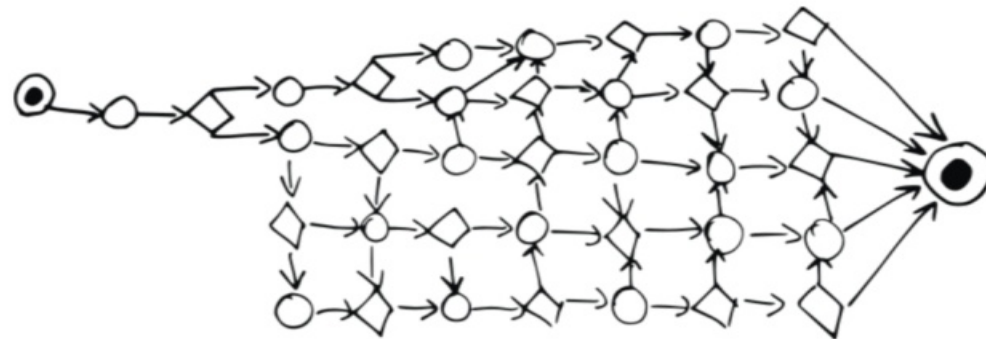


geek & poke

<http://geekandpoke.typepad.com/geekandpoke/2009/03/let-the-clouds-make-your-life-easier.html>



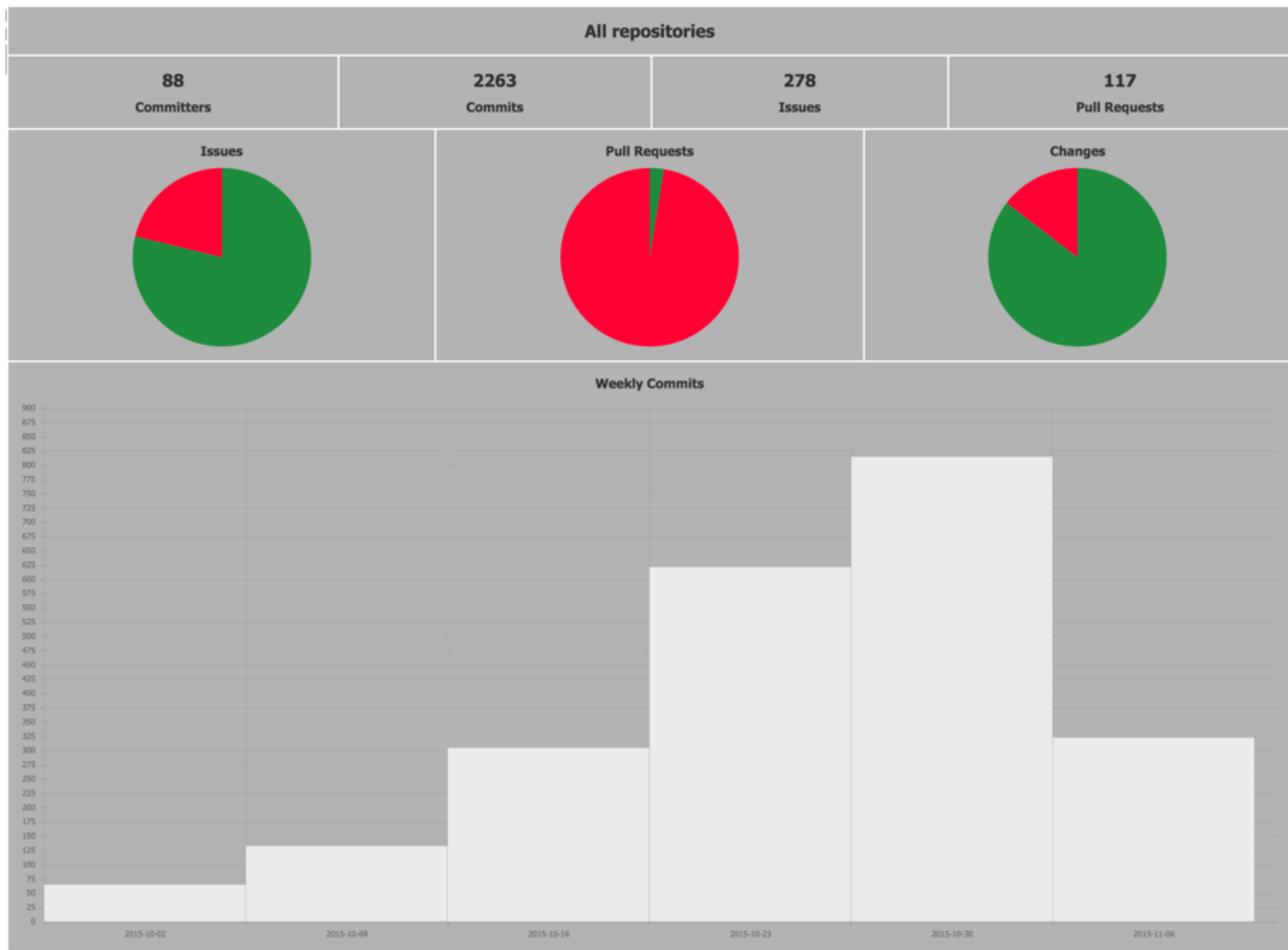
LAST YEAR WE
RECOGNIZED THAT OUR
PROCESSES WERE FAR
TOO COMPLEX

SO WE PUT THEM
INTO THE CLOUD



Cloud Computing & Scalability

Reid Holmes



4
Committers

139
Commits

31
Issues

0
Pull Requests

Links
[GitHub](#)
[Commits](#)
[Issues](#)
[Pull requests](#)

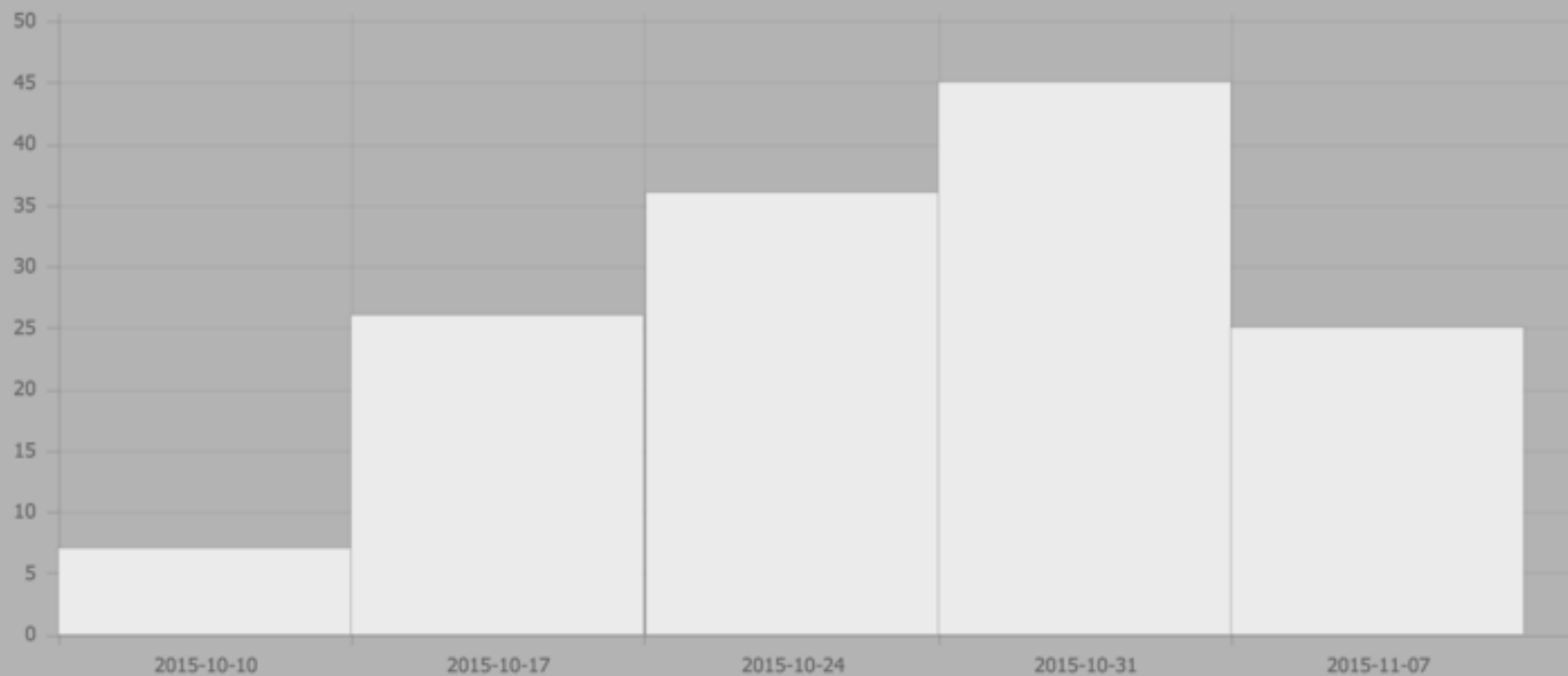
Issues



Changes



Weekly Commits



4
Committers

269
Commits

32
Issues

0
Pull Requests

Links
[GitHub](#)
[Commits](#)
[Issues](#)
[Pull requests](#)

