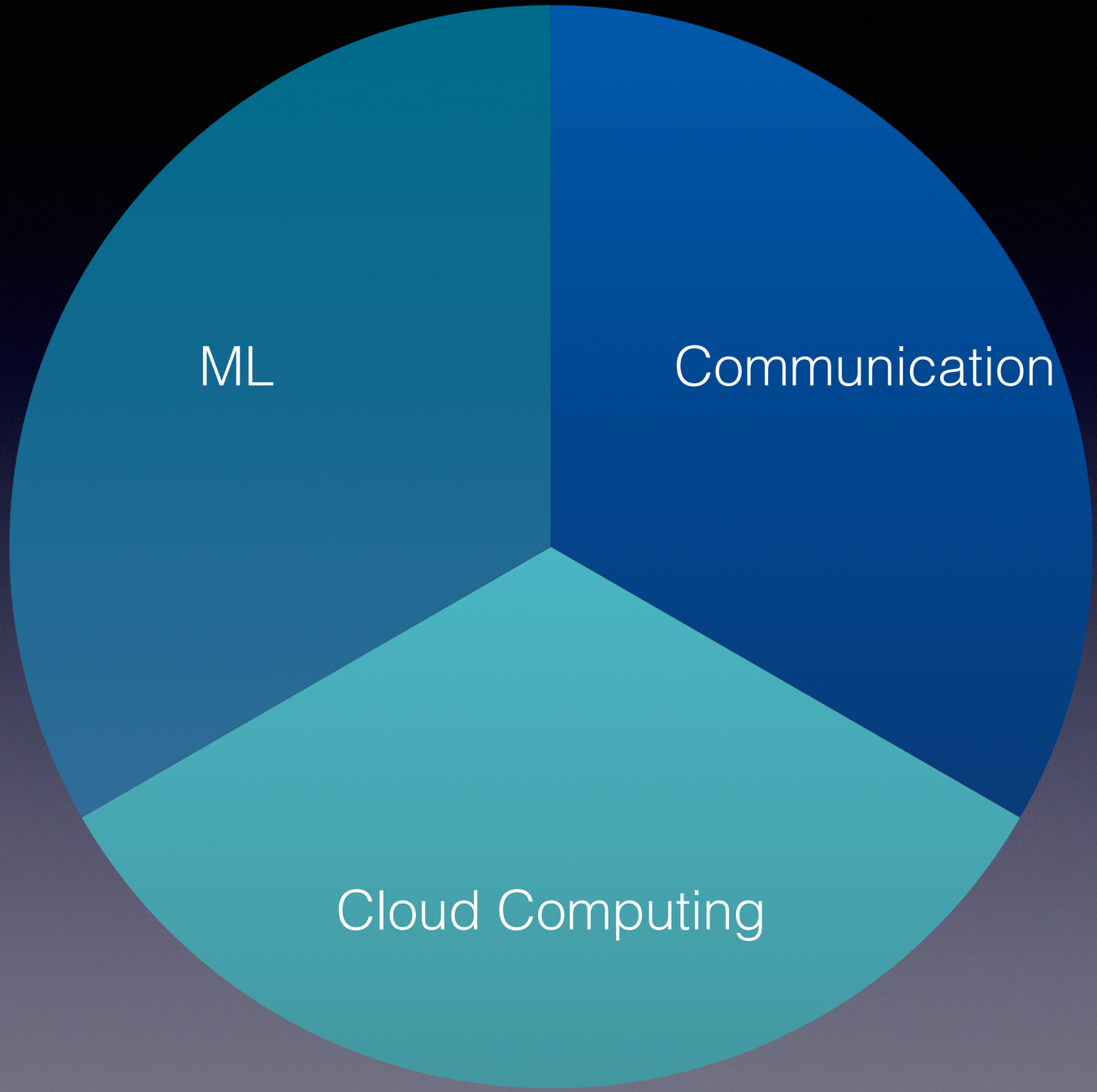
AWS
Deployment, Maintenance, & Best Practices

