Cloud Programming: Lecture2 – Cloud Services & AWS Introduction

National Tsing-Hua University

2015, Spring Semester

Outline

- Introduction of Cloud Services
- Introduction of Amazon AWS
- AWS Global Infrastructure
- AWS Tools for Accessing Services
- AWS Services
- AWS Use Cases

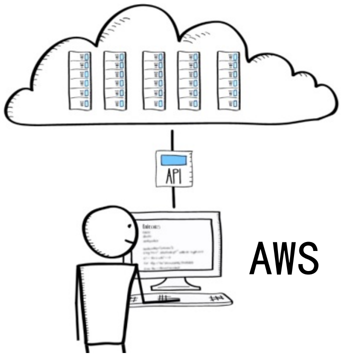
What is Cloud Service?

- **Services** made available to users **on demand via the Internet from a cloud computing provider's servers** as opposed to being provided from a company's own on-premises servers.

Email service

Dropbox

Storage service

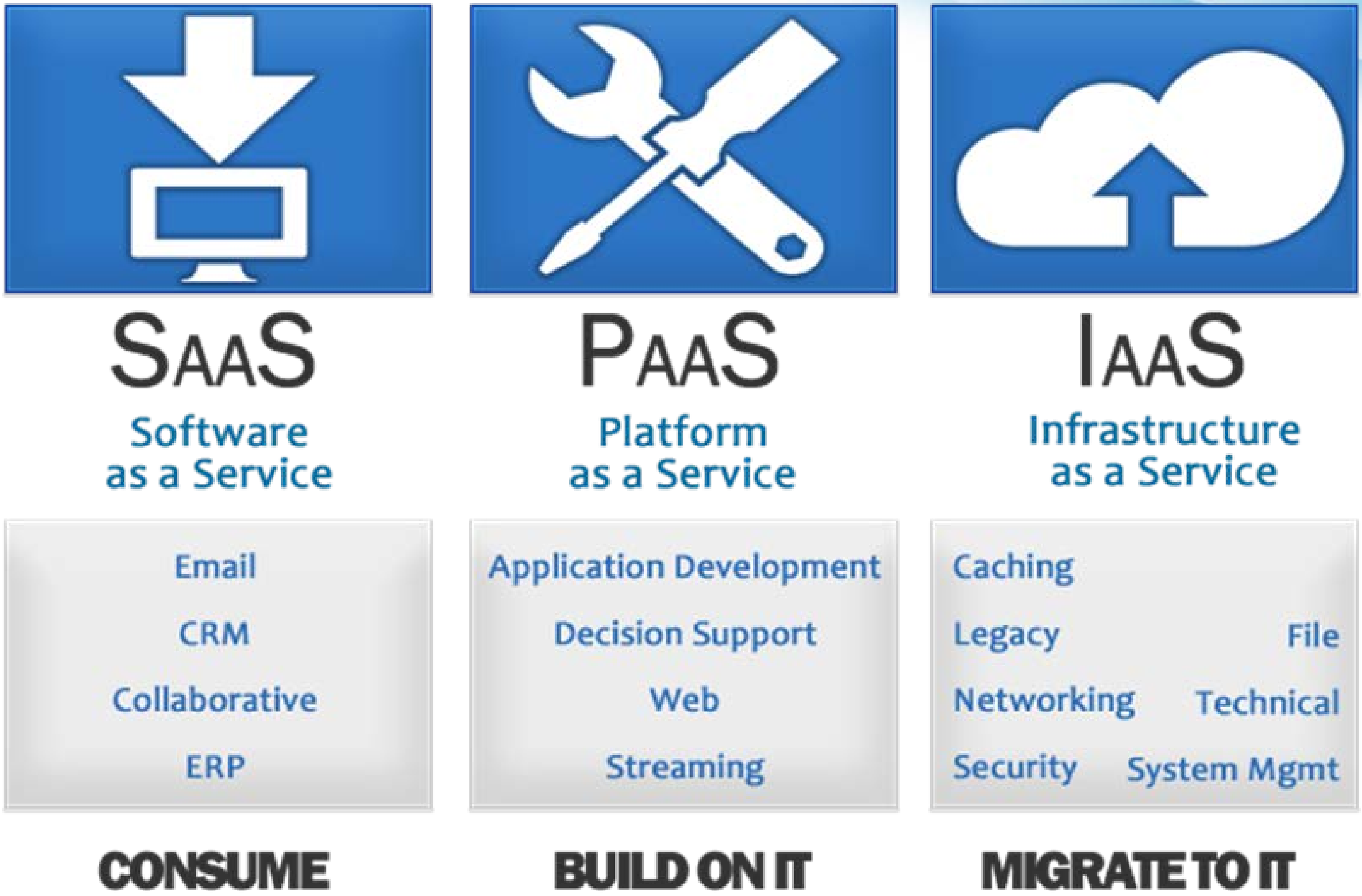


AWS EC2

Computing service

Database service

Classification of Cloud Services



SaaS

Software as a Service

Email

CRM

Collaborative

ERP

CONSUME

PaaS

Platform as a Service

Application Development

Decision Support

Web

Streaming

BUILD ON IT

IaaS

Infrastructure as a Service

Caching

Legacy

File

Networking

Technical

Security

System Mgmt

MIGRATE TO IT

Major Cloud Providers for SaaS

- ***Salesforce.com***
  - Salesforce has turned into the go-to provider for SaaS CRM solutions, with Gartner stating the firm is dominating the market.
  - **Customer relationship management (CRM)** is a system for managing a company's interactions with current and future customers. It often involves using technology to organize, automate and synchronize sales, marketing, customer service, and technical support.
  - SaaS CRM means the software is delivered via the Internet and does not require installation on your computer.
  - Businesses using the software do not purchase the software, but typically pay a recurring subscription fee to the software vendor.

Major Cloud Providers for PaaS

- ***Red Hat OpenShift***
  - This open source-based PaaS provider lets developers customize it as much as they want, and can be provided free as a trial (just 1GB storage is offered, though).
  - *https://www.openshift.com/*
- ***Heroku***
  - This platform supports a ton of programming languages, from Java to Ruby to Python. One of the earliest PaaS providers, it offers third-party apps as well as its own 'dynos' - virtualised containers that run processes in siloed environments.
  - Acquired by salesforce in 2010.
  - https://www.heroku.com/
- ***IBM BlueMix***
  - Focus more on applications development and deployment.
  - Leveraging Cloud Foundry to build, deploy, and manage their cloud applications.
  - Combine the strength of IBM, such as data analytic tools, DB, security services, etc.
  - Also provide some of the IaaS services, but not including virtual machine service.
  - https://console.ng.bluemix.net/

Major Cloud Providers for IaaS

- ***Amazon AWS***
  - First and also the current market leader in IaaS cloud computing
  - Widest breadths of cloud services
  - But also require for IT knowledge to deploy applications on it.
  - *http://aws.amazon.com/*
- ***Microsoft Azure***
  - Similar to AWS, but has better integration with Visual Studio development env.
  - Powered by Microsoft and its Hyper-V virtualization software
  - Has strong relationships with a large number of enterprise IT shops
  - *http://azure.microsoft.com/*
- ***Google Cloud Platform***
  - Start with **Google App Engine**, a development platform
  - Later expanded into the IaaS space with **Google Compute Engine**
  - *https://cloud.google.com/*

Major Cloud Provides for IaaS

- ***HP Cloud & Rackspace***
  - Both base on the open source IaaS solution: OpenStack
  - Could prevent vendor lock-in, and compatible to other OpenStack-based cloud
  - But limited by the development progress of OpenStack
  - http://www.hpcloud.com/
  - http://www.rackspace.com/

Outline

- Introduction of Cloud Services
- Introduction of Amazon AWS
- AWS Global Infrastructure
- AWS Tools for Accessing Services
- AWS Services
- AWS Use Cases

AWS – Amazon Web Services

- A collection of remote computing services, also called web services, that make up a cloud computing platform by Amazon.com.
- Officially launched in 2006, a result of over $2billion dollar investment by the company and was developed with following characteristics in mind:
  - Elastic capacity both up and down.
  - Fast response time.
  - 24/7 availability.
  - Rock solid reliability.