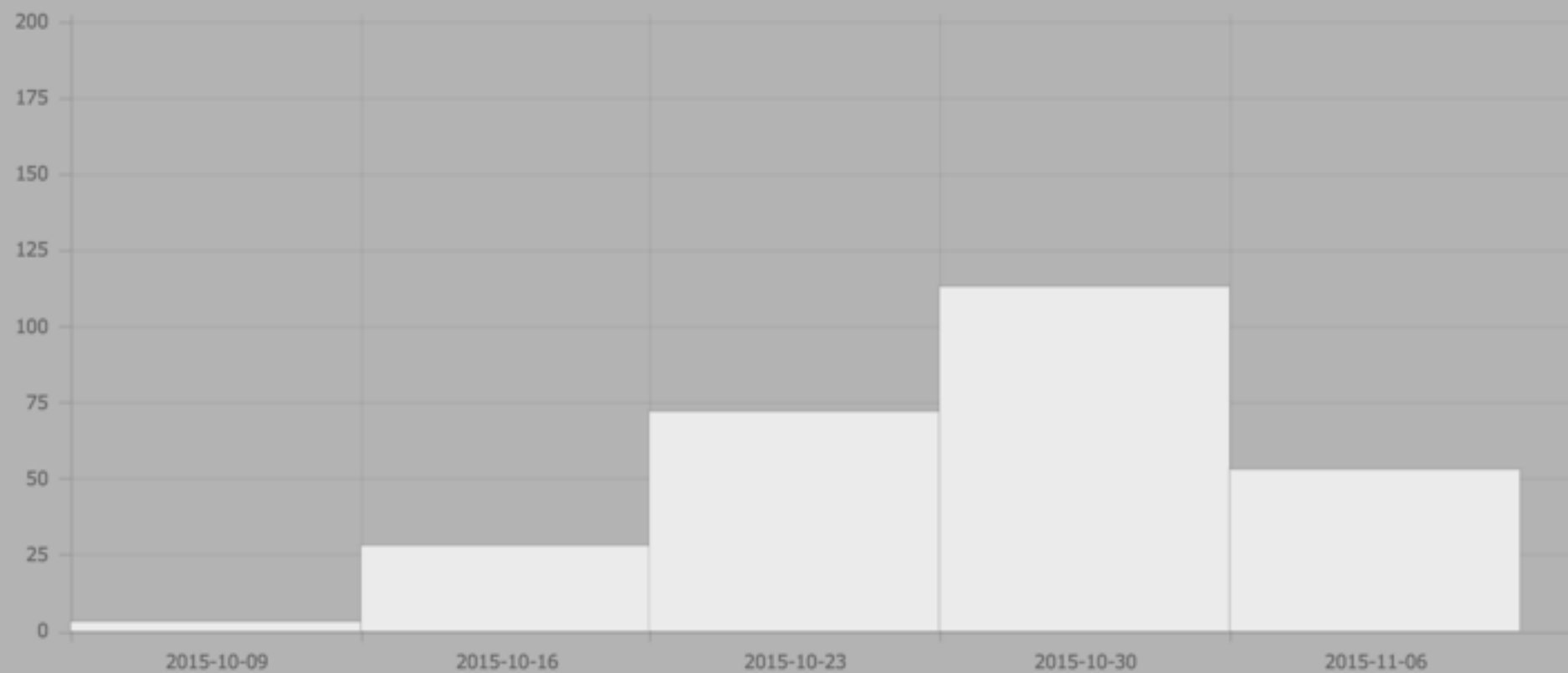
Issues



Changes



Weekly Commits



4
Committers

71
Commits

15
Issues

0
Pull Requests

Links
[GitHub](#)
[Commits](#)
[Issues](#)
[Pull requests](#)