Greg Chase


ML
Communication
Cloud Computing

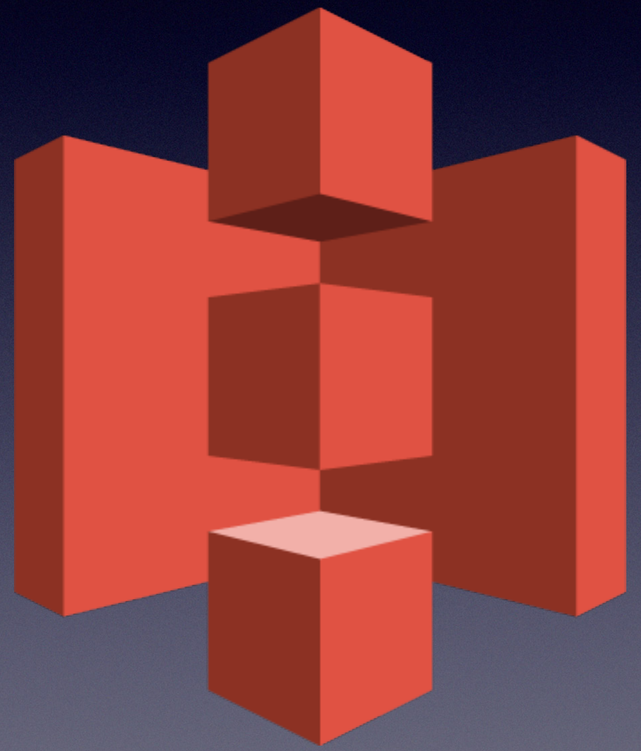
Everyone's an expert
until deploying to production.

Elastic Compute Cloud (EC2)

- "A really big, powerful computer."
- Scale up when necessary

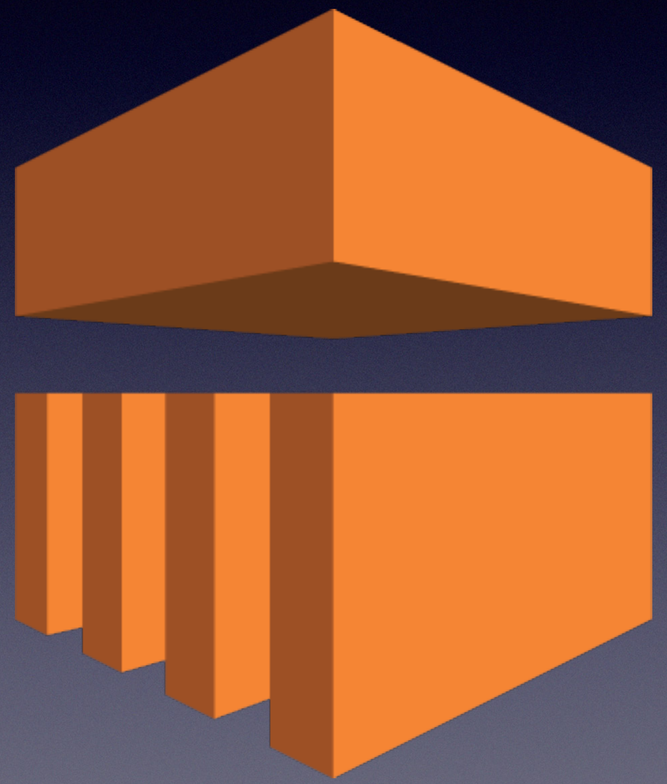
Simple Storage Service (S3)

- "An unlimited hard drive in the cloud."
- Store any file type



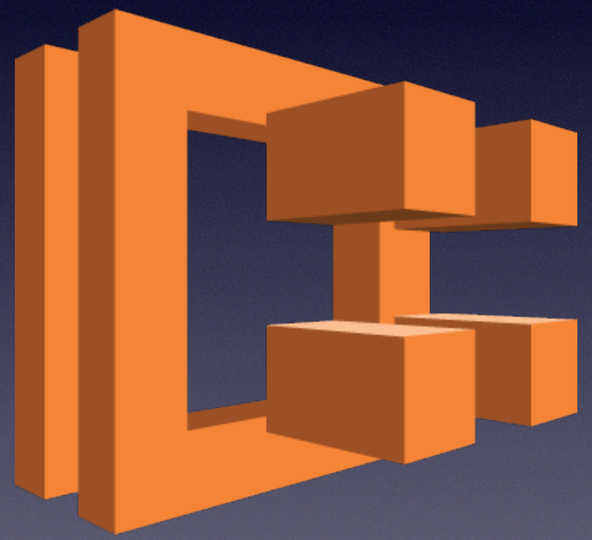
Elastic MapReduce (EMR)

- Distributed Computing
- Apache Spark



Elastic Container Service (ECS)

