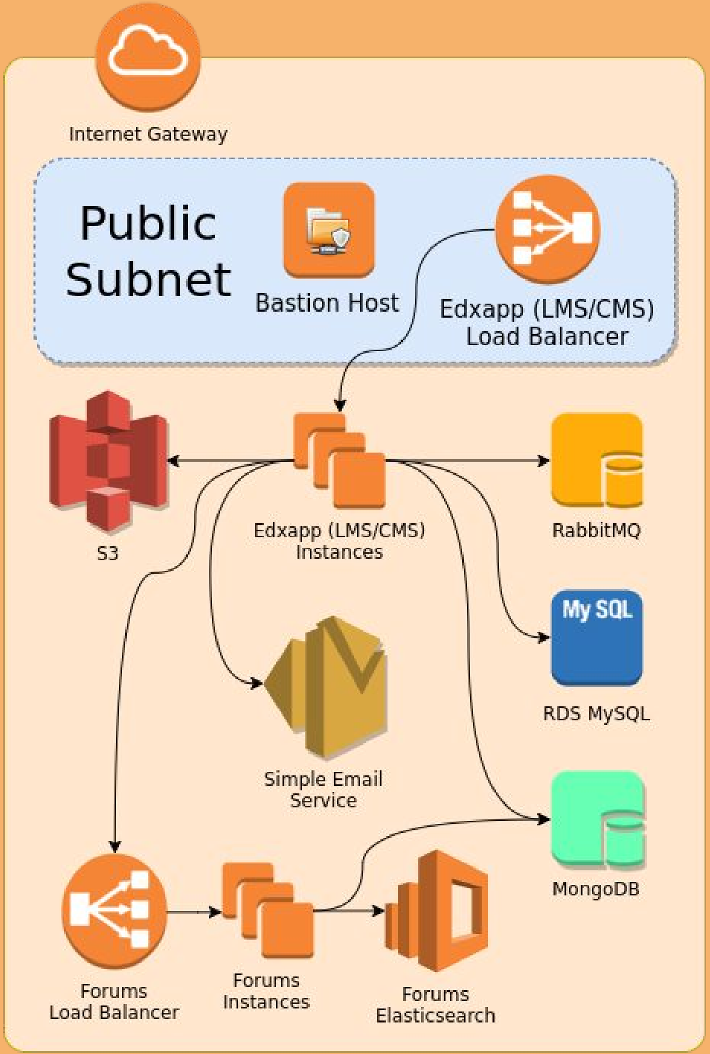
If desired, use managed databases

- ➔ **Less infrastructure** to take care of
- ➔ **Easy to scale**

- ➔ **More expensive** than the bare infrastructure

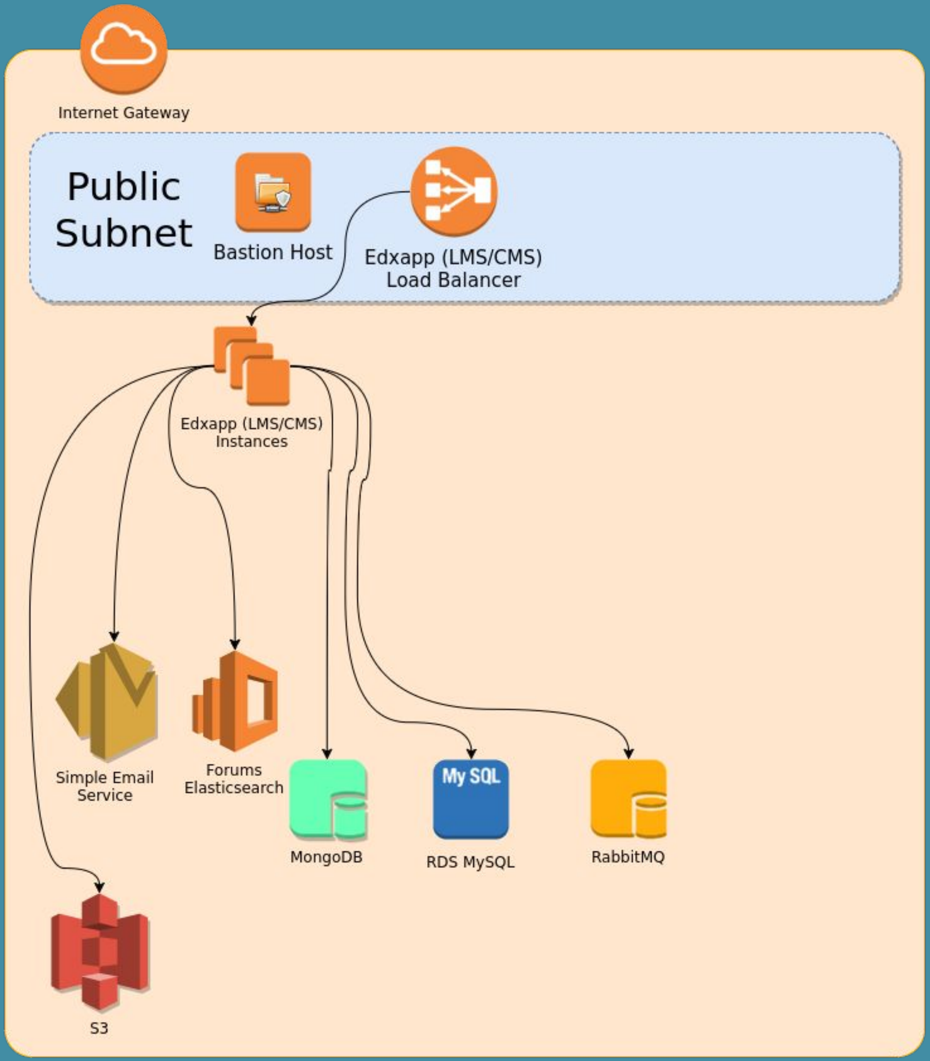
- ➔ Check for education/user privacy **law compliance**



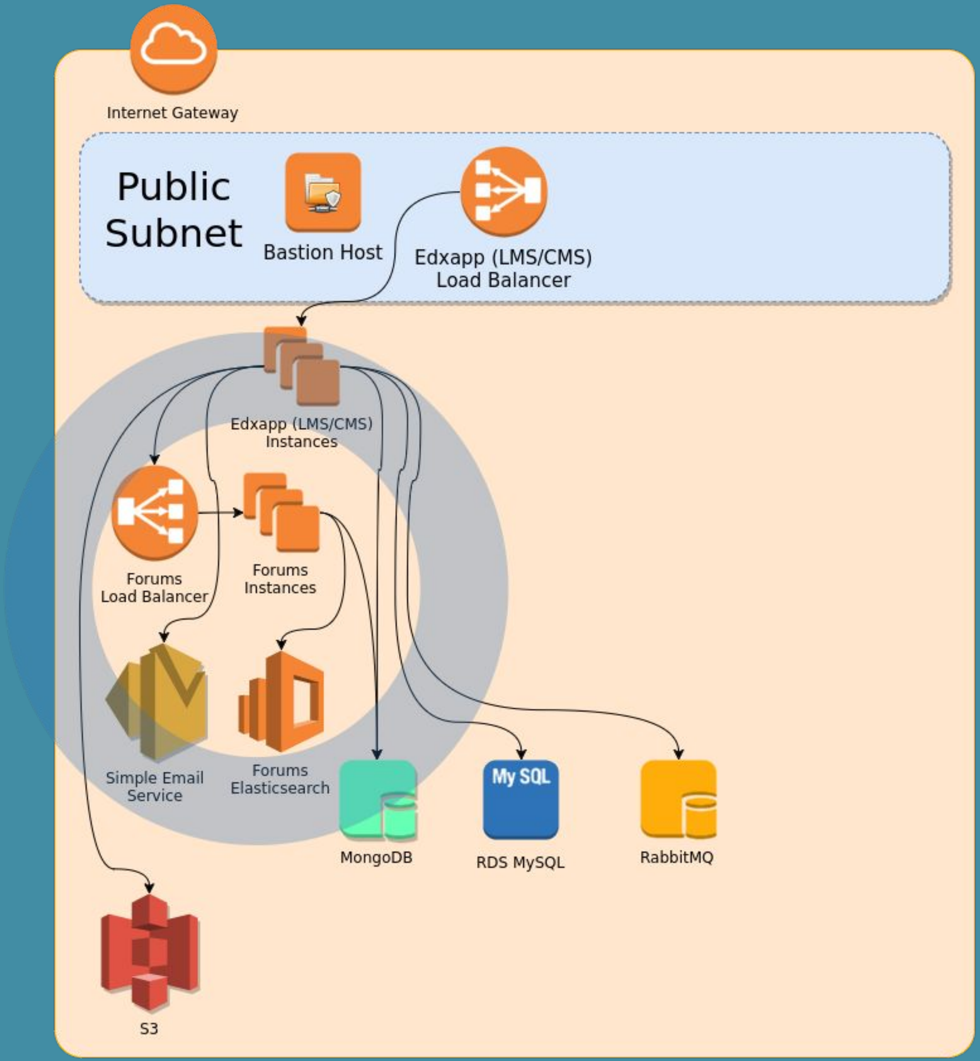

Internet Gateway
Public Subnet
Bastion Host
Edxapp (LMS/CMS) Load Balancer
S3
Edxapp (LMS/CMS) Instances
RabbitMQ
My SQL
RDS MySQL
Simple Email Service
MongoDB
Forums Load Balancer
Forums Instances
Forums Elasticsearch

Example:
A large stack

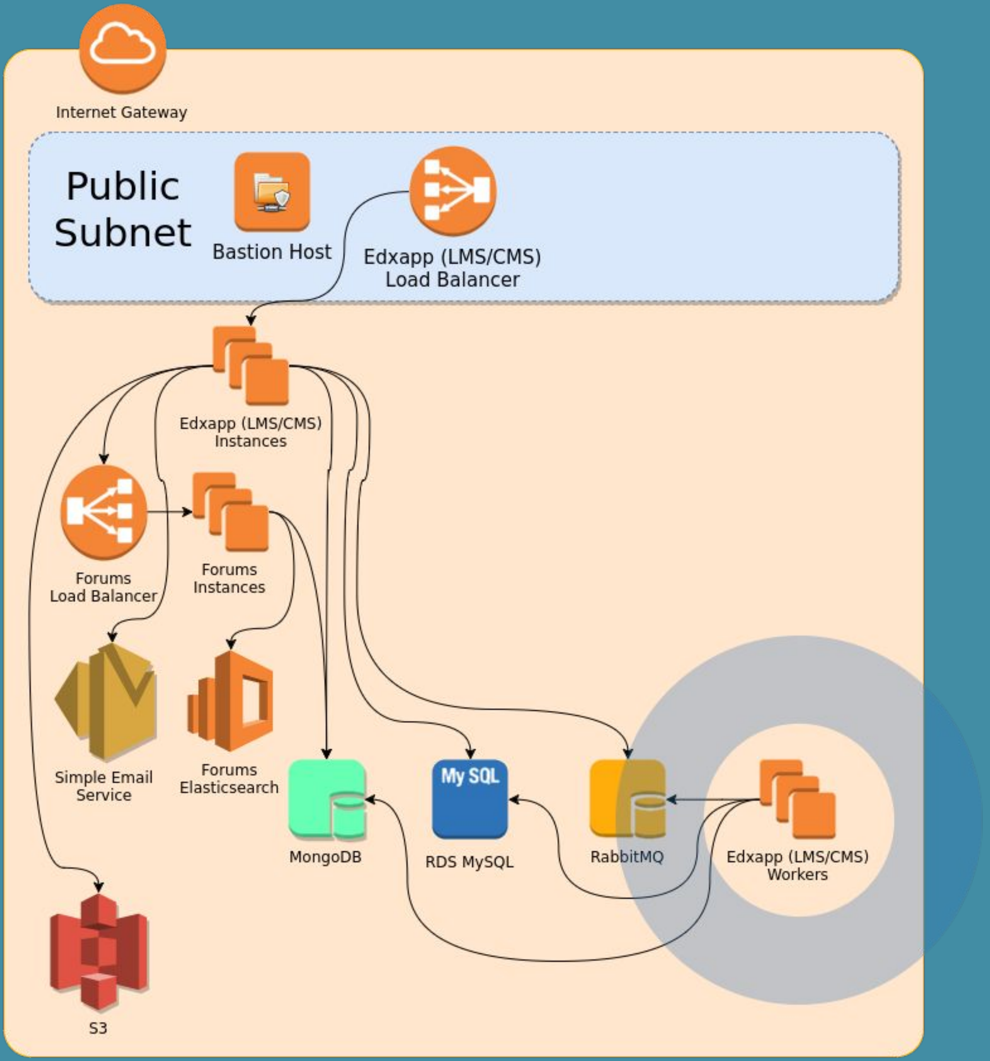
Where we left off:



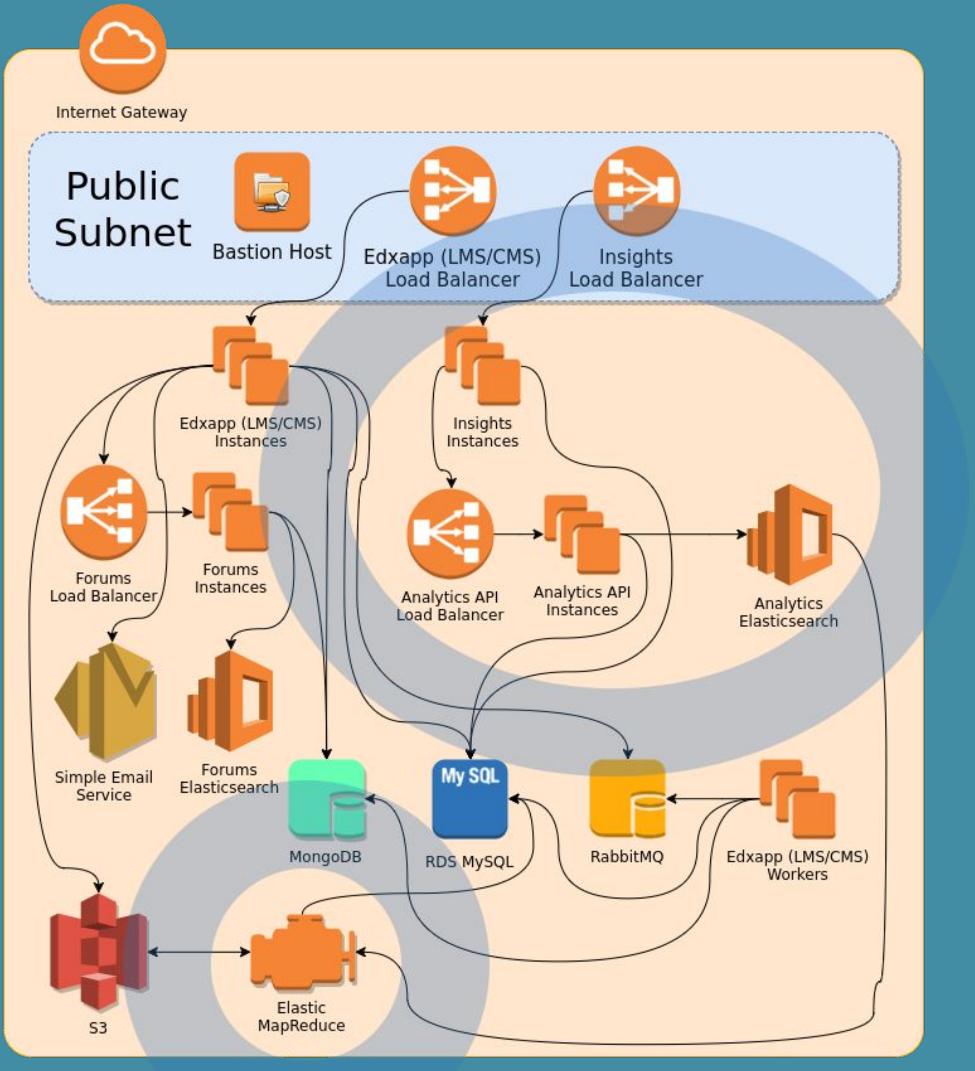
Separate forums...



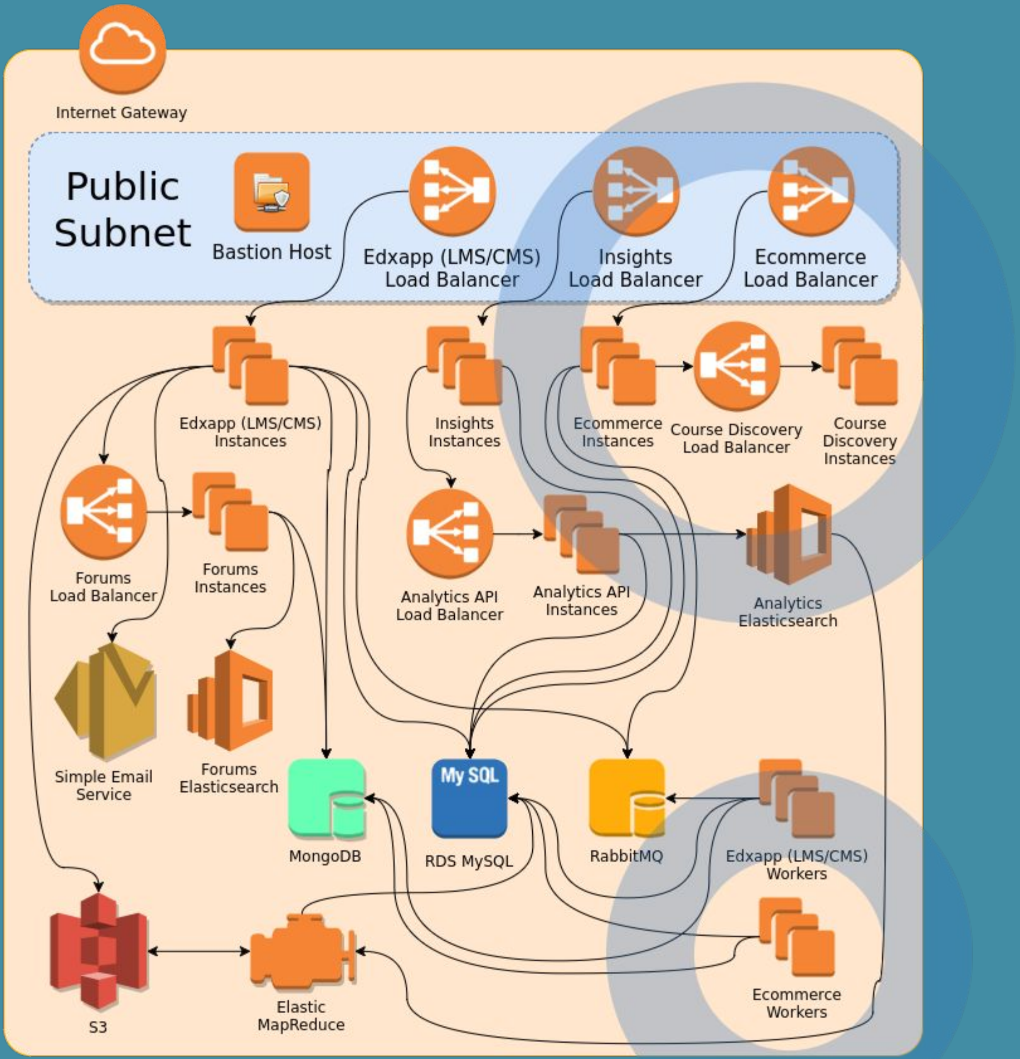
Separate workers...



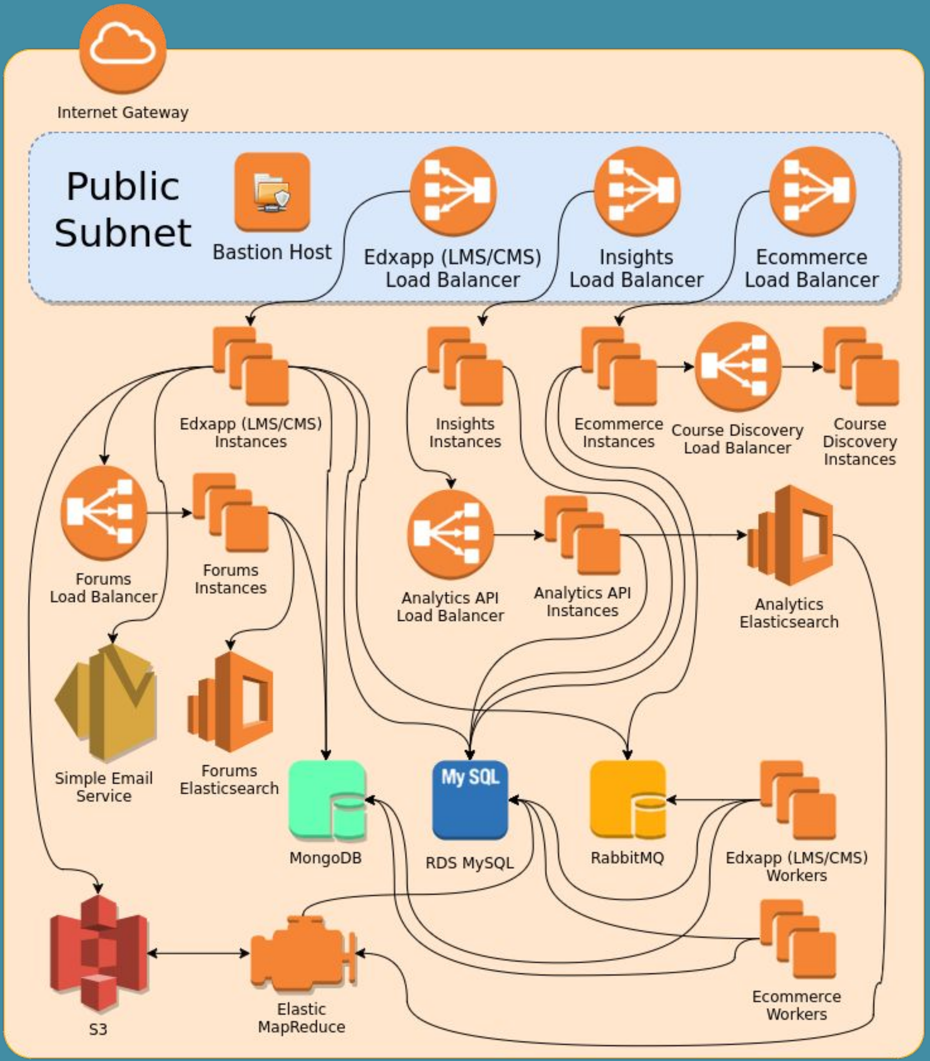
Analytics!