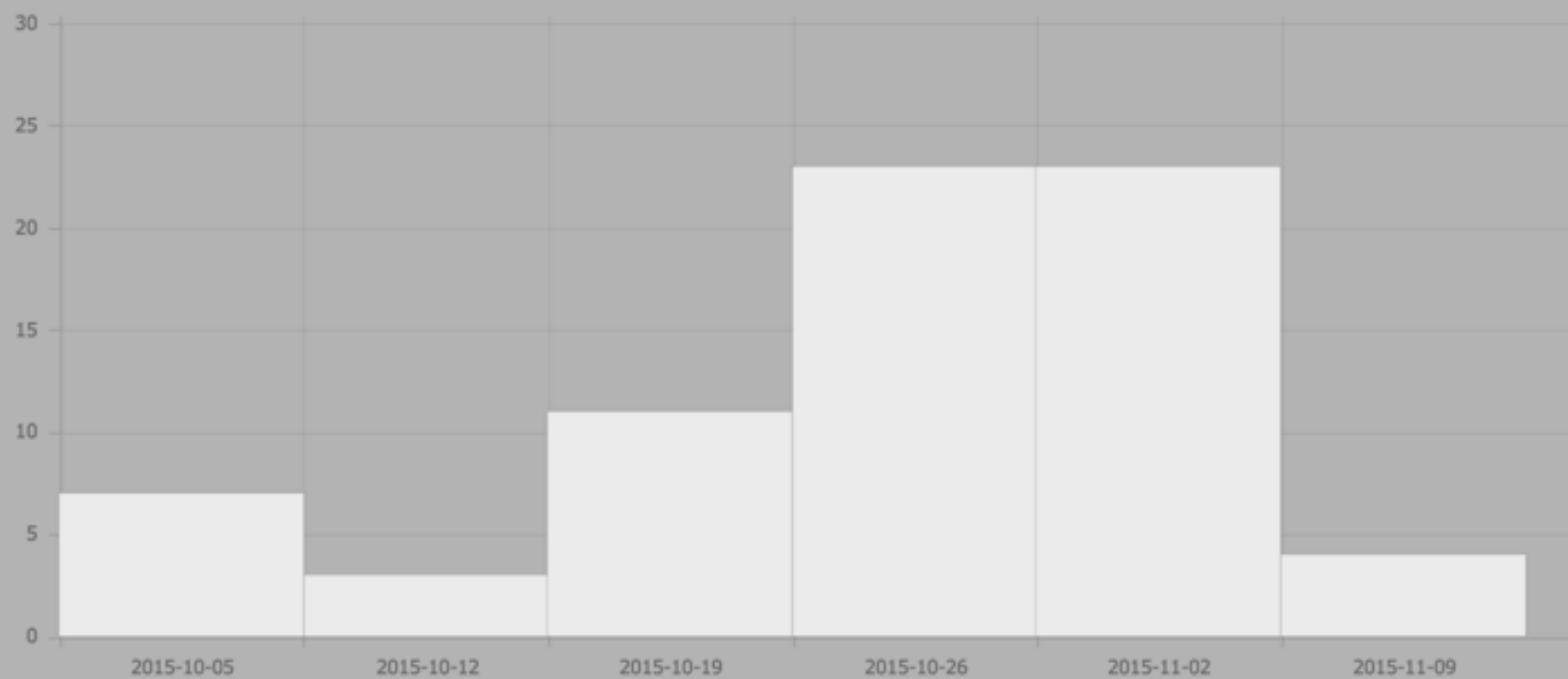
Issues



Changes



Weekly Commits



1
Committers

1
Commits

0
Issues

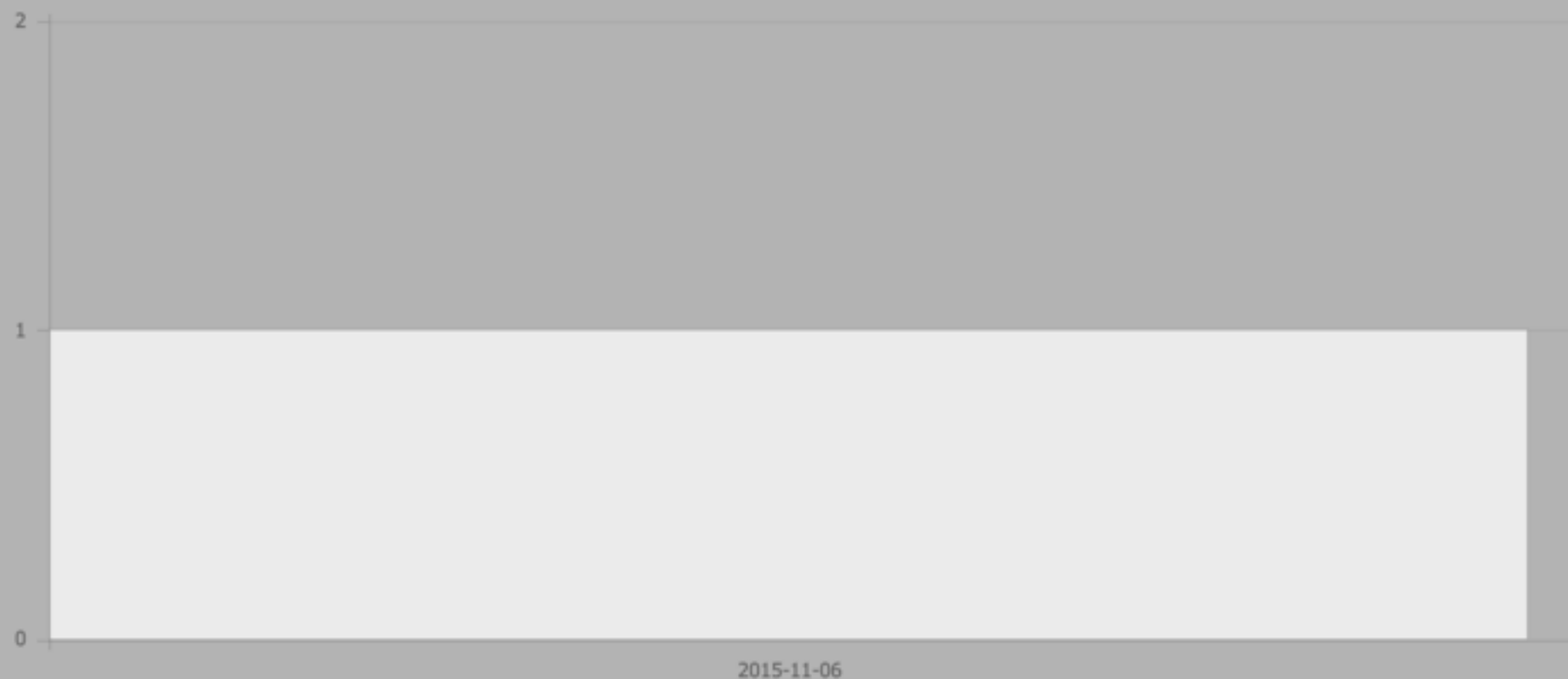
0
Pull Requests

Links
[GitHub](#)
[Commits](#)
[Issues](#)
[Pull requests](#)

Changes



Weekly Commits



2015-11-06



Contributors

Traffic

Commits

Code frequency

Punch card

Network

Members

Sep 27, 2015 – Nov 12, 2015

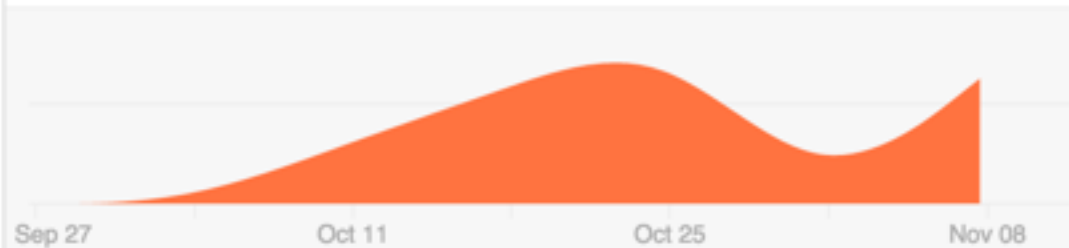
Contributions to master, excluding merge commits

Contributions: **Commits** ▾



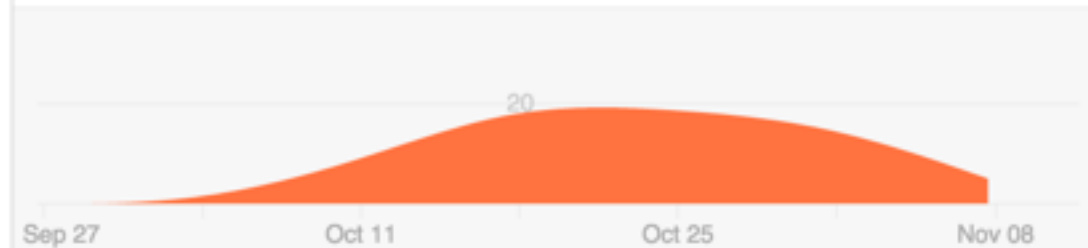
94 commits / 144,737 ++ / 41,466 --

#1



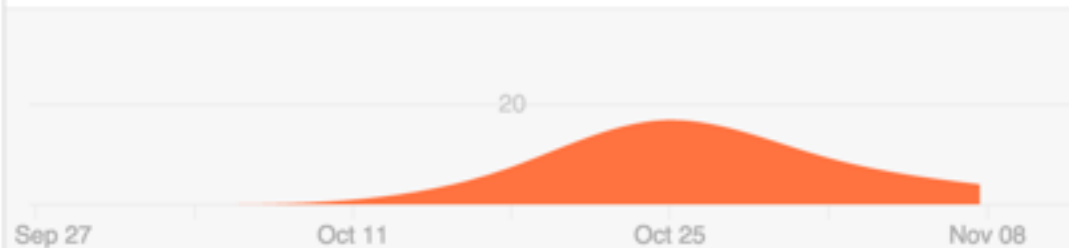
68 commits / 5,816 ++ / 2,522 --

#2



39 commits / 1,666 ++ / 750 --

#3



16 commits / 1,800 ++ / 648 --

#4



Cloud precursors

- ▶ Grid Computing:
 - ▶ Combination of computing resources from multiple administrative domains applied to common tasks.
 - ▶ Usually used to create ‘super computers’ that can work on specific parallel computation tasks.
- ▶ Utility Computing:
 - ▶ Combining computation, storage, and services metered like utilities.

Cloud Computing

- ▶ “Cloud computing is a model for enabling convenient, **on-demand** network access to a **shared pool** of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be **rapidly provisioned and released** with minimal management effort or service provider interaction. This cloud model **promotes availability** and is composed of five essential characteristics, three service models, and four deployment models.” [NIST]

NIST Essential Characteristics

- ▶ On-demand self-service:
 - ▶ Consumers can provision computing capabilities without human interaction.
- ▶ Resource pooling:
 - ▶ Computing resources are pooled to serve multiple consumers.
 - ▶ Location independence.
- ▶ Rapid elasticity
 - ▶ Resources can be easily added and removed.
- ▶ Measured service [services and/or resources]
 - ▶ Metering of storage, processing, bandwidth, etc.