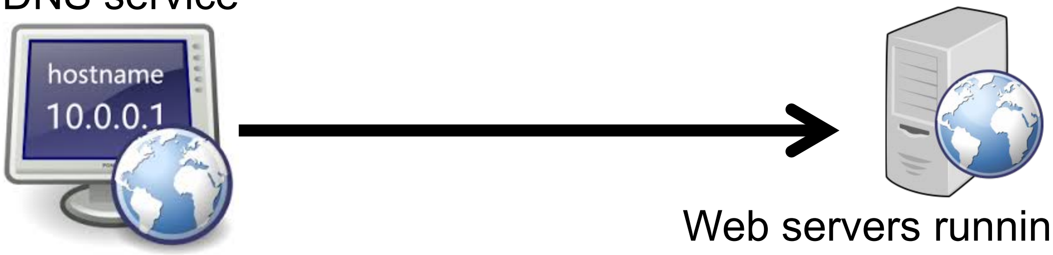
Example Application

- Build and host a basic website on AWS

EC2: Web servers running
Apache HTTP Web service

Example Application

- Build and host a basic website on AWS

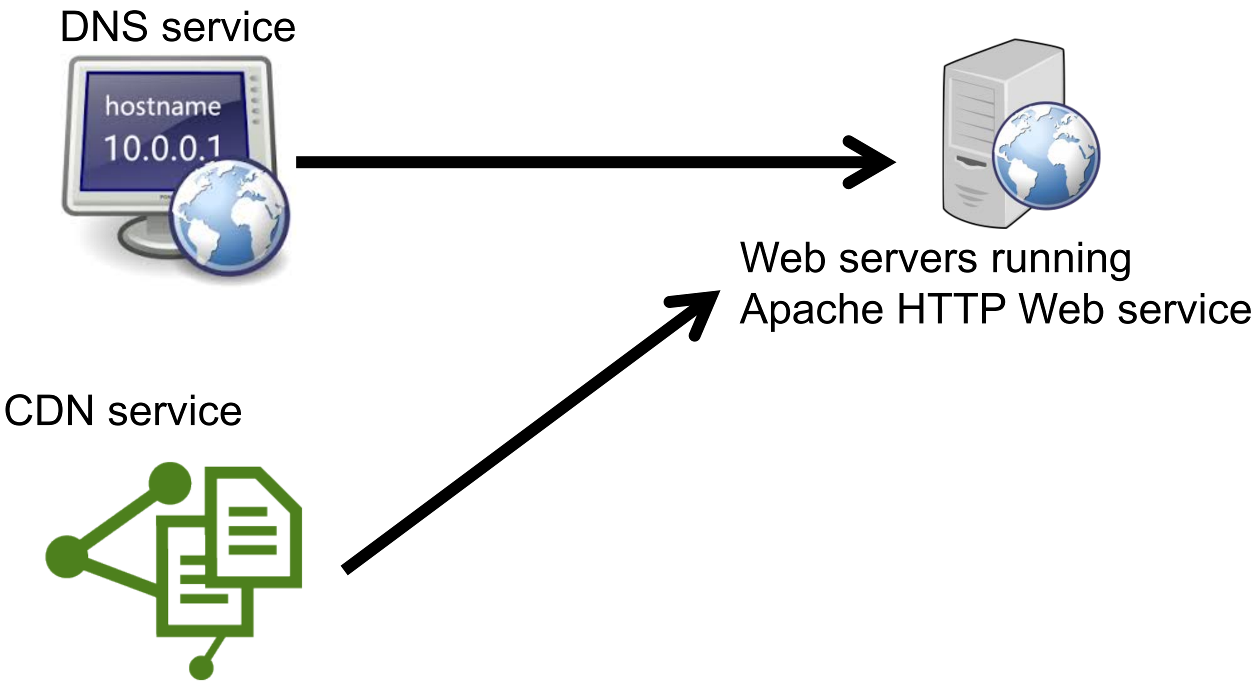


DNS service

Web servers running
Apache HTTP Web service

Example Application

- Build and host a basic website on AWS



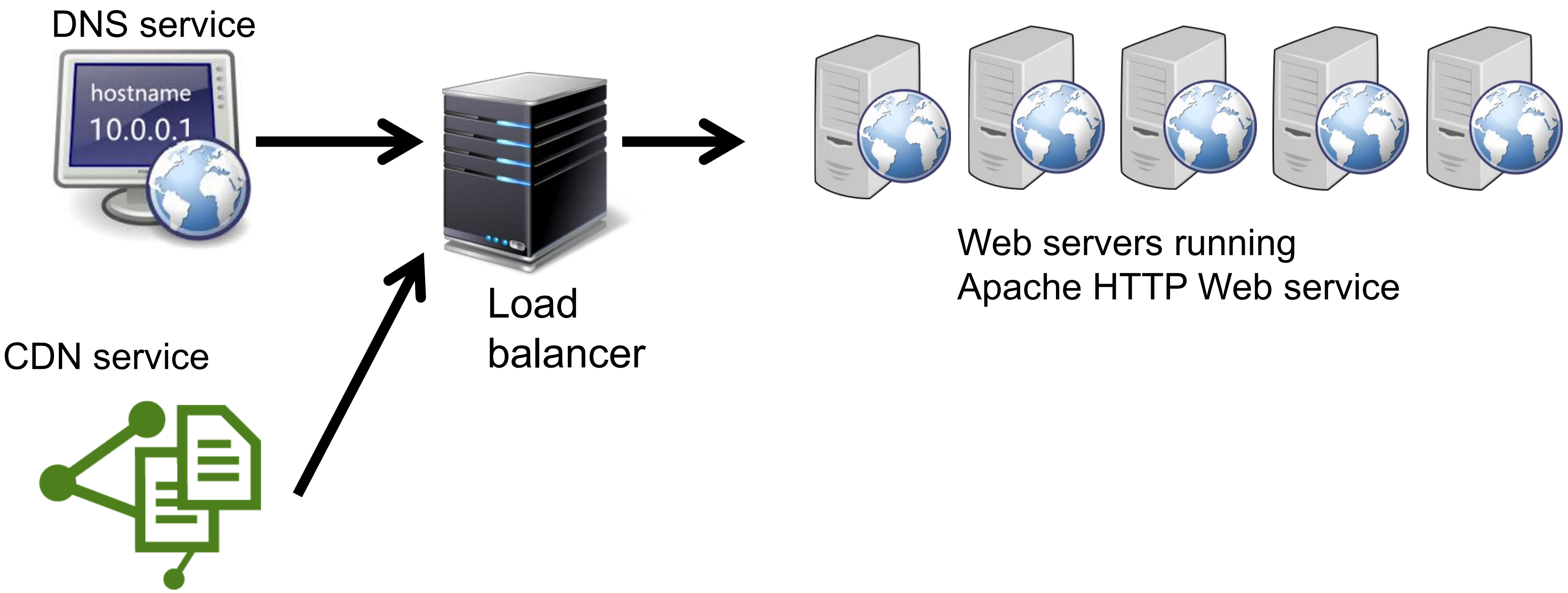
DNS service

Web servers running
Apache HTTP Web service

CDN service

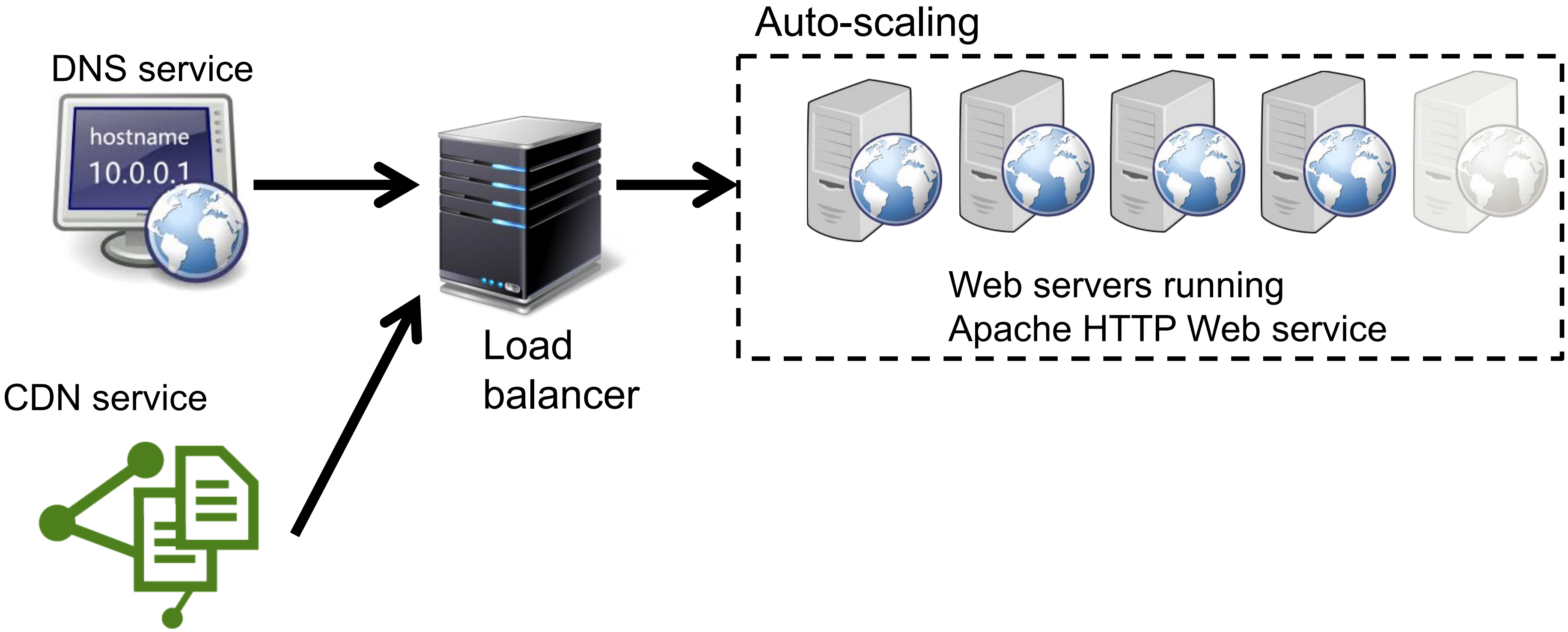
Example Application

- Build and host a basic website on AWS



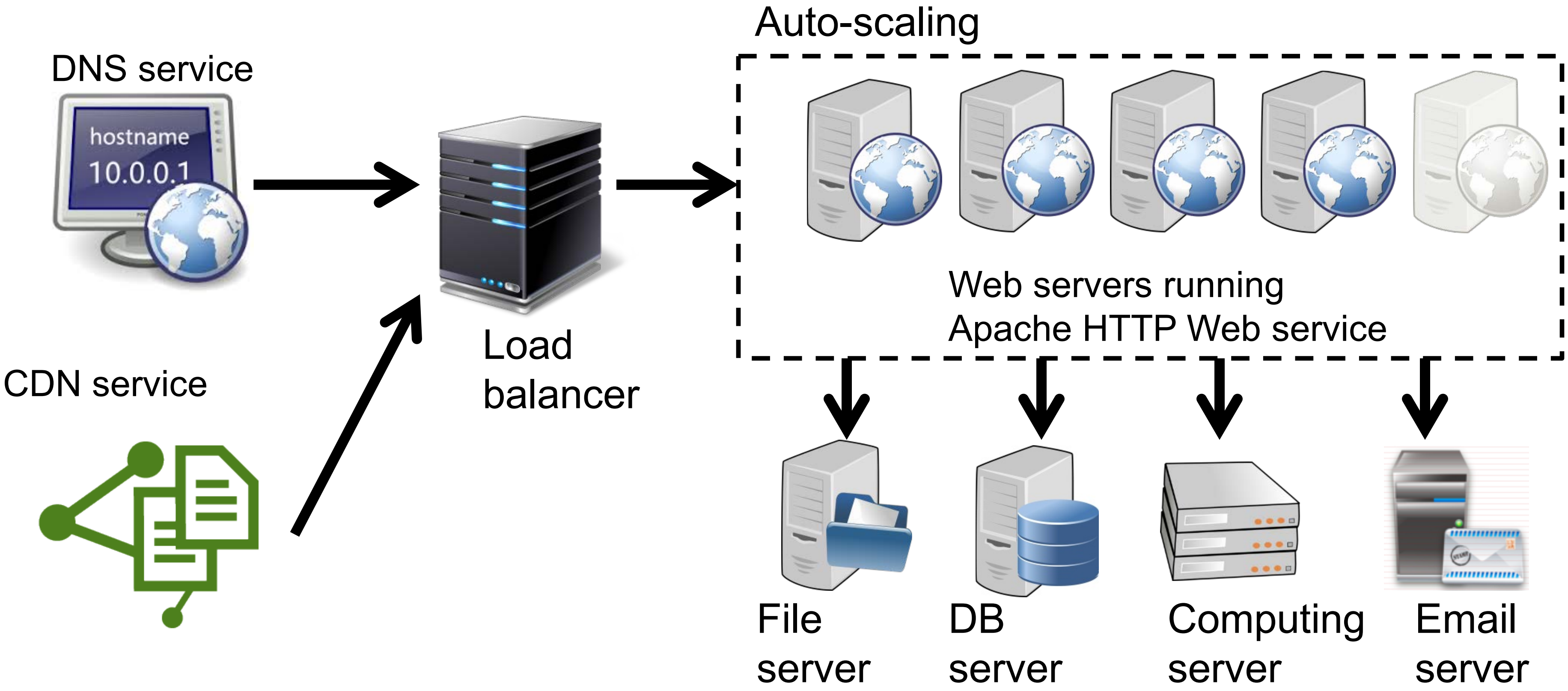
Example Application

- Build and host a basic website on AWS



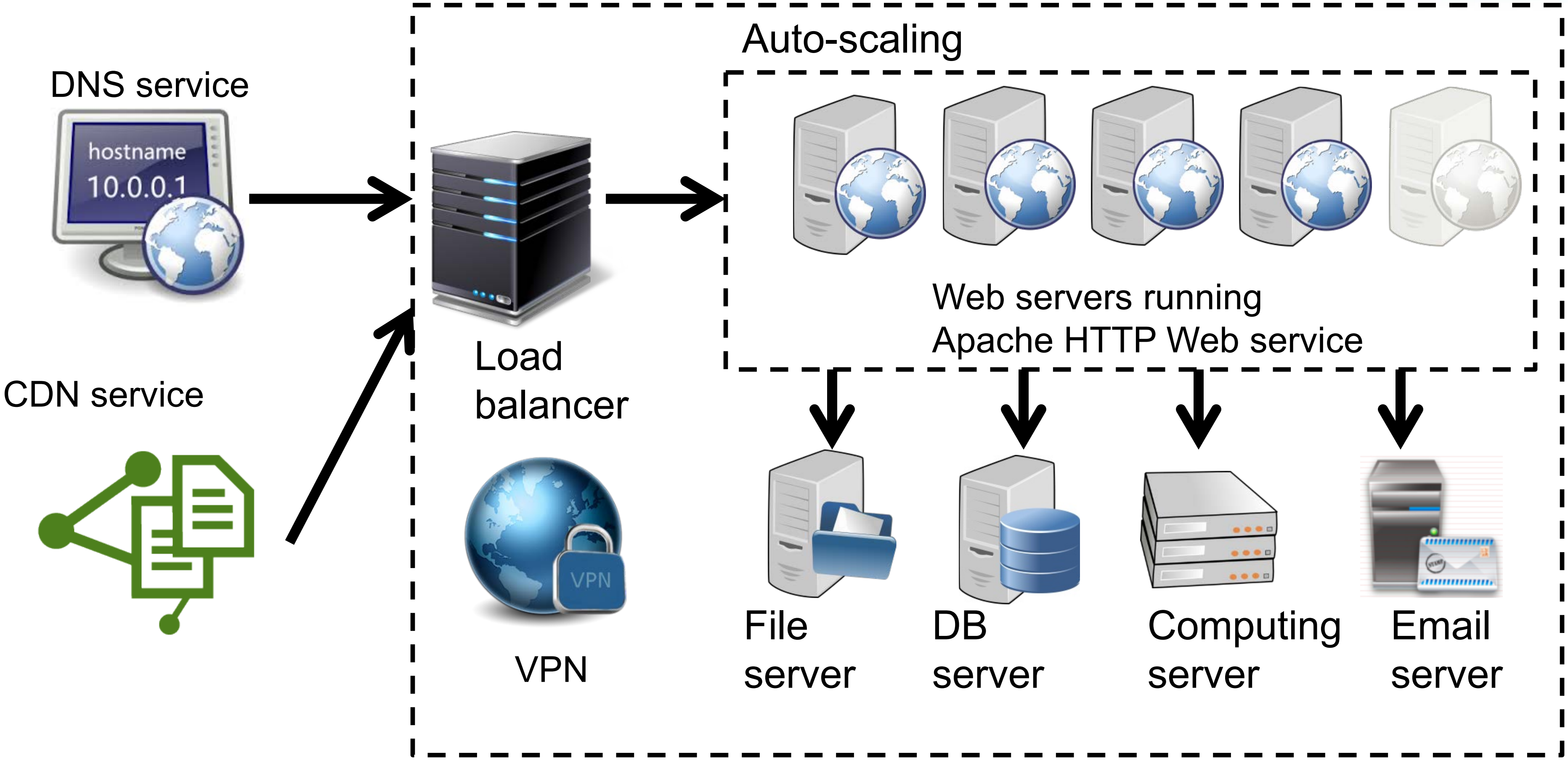
Example Application

- Build and host a basic website on AWS



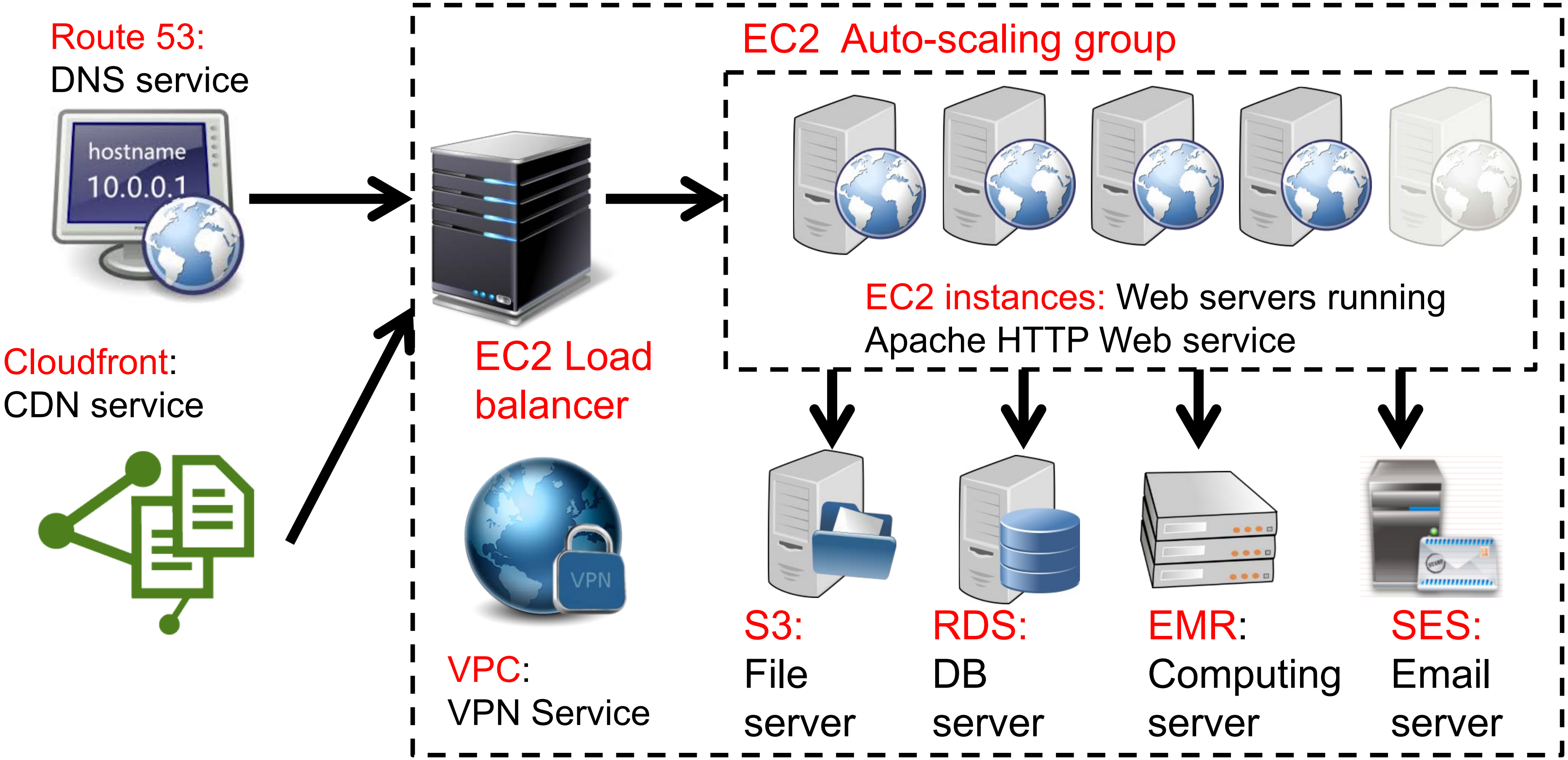
Example Application

- Build and host a basic website on AWS