Internet Gateway
Public Subnet
Bastion Host
Edxapp (LMS/CMS) Load Balancer
Insights Load Balancer
Edxapp (LMS/CMS) Instances
Insights Instances
Forums Load Balancer
Forums Instances
Analytics API Load Balancer
Analytics API Instances
Analytics Elasticsearch
Simple Email Service
Forums Elasticsearch
MongoDB
My SQL
RDS MySQL
RabbitMQ
Edxapp (LMS/CMS) Workers
S3
Elastic MapReduce

€commer¢e $$$

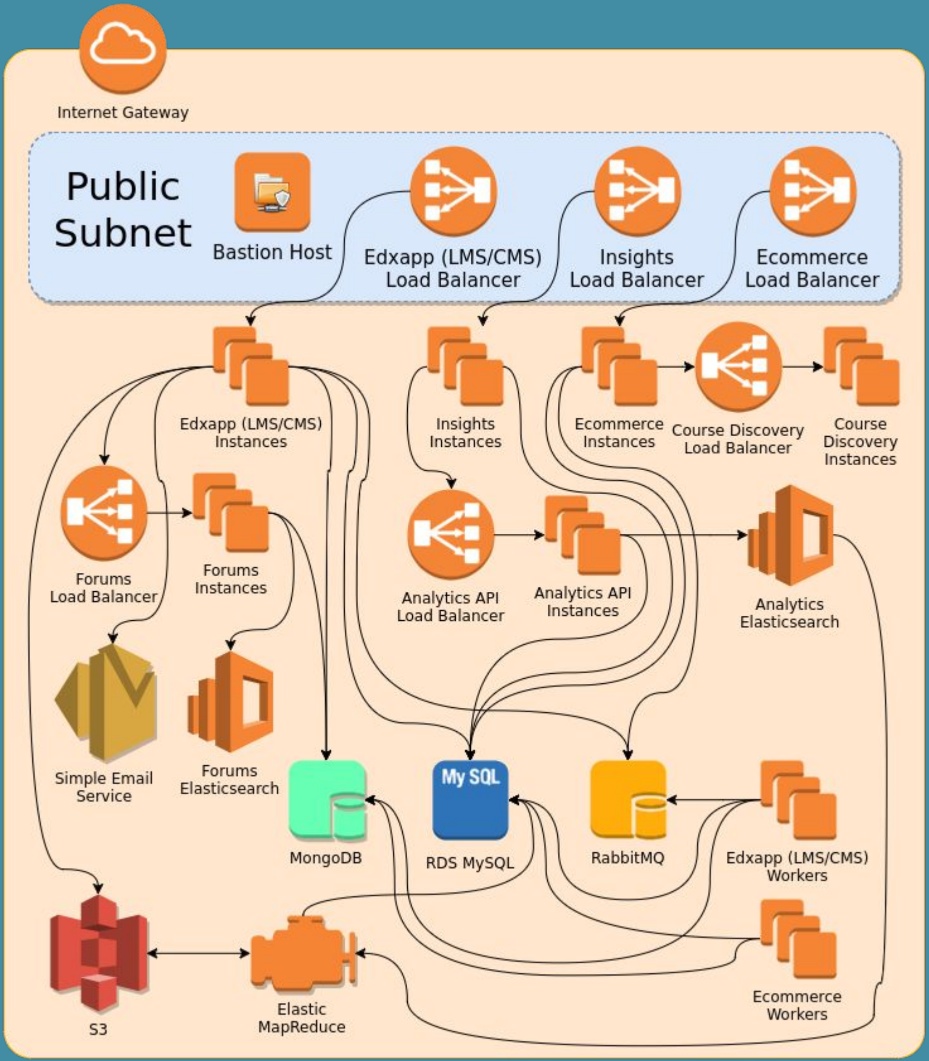

Internet Gateway
Public Subnet
Bastion Host
Edxapp (LMS/CMS) Load Balancer
Insights Load Balancer
Ecommerce Load Balancer
Edxapp (LMS/CMS) Instances
Insights Instances
Ecommerce Instances
Course Discovery Load Balancer
Course Discovery Instances
Forums Load Balancer
Forums Instances
Analytics API Load Balancer
Analytics API Instances
Analytics Elasticsearch
Simple Email Service
Forums Elasticsearch
MongoDB
My SQL
RDS MySQL
RabbitMQ
Edxapp (LMS/CMS) Workers
Ecommerce Workers
S3
Elastic MapReduce

Well...



How do we make this smaller?

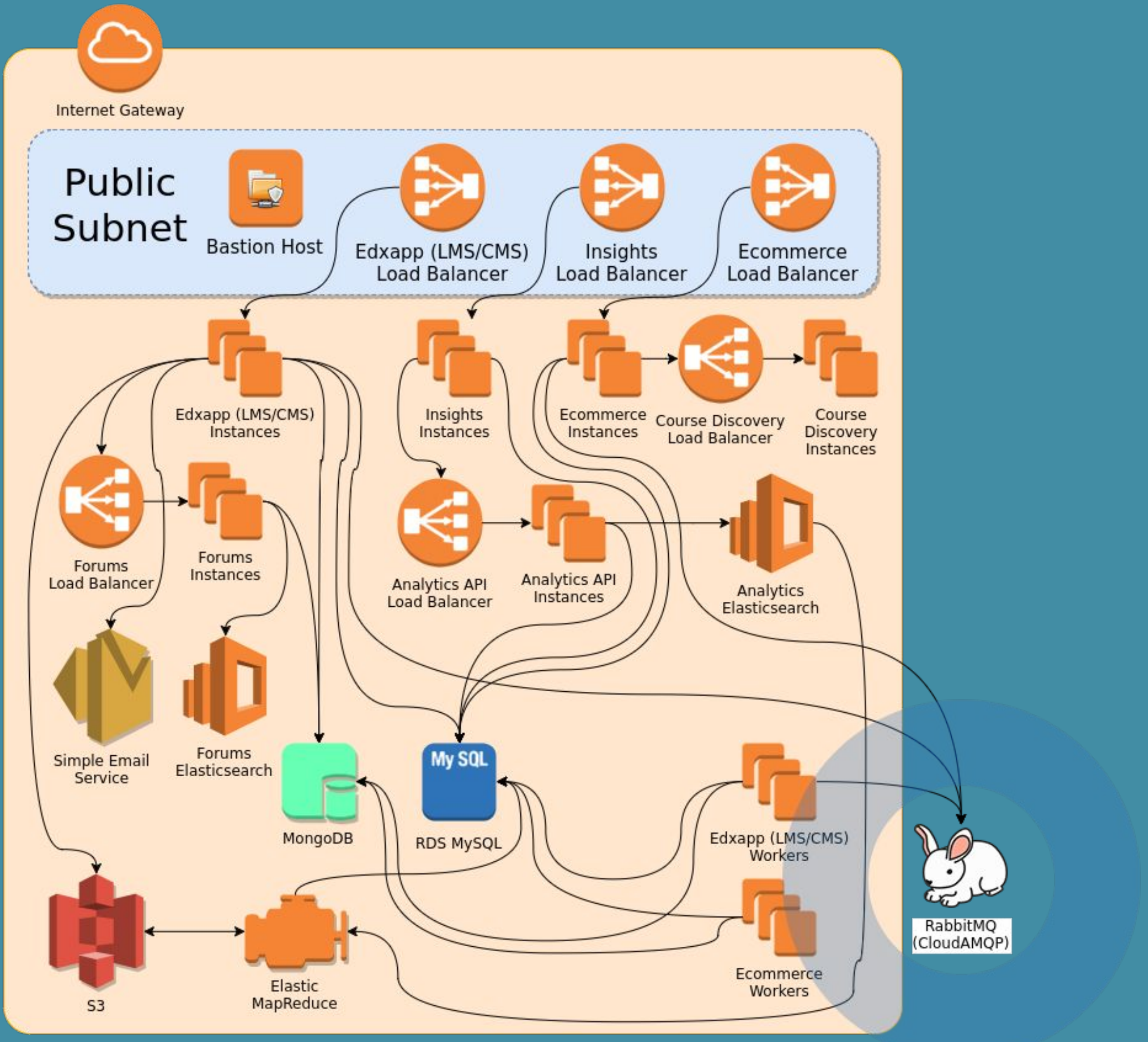
Well...



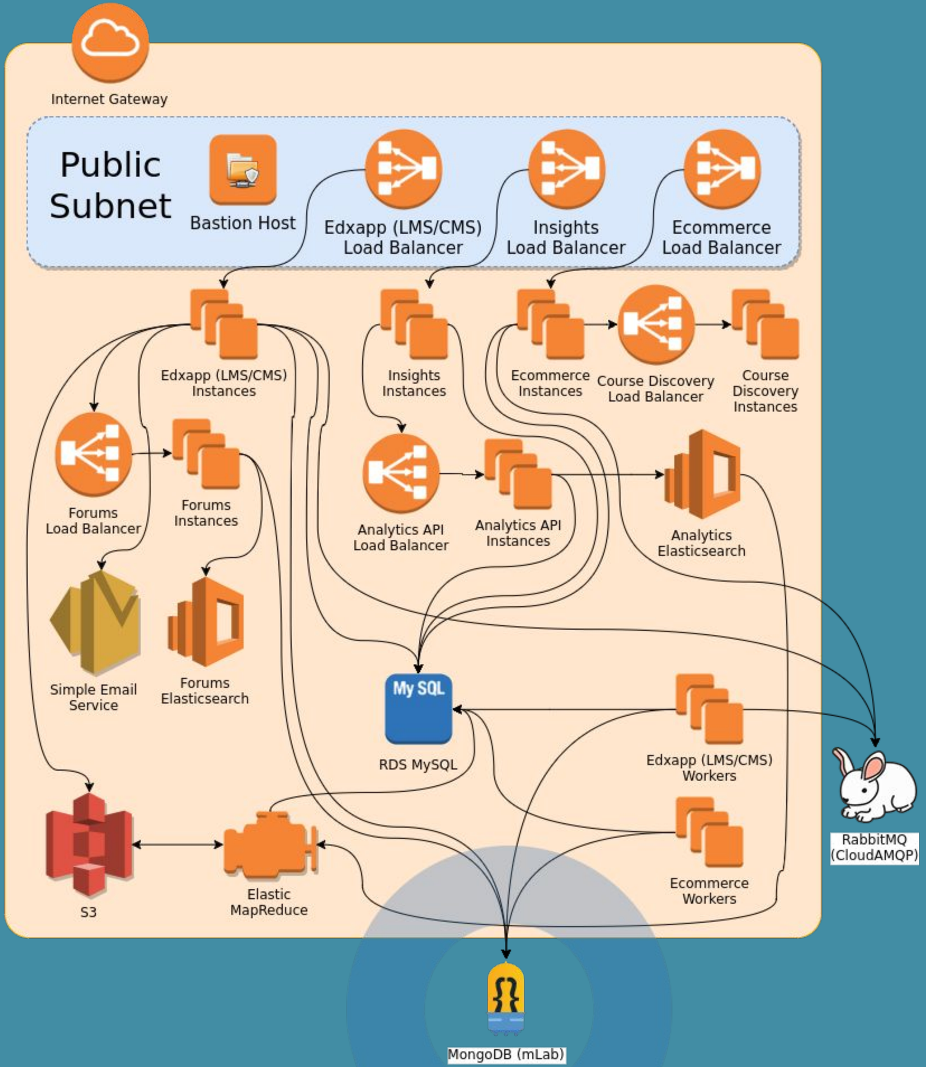
How do we make this smaller?

Maybe try the managed services again…?

External RabbitMQ...

