AWS Migration

Stephen Mouring

Director of Software Engineering

Sequoia Holdings

Introductions

Who Am I

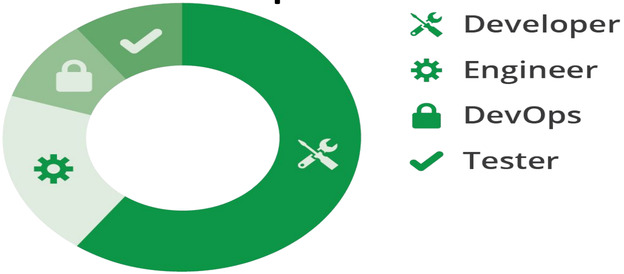
- Software Developer for the customer for 10 years
- Certified AWS Developer
- Certified System Administrator
- Working on DevOps Certification
- Project Lead on Legacy Application Migration Cloud Project
  - First Production application in C2S
  - Involved as an advisor/observer on multiple similar efforts

Who Is Sequoia

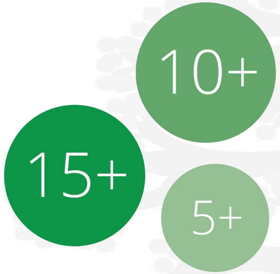
- Employee owned world class IT professional services firm that rewards growth and positive results
- Leading provider of high-end IT, engineering and professional services
- Customers:
  - Intelligence Community, Homeland Security & Defense, Civilian & Justice, Emerging Technology
- Solutions:
  - AWS/C2S Development/Migration, Big Data Development, Cyber Security, Data Science & Analytics
- Founded in 2012; built on uncompromising commitment to our customers and employees
- Strong Leadership Team driven by entrepreneurial creativity and focused on client relationships, service and satisfaction
- Venture Arm Launched in 2015 - Sequoia Apps
  - Startup Accelerator that provides coaching, mentoring and initial seed funding

Sequoia By The Numbers

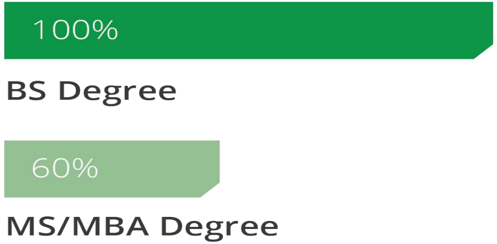
Expertise



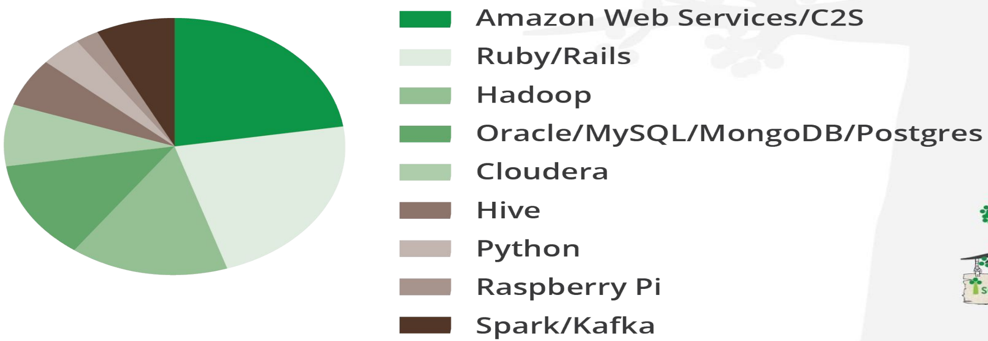
Years of Experience



Education



Technologies



AWS Past Performance

- Sequoia has established itself as a leader in assisting the national security sector with strategy, architecture, migration and development solutions in the AWS cloud
- Experience w/ AWS support includes the US Intelligence Community and DHS
- AWS past performance examples:
  - Application Migration Support
  - New Application Development
  - System Architecture Design and Rollout
  - Managed Services (DevOps)
- Sequoia is a recognized Amazon Consulting Partner
- Detailed Past Performance Upon Request

This Presentation

Focus on C2S

But many lessons apply to the public AWS environment as well

C2S Differences

Fewer things and not all things work the same way (or always very well)

Most on Software Architecture / Design

Some on DevOps

Less on Databases, Networking, and so forth...

Lessons Learned

Lesson Learned #1

Not everything goes in the cloud

Consider Sunk Costs

Example: Greenplum Database

Cloud cost 3 times more than buying hardware annually!

Example: Persistent Clusters

900 servers, 20 hour a day utilization… AWS is cheap for surges, expensive for continual use!

Consider Data Scale

Expensive to hold lots of data for a long time!

Political Pressure

The Cloud is not Magic

Lesson Learned #2

Design for *Stateless* Servers

Servers Die

Sometimes Amazon tells you in advance!

One of the top 10 most unsettling emails of my professional career...

Sometimes they die alone on the weekend...

AWS Paradigm:

Stateless Servers backed by Elastic Load Balancers

Major Architectural Change

In terms of a web application, this means no using the session!

Why Not Sticky Sessions?

Low Side:

Plausible but still dangerous

High Side:

You Get Either Sticky Sessions Or PKI Certificates (Not Both)

State Storage Hierarchy

Evaluate Your “State”

(1) Put it on the client side

Clients re-submit information instead of caching in the session

(2) Use your own cookies

(3) Use a database

(4) Use a distributed cache

Consider EhCache or Tomcat Session plugin

Design for autoscaling

Applications should be **bootstrappable** and **self contained**

Lesson Learned #3

EBS is expensive!

EBS is familiar and easy to migrate

But it costs a *lot*

Our Project: $4 million budget, about half is spent on EBS

Options?

S3

Much cheaper… Less intuitive to work with.

Ephemeral

Disappears on shutdown… but it some cases (HDFS?) and applications (buffering, caching, etc.) it is viable

Storage Architecture Requires Planning

A word on RDS

Lesson Learned #4

Tag early tag often

Tags are powerful

Can be accessed and used programmatically!

Ability to schedule off hours shutdowns

Makes billing far more comprehensible

Support not strong in AWS

Still a manual process… but worth the investment!

We are growing this space

Example

First step: A linux server in AWS

Second step: Batch processing on web

Third step: Solving State problems

Next Steps: Integration with DevOps and Scaling

Take away:

Migration is a gradual process

Start *small* and gradually introduce complexity

Further Resources

Upcoming Whitepaper

Hedge Environment

Our Experts Can Help

Questions?

For More Information

- Contact Us At:
  - 11180 Sunrise Valley Drive, Suite 110 Reston VA
  - 571-267-7945
  - jobs@sequoiainc.com
- Follow Us At:
  - Website: http://sequoiainc.com
  - Blog: http://blogs.sequoiainc.com/
  - Sequoia Apps: http://www.sequoia-apps.com/
  - Twitter: https://twitter.com/sequoiainc
  - Facebook: https://www.facebook.com/sequoiainc
  - LinkedIn: https://www.linkedin.com/company/sequoia-holdings-inc