Introduction to AWS

Matt Trahan, Narrative Science

Bio

- > 1 year at Narrative Science, 7 years at Amazon
- email: mtrahan@narrativescience.com
- twitter: @matt_trahan

Takeaways

- Understand the basics of “Cloud Computing”
- Learn how to launch a web service
- Learn how to make it scalable and available
- AWS Best Practices (at least some of them)

Overview

1. AWS
2. Getting Started
3. My First Instance
4. Databases
5. Automation
6. Availability, Scaling, and Monitoring
7. Wrap up

AWS aka Cloud Computing

On-demand delivery of **IT resources** via the Internet with **pay-as-you-go** pricing

Don’t have to buy physical computers

Programmatic access for automating your infrastructure

Lots of services available to accelerate development

Flexibility to try things out and change your mind

AWS Services

- Compute
- Storage
- Databases
- Networking
- Analytics
- Application Services
- Deployment & Management

What are we going to use today?

EC2

CloudFormation

RDS

Route53

ELB

Autoscaling


amazon
web services™

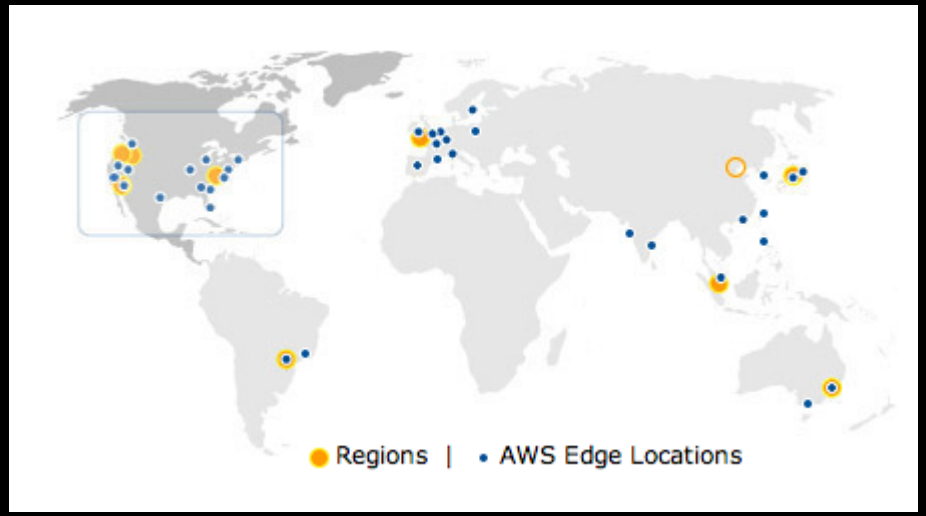
EC2

Repo

https://github.com/mtrahan/fuzzy-octo-computing-machine

EC2
Instance

Regions



Ruby app on EC2 Instance
running database locally

uby app on EC2 Instance
unning database locally

RDS

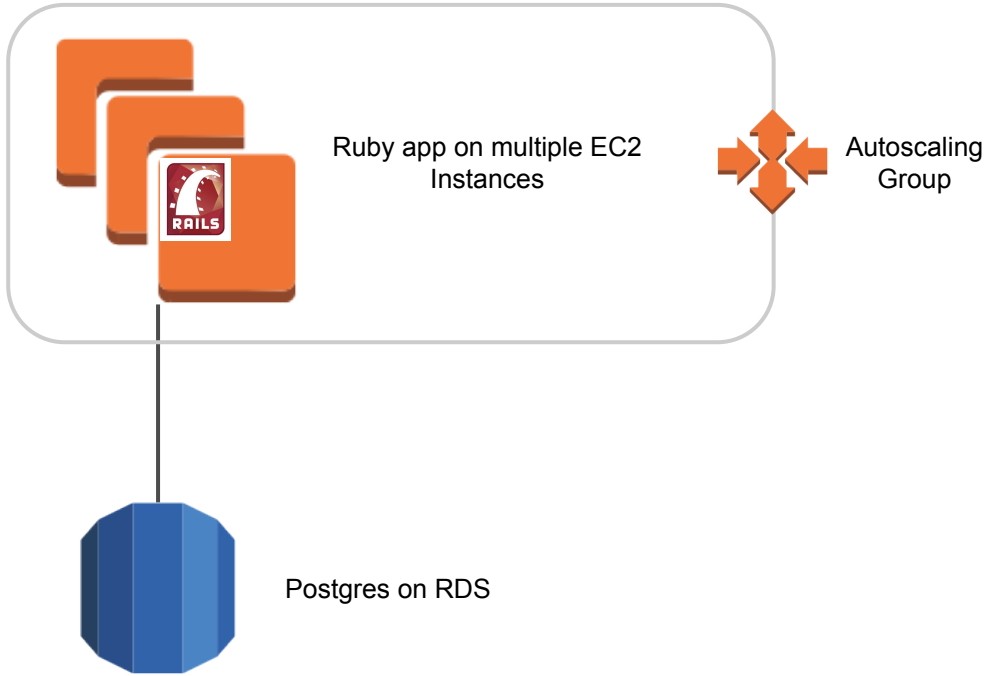
Ruby app on single EC2 Instance

Postgres on RDS