- Distributed Docker
- Supports EC2 & Fargate



Lambda

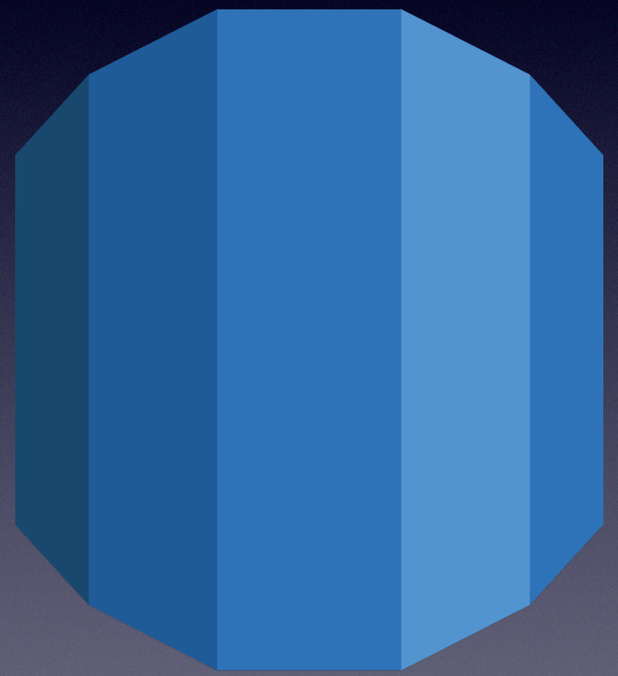
- Serverless Computing
- No EC2, ALB, or ECS
- If utilization < 40%, use Lambda

Simple Queue Service (SQS)

- Send/ receive messages
- Send messages, trigger Lambda

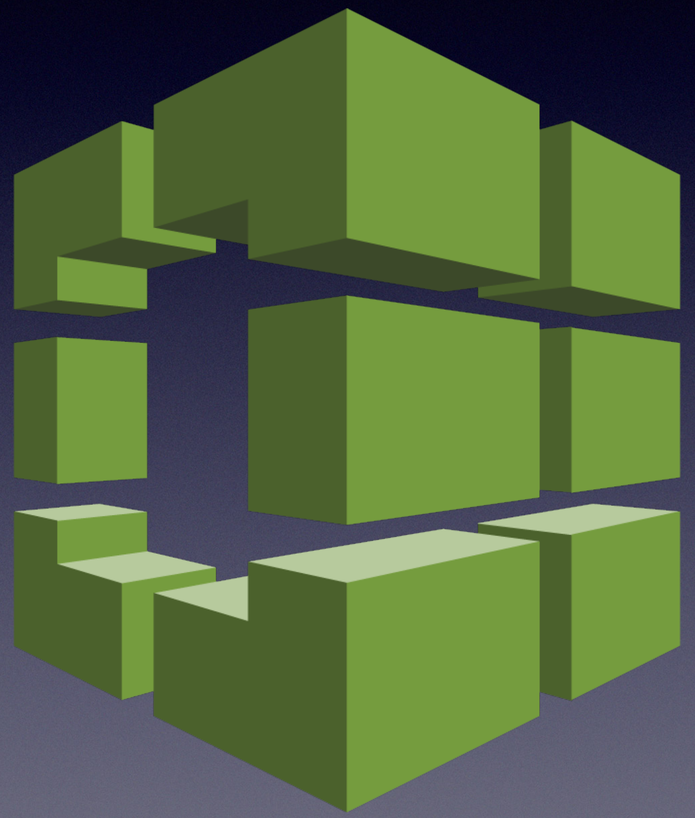
Relational Database Services (RDS)

- Databases in the cloud
- Supports all major databases



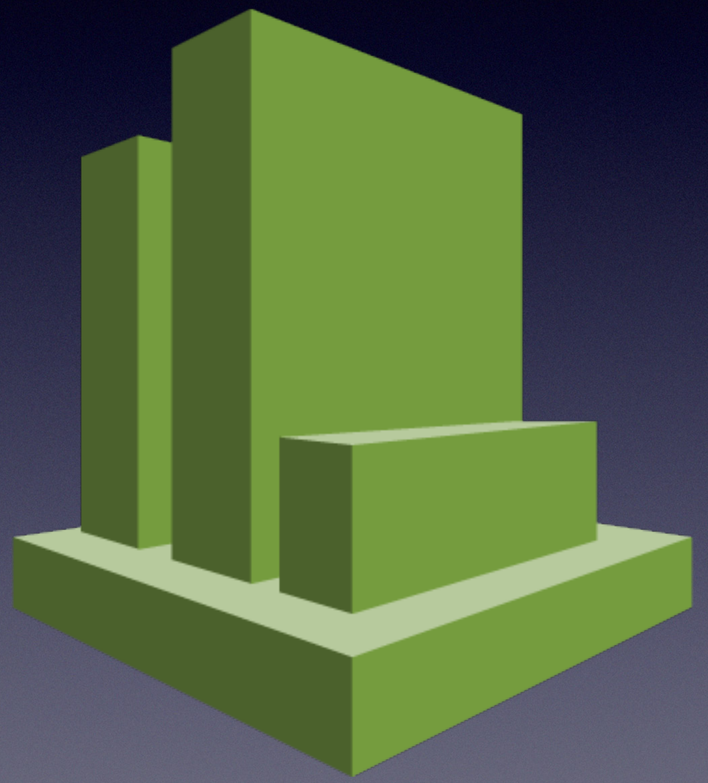
CodeBuild

- CI/CD pipeline
- Run all tests before deployment
- Automatically updates services