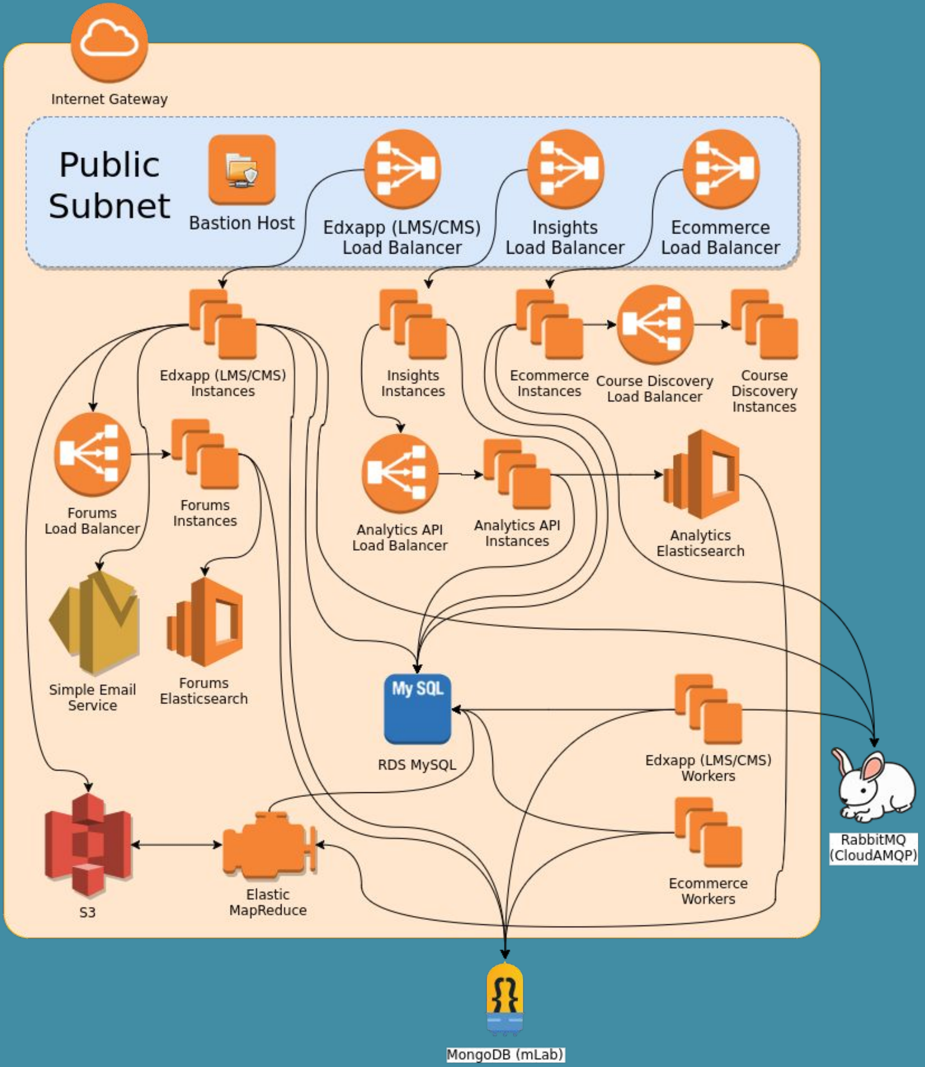


External MongoDB...

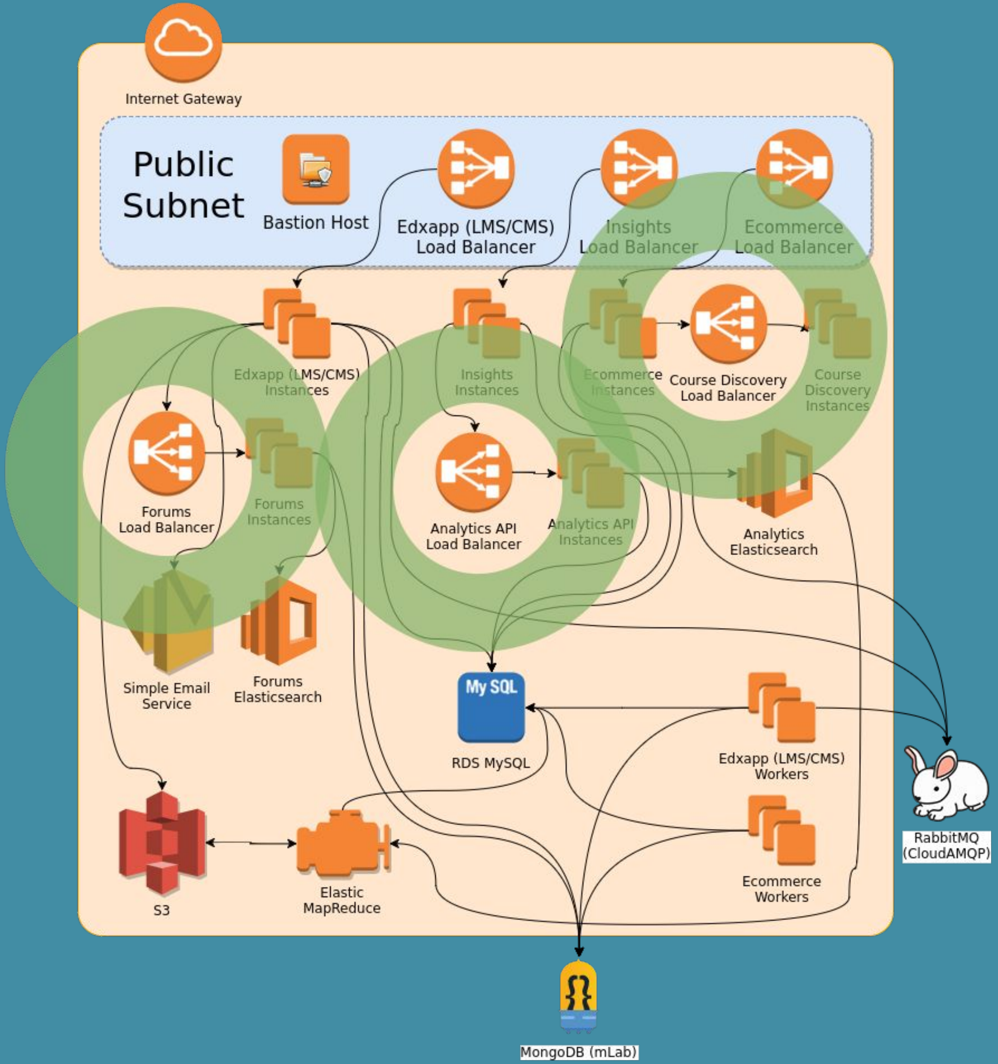


Hmm…

That didn't do very much….



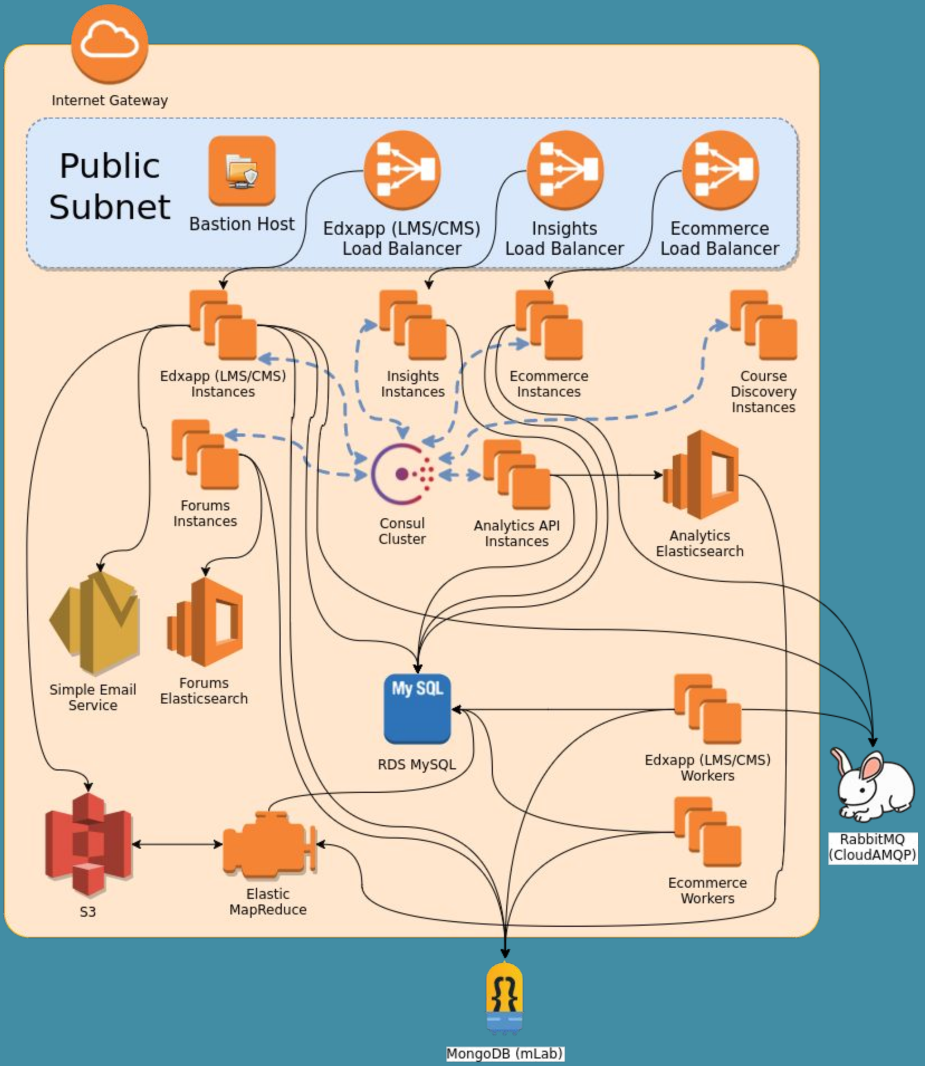
Hmm...



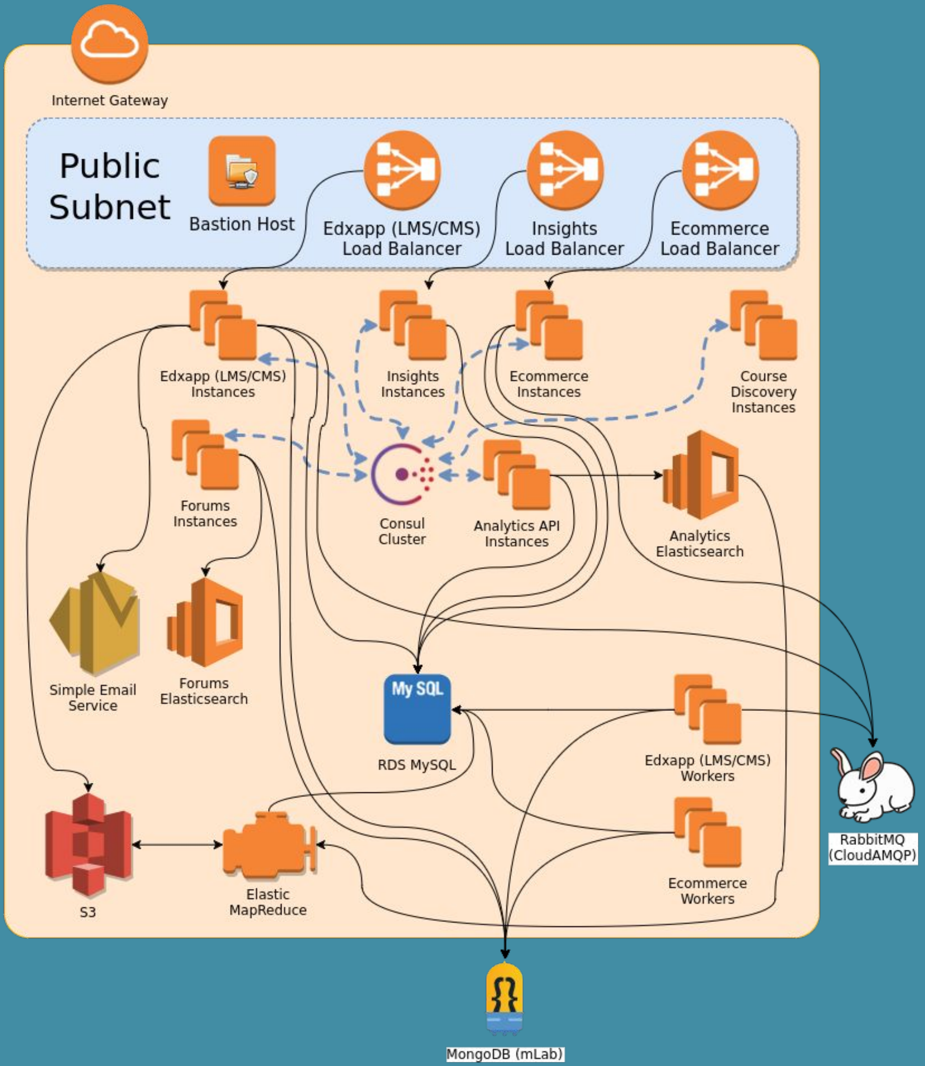
That didn't do very much….