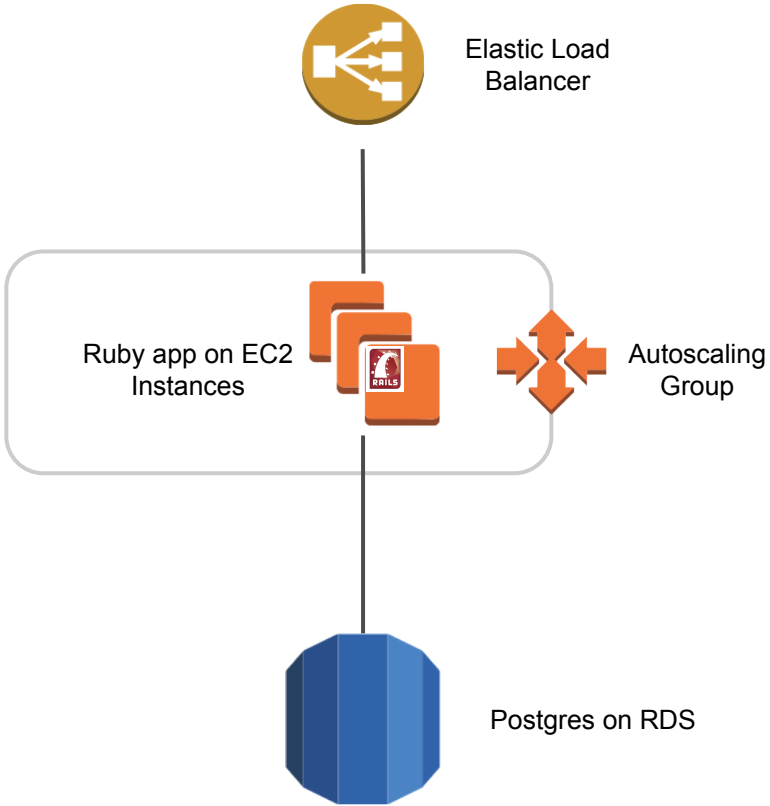
Ruby app on multiple EC2 Instances

Postgres on RDS

AutoScaling

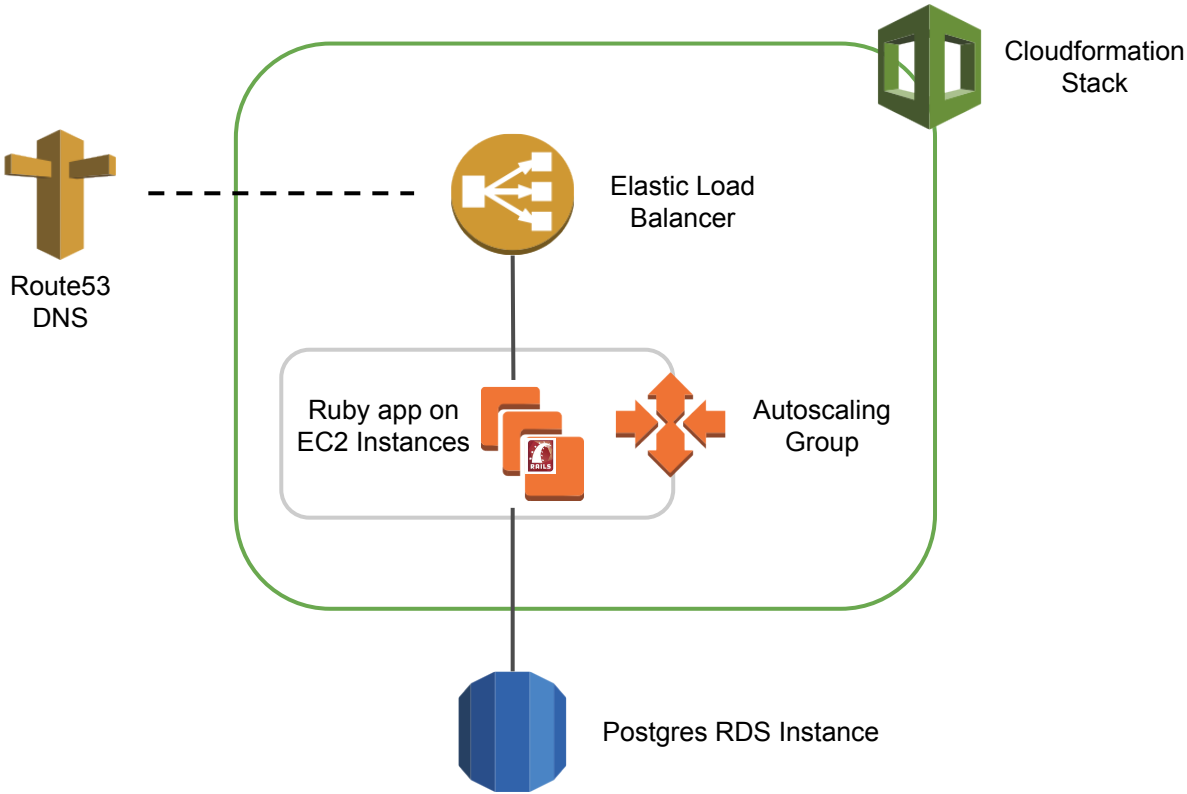

RAILS
Ruby app on multiple EC2 Instances
Autoscaling Group
Postgres on RDS

Elastic Load Balancer

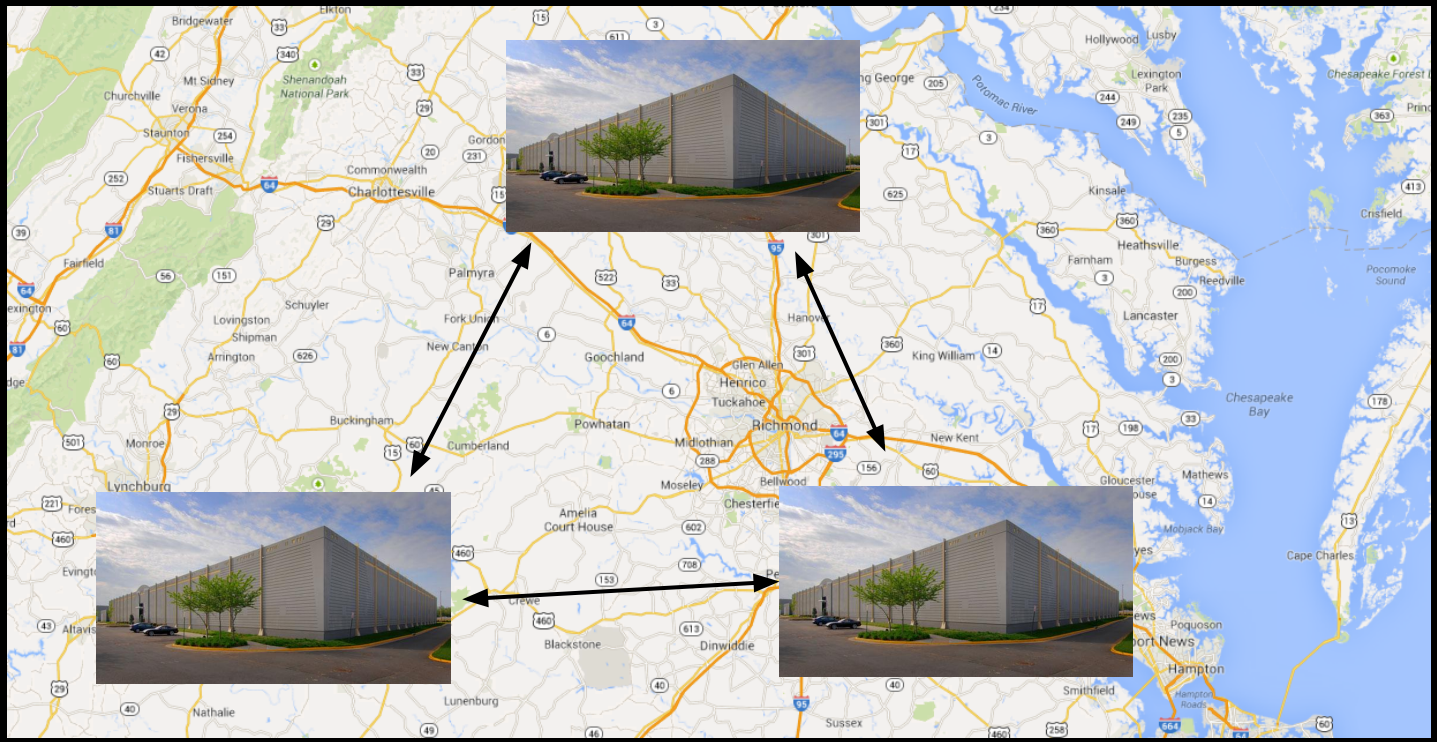

Elastic Load Balancer
Ruby app on EC2 Instances
RAILS
Autoscaling Group
Postgres on RDS

CloudFormation

Route53


Cloudformation
Stack
Route53
DNS
Elastic Load
Balancer
Ruby app on
EC2 Instances
Autoscaling
Group
Postgres RDS Instance

Availability Zones



* Not accurate

RDS

- Multi-Availability Zone Master/Slave
- Simple Scaling
- Backups
- Automatic maintenance

PaaS

- Beanstalk
- Heroku

Thanks!