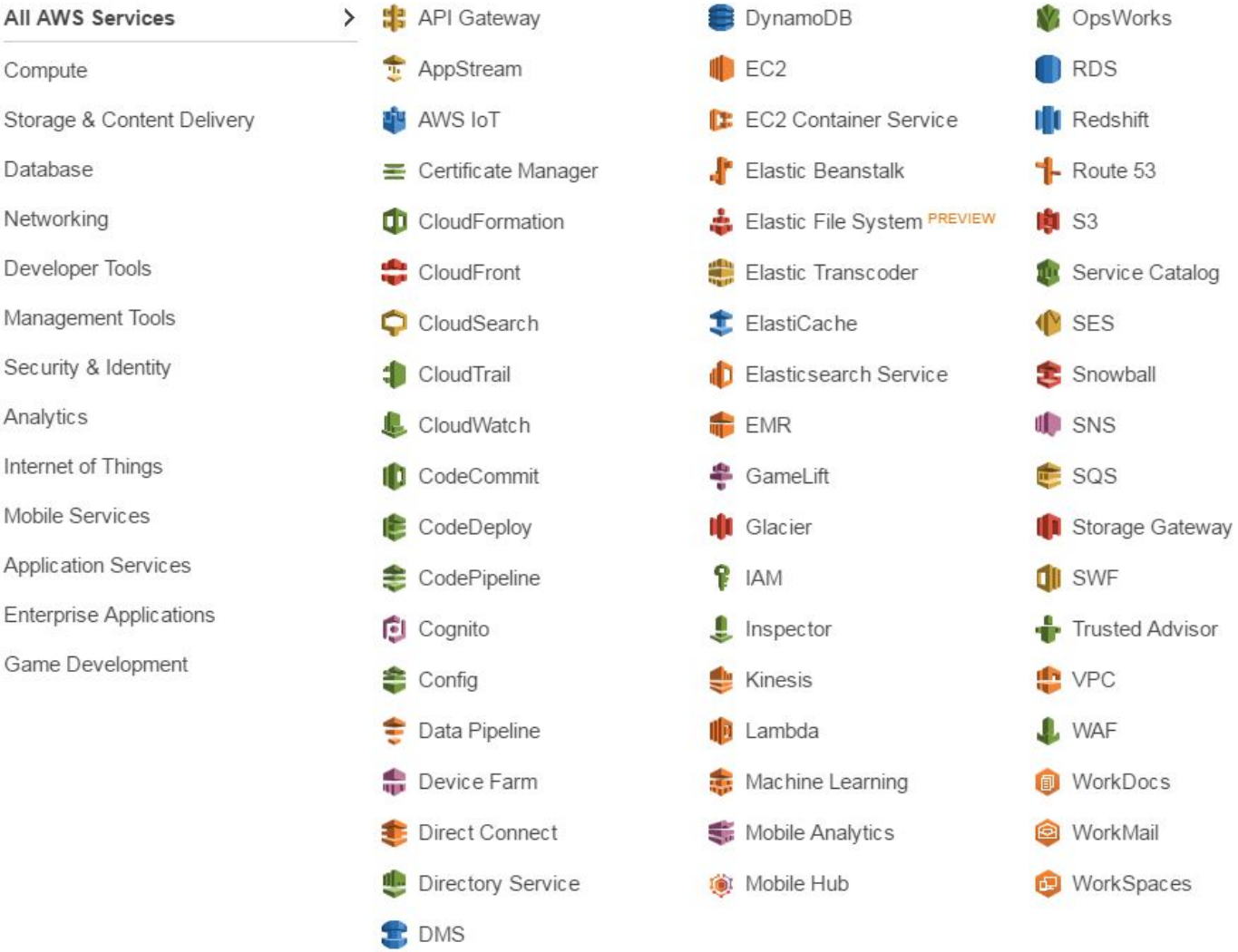
Amazon Web Services

Blazing the Cloud Trail

Andy Cowell has over twenty years of system administration experience in a variety of environments. Currently, he works for Scripps Networks as the manager of the Application Engineering and Automation group, which is responsible for a large number and wide variety of environments, from legacy Windows apps to DevOps-style Linux web deployments, both on-prem and in the cloud. He has been responsible for some of the top destinations on the Internet, such as HGTV.com, Dilbert.com, and Metallica.com. In the past, he has worn many hats, from networking admin to full-time developer, and cut his teeth as a "wizard" with the UTK Computer Science department.

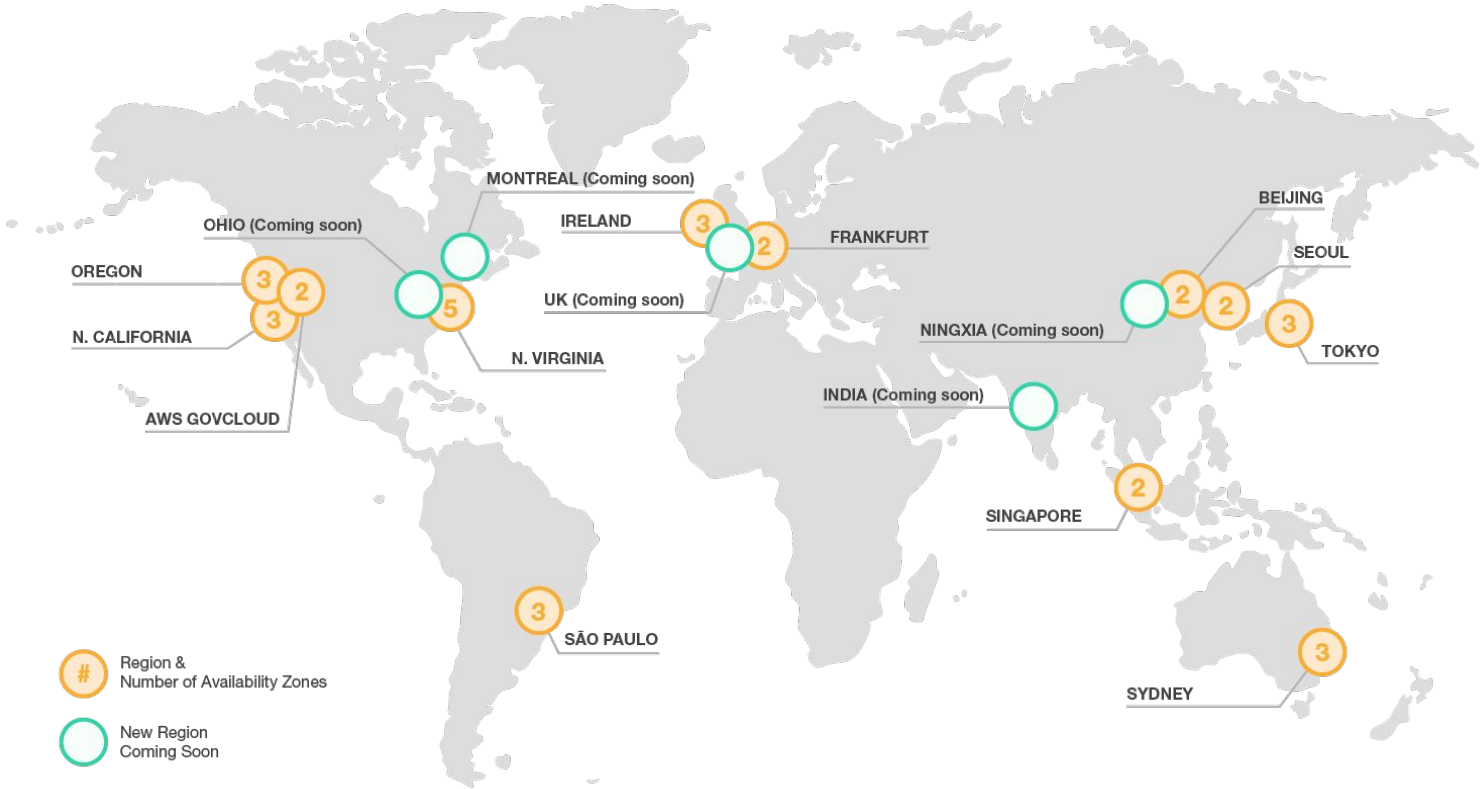
When not wrangling systems, he solders microcontrollers, paints toy soldiers, drinks bourbon, smokes pipes, watches hockey, and rolls 20s.