But maybe we can replace all these load balancers…?

Service Discovery!



Service Discovery!



Very little benefit for a small deployment, though...

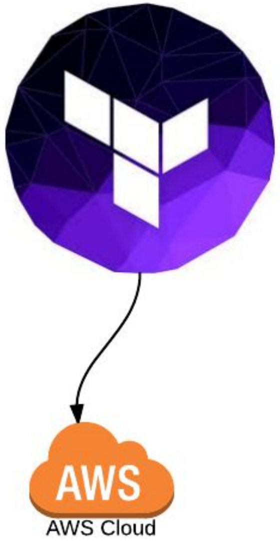
Managing Infrastructure

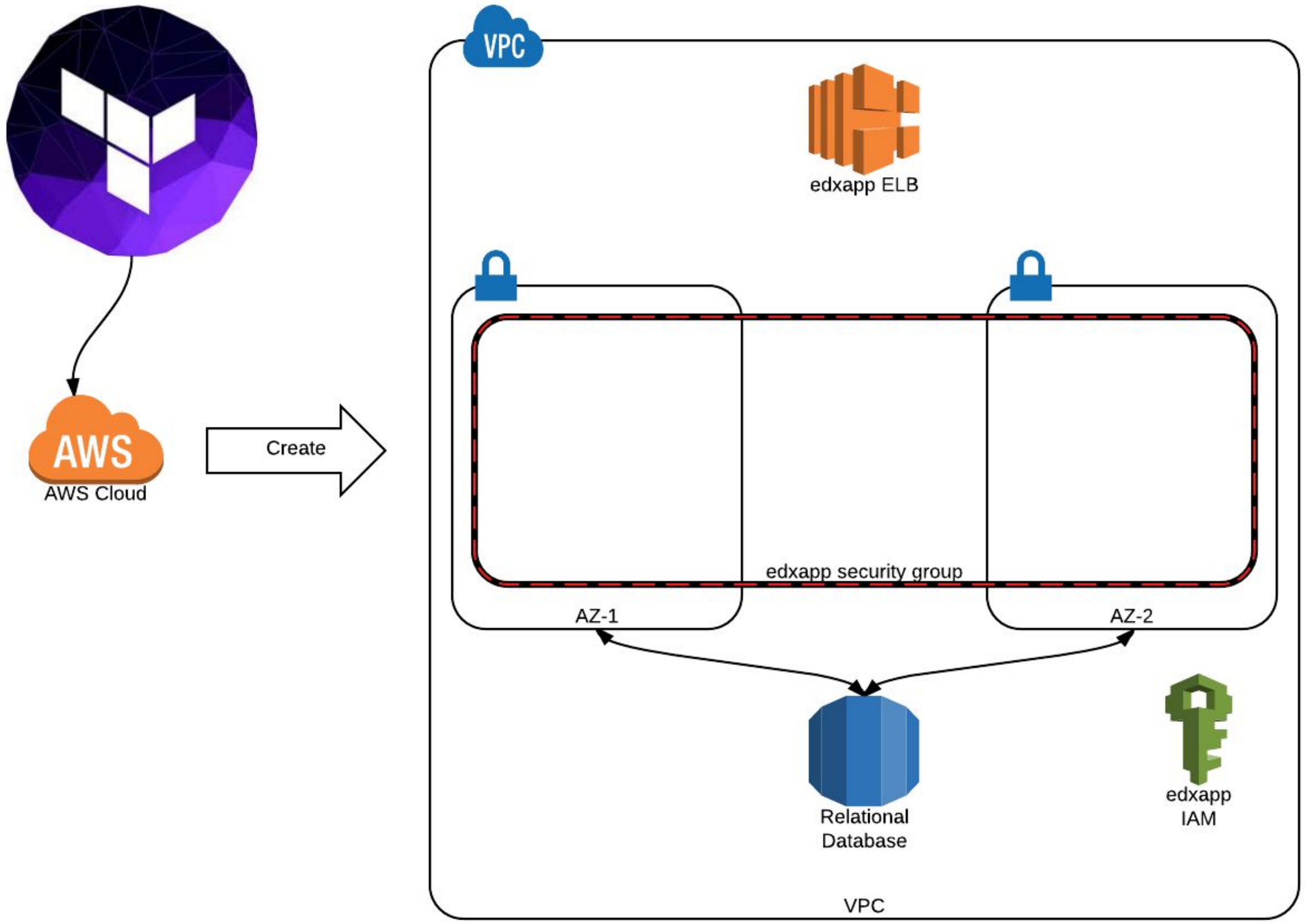
Tools

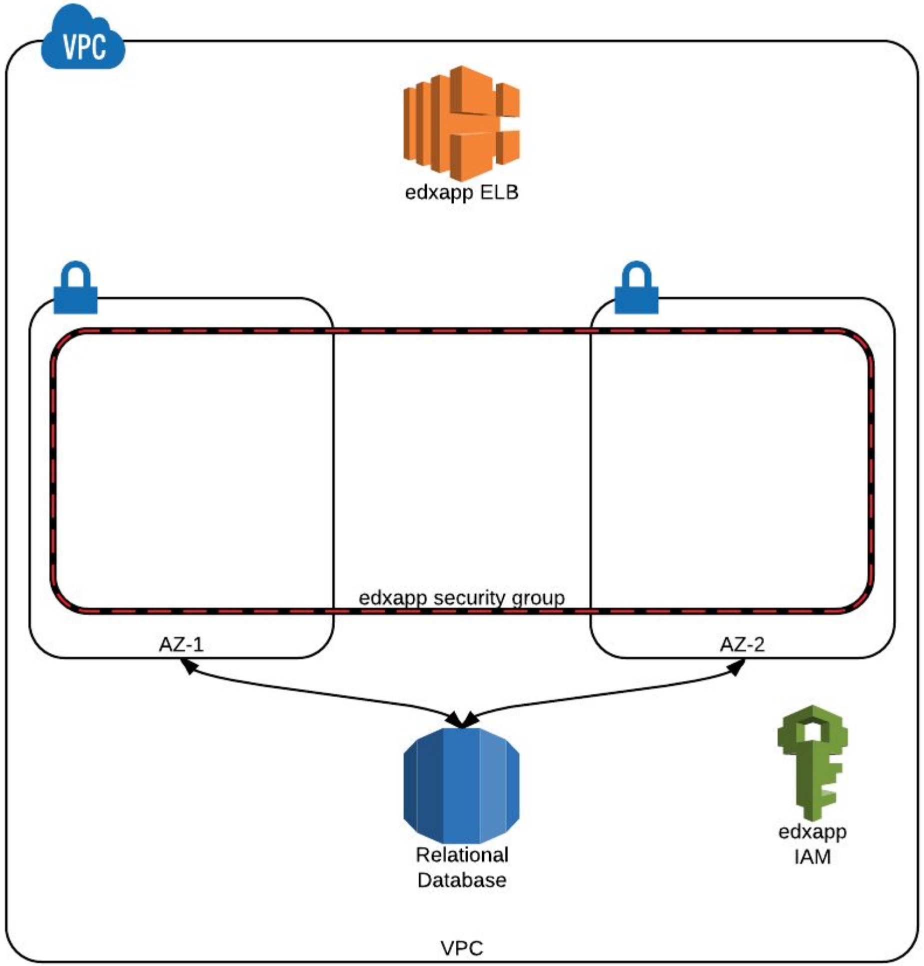
- For things that change slowly
  - Ansible Infrastructure Modules
  - Cloudformation
  - Terraform

