Amazon Web Services

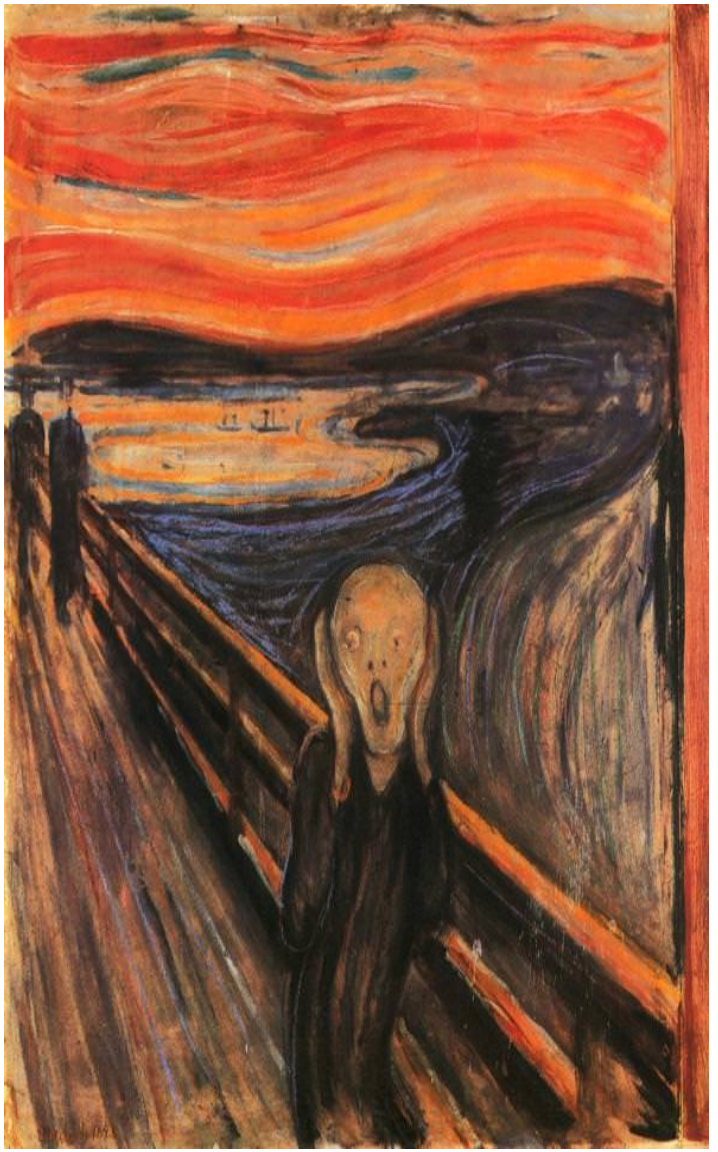
Brian Mason
Netapp

Agenda

- Overview of Services
- Walk through some Use Cases
- Programming to AWS
- Questions