Example Application

- Build and host a basic website on AWS



Route 53: DNS service

Cloudfront: CDN service

EC2 Load balancer

VPC: VPN Service

EC2 Auto-scaling group

EC2 instances: Web servers running Apache HTTP Web service

S3: File server

RDS: DB server

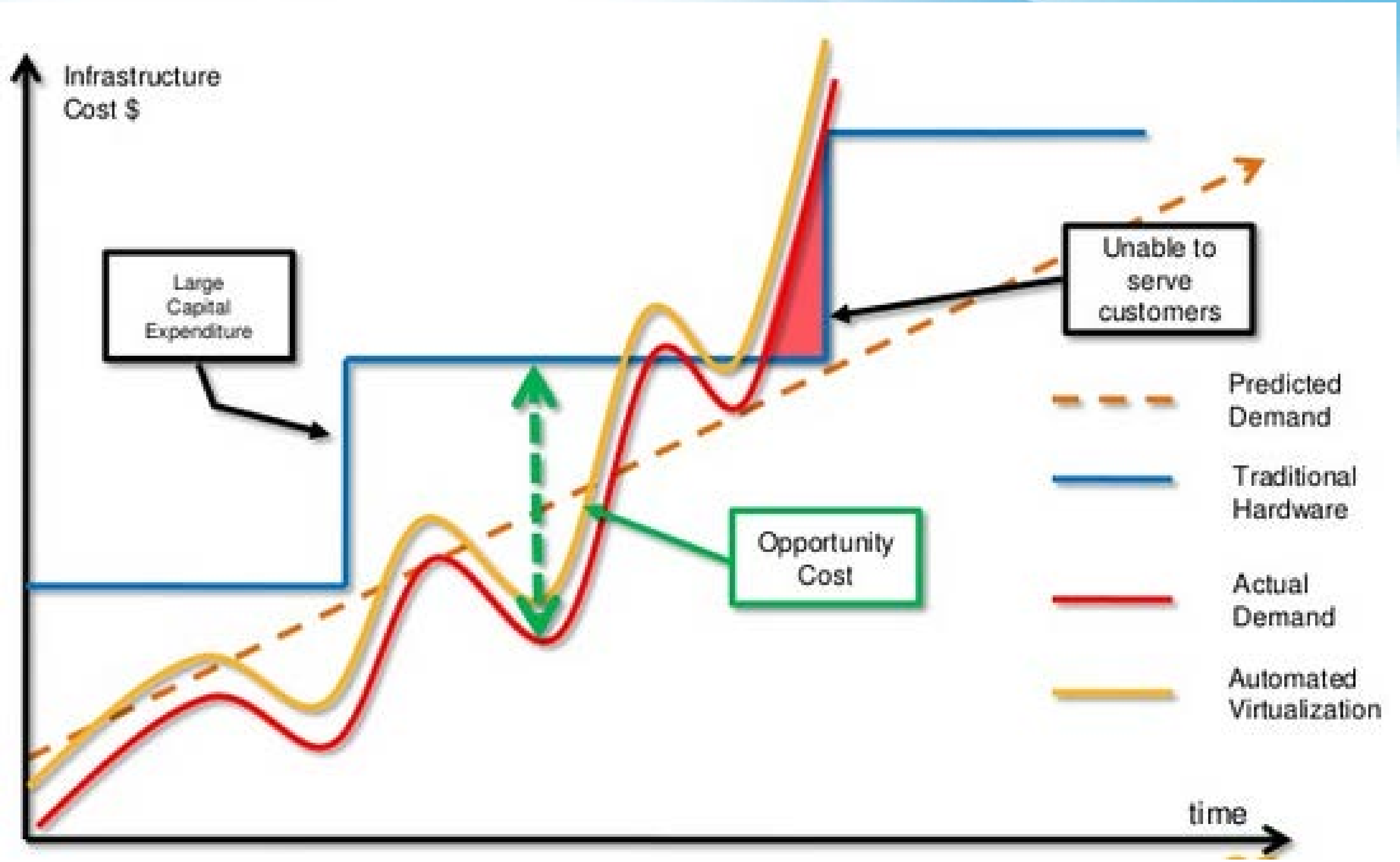
EMR: Computing server

SES: Email server

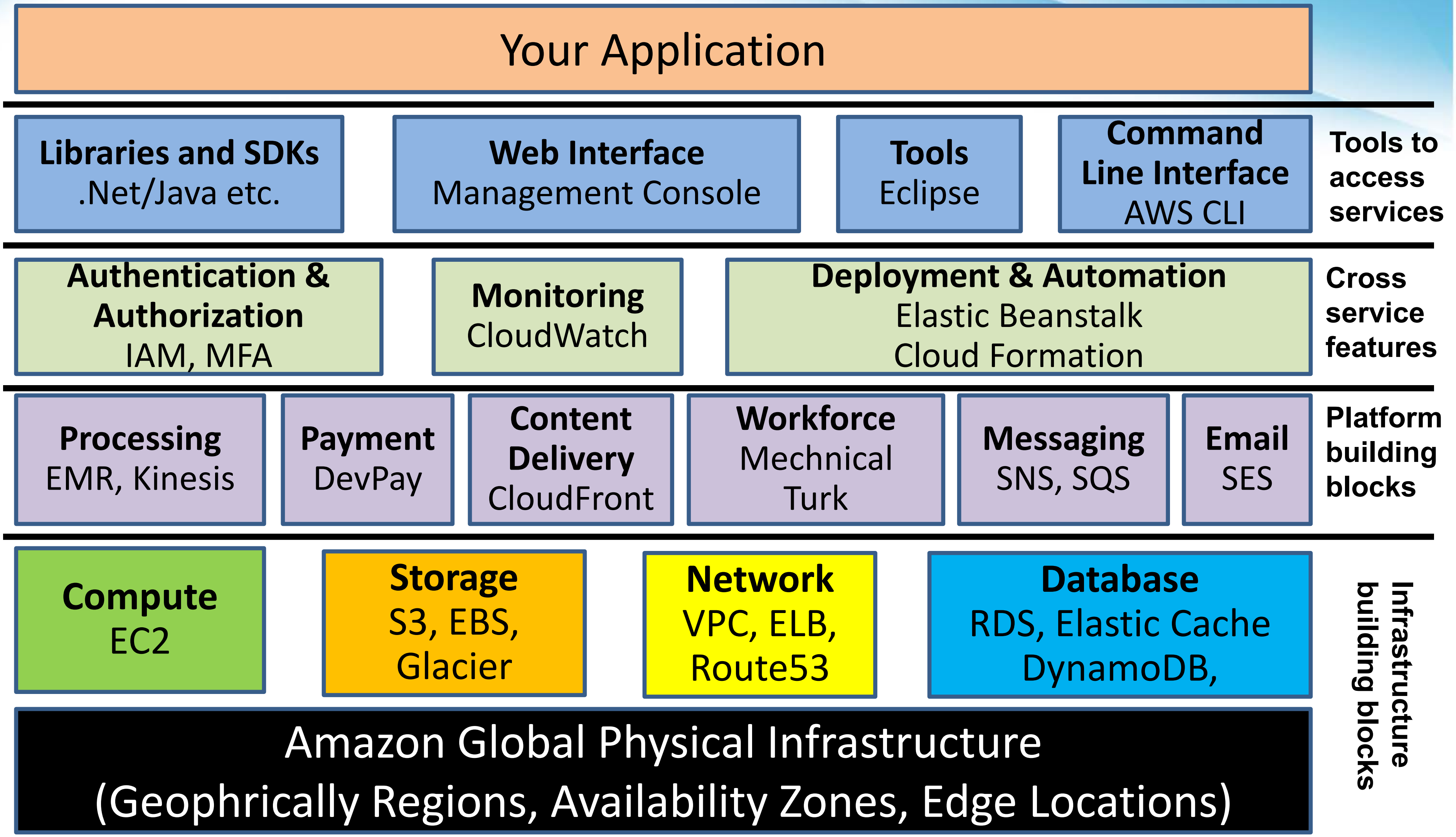
Benefits of Cloud Services

- Development:
  - Enable applications to be rapidly and incrementally composed from services.
  - Allow to use the programming models, operating systems, databases, and architectures with which users are already familiar.
- Deployment & Operations:
  - Provide scalable, elastic and reliable services.
  - Deliver distributed and global infrastructure.
  - Enable continuous availability.
- Cost:
  - Reduce the capital and management cost of IT.
  - Charged by the cost-effective pay-as-you-use pricing model.