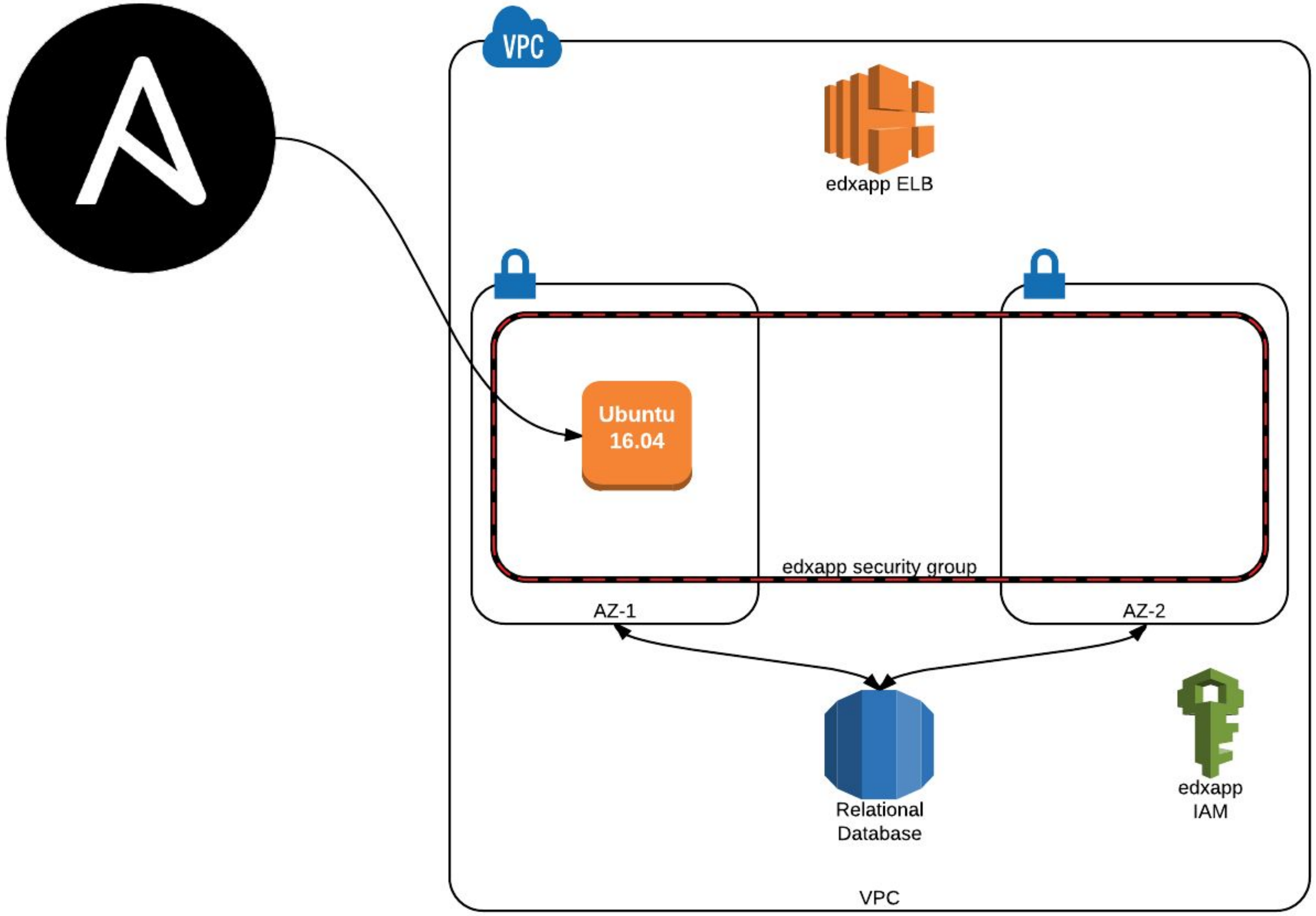
Tools

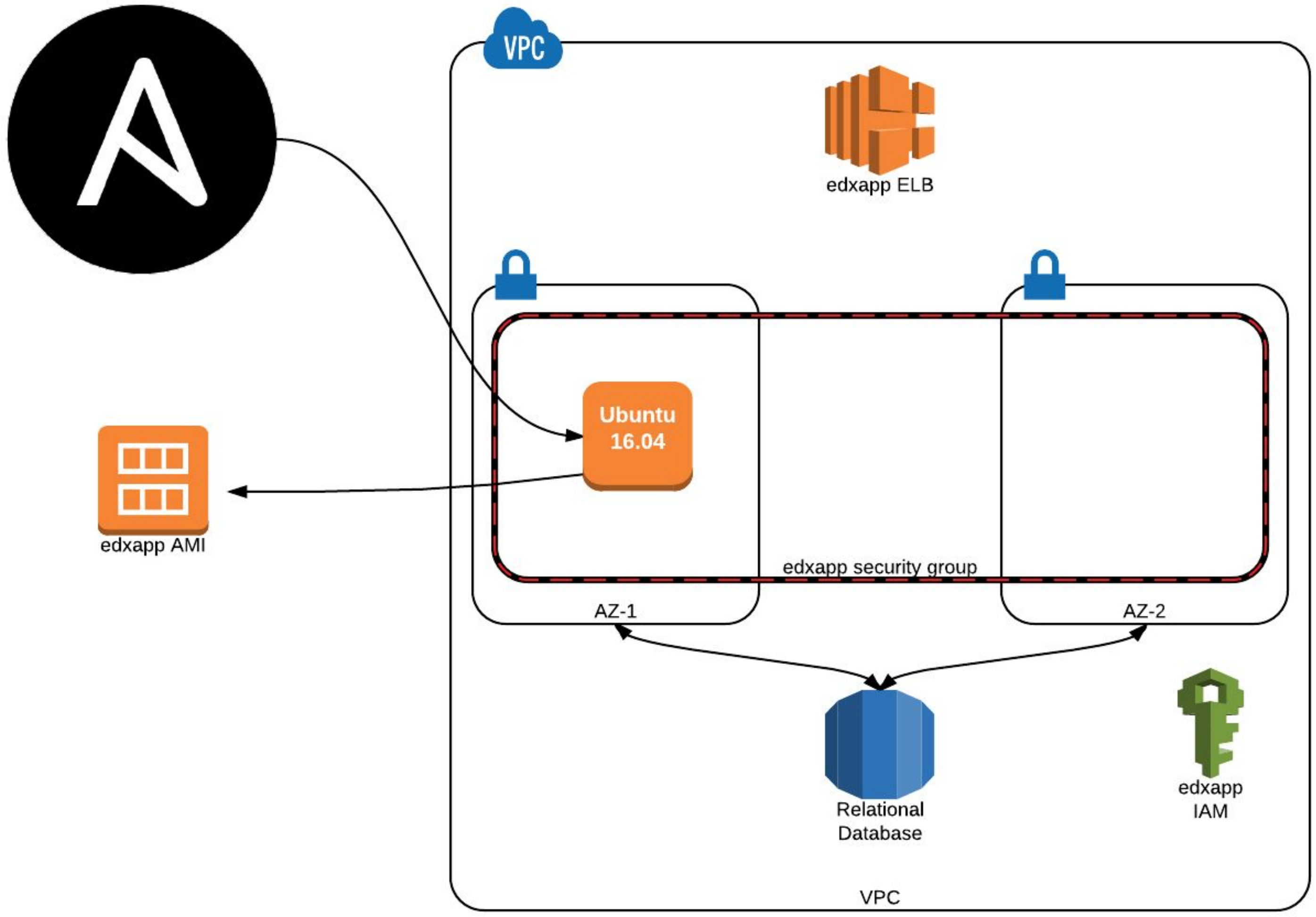
- For things that change more often
    - Configuration Repo
    - Asgard
    - GoCD

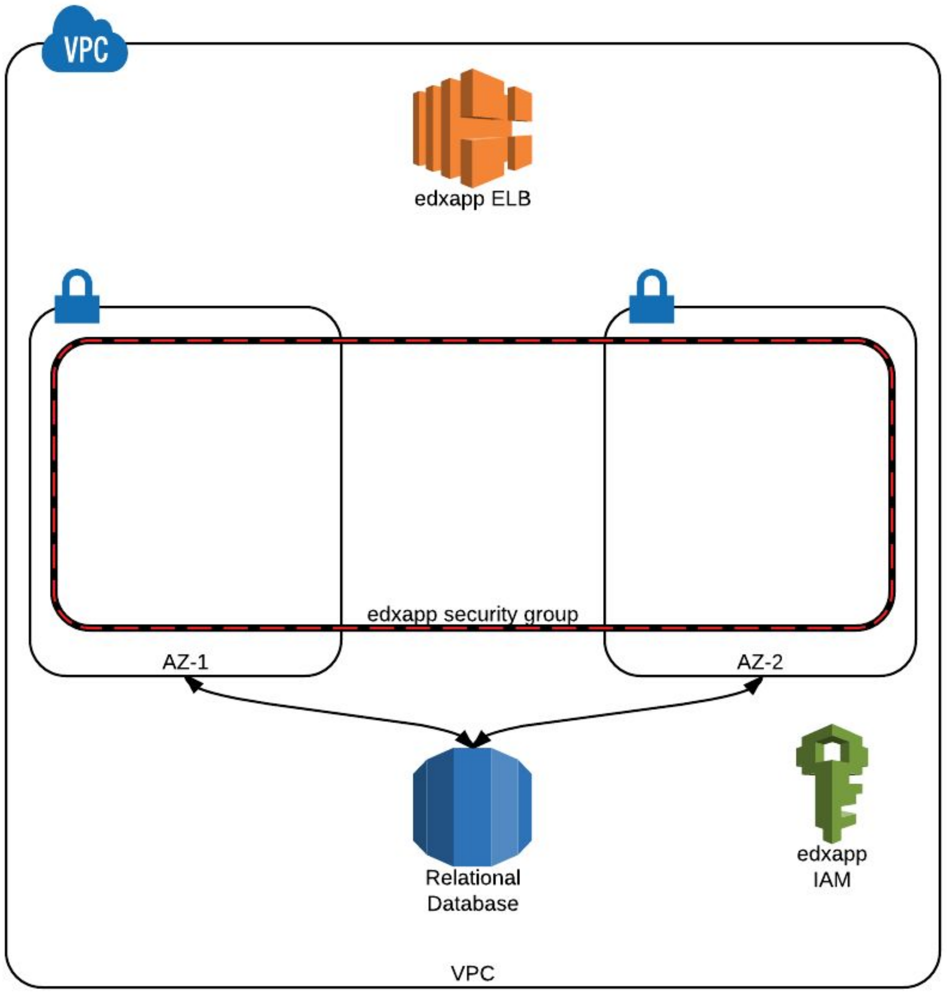

AWS
AWS Cloud

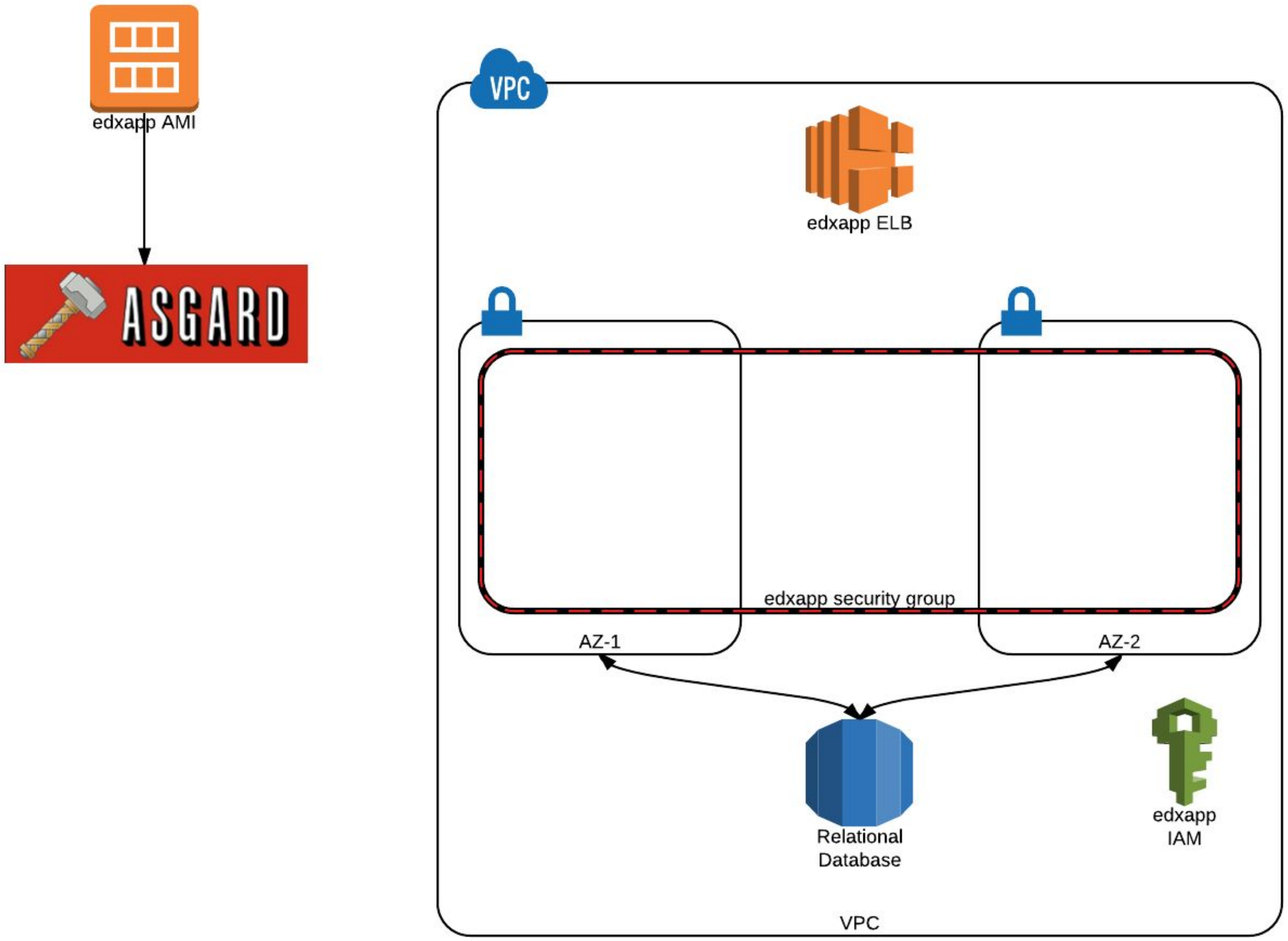

AWS
AWS Cloud
Create
VPC
edxapp ELB
edxapp security group
AZ-1
AZ-2
Relational
Database
edxapp
IAM
VPC