Truly Dizzying Collection of Services

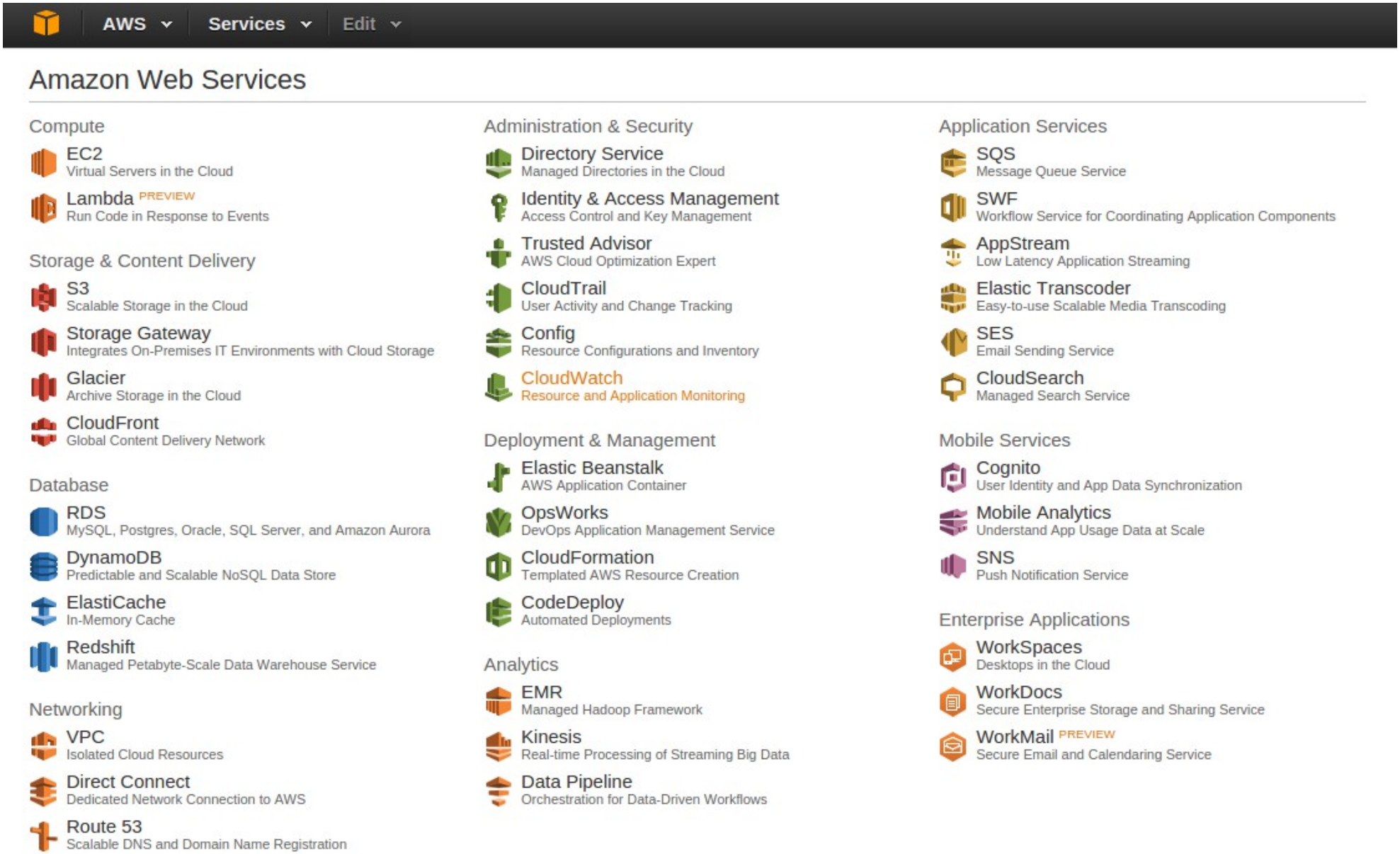
- EC2
- Lambda
- S3
- Storage Gateway
- Glacier
- Cloud Front
- RDS
- Dynamo DB
- ElastiCache
- Redhsift
- VPC
- Direct Donnect
- Route53
- Directory Ser4vices
- Identity Y Access Managment
- Trusted Advisor
- Cloud Trail
- Config



- Cloud Watch
- Elastic Beanstalk
- OpsWork
- Cloud Formation
- Cloud Deploy
- EMR
- Knesis
- Data Pipeline
- SQS
- SWF
- AppStream
- SES
- Cloud Search
- Congnito
- Mobile Analystics
- Workspaces
- WorkDocs
- WorkMail

