



<3



Geezeo Amazon Web Services Presentation

Theron Parlin, CTO & James Elwood, Systems Administrator



“The next generation of financial organization has arrived. It's called Geezeo. And it means you can get to your personal financial information anywhere, at any time” ~ Terry Savage, Chicago Sun Times



Cost Effective

- \$72.00/month per box
- \$0.10 per GB transferred in
- \$0.18 per GB transferred out
- Costs about \$900 per month (12 servers)

Flexible

- We can run different operating systems on different boxes
- We can run any service that we want

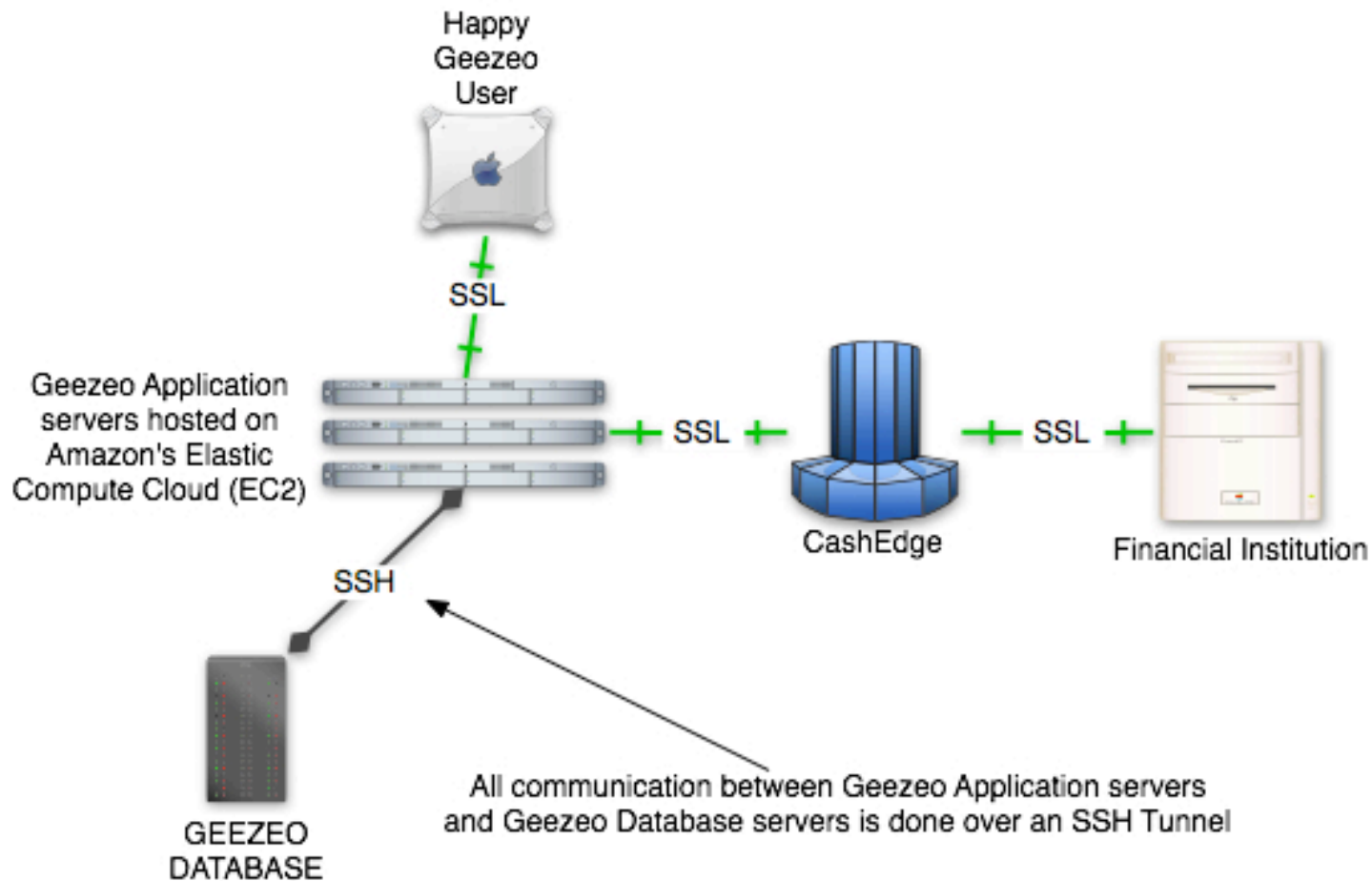
Reliable

- As of yet - no downtime due to AWS
- When they had to bring down a server, they gave us 2 weeks notice
- *we appreciate that very much