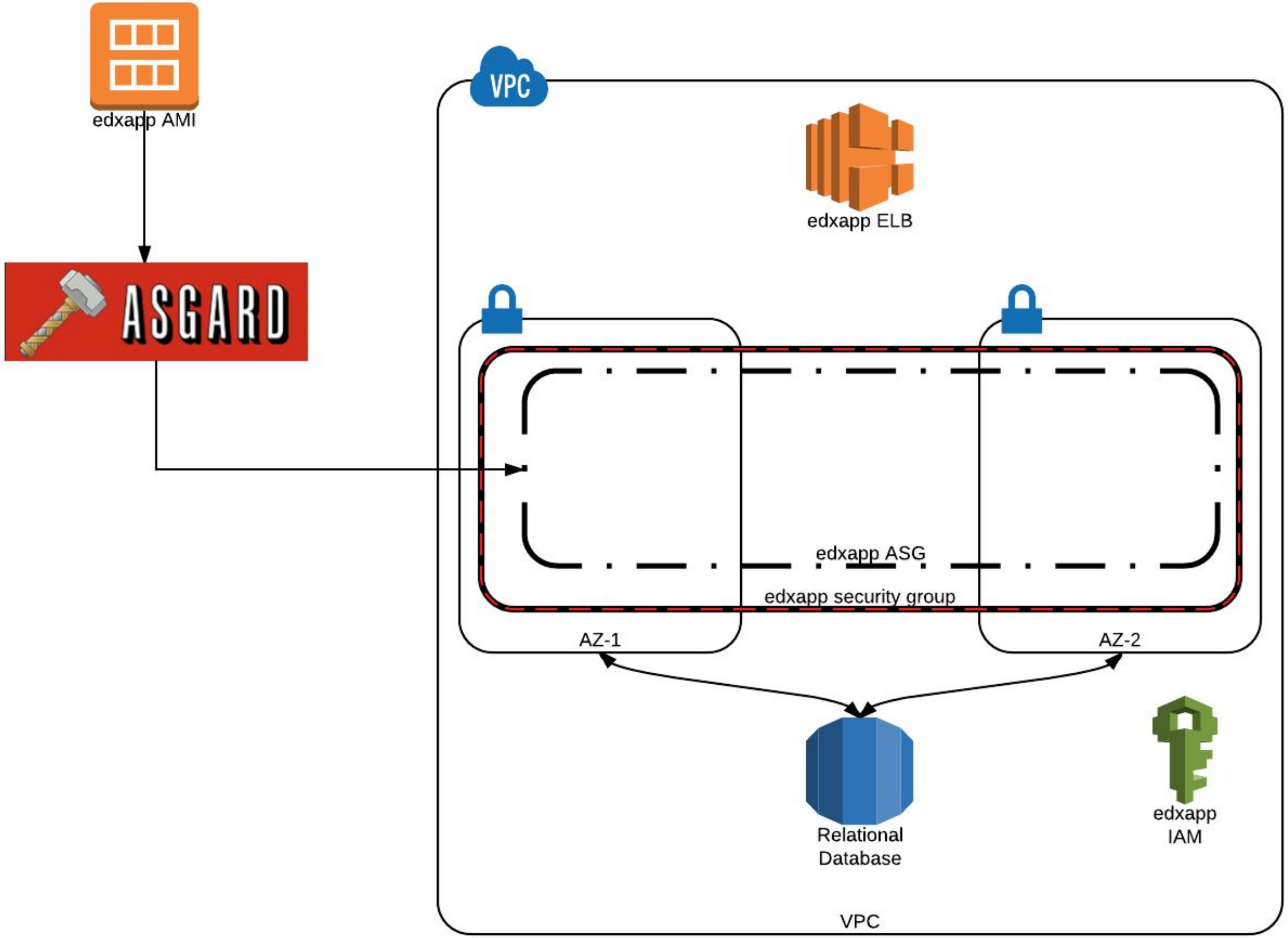

VPC
edxapp ELB
edxapp security group
AZ-1
AZ-2
Relational Database
edxapp IAM
VPC

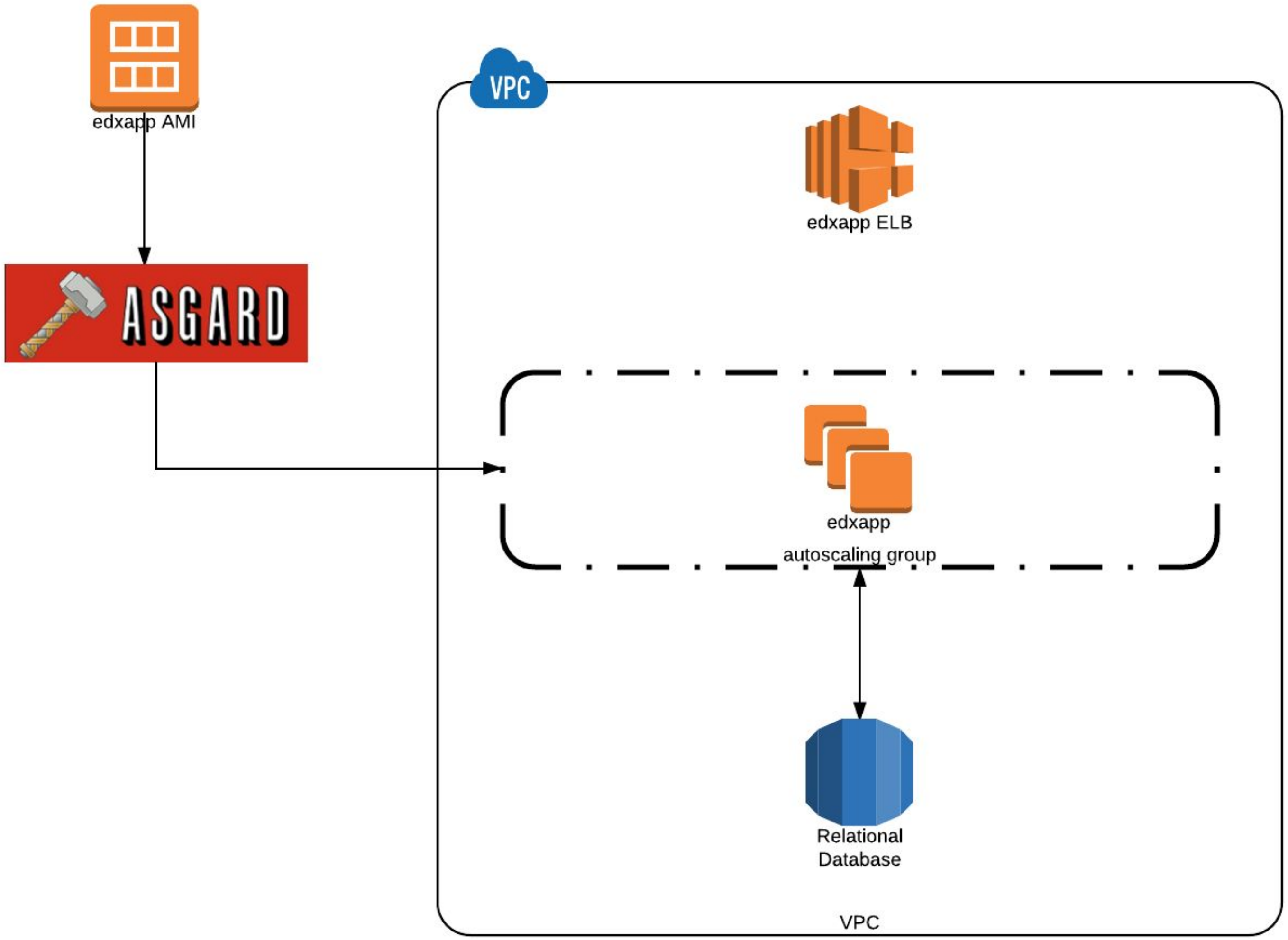

VPC
edxapp ELB
Ubuntu 16.04
edxapp security group
AZ-1
AZ-2
Relational Database
edxapp IAM
VPC

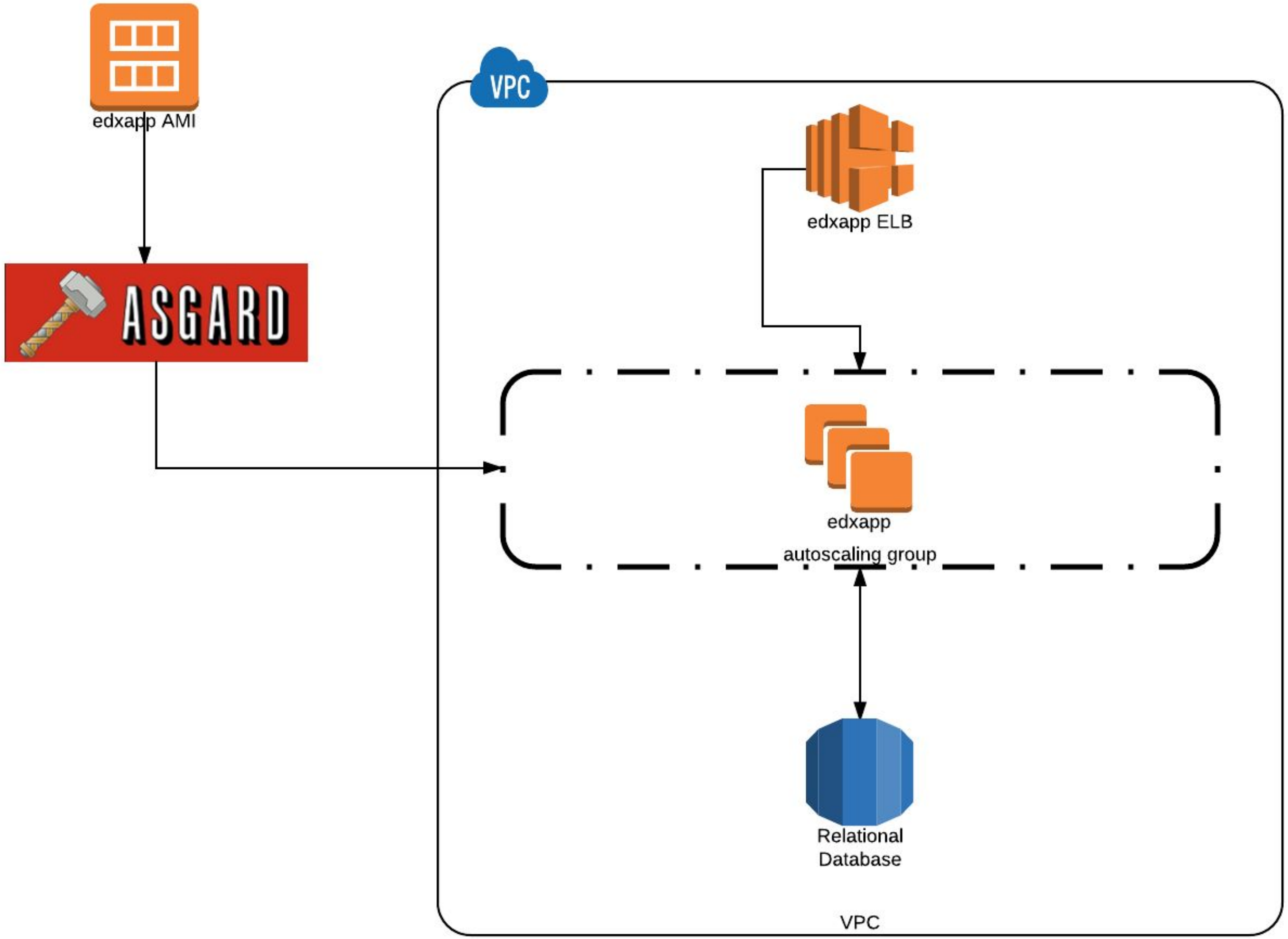

VPC
edxapp ELB
Ubuntu 16.04
edxapp AMI
edxapp security group
AZ-1
AZ-2
Relational Database
edxapp IAM
VPC