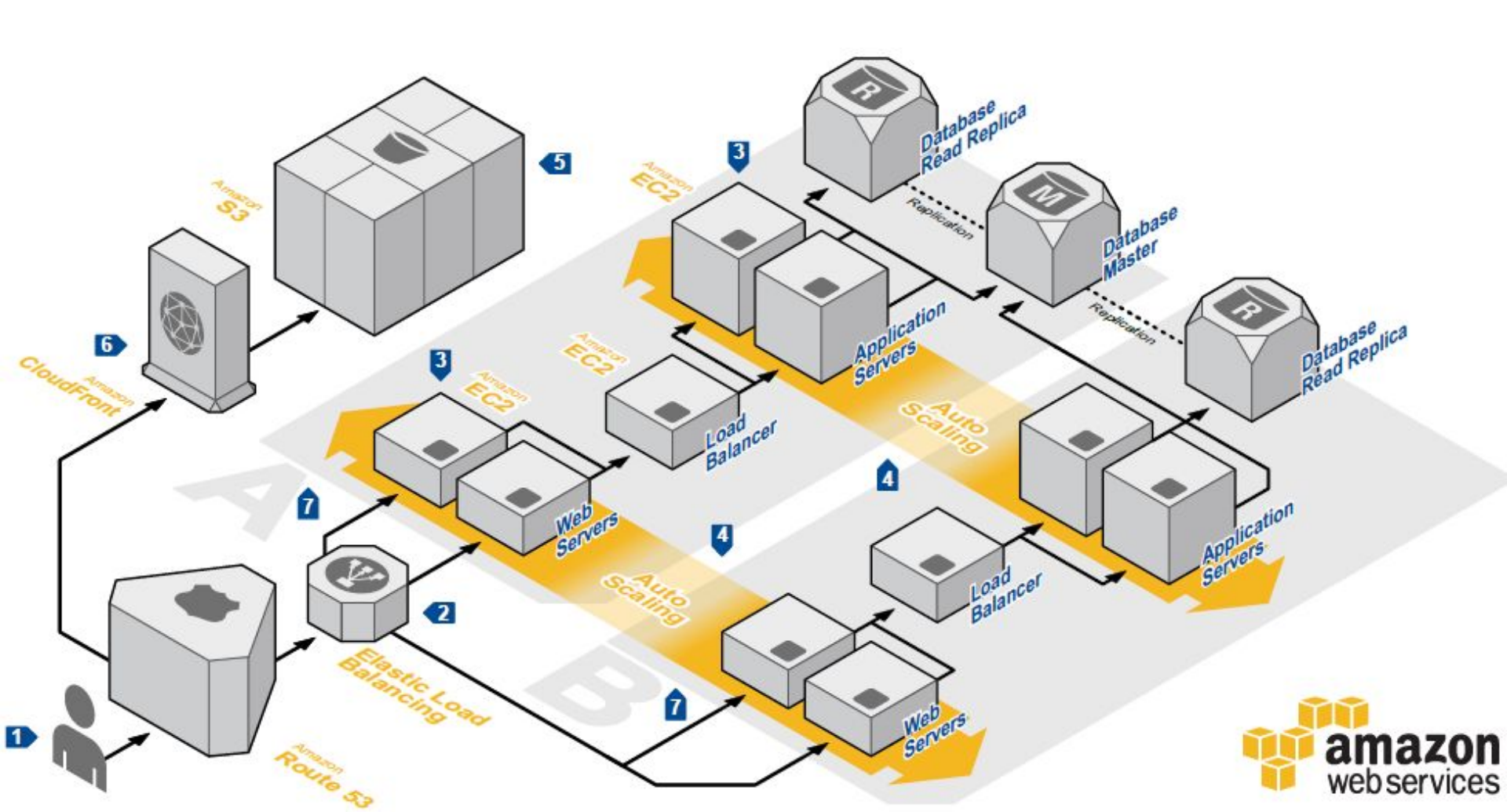
Overview of AWS


All AWS Services
Compute
Storage & Content Delivery
Database
Networking
Developer Tools
Management Tools
Security & Identity
Analytics
Internet of Things
Mobile Services
Application Services
Enterprise Applications
Game Development
API Gateway
AppStream
AWS IoT
Certificate Manager
CloudFormation
CloudFront
CloudSearch
CloudTrail
CloudWatch
CodeCommit
CodeDeploy
CodePipeline
Cognito
Config
Data Pipeline
Device Farm
Direct Connect
Directory Service
DMS
DynamoDB
EC2
EC2 Container Service
Elastic Beanstalk
Elastic File System PREVIEW
Elastic Transcoder
ElastiCache
Elasticsearch Service
EMR
GameLift
Glacier
IAM
Inspector
Kinesis
Lambda
Machine Learning
Mobile Analytics
Mobile Hub
OpsWorks
RDS
Redshift
Route 53
S3
Service Catalog
SES
Snowball
SNS
SQS
Storage Gateway
SWF
Trusted Advisor
VPC
WAF
WorkDocs
WorkMail
WorkSpaces

Regions and AZs

The AWS Cloud operates 33 Availability Zones within 12 geographic Regions around the world.




Amazon S3
5
Amazon EC2
3
Database Read Replica
Replication
Database Master
Replication
Database Read Replica
6
Amazon CloudFront
3
Amazon EC2
Load Balancer
Application Servers
Auto Scaling
7
Web Servers
4
Auto Scaling
4
Load Balancer
Application Servers
2
Elastic Load Balancing
7
Web Servers
1
Amazon Route 53
amazon web services

Infrastructure as a Service

Virtualization

- Elastic Cloud Compute (EC2) service
  - Instances
  - Elastic Block Store (EBS)
    - Volumes
    - Snapshots
  - Amazon Machine Images (AMI)
  - Security Groups
  - Elastic Load Balancers (ELB)
    - Auto-Scaling Groups

Storage Options

- Elastic Block Storage (EBS)
  - Block Volumes
    - General Purpose vs. Provisioned IOPs (and a few others)
    - 1GB to 16TB
  - Snapshots
- Simple Storage Service (S3)
  - Object storage
  - Static website hosting
- Glacier (S3 archive)
- Elastic File System (EFS, Preview)
  - File-based storage
  - NFSv4