Benefits of Cloud Services



AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

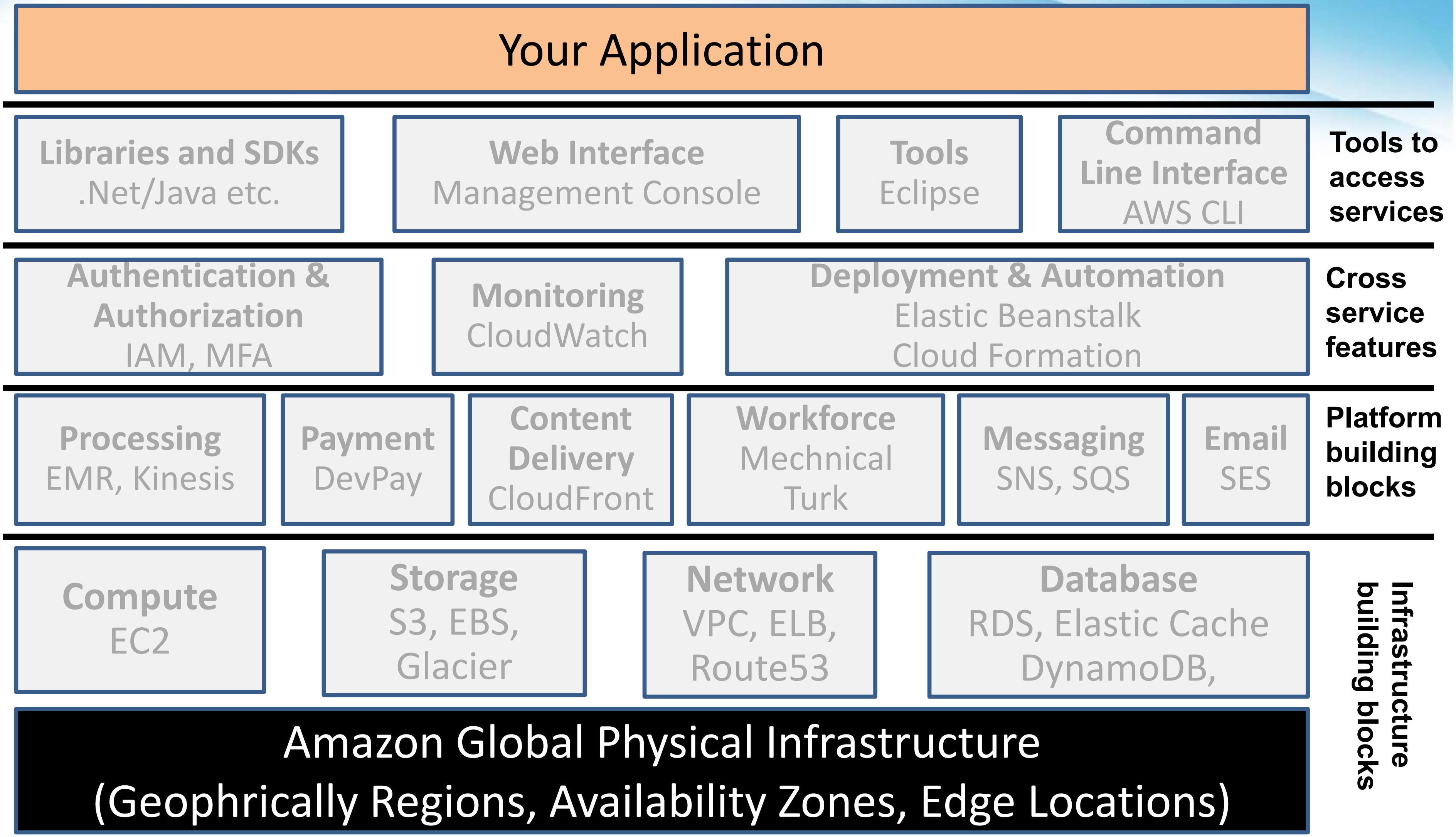
Network
VPC, ELB, Route53

Database
RDS, Elastic Cache
DynamoDB,

Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

Infrastructure building blocks

AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

Network
VPC, ELB, Route53

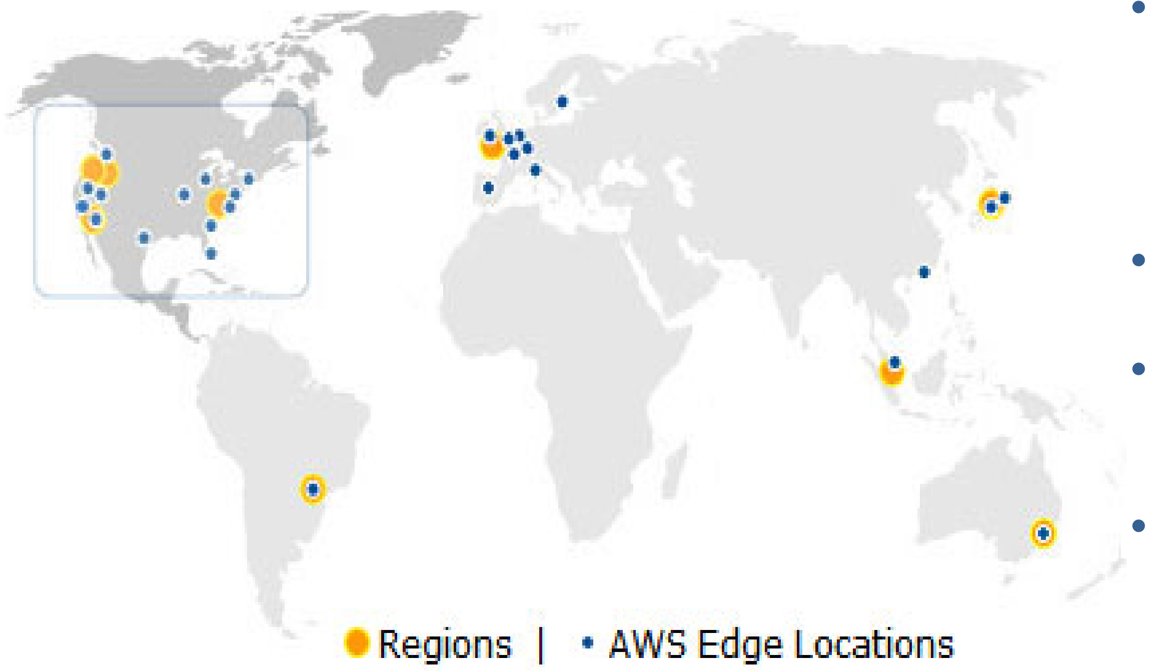
Database
RDS, Elastic Cache
DynamoDB,

Infrastructure building blocks

Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

AWS Global Infrastructure

- 10 AWS **Regions**(geographic area/datacenter)
  - **completely independent.**
  - **Each region has its own hosting services, price and policy.**
- 51 Edge locations store local content for **DNS** and **CDN** services

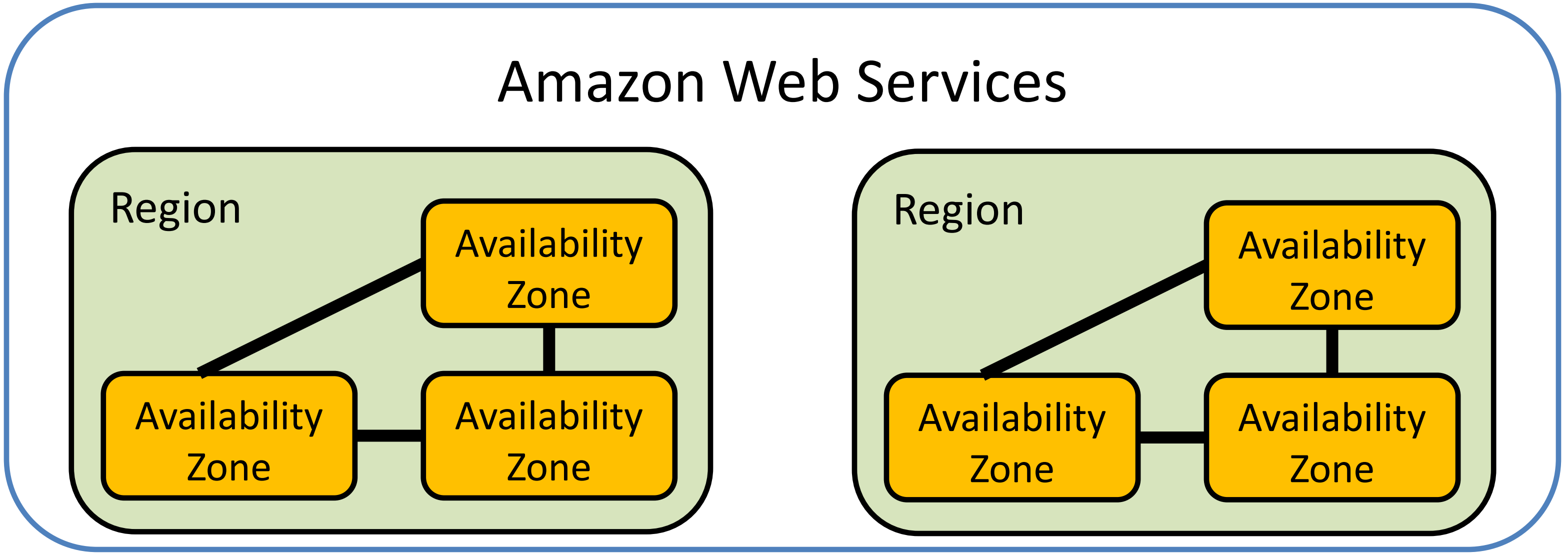


- North America
  - **US East (Northern Virginia)**
  - **US West (Northern California)**
  - **US West (Oregon)**
  - **AWS GovCloud (US) Region**
- South America
  - **São Paulo Region**
- Europe
  - **Ireland Region**
  - **Frankfurt Region**
- Asia Pacific
  - **Singapore Region**
  - **Tokyo Region**
  - **Sydney Region**

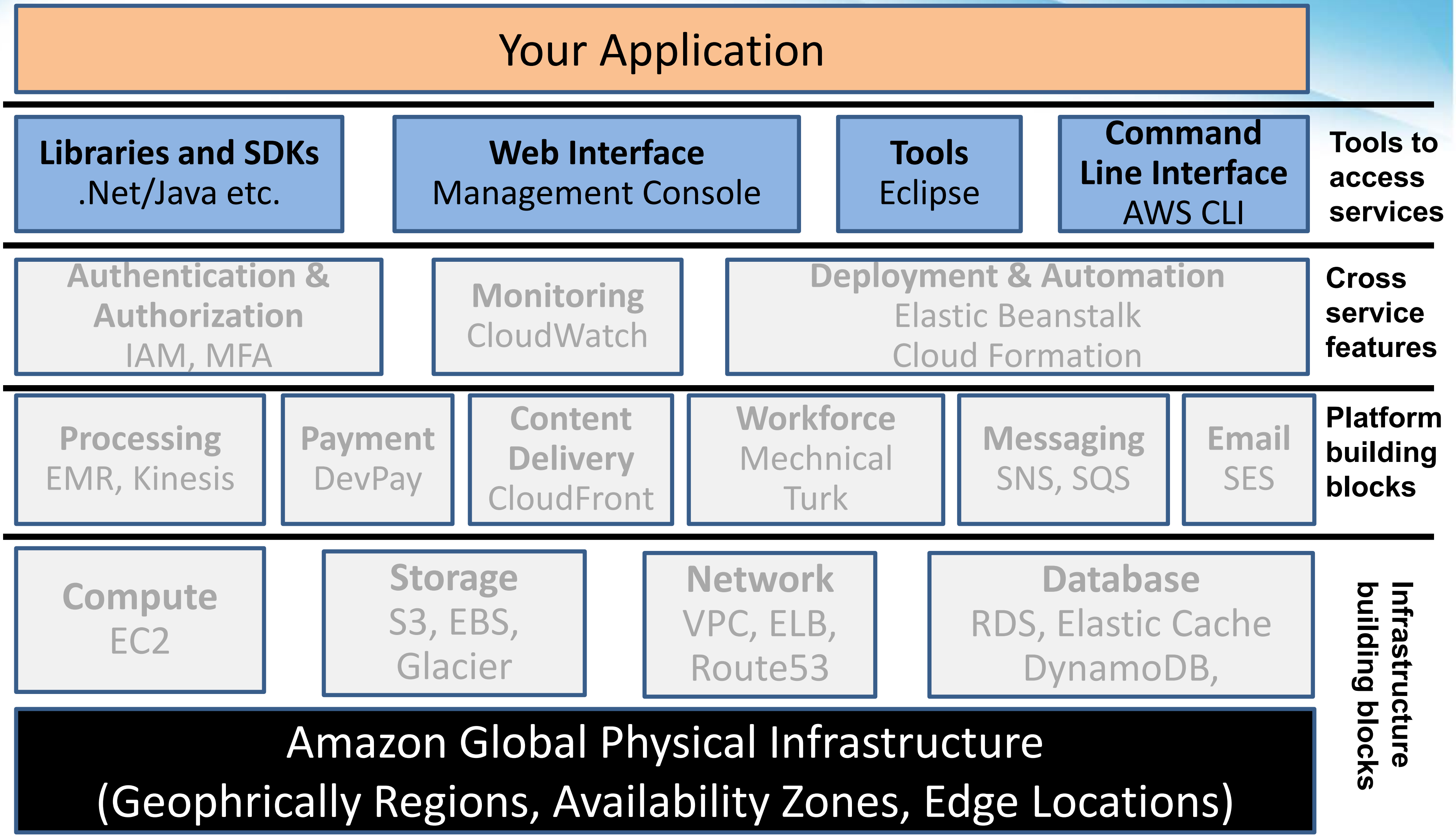
Source: http://aws.amazon.com/about-aws/global-infrastructure/