Benefits

- ▶ Agility
- ▶ Scalability
- ▶ Cost
- ▶ Reliability
- ▶ Security

Technology

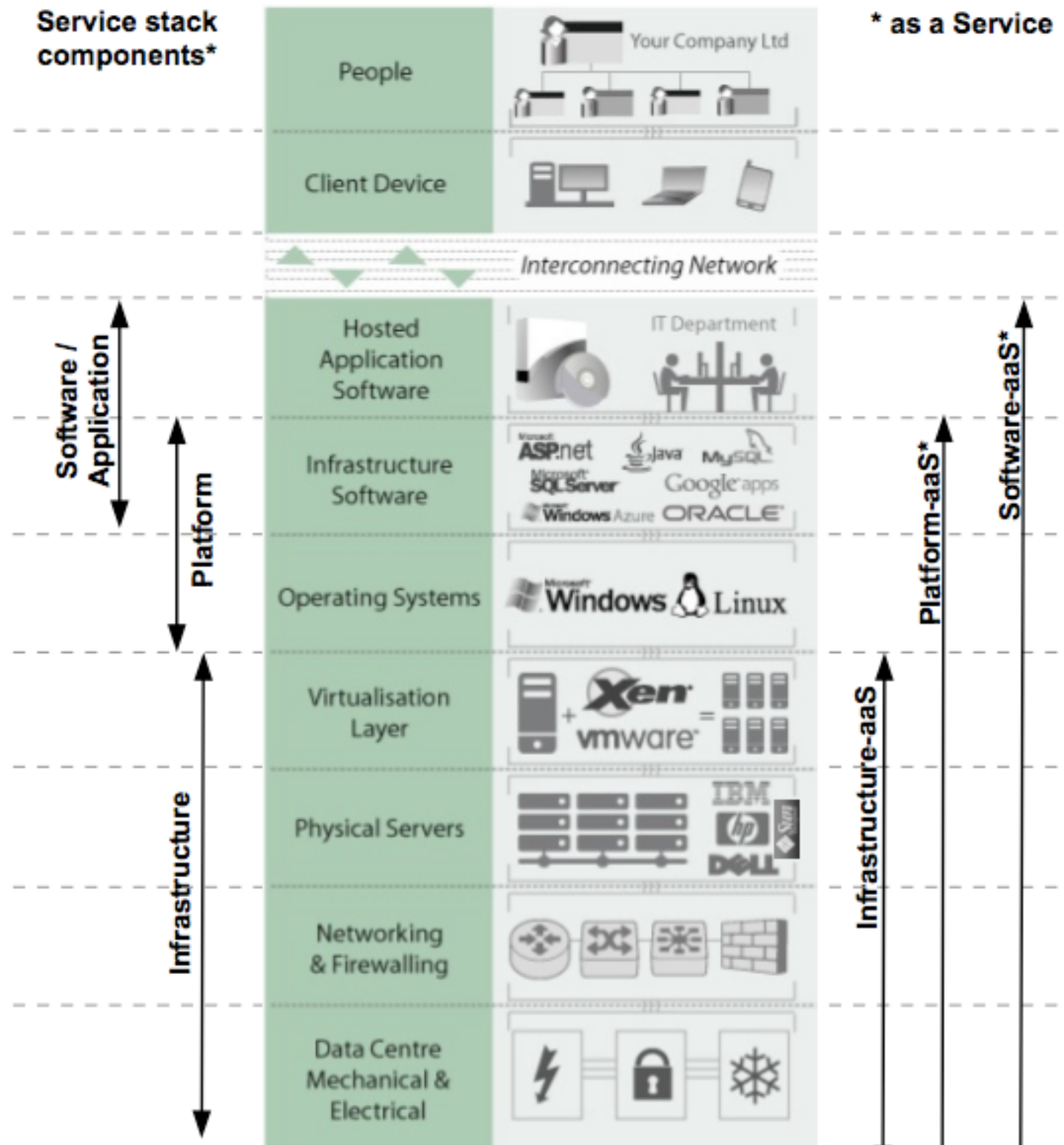
- ▶ Thick and thin clients
- ▶ Broadband
- ▶ Data centres
 - ▶ Large capacity
 - ▶ Globally distributed
- ▶ APIs
 - ▶ Administration
 - ▶ Development
 - ▶ Resource migration

Cloud Layers

- ▶ SaaS
 - ▶ Vendor-controlled remote applications.
 - ▶ Concerns: control, performance, security, privacy.
- ▶ PaaS
 - ▶ Vendor-controlled environment.
 - ▶ Concerns: as for SaaS w/ limited technology choices.
- ▶ IaaS
 - ▶ Vendor-provided resources; consumer provisions VM.
 - ▶ Concerns: more expertise needed to leverage flexibility.

Service Layers Definition

Service stack components*



Notes:

Brand names for illustrative / example purposes only, and examples are not exhaustive.

* Assumed to incorporate subordinate layers.

Cloud Spectrum

**less flexible
more constrained
less effort**

**more flexible
less constrained
more effort**

Google docs



salesforce.com

Windows Azure



amazon web services

Eucalyptus

the rackspace cloud

Layers of Control

In-house Deployment	Hosted Deployment	IaaS Cloud	PaaS Cloud	SaaS Cloud
Data	Data	Data	Data	Data
APP	APP	APP	APP	APP
VM	VM	VM	Services	Services
Server	Server	Server	Server	Server
Storage	Storage	Storage	Storage	Storage
Network	Network	Network	Network	Network
Organization controlled	Organization & service provider share control		Service Provider controlled	

[1] Visualizing the Boundaries of Control in the Cloud. Dec 2009.
<http://kscottmorrison.com/2009/12/01/visualizing-the-boundaries-of-control-in-the-cloud/>

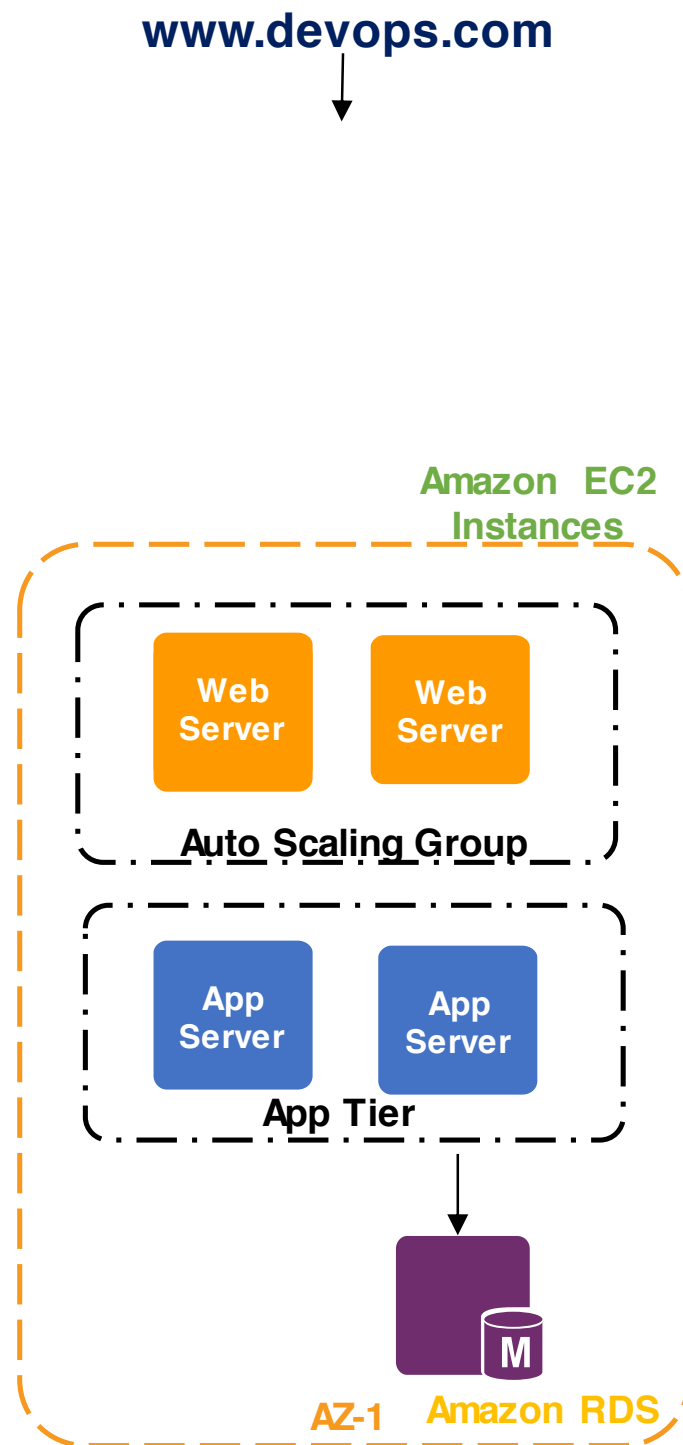
Cloud Consistency

- ▶ CAP Theorem (Eric Brewer UC Berkeley)
 - ▶ Cannot ensure all three simultaneously:
 - ▶ Consistency
 - ▶ Availability
 - ▶ Partition tolerance