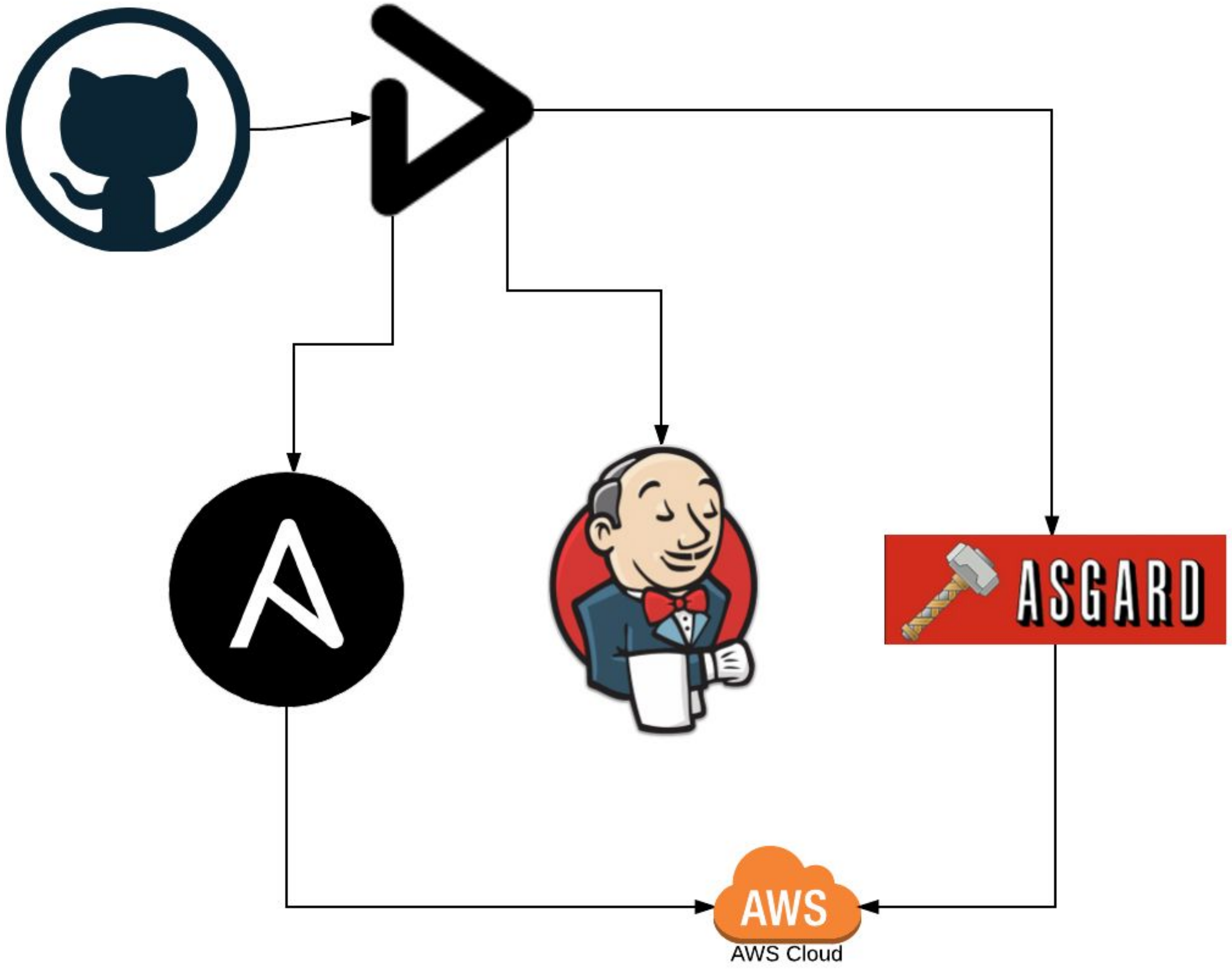

VPC
edxapp ELB
edxapp security group
AZ-1
AZ-2
Relational
Database
edxapp
IAM
VPC

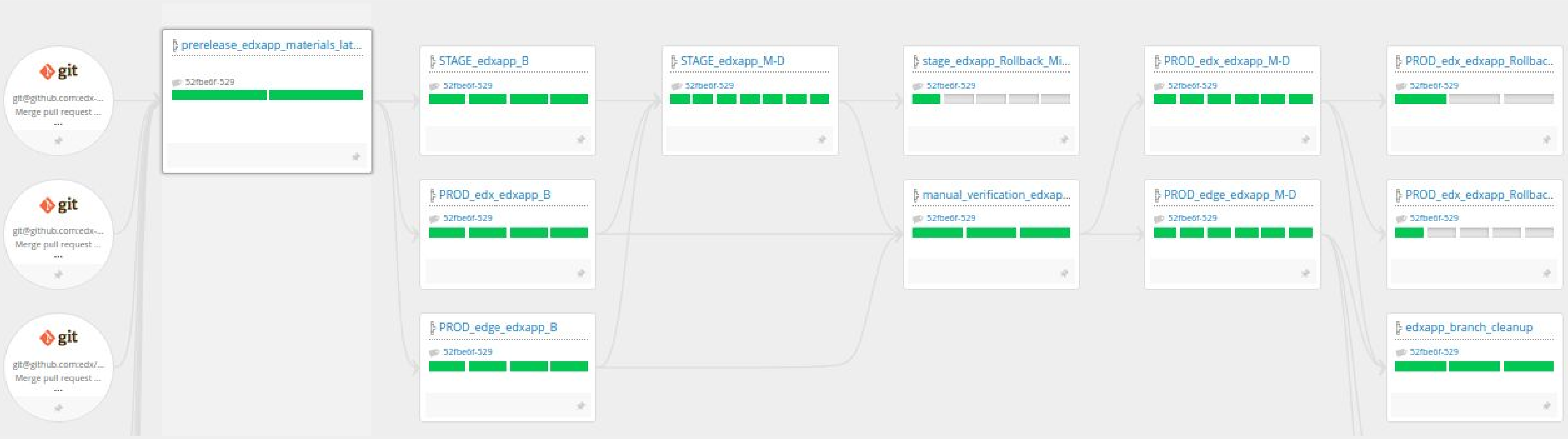

edxapp AMI
ASGARD
VPC
edxapp ELB
edxapp security group
AZ-1
AZ-2
Relational Database
edxapp IAM
VPC


edxapp AMI
ASGARD
VPC
edxapp ELB
edxapp ASG
edxapp security group
AZ-1
AZ-2
Relational
Database
edxapp
IAM
VPC


edxapp AMI
ASGARD
VPC
edxapp ELB
edxapp
autoscaling group
Relational Database
VPC


edxapp AMI
ASGARD
VPC
edxapp ELB
edxapp
autoscaling group
Relational Database
VPC


ASGARD
AWS
AWS Cloud


git
git@github.com:edx-...
Merge pull request ...
git
git@github.com:edx-...
Merge pull request ...
git
git@github.com:edx/...
Merge pull request ...
prerelease_edxapp_materials_lat...
52fbe6f-529
STAGE_edxapp_B
52fbe6f-529
PROD_edx_edxapp_B
52fbe6f-529
PROD_edge_edxapp_B
52fbe6f-529
STAGE_edxapp_M-D
52fbe6f-529
stage_edxapp_Rollback_Mi...
52fbe6f-529
manual_verification_edxap...
52fbe6f-529
PROD_edx_edxapp_M-D
52fbe6f-529
PROD_edge_edxapp_M-D
52fbe6f-529
PROD_edx_edxapp_Rollbac...
52fbe6f-529
PROD_edx_edxapp_Rollbac...
52fbe6f-529
edxapp_branch_cleanup
52fbe6f-529

Fork management tips

~~Fork management tips~~

The Four Horsemen Of Technical Debt

1. KEEP YOUR DIFFS SMALL

- Tiny stuff adds up quickly
- Investing upfront pays off here - deal with it *before* it's a problem
- Named release rebasing will start costing you a **fortune** otherwise
- **Huge diffs are simply a nightmare**

2. Upstream **Everything**

- Distribute the burden of Maintenance
  - Someone broke your feature’s tests?
  - They have to fix it before merging
- Build better software
  - Forces your architecture and code to be high quality
- Be a good citizen

3. Don’t edit **Ansible output**

- lms.env.json
- lms.auth.json
- cms.env.json
- cms.auth.json

- Do it like edX: **Use configuration management**
- These files may not always be the config destination
- edX configuration keeps up as the schema changes
- Things are actually much simpler this way

3. Don't edit **Ansible output**

- lms.env.json
- lms.auth.json
- cms.env.json
- cms.auth.json

- Do it like edX: **Use configuration management**
- These files may not always be the config destination
- edX configuration keeps up as the schema changes
- Things are actually much simpler this way

3.5. Also, don't edit settings code directly

- lms/envs/common.py
- cms/envs/aws.py
- ...etc.

4. Avoid **template overrides** in themes

- Sometimes you have to do it
- ...but try to make style modifications as much as possible
- **Context changes** in views will have you fixing your templates every release

Questions?

Brandon DeRosier
brandon@opencraft.com

Feanil Patel
feanil@edx.org

Not part of presentation: Cloud Agnostic Stack Choices

- The OpenEdx platform uses many off the shelf tools.
- You may want to use SaaS providers
  - Rabbit as a service
  - Mongo as a service
  - Elasticsearch as a service

Infrastructure as Code Options

The next 3 pages talk about different tools for managing infrastructure as code.

Ansible Modules

- Support
  - Number of Modules to manage ec2 resources is growing
  - Interfaces between modules not consistent
  - Modules not available for most other clouds
- Imperative nature
  - Great for orchestration
  - Bad for predicting what's going to happen
- If you're already using ansible, you don't have to learn a new tool

Cloudformation

- Declarative Syntax
  - Declare your resources in a file
  - Provide it to Cloudformation
  - Cloudformation ensures that all declared resources exist
- Con: Only works with AWS
- Con: Works best in one file
  - Can do multi files but is complicated
- Pro: Supports Planning for making changes
  - You update your template and it can show you what it will do to your resources

Terraform

- Similar in capability to current state of cloudformation
  - Declarative
  - Planning Capability
- Differences
  - Can easily span multiple files
  - Can work with multiple cloud providers
    - Provider plugin system is very flexible

GoCD: https://gocd.io

- Gomatic - DSL to make GoCD pipelines
  - https://github.com/edx/edx-gomatic/
  - https://github.com/edx/gomatic/tree/edx
- Tubular - Convenience scripts and utilities
  - Can be used independently of GoCD
  - https://github.com/edx/tubular

Configuration Repo

- Ansible playbooks for building App specific machines
- Optimized for separate machines per application
    - Though roles can be combined flexibly
- https://github.com/edx/configuration

Asgard

- Used to managed deployments at edX
- Allows for Imaged based blue/green deployments
- Con: No longer supported by Netflix
- https://github.com/edx/asgard