Availability Zones

- Each region has multiple **availability zones.**
- Total of **26** zones are available.
- Each **Availability Zone is isolated**, but the Availability Zones in a region are **connected through low-latency links**.
- Provide fault tolerance by running instances on multiple zones.



AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

Network
VPC, ELB, Route53

Database
RDS, Elastic Cache
DynamoDB,

Infrastructure building blocks

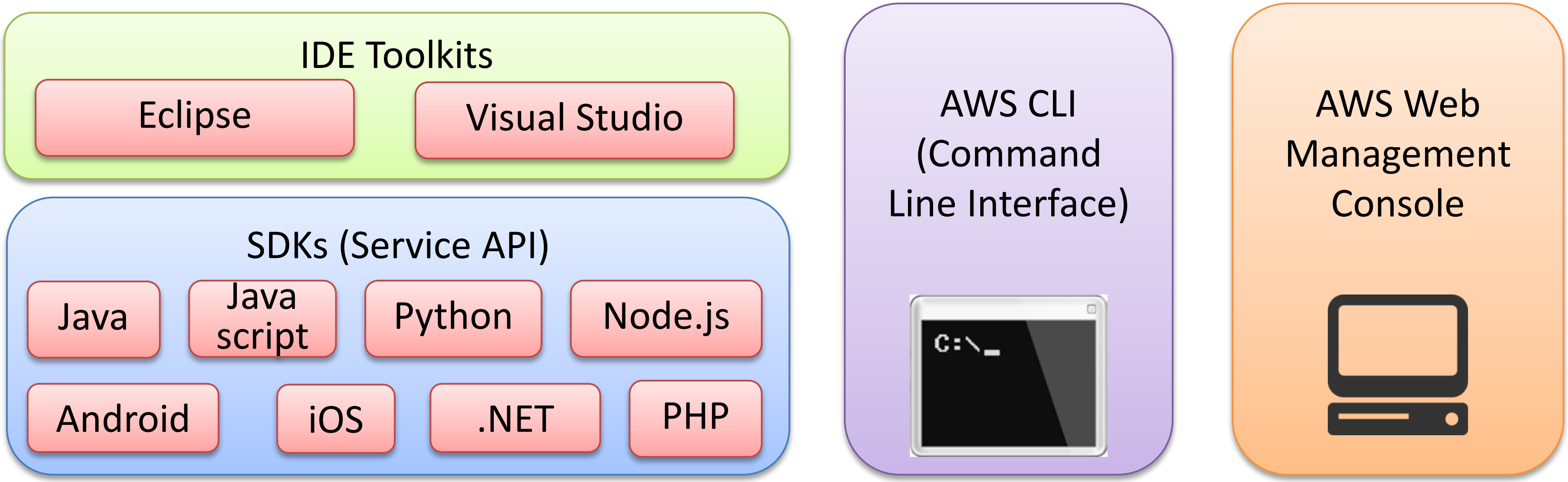
Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

Tools to Access AWS Services

Request temporary credential and assume an IAM role

AWS user with AWS access and secrete keys

Login through AWS console webpage



IDE Toolkits

- Eclipse
- Visual Studio

SDKs (Service API)

- Java
- Java script
- Python
- Node.js
- Android
- iOS
- .NET
- PHP

AWS CLI (Command Line Interface)

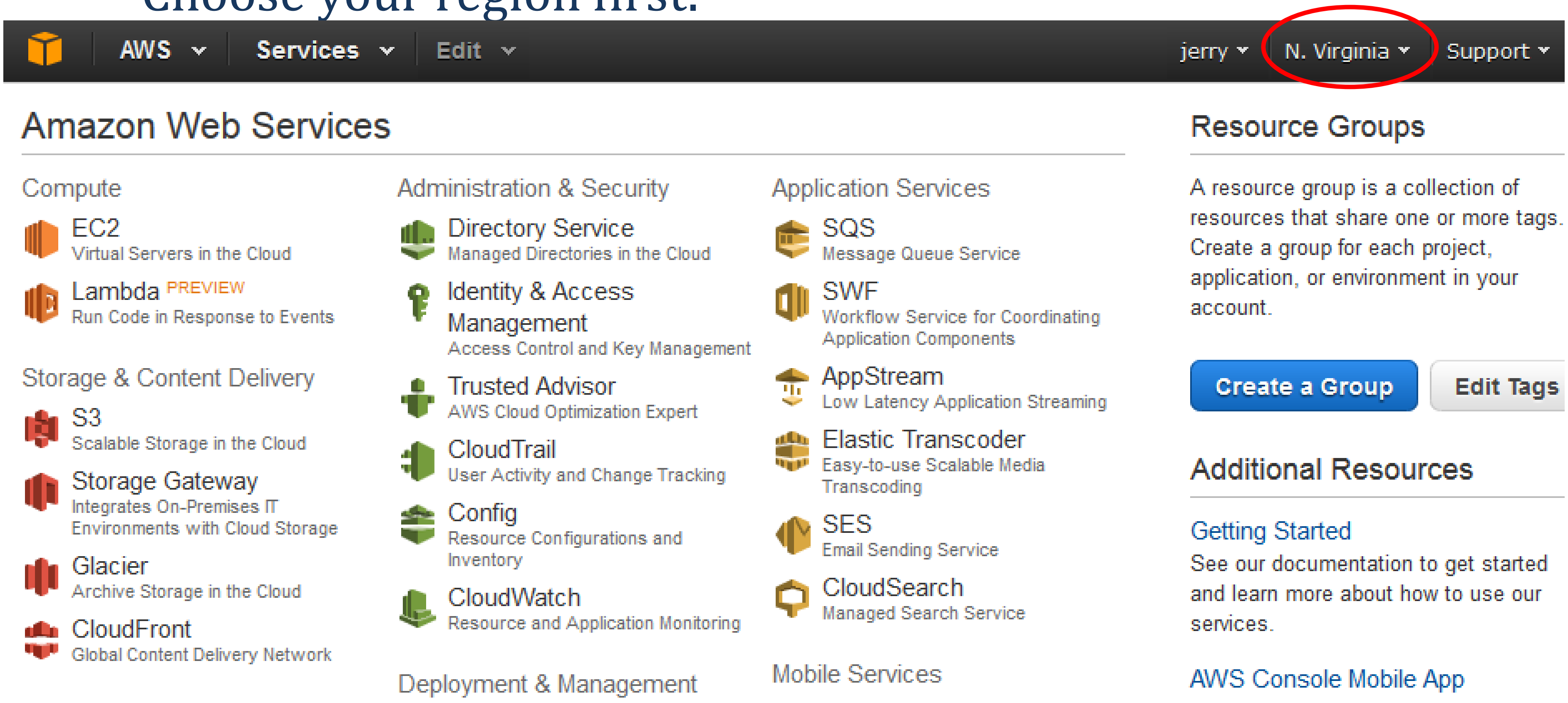
AWS Web Management Console

Source: http://aws.amazon.com/tools/

AWS Web Management Console

- After sign-in your AWS console at https://www.amazon.com
- Choose your region first.

region



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

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

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

Resource Groups

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

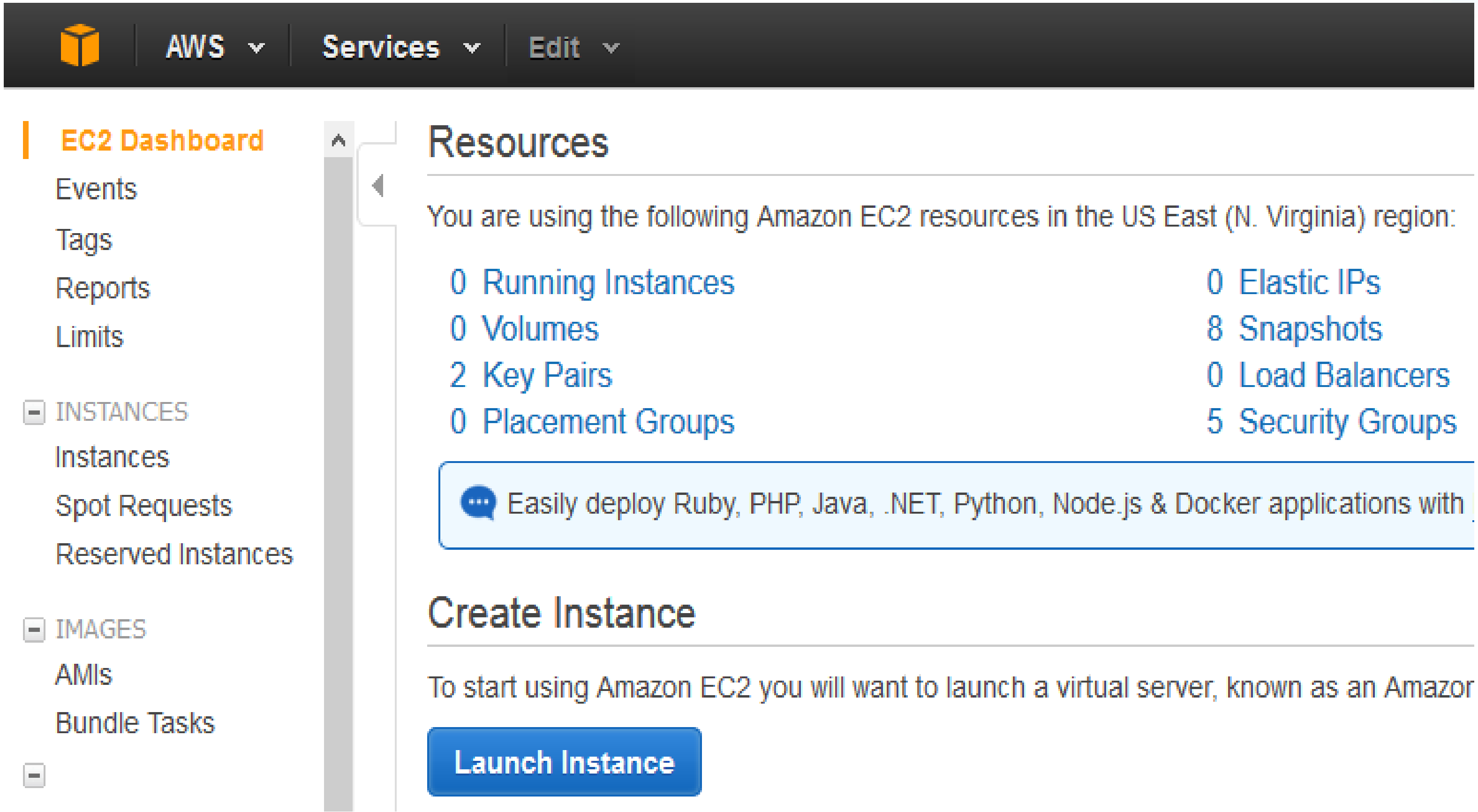
Additional Resources

Getting Started
See our documentation to get started and learn more about how to use our services.