Cloud Security NFPs

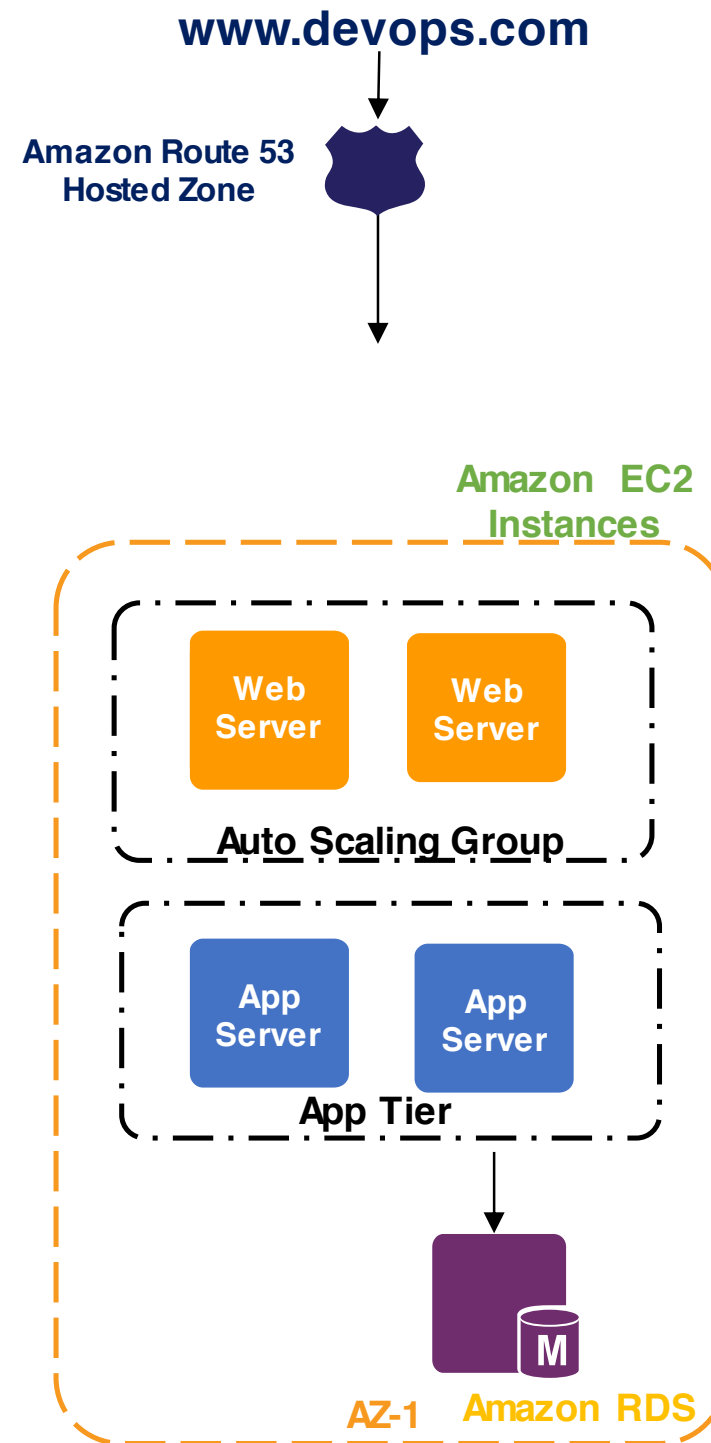
- ▶ Users want assurances of:
 - ▶ Confidentiality
 - ▶ Integrity
 - ▶ Authenticity
 - ▶ Anonymity
 - ▶ Privacy
- ▶ Data remanence is problematic:
 - ▶ How can you purge data from the cloud?

Basic 3-Tier Architecture



Example From Chris Parnin:
<https://github.com/CSC-DevOps/Course>

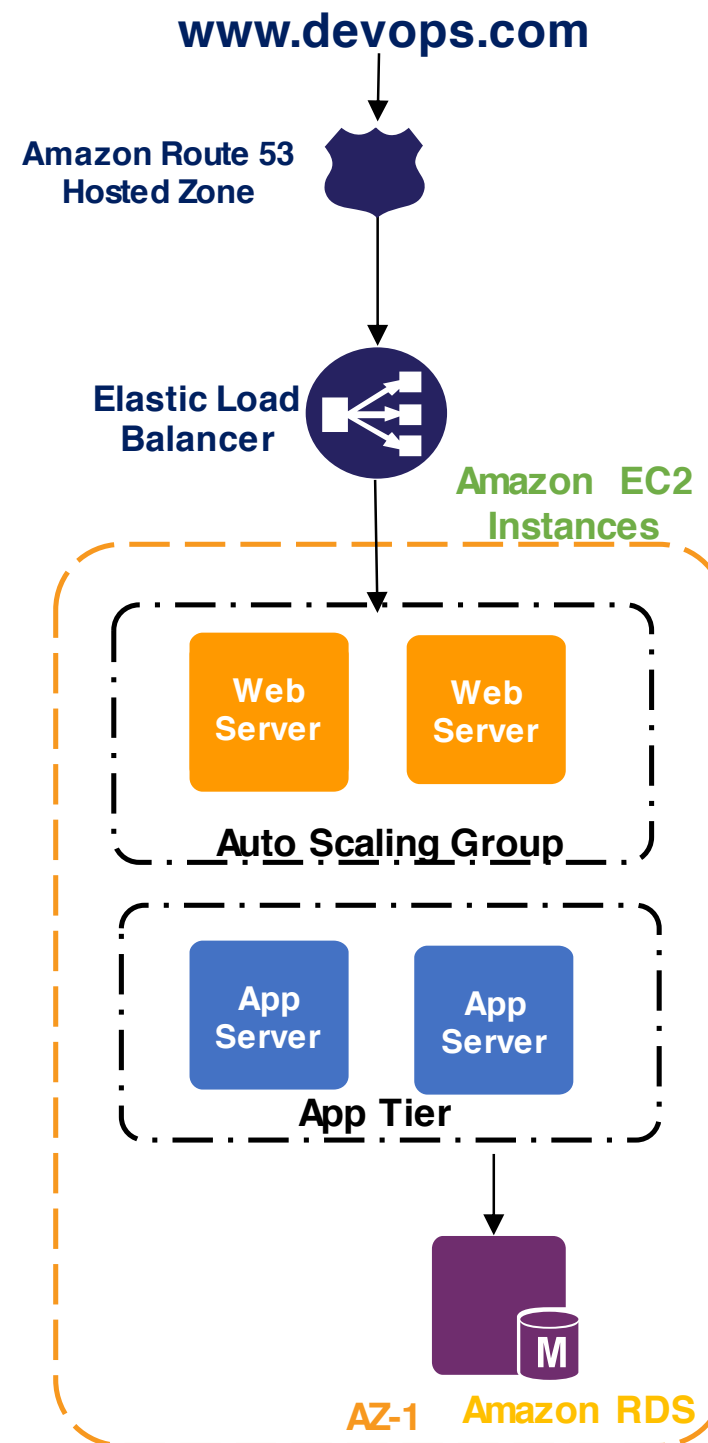
DNS Services



Example From Chris Parnin:
<https://github.com/CSC-DevOps/Course>



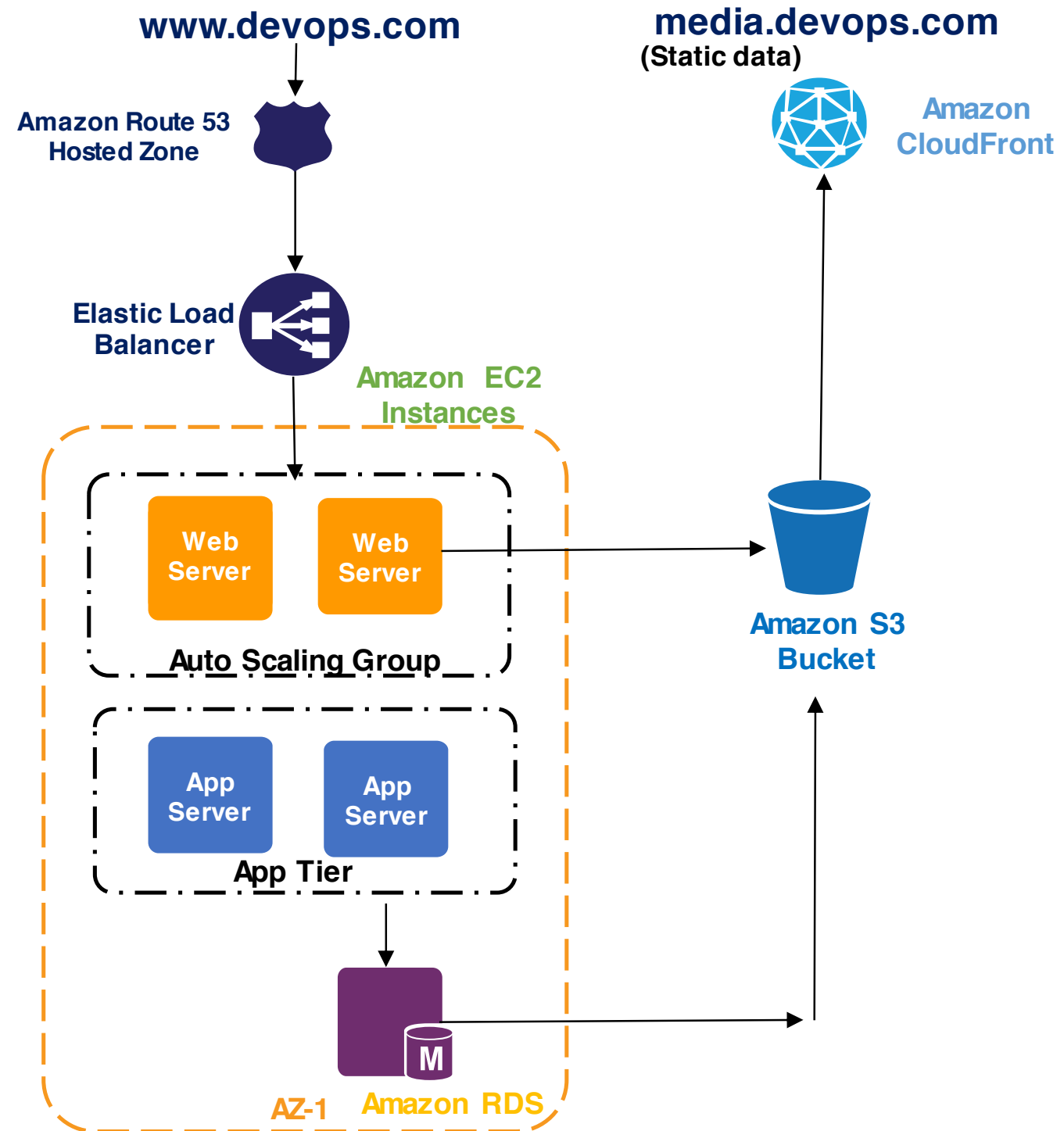
Load Balancing



Example From Chris Parnin:
<https://github.com/CSC-DevOps/Course>



Content Delivery



Example From Chris Parnin:
<https://github.com/CSC-DevOps/Course>

On Latency

Memory Reference

.0001ms

4k from SSD

.15ms

Round trip (same datacenter)