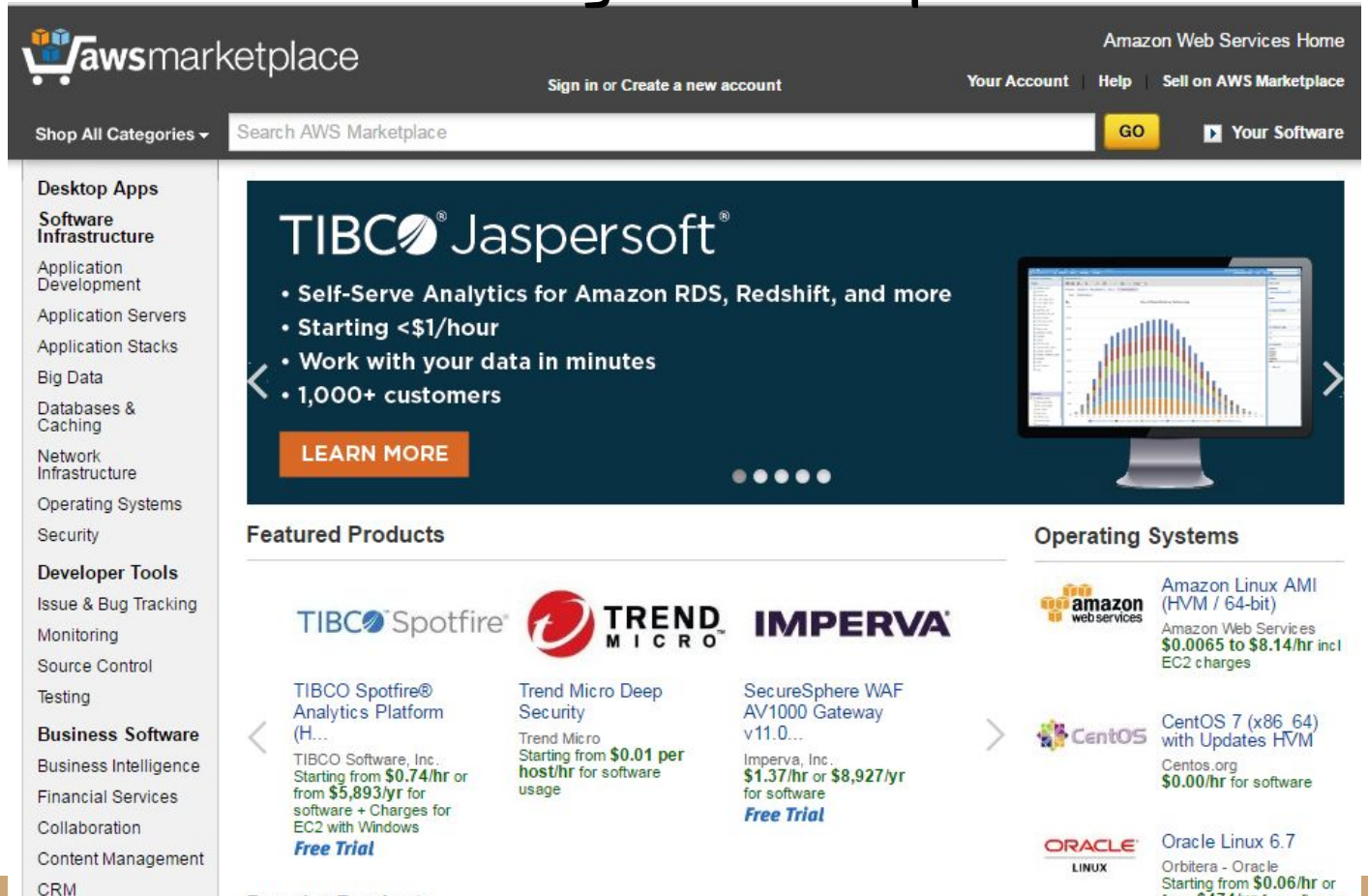
Storage Options

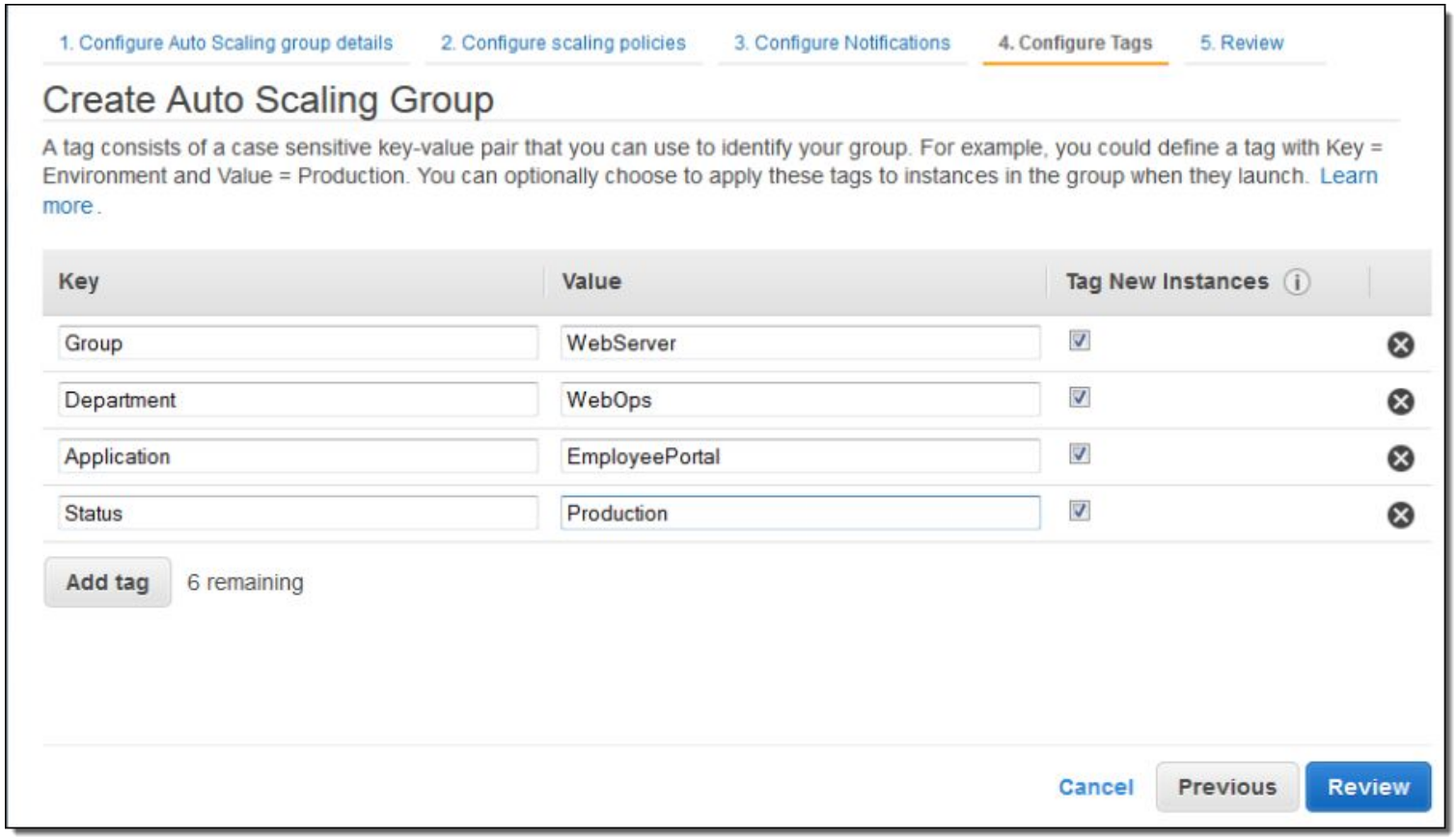
- Storage Gateway
  - On-prem storage appliance backed by the cloud
- Snowball
  - Let UPS haul your data to AWS’s data center

Virtual Private Clouds

- Virtual Private Cloud (VPC)
  - Subnets
    - Route tables
    - Internet gateways
    - Network ACLs
  - Elastic IPs
  - NAT Gateways
  - VPC Peering
  - VPN
- Direct Connect

Amazon Machine Image Marketplace

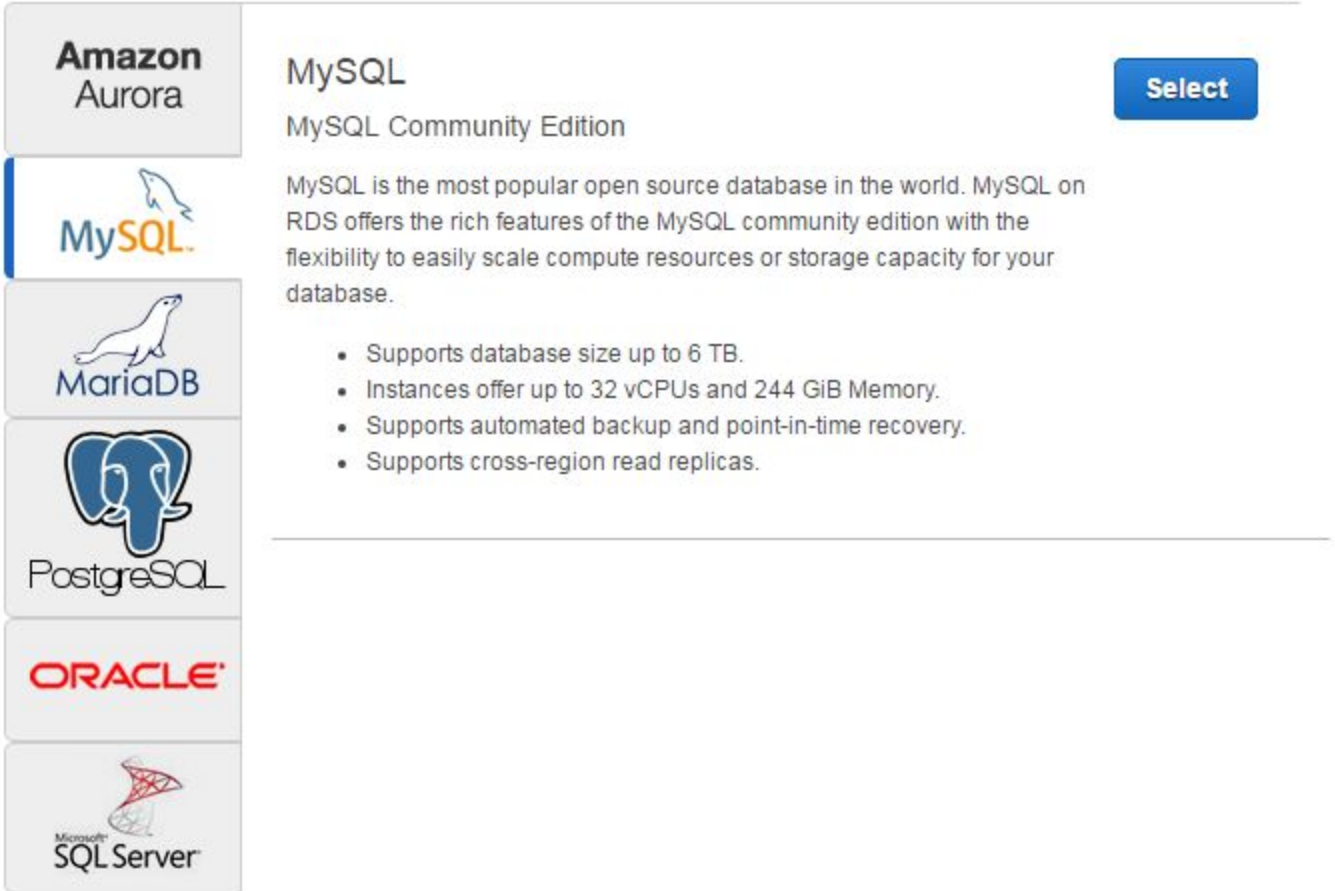



1. Configure Auto Scaling group details
2. Configure scaling policies
3. Configure Notifications
4. Configure Tags
5. Review
Create Auto Scaling Group
A tag consists of a case sensitive key-value pair that you can use to identify your group. For example, you could define a tag with Key = Environment and Value = Production. You can optionally choose to apply these tags to instances in the group when they launch. Learn more.
Key
Value
Tag New Instances
Group
WebServer
Department
WebOps
Application
EmployeePortal
Status
Production
Add tag
6 remaining
Cancel
Previous
Review