AWS Console Mobile App

AWS Web Management Console

- EC2:



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances	0 Elastic IPs
0 Volumes	8 Snapshots
2 Key Pairs	0 Load Balancers
0 Placement Groups	5 Security Groups

Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazor

AWS SDKs API

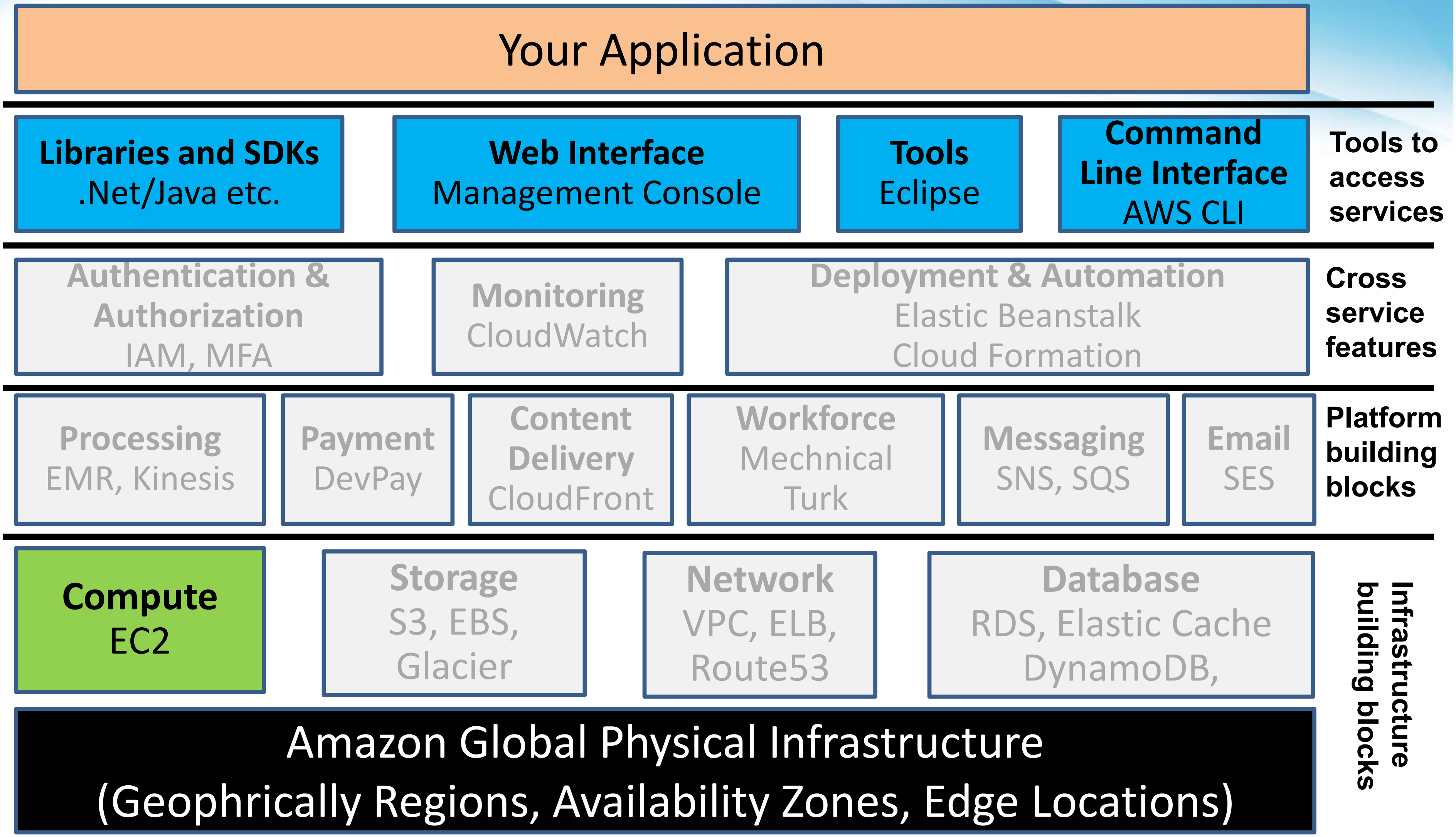
- Define the API of each service. SDK is provided in different programming languages and platforms for programmers:
  - Mobile phone: Android, iOS
  - Browser: Javascript, Node.js
  - General-purpose languages: .NET(C#), Java, Python, Ruby, PHP
- Each SDK has a complete user guide and API reference
  - Example to launch a EC2 instance

Python	conn.run_instances('<ami-image-id>', key_name='myKey', instance_type='c1.xlarge', security_groups=['your-security-group-here'])
Java	RunInstancesResult runInstancesResult = amazonEC2Client.runInstances(runInstancesRequest);
Javascript	ec2.runInstances(params, function(err, data)

AWS Command Line Interface

- A unified tool to manage your AWS services and control multiple AWS services from the command line and automate them through scripts.
  - Including EC2, IAM, S3, SNS, SWF, etc.
  - Example to launch a t1.micro instance in EC2
    - aws ec2 run-instances --image-id ami-*xxxxxxxx* --count 1 --instance-type t1.micro --key-name *MyKeyPair* --security-groups *MySecurityGroup*
  - Example to create a S3 bucket
    - aws s3 mb s3://*bucket-name*
- Must install AWS CLI on your machine, and setup the AWS secrete key & access key.
  - Documentation: http://aws.amazon.com/cli/

AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

Network
VPC, ELB, Route53

Database
RDS, Elastic Cache
DynamoDB,

Infrastructure building blocks

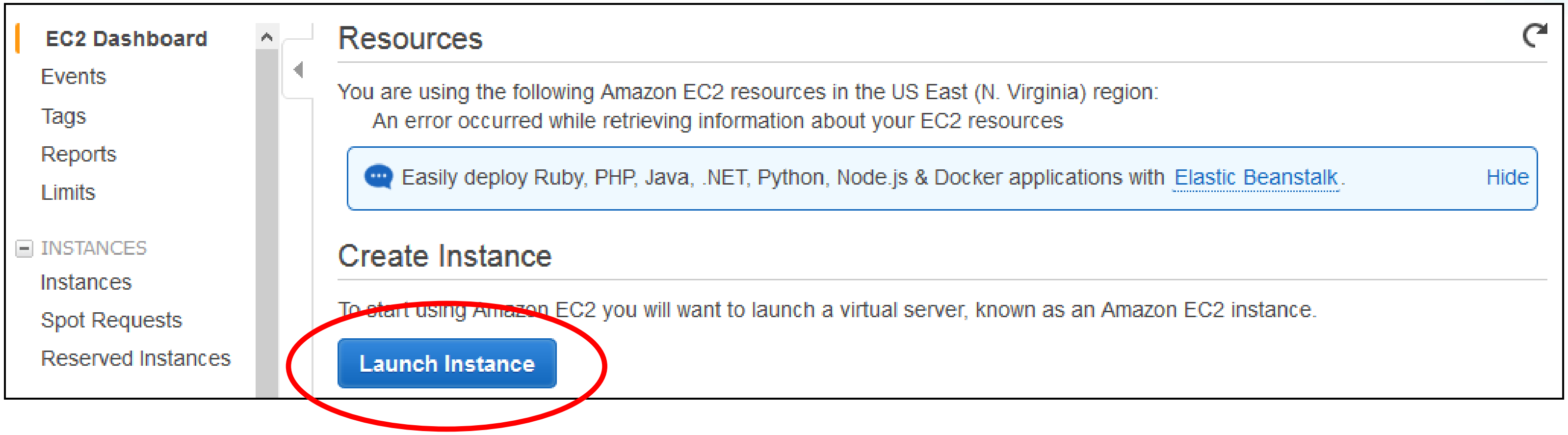
Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

Computing Service: EC2

- Amazon EC2 = Virtual Machine
- On-demand computer power
  - Obtain and boot new server instances in minutes
  - Quick scale capacity (memory, disk, cores) up or down
  - Charge by hours: from $0.02 per hour
  - Offer by On-Demand, Reserved and Spot Pricing
- Key features
  - Support for Windows, Linux, FreeBSD, and OpenSolaris
  - Full control and access to operating system
  - Deploy across availability zones for reliability
  - Monitor status and usage

Computing Service: EC2

- Process to launch a EC2 instance



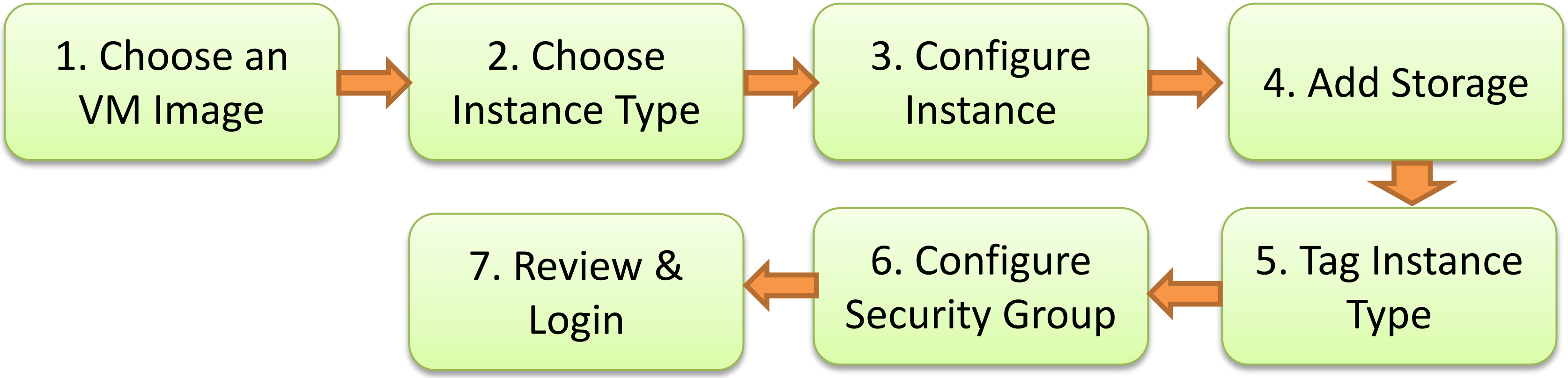
Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:
An error occurred while retrieving information about your EC2 resources

Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with Elastic Beanstalk. Hide

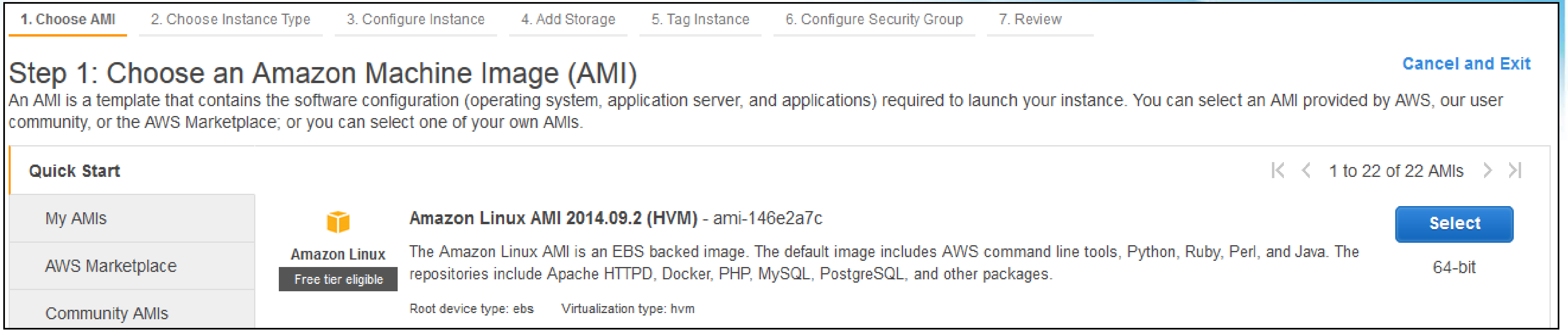
Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.