.5ms

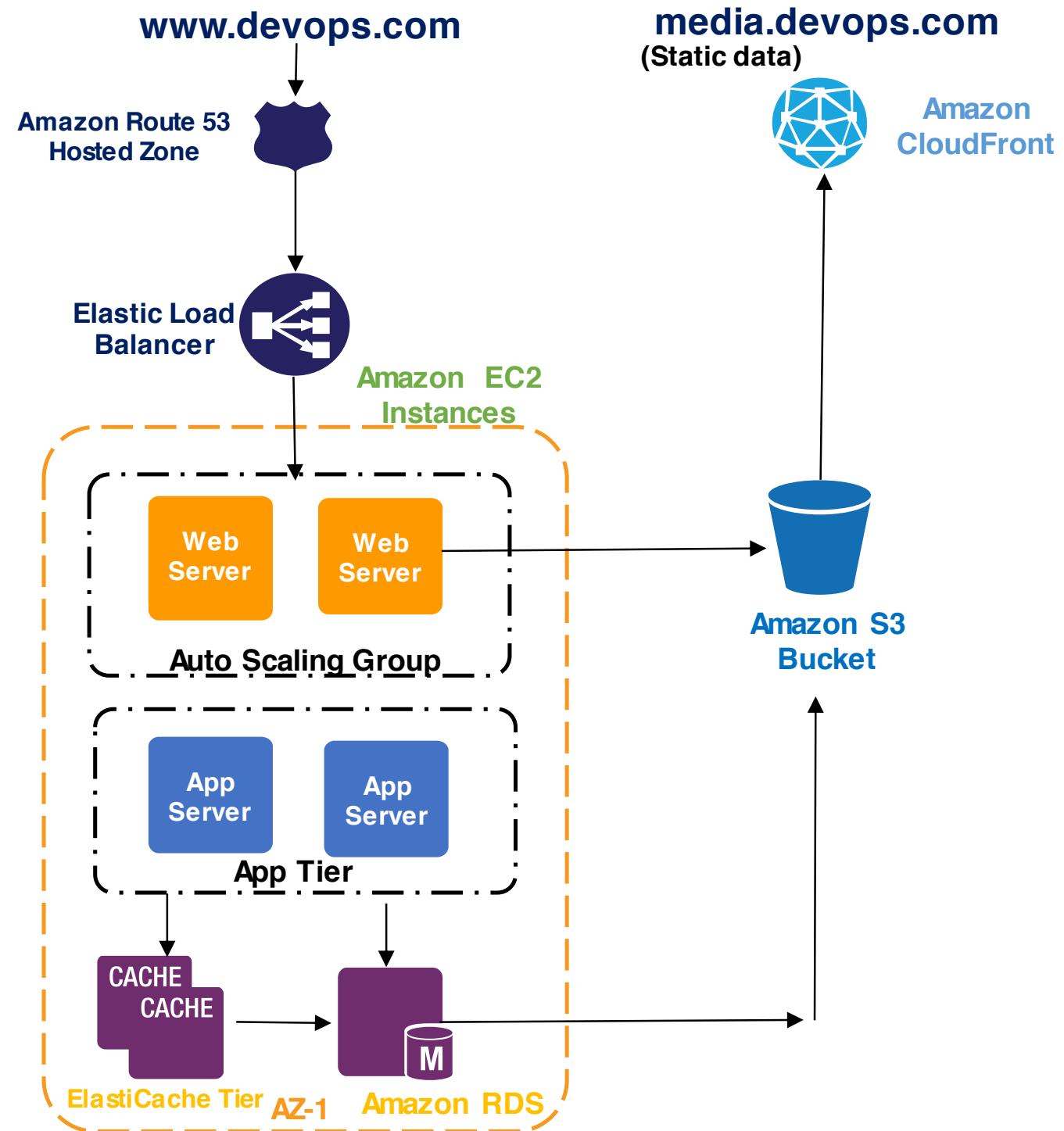
Disk seek

10ms

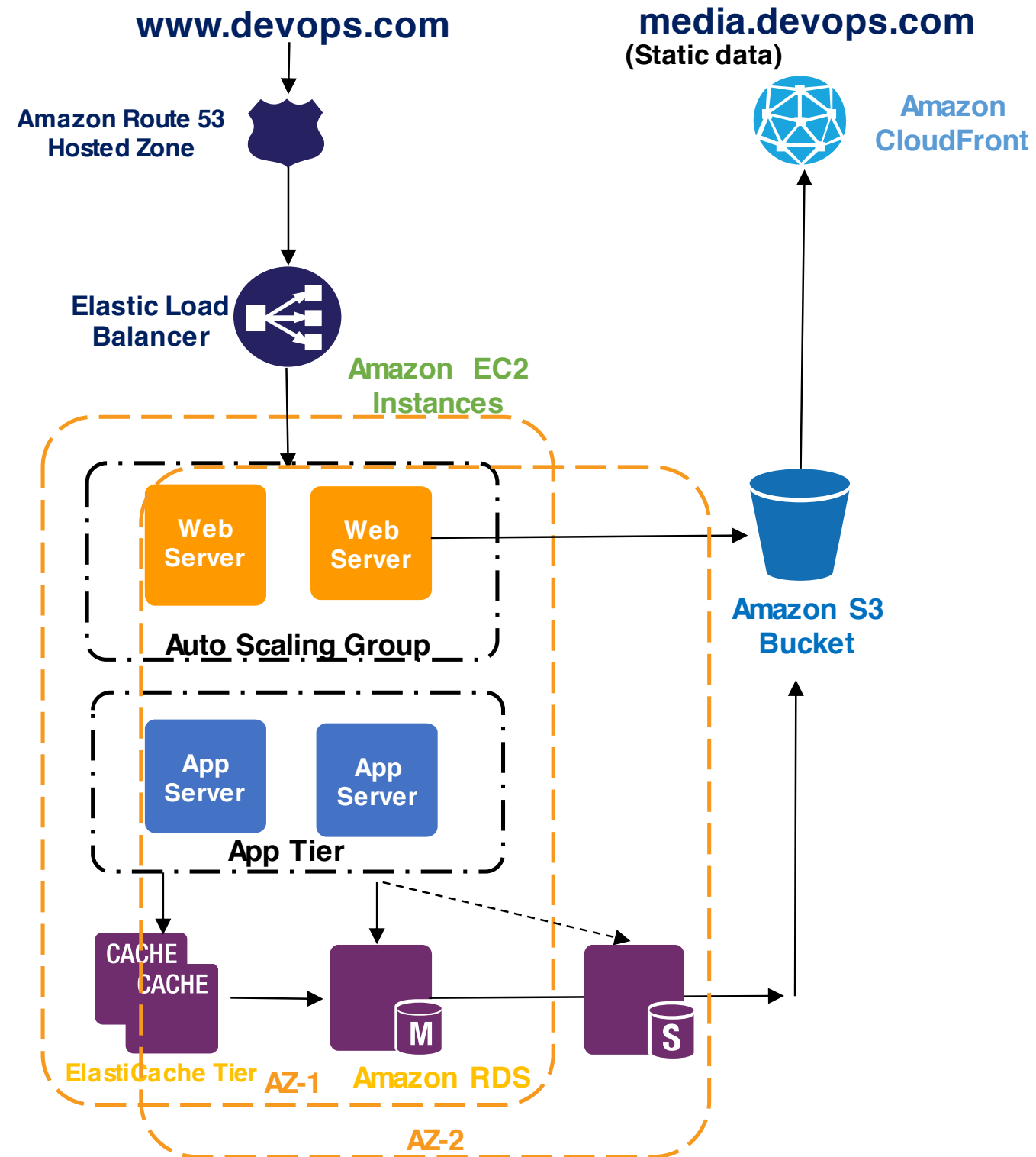
<https://gist.github.com/jboner/2841832>