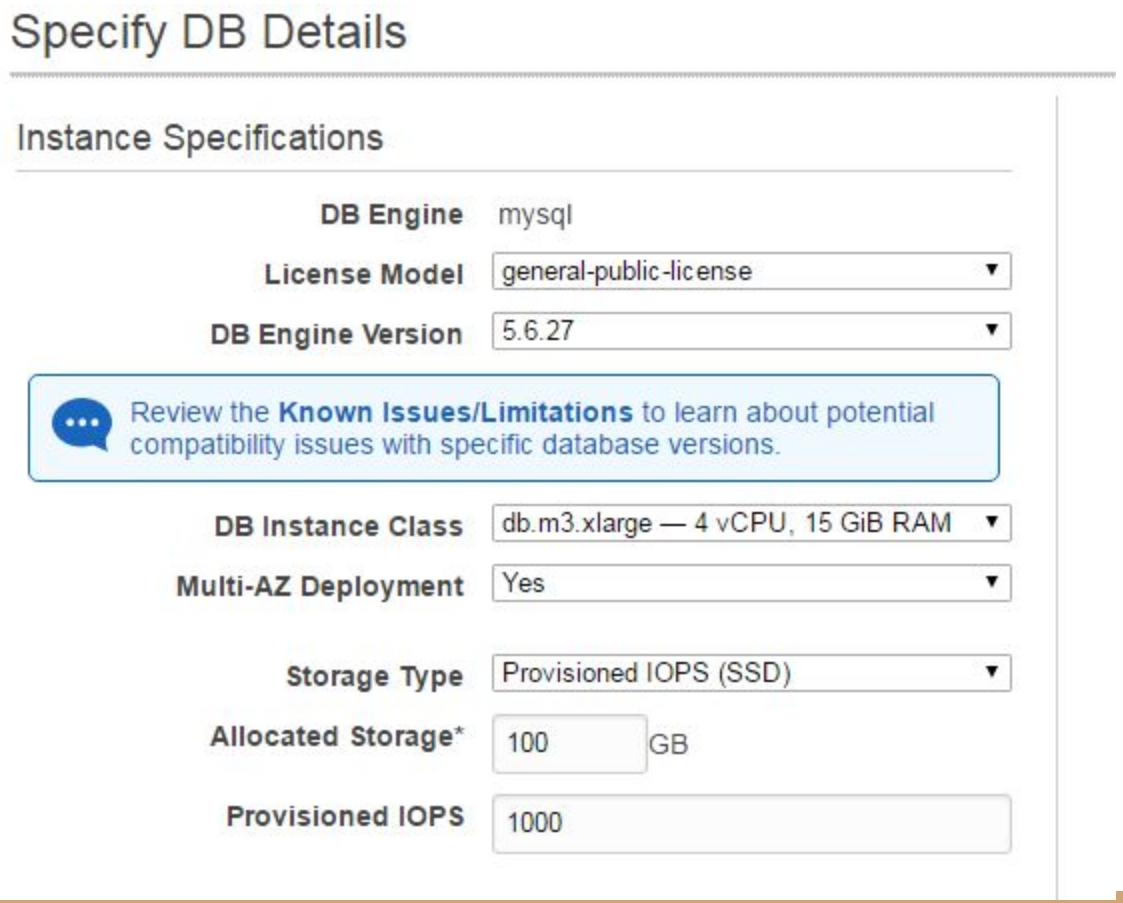
Tagging

Managed Services

Relational Database Service



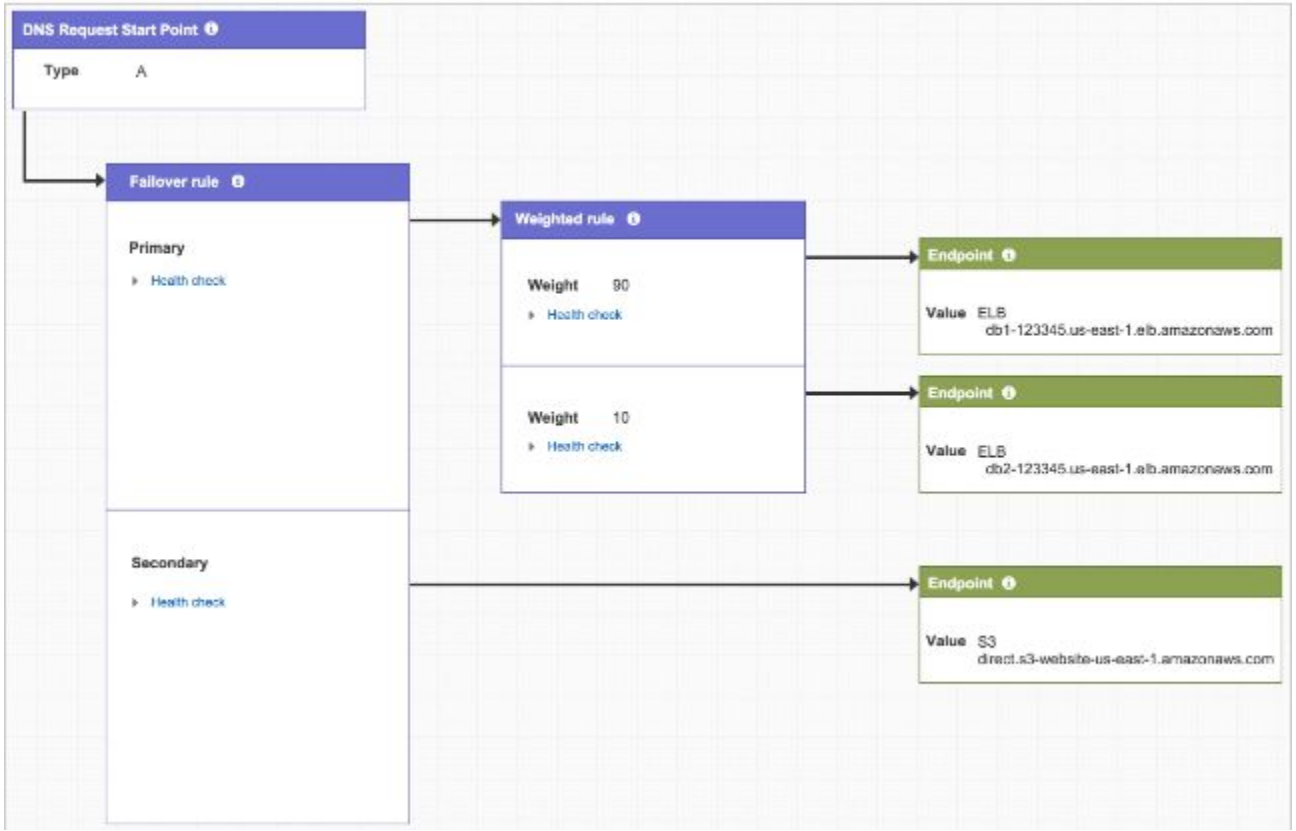
Relational Database Service



DNS - Route 53

- AWS Route 53 is managed DNS
  - Nice Alias feature to dynamic resources inside AWS
  - Traffic policy features

DNS - Route 53


DNS Request Start Point
Type A
Failover rule
Primary
Health check
Secondary
Health check
Weighted rule
Weight 90
Health check
Weight 10
Health check
Endpoint
Value ELB
db1-123345.us-east-1.elb.amazonaws.com
Endpoint
Value ELB
db2-123345.us-east-1.elb.amazonaws.com
Endpoint
Value S3
direct.s3-website-us-east-1.amazonaws.com

Elastic Map Reduce (EMR)

Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

You do not appear to have any clusters. Create one now:

Create cluster

How Elastic MapReduce Works

Upload

Upload your data and processing application to S3.

Create

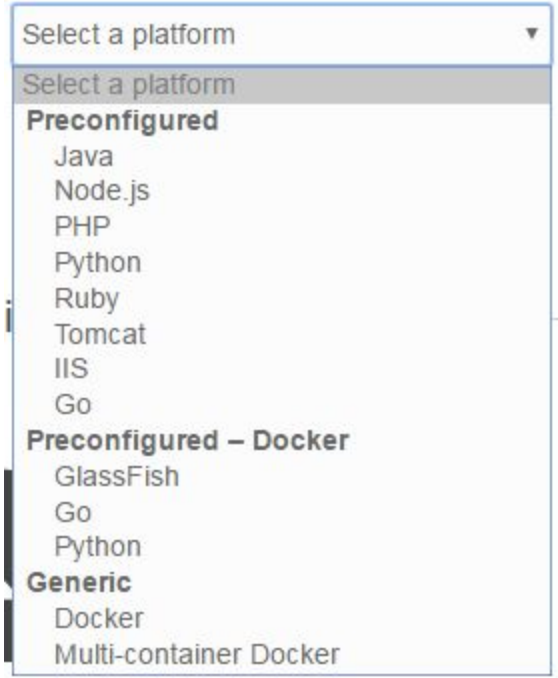
Configure and create your cluster by specifying data inputs, outputs, cluster size, security settings, etc.

Monitor

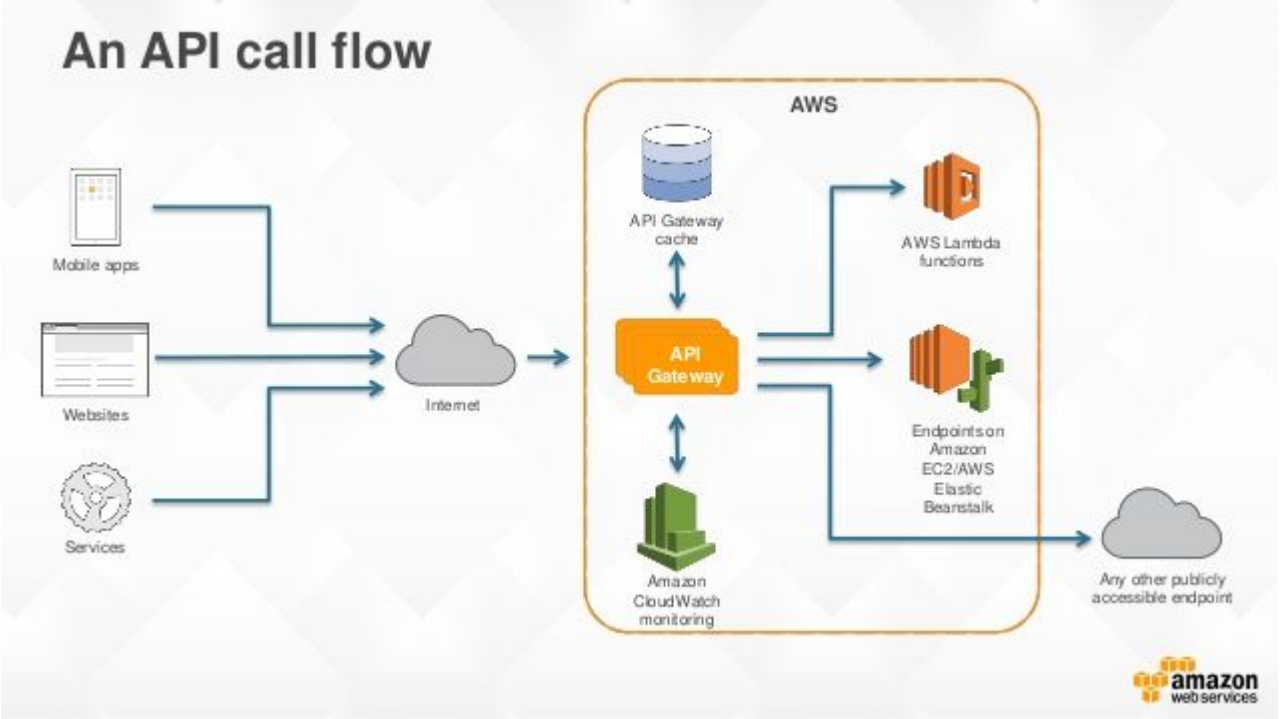
Monitor the health and progress of your cluster. Retrieve the output in S3.