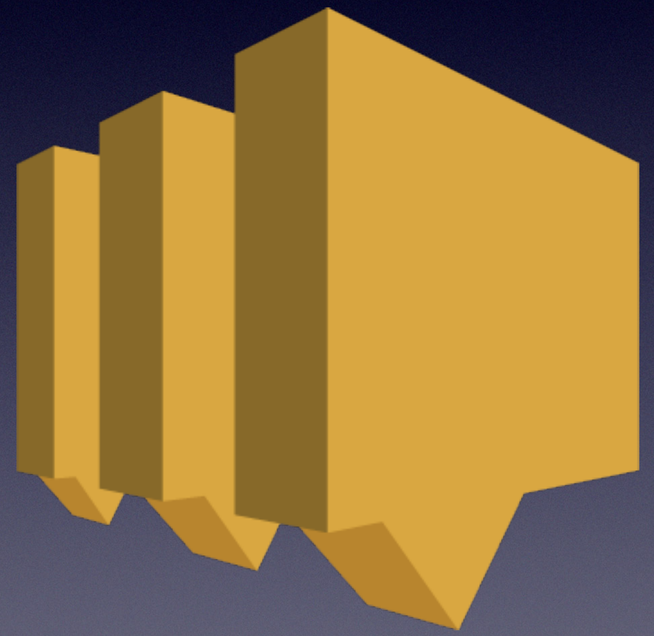
CloudWatch

- Monitor infrastructure 24/7
- Create alarms and dashboards
- CloudWatch Insights
  - Query logs



Simple Notification Service (SNS)

- Event based push notifications
  - Email
  - Text message
- Tied to CloudWatch events



X-Ray

- Debug distributed applications
  - Locate root cause of failure
- Monitor Lambda functions
  - Query like CloudWatch Insights

5 AWS Best Practices

1. TURN YOUR SHIT OFF!

- AWS charges you for *everything*.
- Shut down resources when not in use.

2. Start Small, Then Scale Up

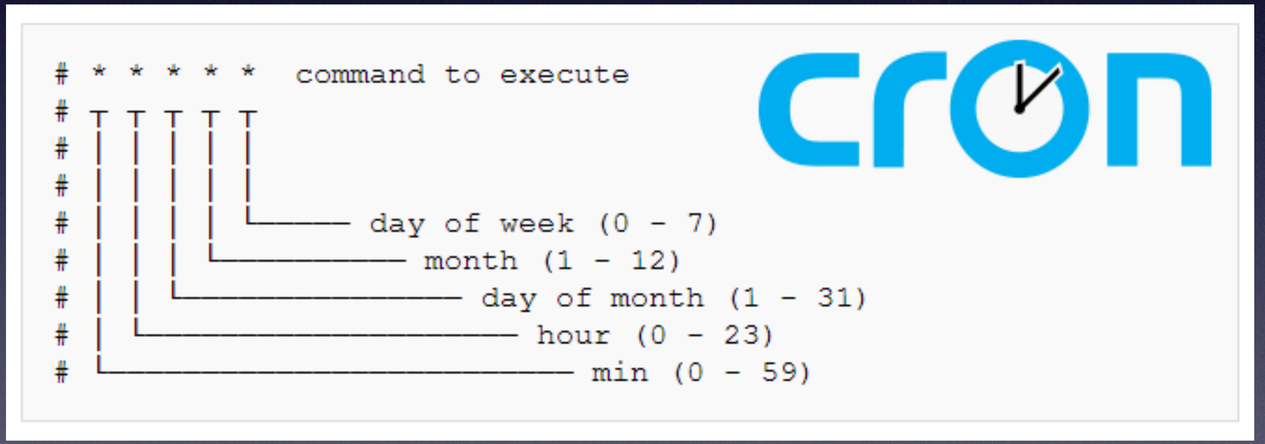
- Develop on a small instance
- Scale up when ready

3. Never, EVER Use `.env` Files

- Unencrypted, and stored per user
- Store keys in Secrets Manager or Parameter Store
  - Single source of truth

4. NEVER Use `cron` In Production

- Cron can't track failures
  - Can't send alerts
- Additional layer of abstraction
- Acceptable for scheduled ECS tasks



5. Be Patient

- AWS is deep
- **Don't avoid learning cloud**