Secure

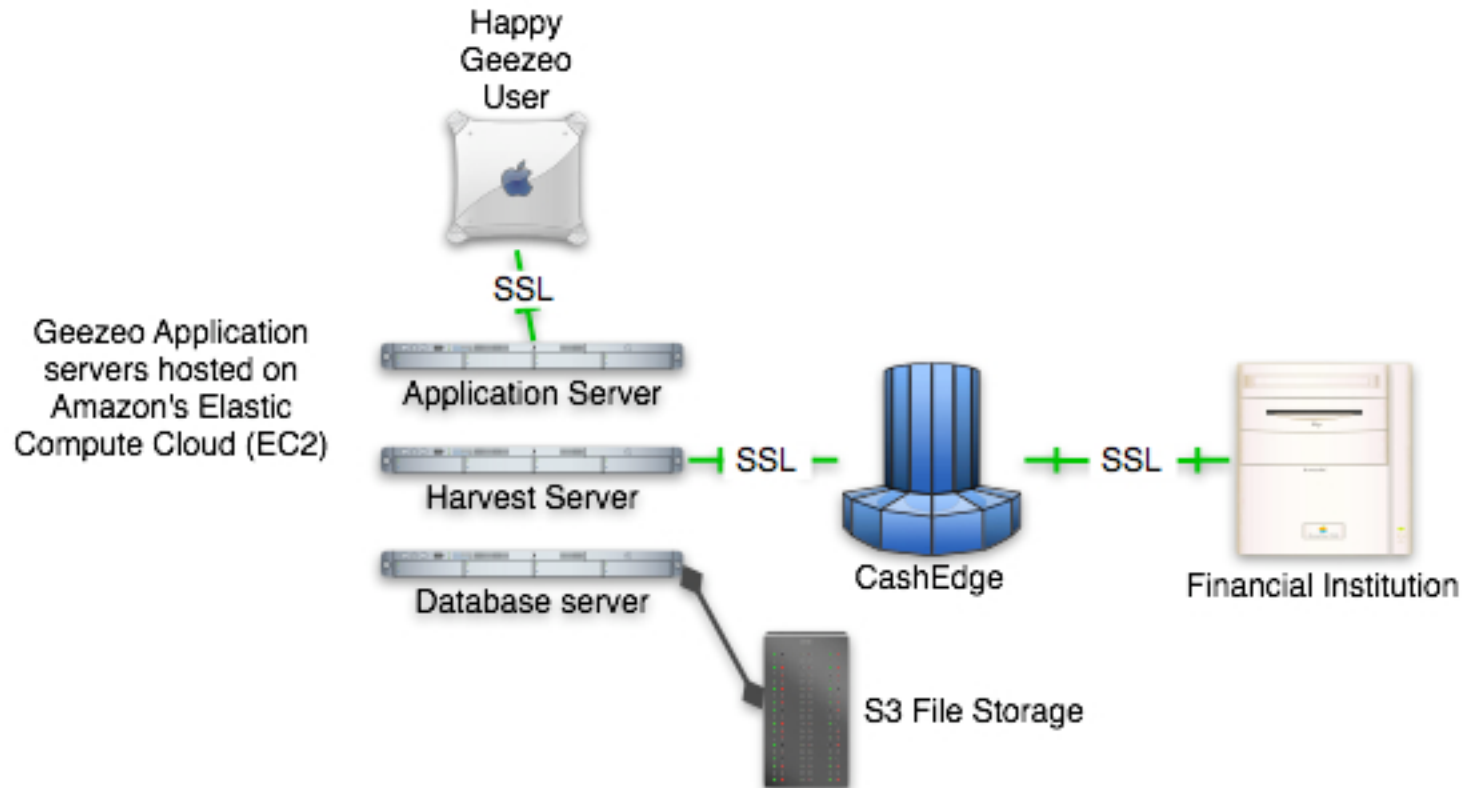
- Must "turn on" the ports you want open
- There are two layers of networking (public and private IPs)

The Past - First implementation



Here we have an off site database, which turned out to be brutally slow.