Elastic Beanstalk

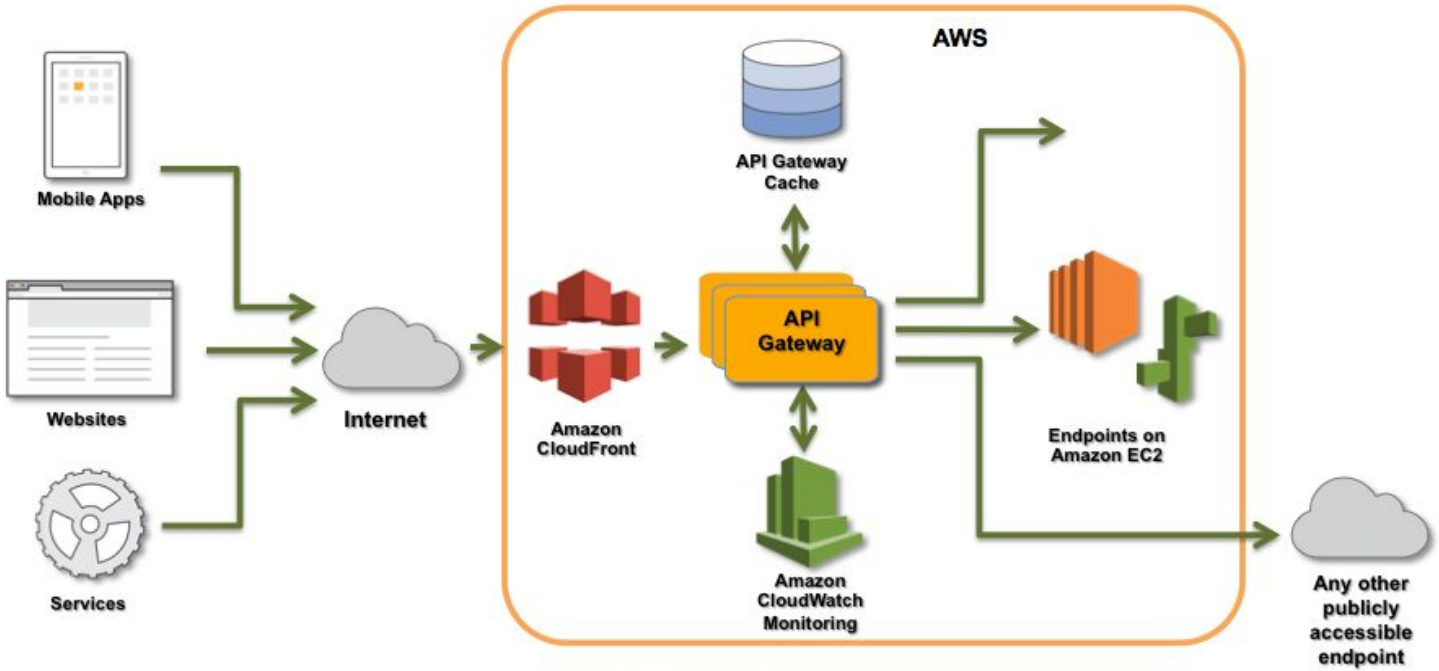
- Platform as a Service



API Gateway

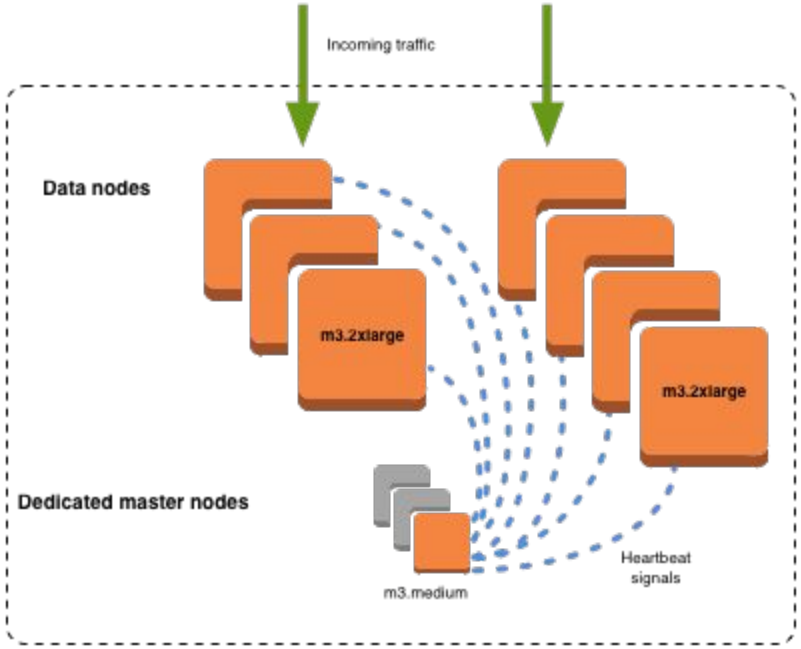

An API call flow
AWS
Mobile apps
Websites
Services
Internet
API Gateway cache
API Gateway
Amazon CloudWatch monitoring
AWS Lambda functions
Endpoints on Amazon EC2/AWS Elastic Beanstalk
Any other publicly accessible endpoint
amazon web services

CloudFront

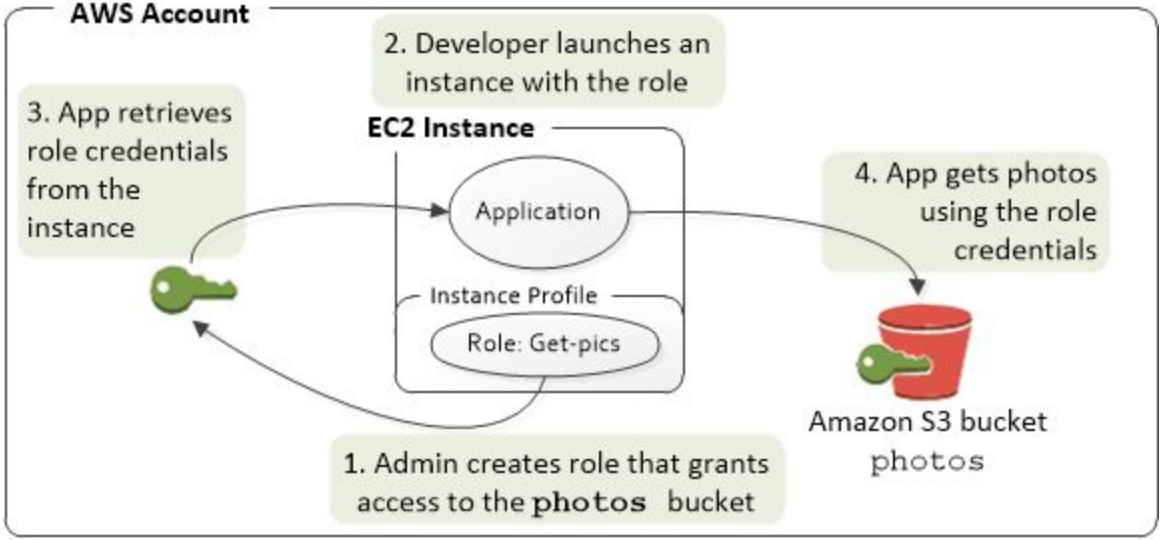


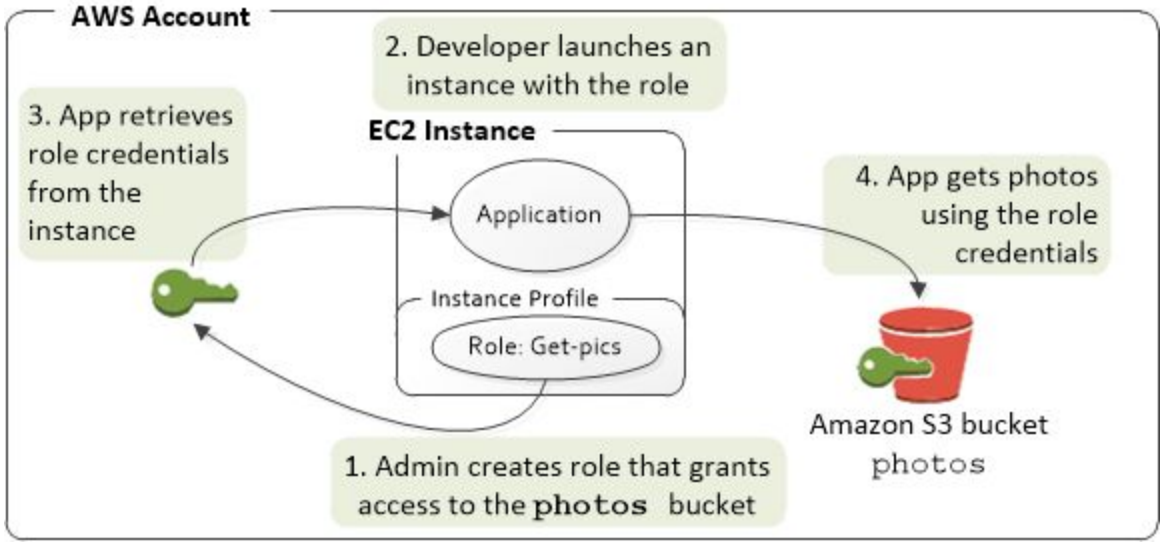
An Amazon API Gateway Call Flow

Elasticsearch



Identity and Access Management (IAM)