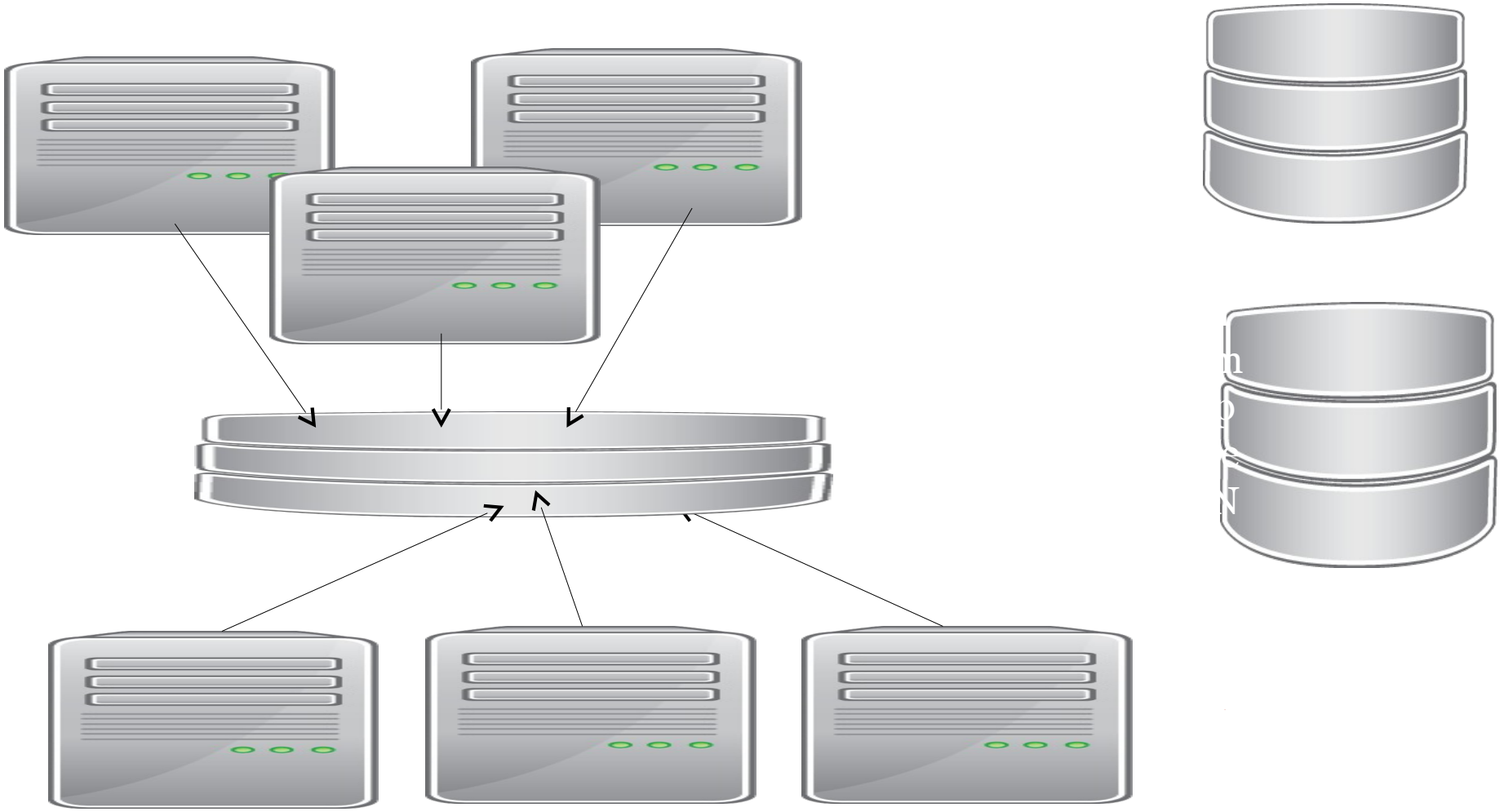
Grouping of Services

- Compute
- Storage
- Database
- Networking
- Administration & Security
- Deployment & Management
- Analytics
- Application Services
- Mobile Services
- Enterprise Application