The Present - Current implementation



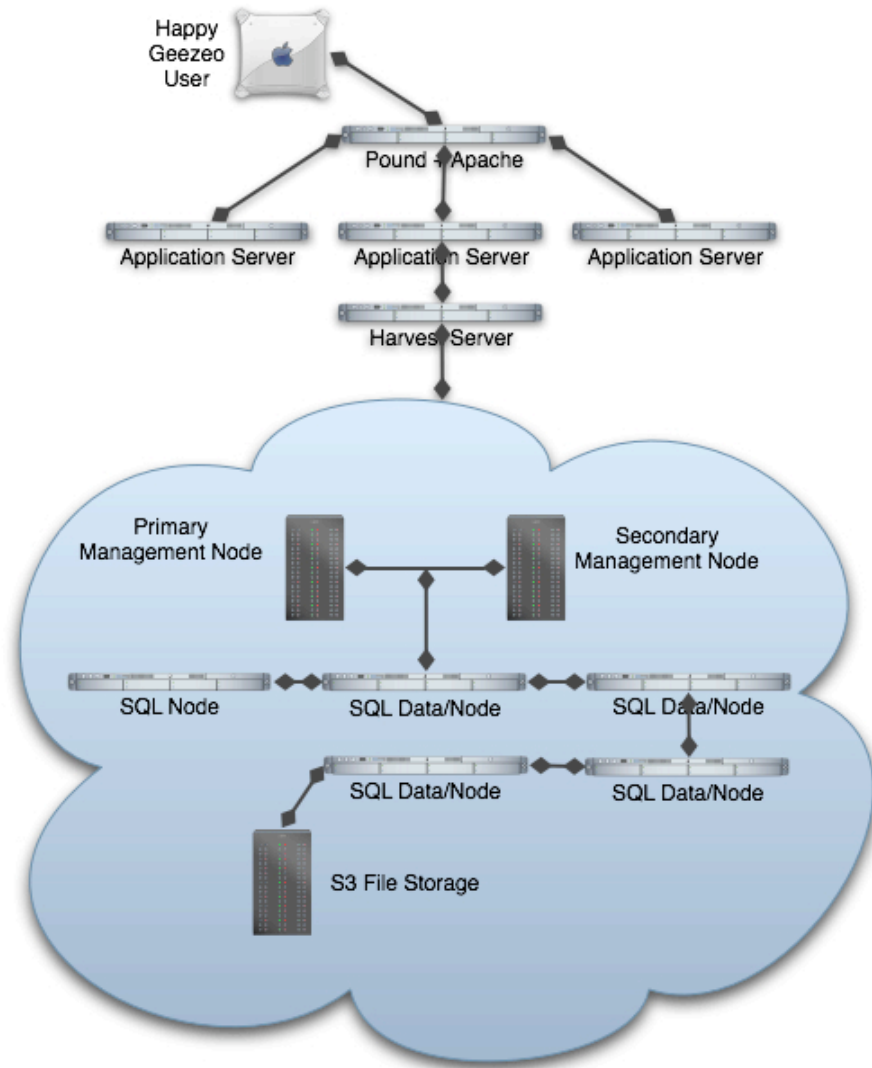
Here we have 1 Application server, 1 harvest server and a database server that we back up every 15 minutes to s3.

The Future - The way we want it



1. Pound + Apache front end
2. Three Application servers
3. Still 1 Harvest server
4. A Database Cluster

That's right, I said it! This setup is currently running in our QA environment.



Play with it

- EC2 is a sandbox
- Try new things
- It's not work if you're having fun

Build it

- Pre-made images make things quick
- Bundling makes for easy image backups
- Add/Remove the services

Break it

- Test your configuration
- What works?
- What doesn't?

One lonely DB server backing up every 5 minutes to S3

- One massive single point of failure
- Doesn't scale too well

Master/Slave replication

- Scales better
- Dev team breaks it every time they change the data schema

Cluster

- Scales best
- No single point of failure
- Dev team has yet to break it...yet.

Flexibility

- What we want
- When we want
- How we want it

Low Overhead

- No physical facility
- No hardware woes (dead fans, fried procs, flaky NICs)

Stability

- A good image will stay running until we kill it
- Instances are distributed over the grid

We can build servers at Starbucks :-)

Various RAM configurations

- MySQL clustering eats memory
- Applications could see a performance gain with more
- Scale back if its not needed

Various processor configurations

- Again with the performance gain
- Ensure the right setup for the task

Better management tools

- Descriptive tagging of images and instances
- Stats on the servers (uptime, networking, etc)

Got Questions?



Theron Parlin tparlin@geezeo.com
James Elwood jelwood@geezeo.com