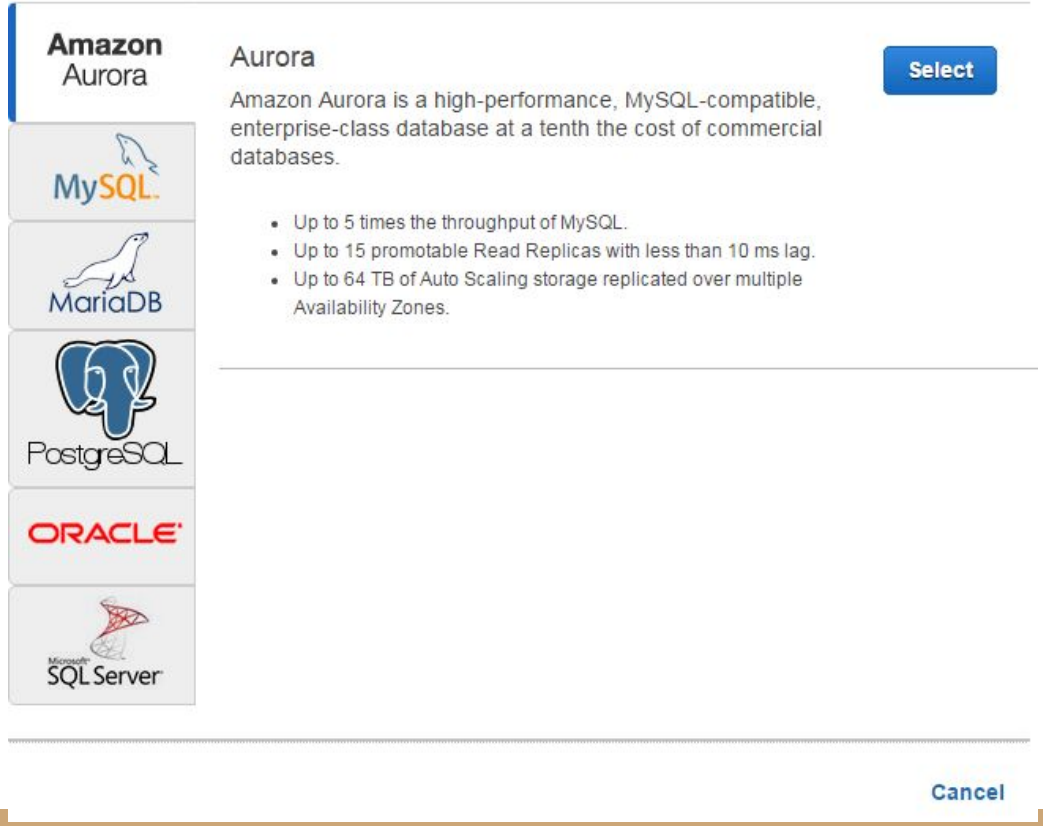
Using roles inside AWS

Utilities

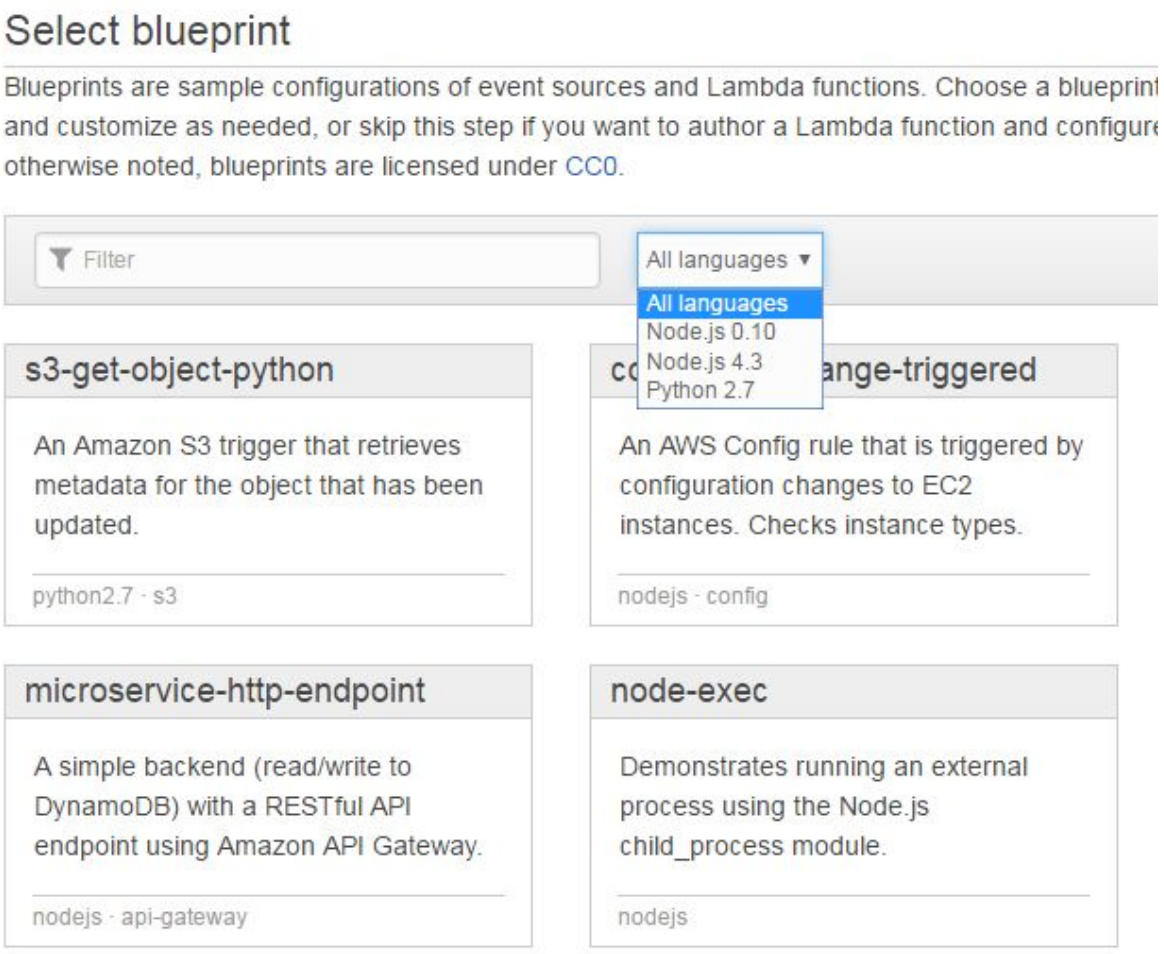
- Simple Queue Service (SQS)
- Simple Notification Service (SNS)
- Simple E-mail Service (SES)
- CloudWatch
- CloudTrail
- Simple Workflow Service (SWS)

Custom Services

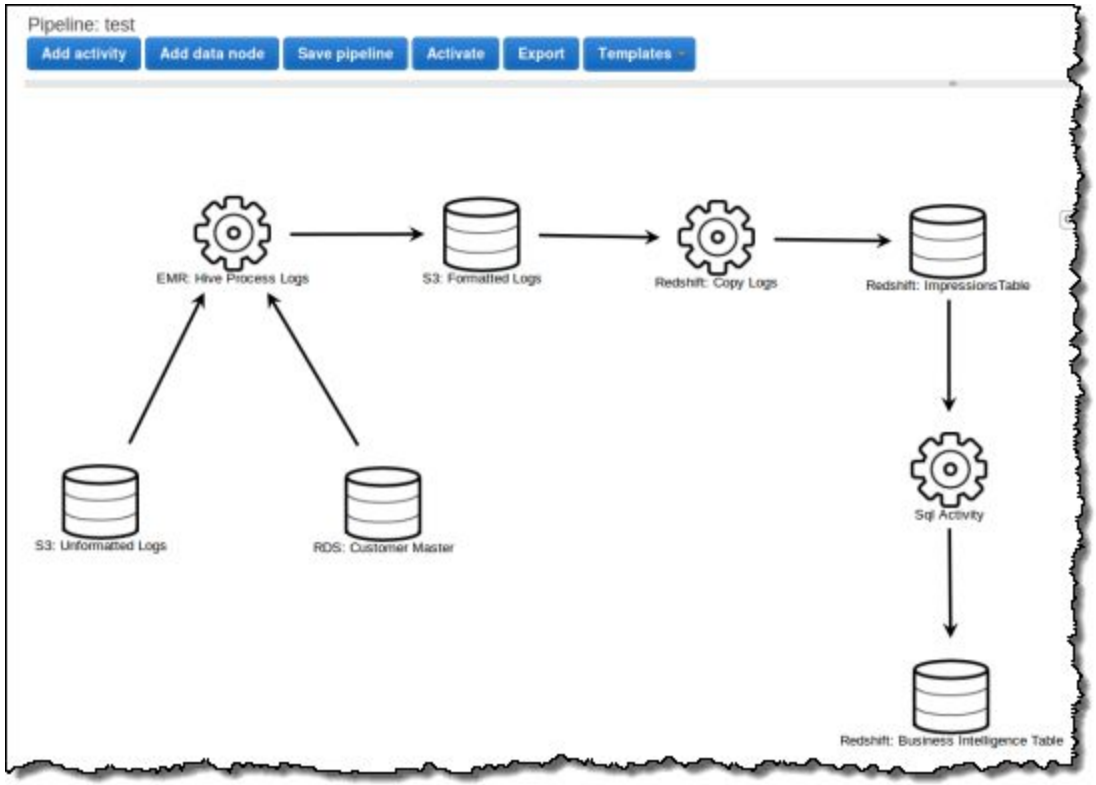
RDS - Aurora


Amazon Aurora
MySQL
MariaDB
PostgreSQL
ORACLE
Microsoft SQL Server
Aurora
Amazon Aurora is a high-performance, MySQL-compatible, enterprise-class database at a tenth the cost of commercial databases.
Up to 5 times the throughput of MySQL.
Up to 15 promotable Read Replicas with less than 10 ms lag.
Up to 64 TB of Auto Scaling storage replicated over multiple Availability Zones.
Select
Cancel

Lambda


Select blueprint
Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint
and customize as needed, or skip this step if you want to author a Lambda function and configure
otherwise noted, blueprints are licensed under CC0.
Filter
All languages
All languages
Node.js 0.10
Node.js 4.3
Python 2.7
s3-get-object-python
An Amazon S3 trigger that retrieves metadata for the object that has been updated.
python2.7 · s3
ange-triggered
An AWS Config rule that is triggered by configuration changes to EC2 instances. Checks instance types.
nodejs · config
microservice-http-endpoint
A simple backend (read/write to DynamoDB) with a RESTful API endpoint using Amazon API Gateway.
nodejs · api-gateway
node-exec
Demonstrates running an external process using the Node.js child_process module.
nodejs