AWS
Services
Edit
Amazon Web Services
Compute
EC2
Virtual Servers in the Cloud
Lambda PREVIEW
Run Code in Response to Events
Storage & Content Delivery
S3
Scalable Storage in the Cloud
Storage Gateway
Integrates On-Premises IT Environments with Cloud Storage
Glacier
Archive Storage in the Cloud
CloudFront
Global Content Delivery Network
Database
RDS
MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
DynamoDB
Predictable and Scalable NoSQL Data Store
ElastiCache
In-Memory Cache
Redshift
Managed Petabyte-Scale Data Warehouse Service
Networking
VPC
Isolated Cloud Resources
Direct Connect
Dedicated Network Connection to AWS
Route 53
Scalable DNS and Domain Name Registration
Administration & Security
Directory Service
Managed Directories in the Cloud
Identity & Access Management
Access Control and Key Management
Trusted Advisor
AWS Cloud Optimization Expert
CloudTrail
User Activity and Change Tracking
Config
Resource Configurations and Inventory
CloudWatch
Resource and Application Monitoring
Deployment & Management
Elastic Beanstalk
AWS Application Container
OpsWorks
DevOps Application Management Service
CloudFormation
Templated AWS Resource Creation
CodeDeploy
Automated Deployments
Analytics
EMR
Managed Hadoop Framework
Kinesis
Real-time Processing of Streaming Big Data
Data Pipeline
Orchestration for Data-Driven Workflows
Application Services
SQS
Message Queue Service
SWF
Workflow Service for Coordinating Application Components
AppStream
Low Latency Application Streaming
Elastic Transcoder
Easy-to-use Scalable Media Transcoding
SES
Email Sending Service
CloudSearch
Managed Search Service
Mobile Services
Cognito
User Identity and App Data Synchronization
Mobile Analytics
Understand App Usage Data at Scale
SNS
Push Notification Service
Enterprise Applications
WorkSpaces
Desktops in the Cloud
WorkDocs
Secure Enterprise Storage and Sharing Service
WorkMail PREVIEW
Secure Email and Calendaring Service

Use Case 1

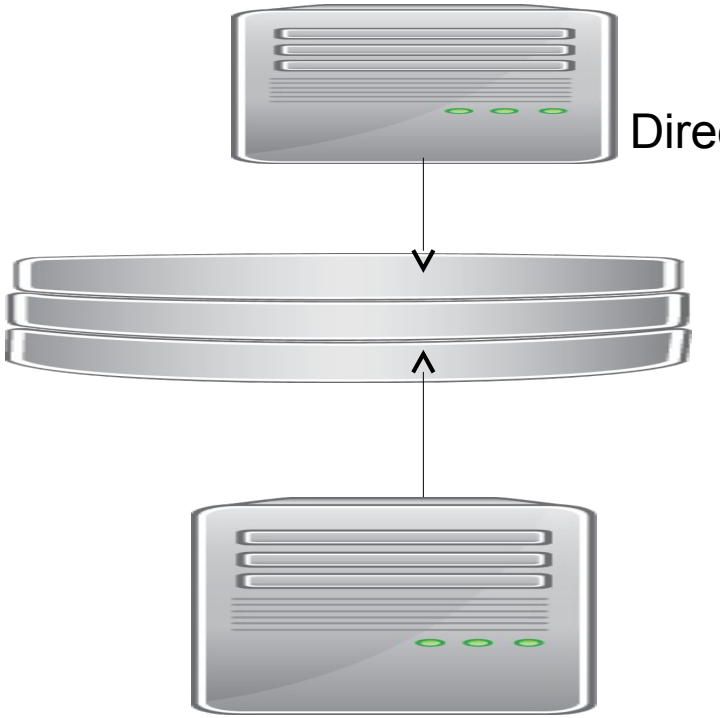
- User wants a scale out service
- Has drank the AWS Kool-aid
- Image Processing Service
- Has following Components
  - Web Front End
  - REST API
  - Processing Nodes
  - Meta Data storage
  - Image Storage



Route 53 Round Robin DNS

Use Case 2

- User wants scale out services
- User Likes AWS
- NO WAY DO WE TRUST IT WITH OUR DATA