Caches



Availability Zones



Example From Chris Parnin:
<https://github.com/CSC-DevOps/Course>

On Latency

Memory Reference	.0001ms
------------------	---------

4k from SSD	.15ms
-------------	-------

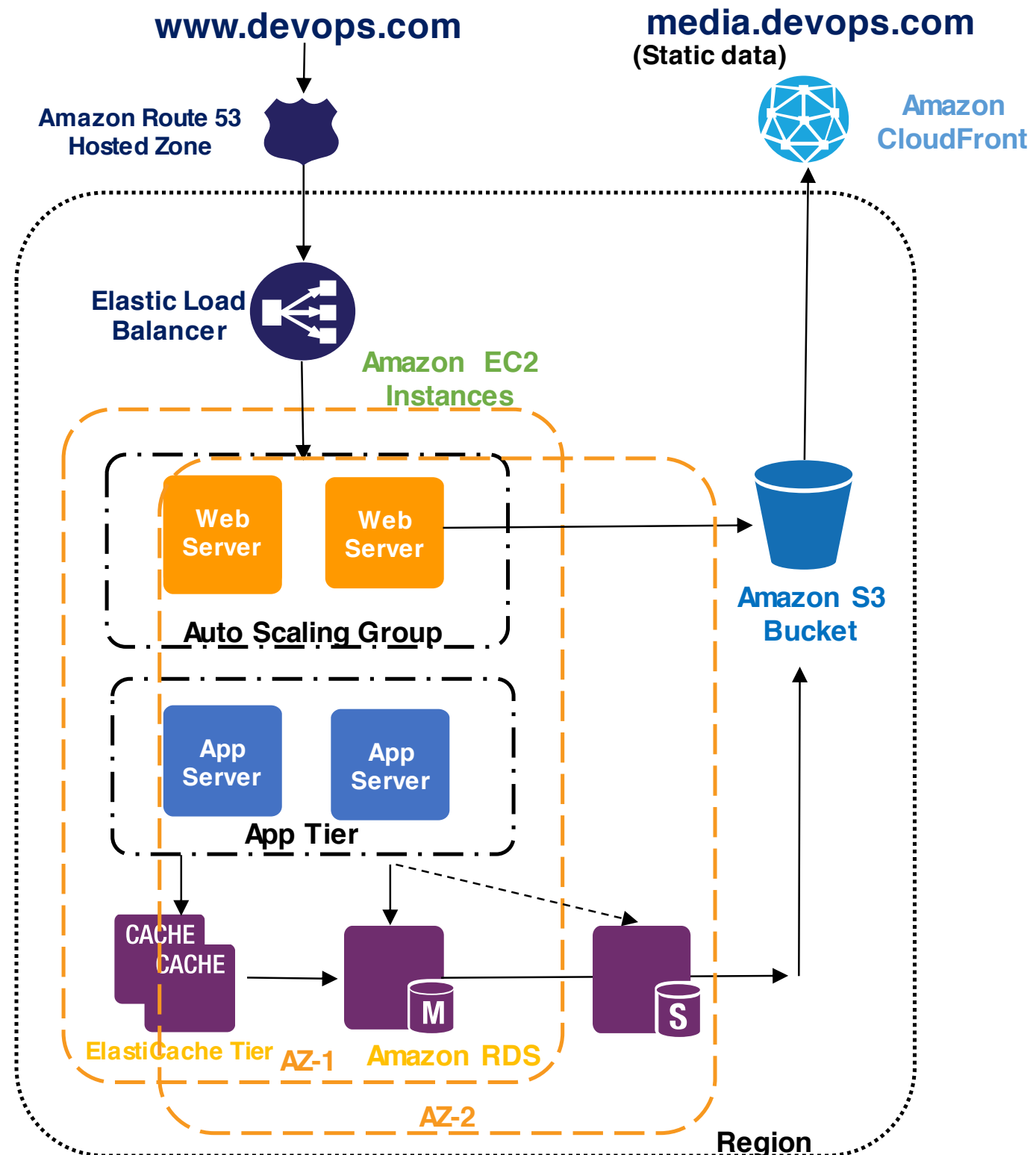
Round trip (same datacenter)	.5ms
------------------------------	------

Disk seek	10ms
-----------	------

Round trip (international)	150ms
----------------------------	-------

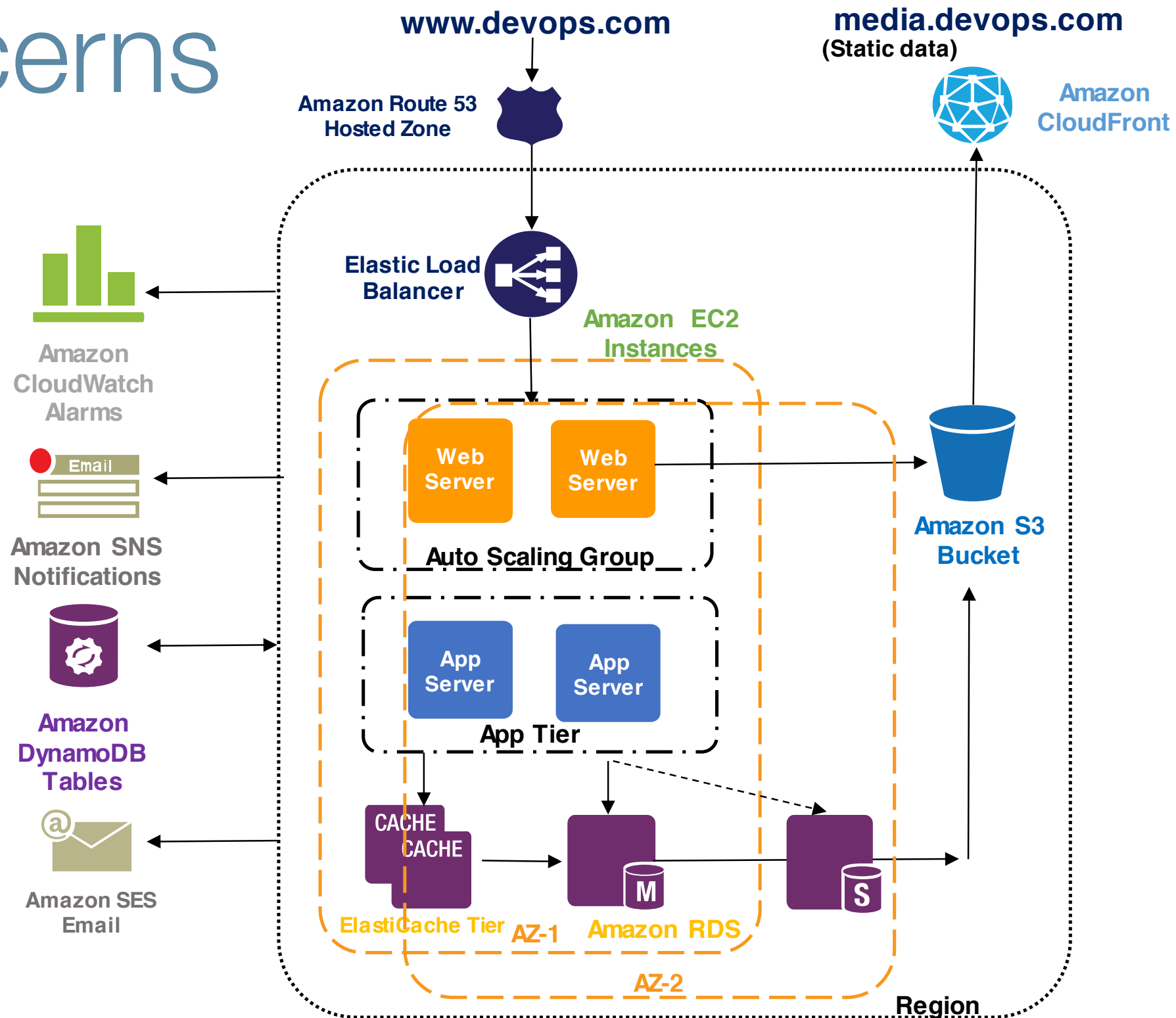
<https://gist.github.com/jboner/2841832>