Data Pipeline



Kinesis

- Real-time data stream analytics

Redshift

- Data warehousing

CLI and API

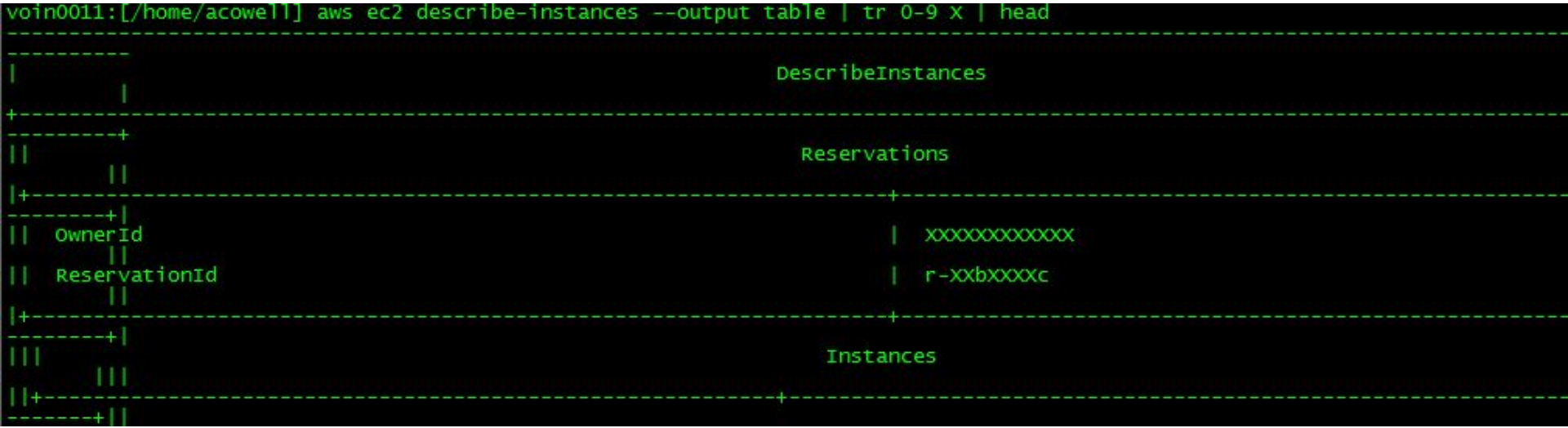
```
voin0011:[/home/acowell] aws help
AWS()                                                                   AWS()



NAME
       aws -

DESCRIPTION
       The  AWS  Command  Line  Interface is a unified tool to manage your AWS
       services.

SYNOPSIS
            aws [options] <command> <subcommand> [parameters]

       Use aws command help for information on a specific command.

OPTIONS
       --debug (boolean)

       Turn on debug logging.

       --endpoint-url (string)

       Override command's default URL with the given URL.

       --no-verify-ssl (boolean)

       Override default behavior of verifying SSL certificates.

       --no-paginate (boolean)

       Disable automatic pagination.

       --output (string)

       The formatting style for command output.

       o json

       o text

       o table

       --query (string)

       A JMESPath query to use in filtering the response data.
```

```
voin0011:[/home/acowell] cat ~/.aws/config | tr A-Z0-9 X
[default]
region = us-east-1
# region = eu-west-1
aws_access_key_id = XXXXXXXXXXXXXXXXXXX
aws_secret_access_key = tXqhX+XXkXXXXXXXXmXXrXgX/pXpgXXXpXXXalXX

[profile apac]
region = ap-southeast-1

[profile cs]
region = us-east-1
voin0011:[/home/acowell] cat ~/.aws/credentials | tr A-Z0-9 X
[default]
# region = us-west-X
# region = eu-west-1
# region = eu-east-1
aws_access_key_id = XXXXXXXXXXXXXXXXXXX
aws_secret_access_key = tXqhX+XXkXXXXXXXXmXXrXgX/pXpgXXXpXXXalXX

[apac]
aws_access_key_id = XXXXXXXXXXXXXXXXXXX
aws_secret_access_key = XXgXn/XXcXbgsX/dXXnqXXyXXizXXq/XXXdXm/+X

[cs]
aws_access_key_id = XXXXXXXXXXXXXXXXXXX
aws_secret_access_key = spXXXXntXXXuXkXXXXbXoyoqoeXXlXXswexkXvXX
```


AVAILABLE SERVICES
o autoscaling
o cloudformation
o cloudsearch
o cloudsearchdomain
o cloudtrail
o cloudwatch
o cognito-identity
o cognito-sync
o configure
o datapipeline
o directconnect
o dynamodb
o ec2
o elasticache
o elasticbeanstalk
o elastictranscoder
o elb
o help
o iam

```
voin0011:[/home/acowell] aws ec2 help
EC2()                                                                EC2()



NAME
       ec2 -

DESCRIPTION
       Amazon  Elastic Compute Cloud (Amazon EC2) provides resizable computing
       capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 elim-
       inates your need to invest in hardware up front, so you can develop and
       deploy applications faster.

AVAILABLE COMMANDS
       o accept-vpc-peering-connection

       o allocate-address

       o assign-private-ip-addresses

       o associate-address

       o associate-dhcp-options

       o associate-route-table

       o attach-internet-gateway

       o attach-network-interface

       o attach-volume

       o attach-vpn-gateway

       o authorize-security-group-egress

       o authorize-security-group-ingress

       o bundle-instance

       o cancel-bundle-task

       o cancel-conversion-task
```

```
voin0011:[/home/acowell] aws ec2 describe-instances | tr 0-9 x
{
    "Reservations": [
        {
            "OwnerId": "xxxxxxxxxxxx",
            "ReservationId": "r-xxbxxxxc",
            "Groups": [],
            "Instances": [
                {
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "PublicDnsName": null,
                    "State": {
                        "Code": xx,
                        "Name": "running"
                    },
                    "EbsOptimized": false,
                    "LaunchTime": "xxxx-xx-xxTxx:xx:xx.xxxZ",
                    "PublicIpAddress": "xx.xxx.xx.xxx",
                    "PrivateIpAddress": "xx.xx.x.xxx",
                    "ProductCodes": [],
                    "VpcId": "vpc-xxxfxxxe",
                    "StateTransitionReason": null,
                    "InstanceId": "i-xfxxxxdx",
                    "ImageId": "ami-fxfxxxxx",
                    "PrivateDnsName": "ip-xx-xx-x-xxx.ecx.internal",
                    "KeyName": "ns-nVA",
                    "SecurityGroups": [
                        {
                            "GroupName": "JenkinsMasterTesting",
                            "GroupId": "sg-dxxxxdae"
                        }
                    ],
                    "ClientToken": "oHzhsxxxxxxxxxxxxx",
                    "SubnetId": "subnet-axxxxcdx",
                    "InstanceType": "tx.small",
                    "NetworkInterfaces": [
                        {
                            "Status": "in-use",
                            "SourceDestCheck": true,
                            "VpcId": "vpc-xxxfxxxe",
                            "Description": "Primary network interface",
                            "Association": {
                                "PublicIp": "xx.xxx.xx.xxx",
                                "PublicDnsName": null,
                                "IpOwnerId": "amazon"
```


voin0011:[/home/acowell] aws ec2 describe-instances --output table | tr 0-9 X | head
DescribeInstances
Reservations
OwnerId
XXXXXXXXXXX
ReservationId
r-XXbXXXXc
Instances

```
voin0011:[/home/acowell] aws ec2 describe-instances --output text | tr 0-9 X | head
RESERVATIONS    XXXXXXXXXXXX    r-XXbXXXXc
INSTANCES       X       xxx_xx  oHzhsXXXXXXXXXXXXXX      False   xen     ami-fxfxxxxx    i-xfxxxxdx      tx.small
XX-XX-XXTXX:XX:XX.XXXZ  ip-XX-XX-X-XXX.ecX.internal     XX.XX.X.XXX     None    XX.XXX.XX.XXX   /dev/xvda       ebs
ne      subnet-aXXXXcdx hvm     vpc-XXXfXXXe
BLOCKDEVICEMAPPINGS     /dev/xvda
EBS     XXXX-XX-XXTXX:XX:XX.XXXZ        True    attached        vol-cXXXXdXX
IAMINSTANCEPROFILE      arn:aws:iam::XXXXXXXXXXX:instance-profile/ns-testing-sX-read   AIPAJZRXXJFWMJEXXPXYO
MONITORING      disabled
NETWORKINTERFACES       Primary network interface       eni-dXdXXXXX    XXXXXXXXXXX     XX.XX.X.XXX     True    in-use
XXcdX   vpc-XXXfXXXe
ASSOCIATION     amazon  None    XX.XXX.XX.XXX
ATTACHMENT      XXXX-XX-XXTXX:XX:XX.XXXZ        eni-attach-fxfxcXXX     True    X       attached
GROUPS  sg-dXXXXdae     JenkinsMasterTesting
```

```
voin0011:[/home/acowell] aws ec2 describe-instances --output text | egrep INSTANCES | while read i ; do \
>    echo $i | awk '{print $8":", $9}' | tr 0135-9 X ; done
i-xfxxxxdx: t2.small
i-xbbxaxex: t2.medium
i-exfxcxx4: t2.large
i-4xxxcxcx: t2.micro
i-xxxx4bxx: t2.micro
i-xcacb4ex: t2.micro
i-dxxbbx4d: t2.medium
i-xx4xxxfx: t2.micro
i-xxxxefae: t2.medium
i-fx4xxex2: m4.2xlarge
i-fbdacxxc: t2.micro
i-xxaxxbxx: t2.small
i-2exx4bax: t2.medium
i-fddcacx4: t2.micro
```

boto3

Boto 3 - The AWS SDK for Python

Boto3 is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python, which allows Python developers to write software that makes use of services like Amazon S3 and Amazon EC2. You can find the latest, most up to date, documentation at Read the Docs, including a list of services that are supported.

Quick Start

First, install the library and set a default region:

```
$ pip install boto3
```

Restore Glacier objects in an Amazon S3 bucket

The following example shows how to initiate restoration of glacier objects in an Amazon S3 bucket, determine if a restoration is on-going, and determine if a restoration is finished.