AWS
Direct Connect

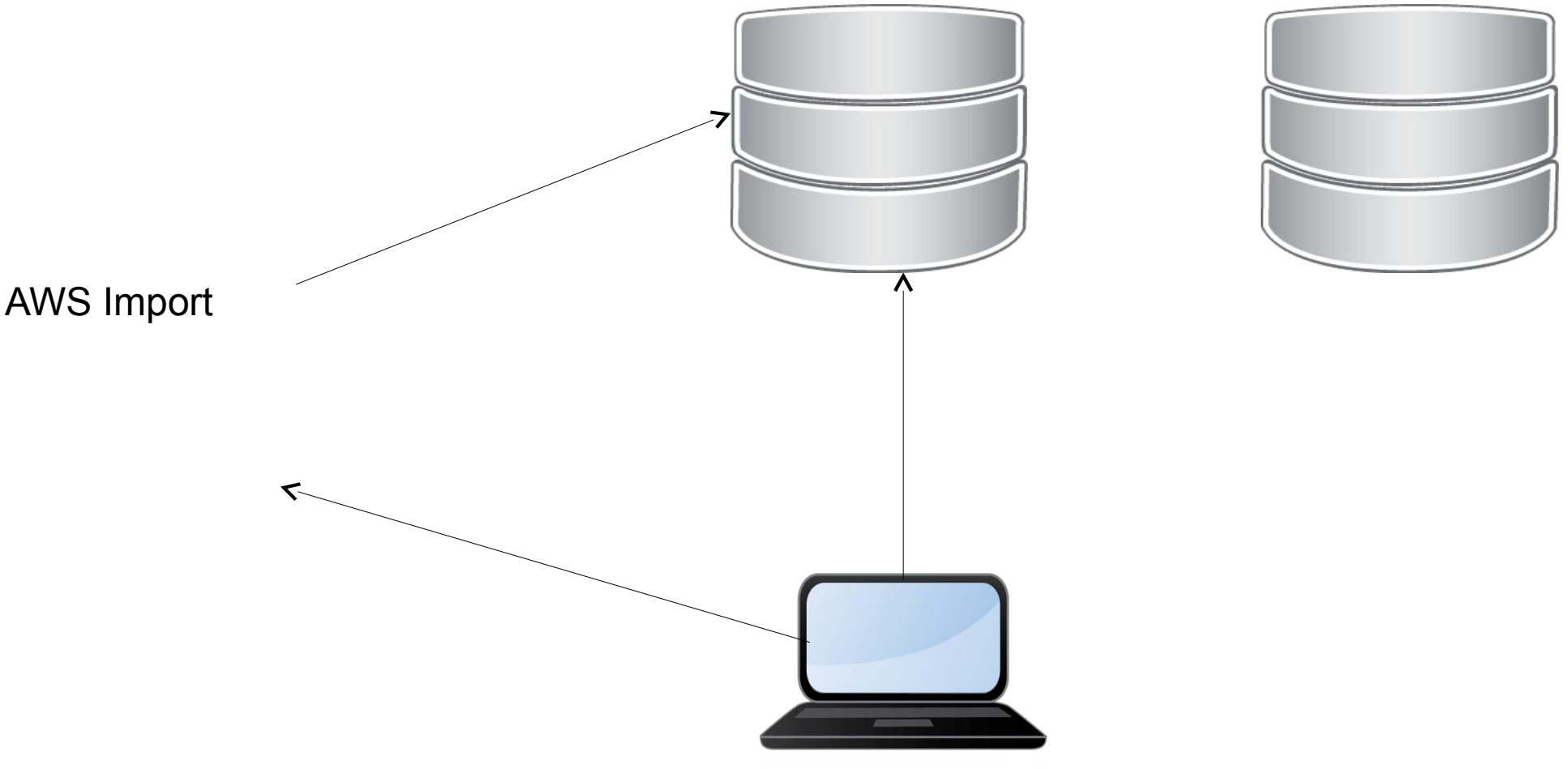
Route 53 Round Robin DNS

Such as
Netapp Storage
(Shameless Company Plug)

Use Case 3

- Archival Data Storage
- S3 Store – Object Store Online
- Glacier – Object Store Offline / cold