1. Choose an VM Image → 2. Choose Instance Type → 3. Configure Instance → 4. Add Storage → 5. Tag Instance Type → 6. Configure Security Group → 7. Review & Login

Step1. Choose an Amazon Machine Image (AMI)



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

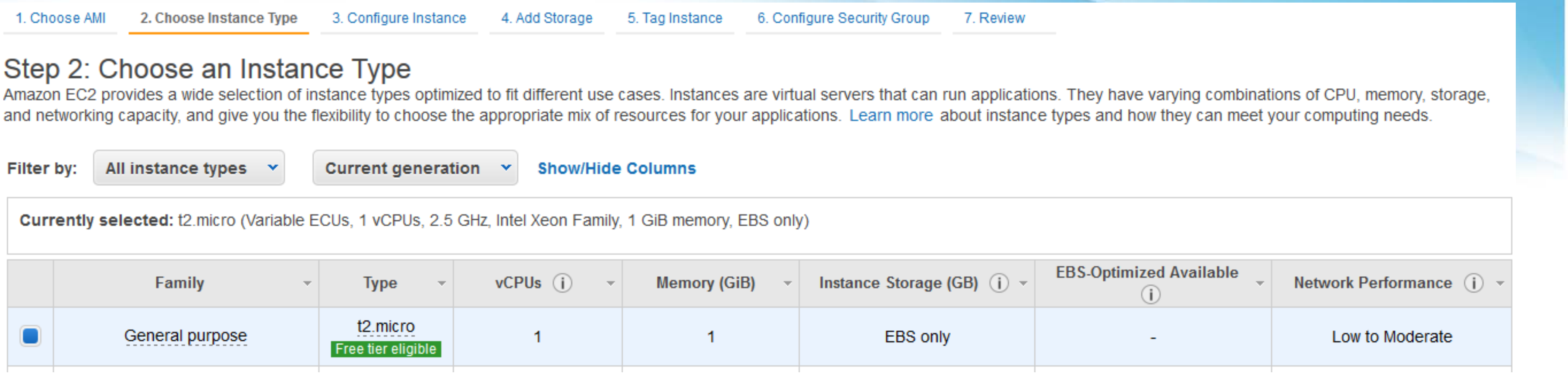
Amazon Linux AMI 2014.09.2 (HVM) - ami-146e2a7c

The Amazon Linux AMI is an EBS backed image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Apache HTTPD, Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm

- VM **instance** is an **active running** virtual machine.
- VM **image** is a **template** for creating new instances.
  - Contain the OS, software and disk content.
  - But **without** specifying the resource capacity
  - AWS has image for **Windows, Linux, FreeBSD,** and **OpenSolaris.**
- Users can create new image by snapshotting a VM instance.
- Documentation for creating and using AMI:
  - http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html

Step2. Choose Instance Type



1. Choose AMI | 2. Choose Instance Type | 3. Configure Instance | 4. Add Storage | 5. Tag Instance | 6. Configure Security Group | 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

Filter by: All instance types | Current generation | Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs ⓘ	Memory (GiB)	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ
■	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate

- Board set of EC2 instance types…
  - General Purpose: T2, M3
  - Compute optimized: C3, C4
  - Storage and I/O optimized: I2, HS1,
  - Memory optimized: R3
  - GPU enabled: G2
- Sizes: micro, small, medium, large, xlarge, 2xlarge, etc.
  - **Large size has higher resource capacity but more expensive**
- Instance selection is a combination of both type and size
  - E.g. t2.micro, t2.small, m3.medium

Source: http://aws.amazon.com/ec2/instance-types/

Step3. Configure Instance

- Number of instances:
  - to create one or multiple VM instances
- Purchasing option:
  - to use spot instance or on-demand
- Network settings:

For fault tolerence

  - **IP**, subnet, network interface, etc.
  - **Subnet refers to availability zone**.
- IAM role:
  - to assume the permission of a given **role**.
- Advanced details:
  - specify user data to configure an instance or **run a configuration script during launching**
  - used for automation

3 EC2 Purchasing Options

- On-demand
  - **Fixed price**, pay per hour.
  - Guarantee **availability** until terminated by users.
- Spot instance
  - Based on **bidding** auction.
  - Cost is terminated by **market price**. (Not bidding price.)
  - Instances can be taken away **without notice at anytime**.
  - Normally 1/5 of the on-demand cost.
  - Users applications must have **fault tolerance** ability.
- Reserved instance
  - Fixed price but with **discount upto 75%**
  - Reservation **in advance**.
  - 1 year or 3 year term.
  - Users have to **predict future usage**.
- Reference: http://aws.amazon.com/ec2/purchasing-options/

Step4. Add Storage

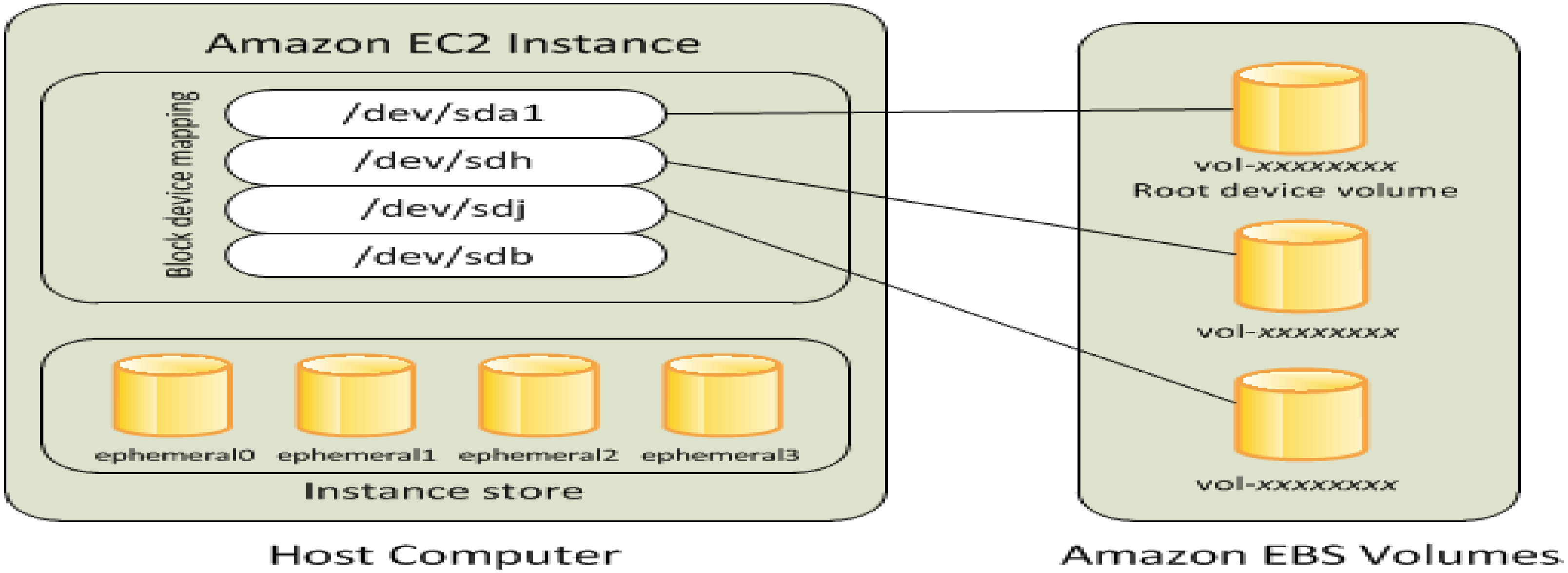
- Two types of **block storage** are available to EC2 instance
  - In general, we prefer EBS because of its flexibility and persistency.
  - All default VM images provided by AWS use ECS for root volume.
  - Many (**but not all**) Amazon EC2 instance types can choose to attach instance store before launching.

	Elastic Block Storage (EBS)	Instance Store
Enabling technique	Virtualized network volume can be **re-attached** to another EC2 instance	Physical volume on local host computer
Lifetime	Persistent, independent of EC2 instance	Temporary
Size	Limited by 1TB, and **re-adjustable** before attachment	Limited by 10GB, and is fixed after launching the instance
Boot time	< 1min.	< 5 min.
Charge	**By # IO requests and storage size**	Same as EBS, but slightly cheaper
IOPS (Perf.)	EBS could be less stable than instance store due to **network and virtualization overhead**	

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ComponentsAMIs.html#storage-for-the-root-device

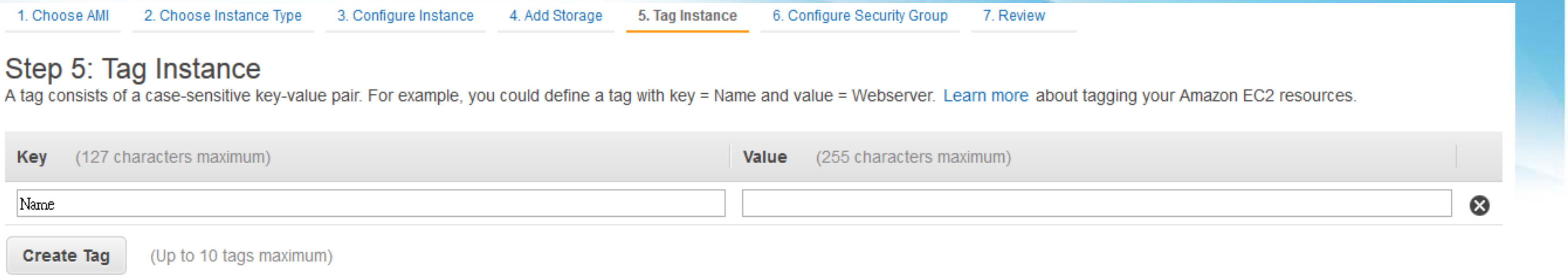
EBS vs. Instance Store

- An instance store consists of one or more instance store volumes, and available volumes are automatically mapped to an instance store-backed AMI in launching.
- Unattached EBS volume can be created independently from EC2 console.



http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html

Step5. Tag Instance Type



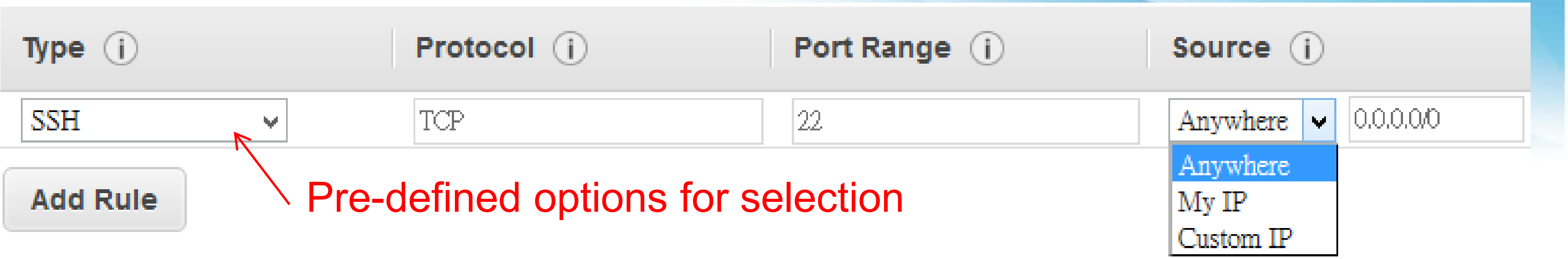
Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	

- Tags enable you to categorize your AWS resources. Help you to **manage and monitor resource usage**.
- Each tag consists of **a key and an optional value**. Tags don't have any semantic meaning to Amazon EC2 and are interpreted strictly as a string of characters.
- **Tags are not automatically assigned** to your resources, you have to tag your resource in creation. Tags can be edited and removed anytime.
- **E.g.: In this class, we require everyone to tag their resources with an unique key**