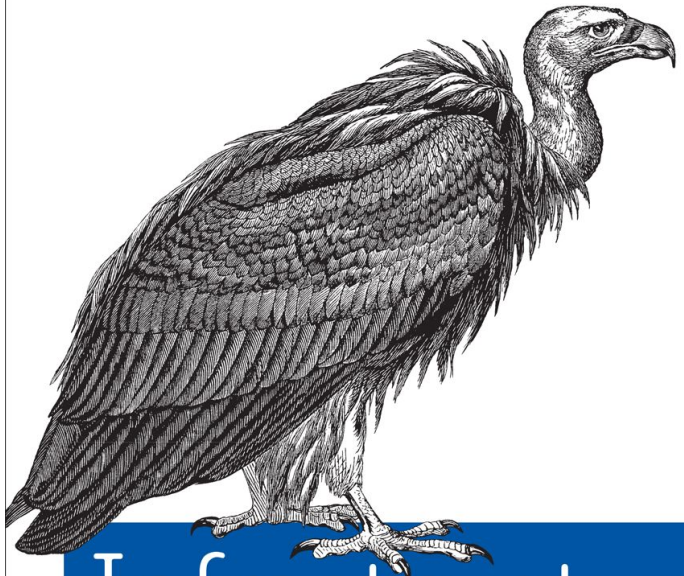
```
import boto3

s3 = boto3.resource('s3')
bucket = s3.Bucket('glacier-bucket')
for obj_sum in bucket.objects.all():
    obj = s3.Object(obj_sum.bucket_name, obj_sum.key)
    if obj.storage_class == 'GLACIER':
        # Try to restore the object if the storage class is glacier and
        # the object does not have a completed or ongoing restoration
        # request.
        if obj.restore is None:
            print('Submitting restoration request: %s' % obj.key)
            obj.restore_object()
        # Print out objects whose restoration is on-going
        elif 'ongoing-request="true"' in obj.restore:
            print('Restoration in-progress: %s' % obj.key)
        # Print out objects whose restoration is complete
        elif 'ongoing-request="false"' in obj.restore:
            print('Restoration complete: %s' % obj.key)
```

Infrastructure as Code

CloudFormation

https://s3.amazonaws.com/cloudformation-templates-us-east-1/WordPress_Single_Instance.template

O'REILLY®

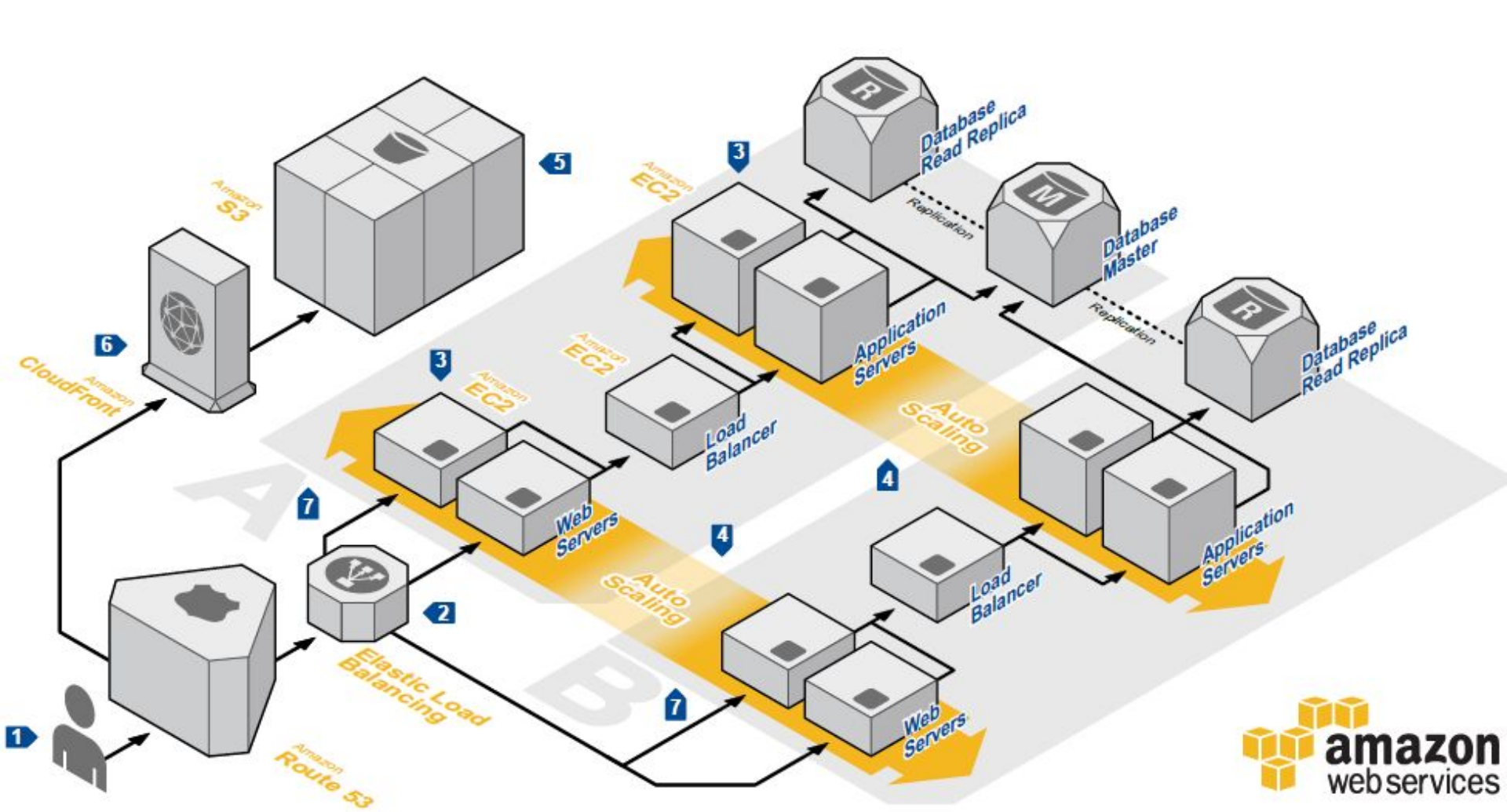


Infrastructure as Code

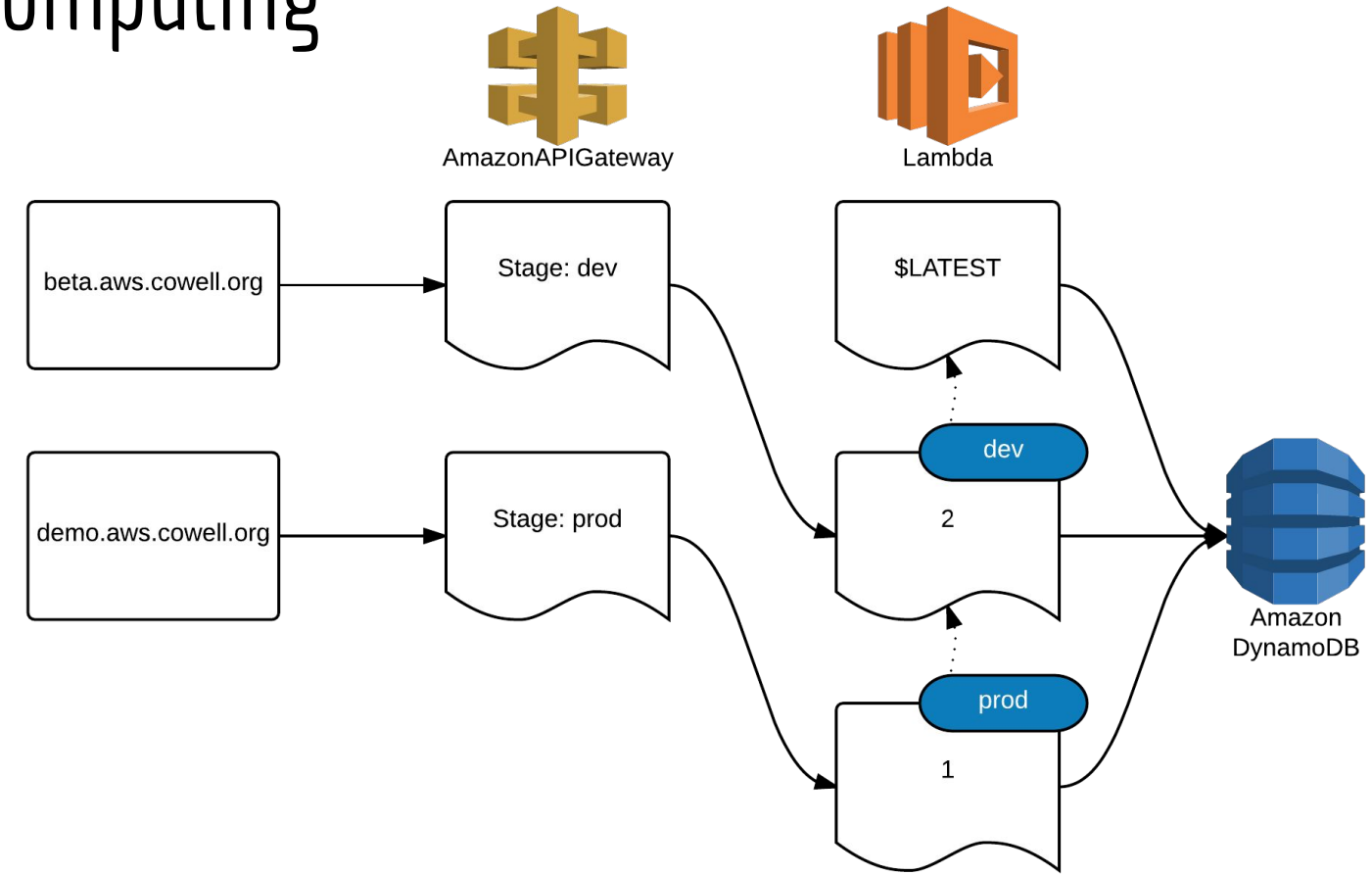
MANAGING SERVERS IN THE CLOUD

Kief Morris

AWS as a Platform


Amazon S3
5
Amazon EC2
3
Database Read Replica
Replication
Database Master
Replication
Database Read Replica
6
Amazon CloudFront
3
Amazon EC2
Load Balancer
Application Servers
Auto Scaling
7
Web Servers
4
Auto Scaling
4
Load Balancer
Application Servers
2
Elastic Load Balancing
7
Web Servers
1
Amazon Route 53
amazon web services

Serverless Computing




SERVERLESS
FRAMEWORK
THE SERVERLESS APPLICATION FRAMEWORK
SERVERLESS.COM
POWERED BY
amazon
web services™

The End