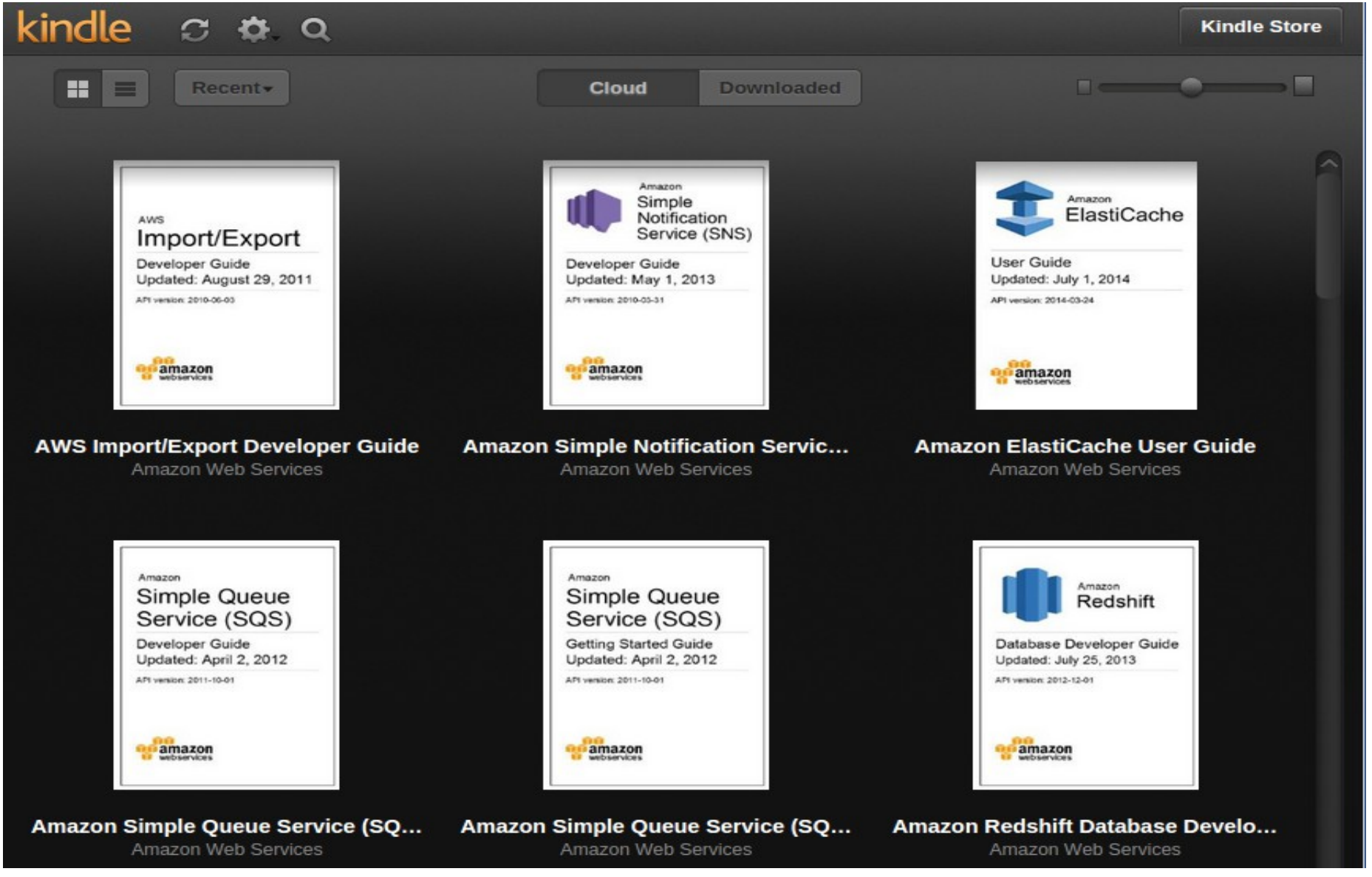
Archival Storage Options



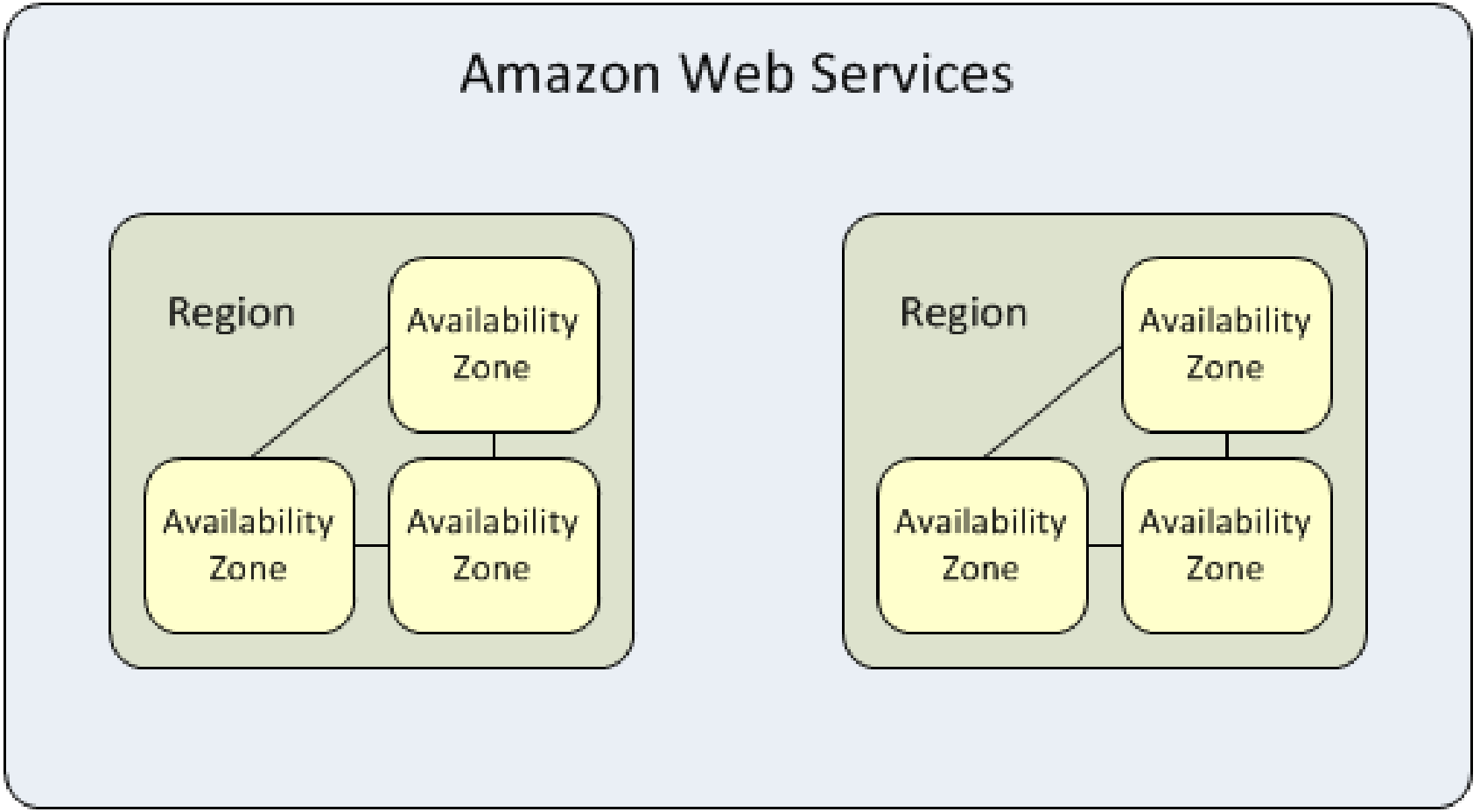
Programming to AWS

- Services expose Web API
  - Most use REST
  - Still a couple lurking SOAP API
- Amazon has SDKs!!
  - Java, Ruby, .niet, PHP and Java script
  - Java sdk is comprehensive and EASY
- Java SDK goes beyond REST API
  - Java Beans <-> DynmoDB for example

Well Documented (Kindle, PDF, HTML, Javadoc…)



Regions and Availability Zones



AWS Regions

- US East -N Virgina (us-east-1)
- US West -Oregon (us-west-2)
- US West - N California (us-west-1)
- EU -Ireland (eu-west-1)
- EU – Frankfurt (eu-central-1)
- Asia Pacific -Singapore (ap-southeast-1)
- Asia Pacific -Tokyo (ap-northeast-1)
- Asia Pacific -Sydney (ap-southeast-2)
- South America - San Paulo (sa-east-1)

Java SDK

- SDK has clients for each service
- They use a common pattern and authentication
- Isolates REST and SOAP APIs from user
- Has some value add above the Web APIs

Code Example

```
ProfileCredentialsProvider pcp;

pcp=new ProfileCredentialsProvider();

AWSCredentials credentials =
  pcp.getCredentials();

AmazonDynamoDBClient client;

client = new AmazonDynamoDBClient(pcp);

client.setRegion(Region.getRegion(Regions.US_WEST
  _2))

DynamoDB dynamoDB = new DynamoDB(client);

Table arrayTable = dynamoDB.getTable("arrays");
```

Q&A?