Regions



Example From Chris Parnin:
<https://github.com/CSC-DevOps/Course>

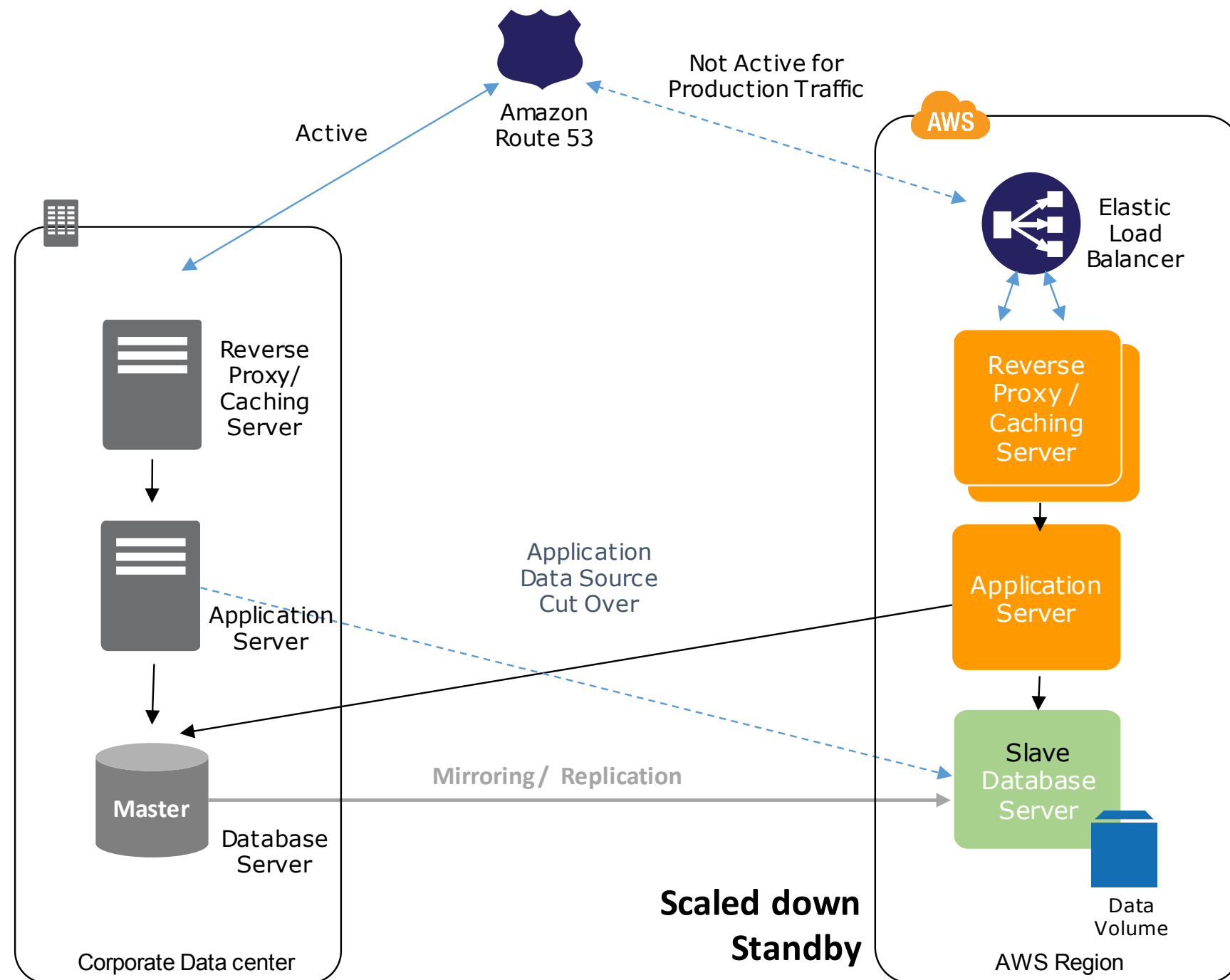
Other Concerns



Example From Chris Parnin:
<https://github.com/CSC-DevOps/Course>

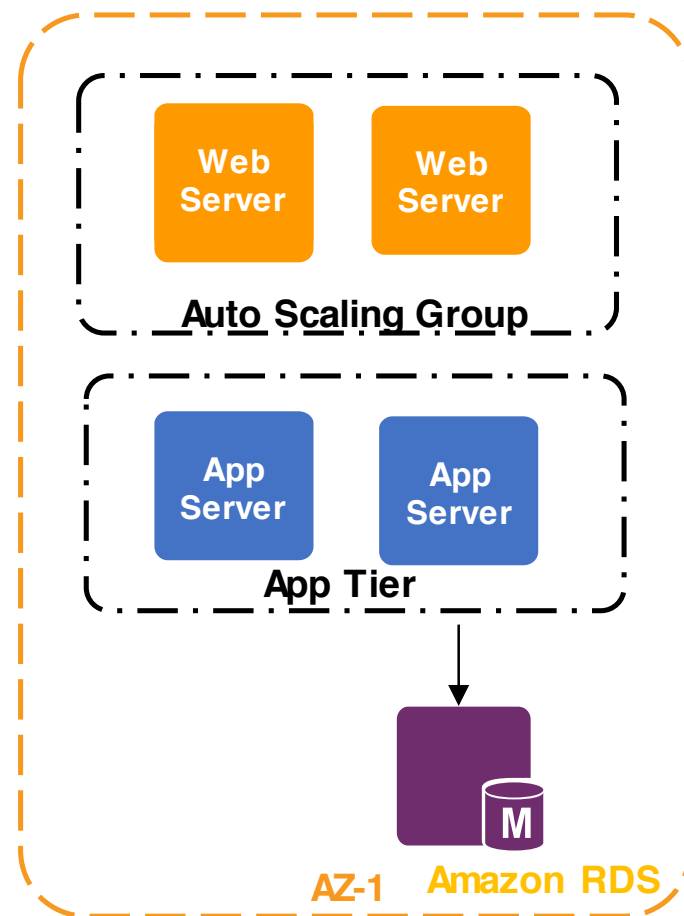


Disaster Recovery



Example From Chris Parnin:
<https://github.com/CSC-DevOps/Course>

You know how
to do this:



But what
about these:

DNS

Amazon Route 53
Hosted Zone



Load
Balancing

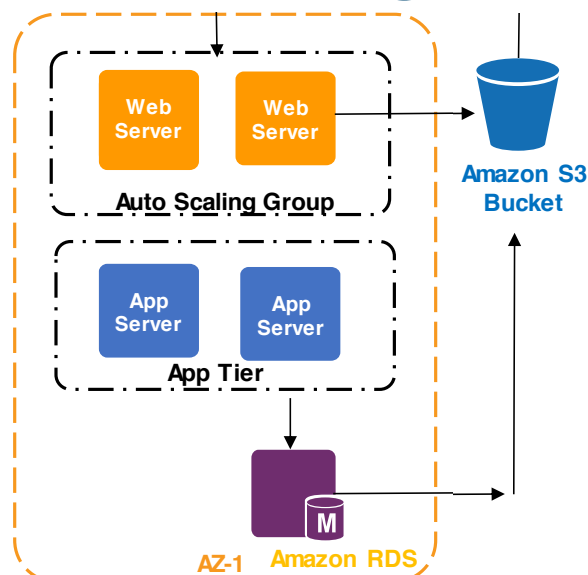
Elastic Load
Balancer



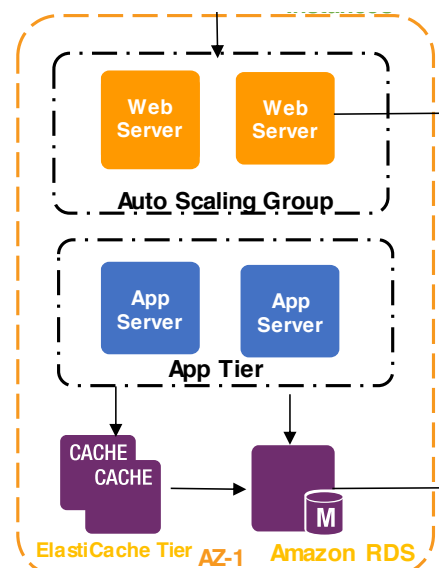
For each of these:

- * You need to know they exist.
- * And what they do.
- * And how to use them.
- * And how to configure them.
- * And how to verify them.
- * And how to monitor them.

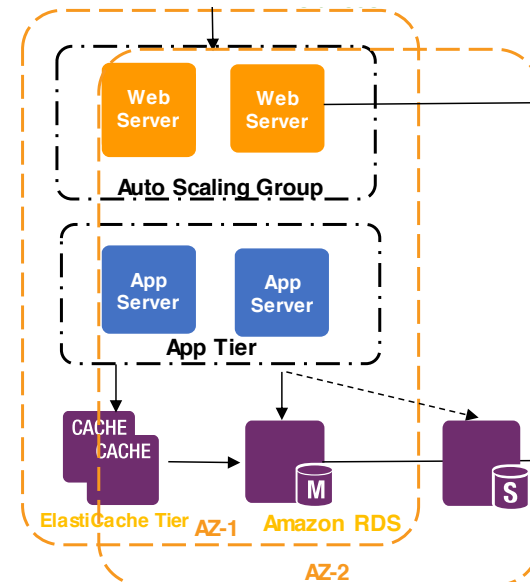
Static
Caching



Dynamic
Caching



Availability
Zones



Regions