Step6. Configure Security Group



Type	Protocol	Port Range	Source	
SSH	TCP	22	Anywhere	0.0.0.0/0

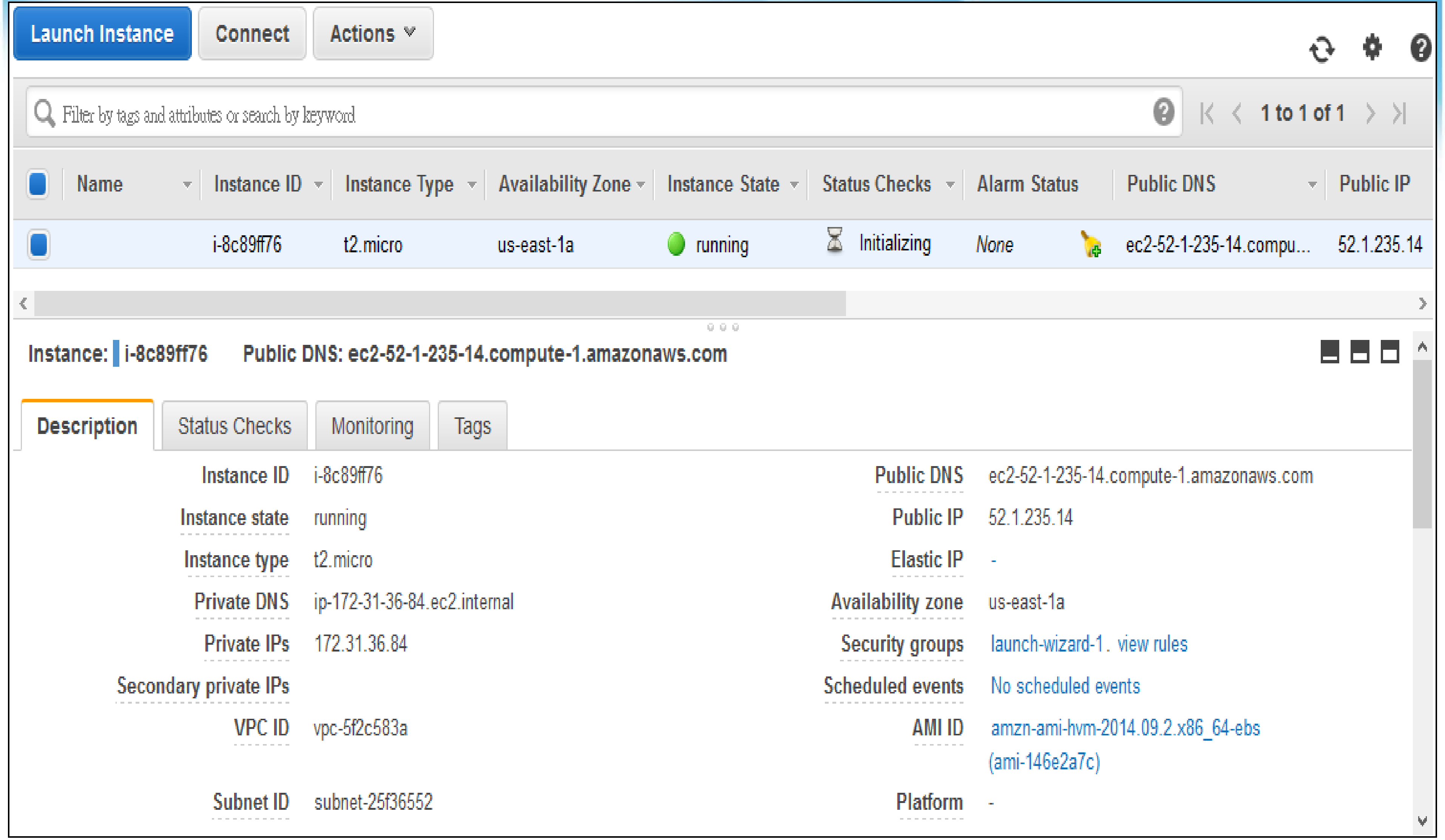
Pre-defined options for selection

- A security group is a set of firewall rules that control the traffic for your instance.
- You can create a **new** security group by adding your own firewall rules.
  - E.g.: if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports.
- Or you can select an **existing** security group

Step7. Review & Login

- After review and confirm all your previous setting, you must setup the key-pair to login the instances.
- By default all VM instances all protected by the **public-key cryptography**.
    - Because we cannot obtain the root password of the machine image provided by Amazon.
    - Only a **default user account with sudo permissio**n is created initially.
- Three key pair options
    - Create a new one: remember to download the key pair immediately.
    - Choose an existing one: commonly used for the same user account.
    - Proceed without key pair: only for the OS image you created with known root password.
- SSh login command
    - ssh -i my-key-pair.pem ec2-default-user@ec2-hostname-or-public-ip

You Have Created Your First EC2 Instance



☐	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
■		i-8c89ff76	t2.micro	us-east-1a	● running	Initializing	None	ec2-52-1-235-14.compu...	52.1.235.14

Instance: i-8c89ff76 **Public DNS:** ec2-52-1-235-14.compute-1.amazonaws.com

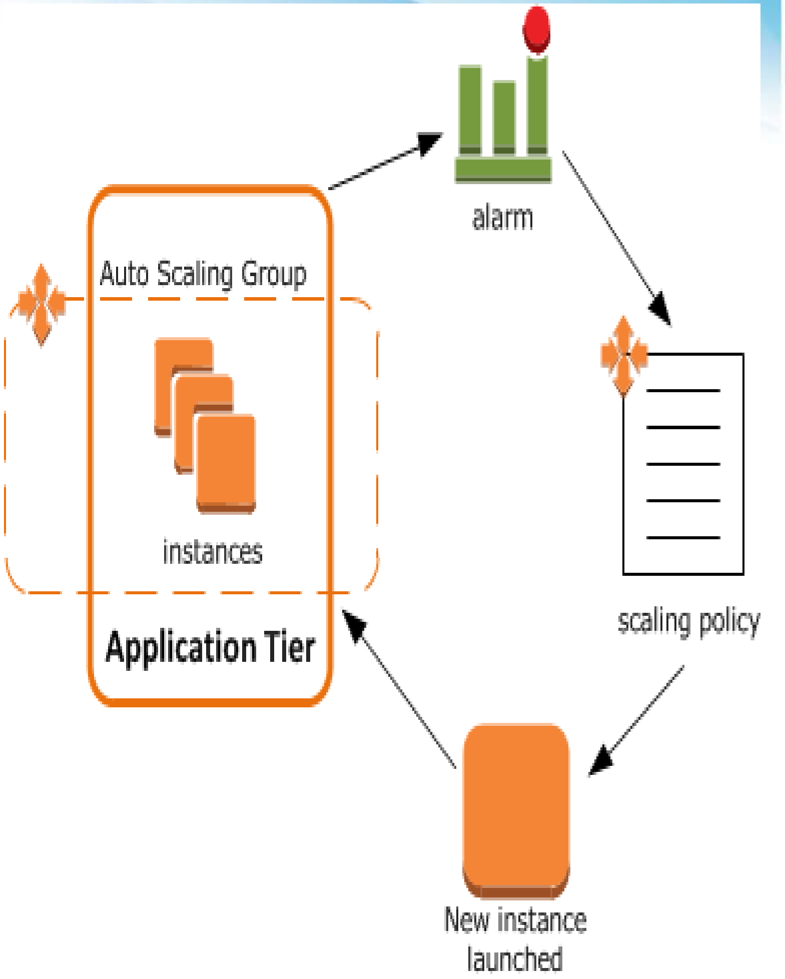
Instance ID	i-8c89ff76	Public DNS	ec2-52-1-235-14.compute-1.amazonaws.com
Instance state	running	Public IP	52.1.235.14
Instance type	t2.micro	Elastic IP	-
Private DNS	ip-172-31-36-84.ec2.internal	Availability zone	us-east-1a
Private IPs	172.31.36.84	Security groups	launch-wizard-1. view rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-5f2c583a	AMI ID	amzn-ami-hvm-2014.09.2.x86_64-ebs (ami-146e2a7c)
Subnet ID	subnet-25f36552	Platform	-

Actions on EC2 Instance

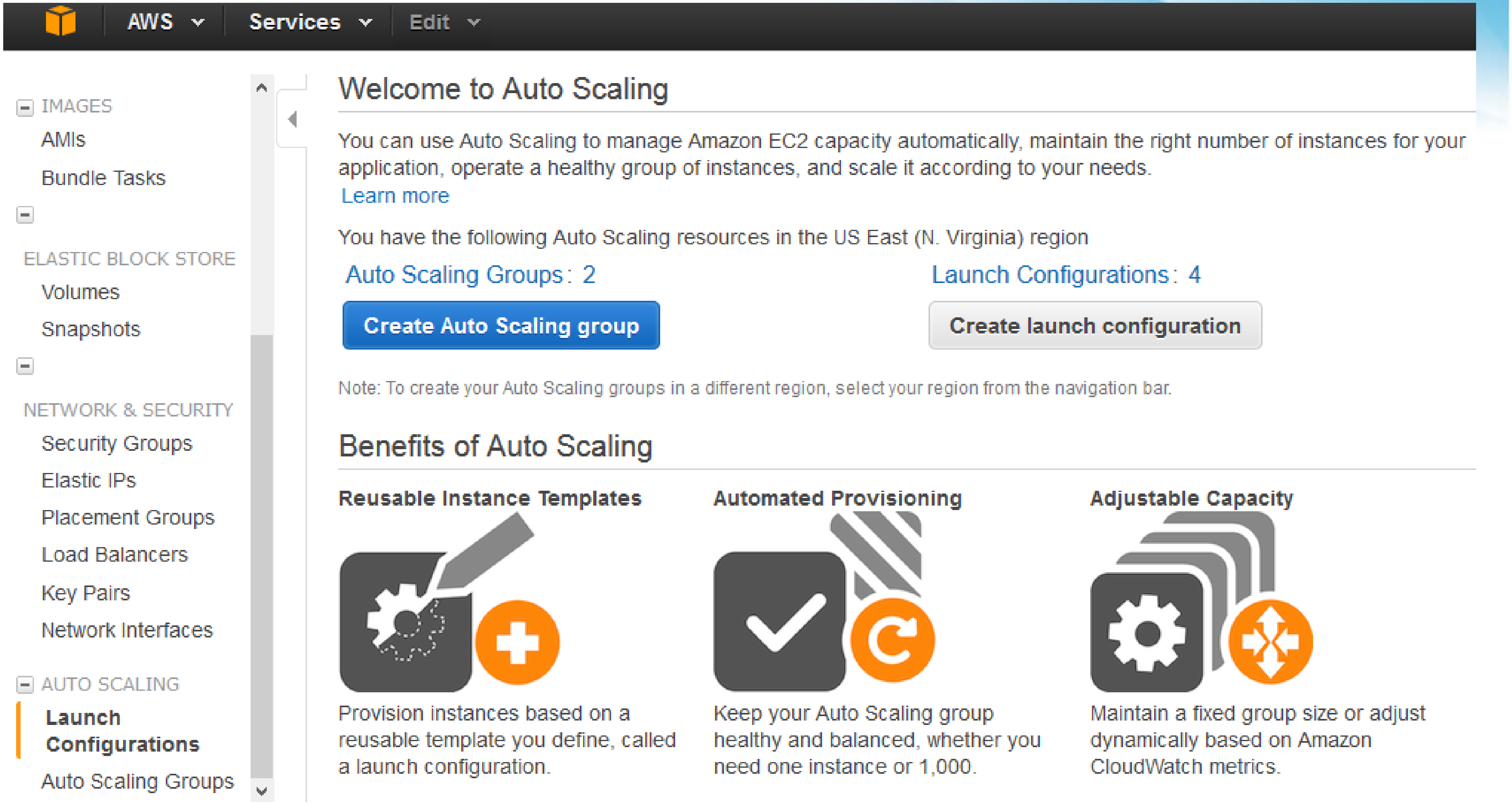
- Connect: must through ssh.
- Get Windows Password: only for windows image.
- Launch More Like This: use the exactly same setting.
- Instance State:
  - Stop: Suspend the VM. System state and data remain, but private IP will be released.
  - Reboot: Same as reboot a OS on a physical machine.
  - Terminate: Everything is lost, unless you have created a snapshot image for the VM. The data stored on EBS could remain.
- Instance Setting: Add/Edit Tag, Attach to Auto Scaling Group, etc.
- Image: create a snapshot image of the VM instance.
- Networking: change network setting
- CloudWatch Monitoring: Enable/Disable cloud watch, Add/Edit alarms.

EC2 Auto-Scaling Feature

- Scale out Amazon EC2 instances seamlessly and automatically when demand increases.
- Shed unneeded Amazon EC2 instances automatically and save money when demand subsides.
- Scale dynamically based on your Amazon **CloudWatch metrics**, or predictably according to a schedule that you define.
- Replace unhealthy or unreachable instances to maintain **higher availability** of your applications.



EC2 Auto-Scaling Feature



Welcome to Auto Scaling

You can use Auto Scaling to manage Amazon EC2 capacity automatically, maintain the right number of instances for your application, operate a healthy group of instances, and scale it according to your needs.

Learn more

You have the following Auto Scaling resources in the US East (N. Virginia) region

Auto Scaling Groups: 2

Launch Configurations: 4

Note: To create your Auto Scaling groups in a different region, select your region from the navigation bar.

Benefits of Auto Scaling

Reusable Instance Templates

Provision instances based on a reusable template you define, called a launch configuration.

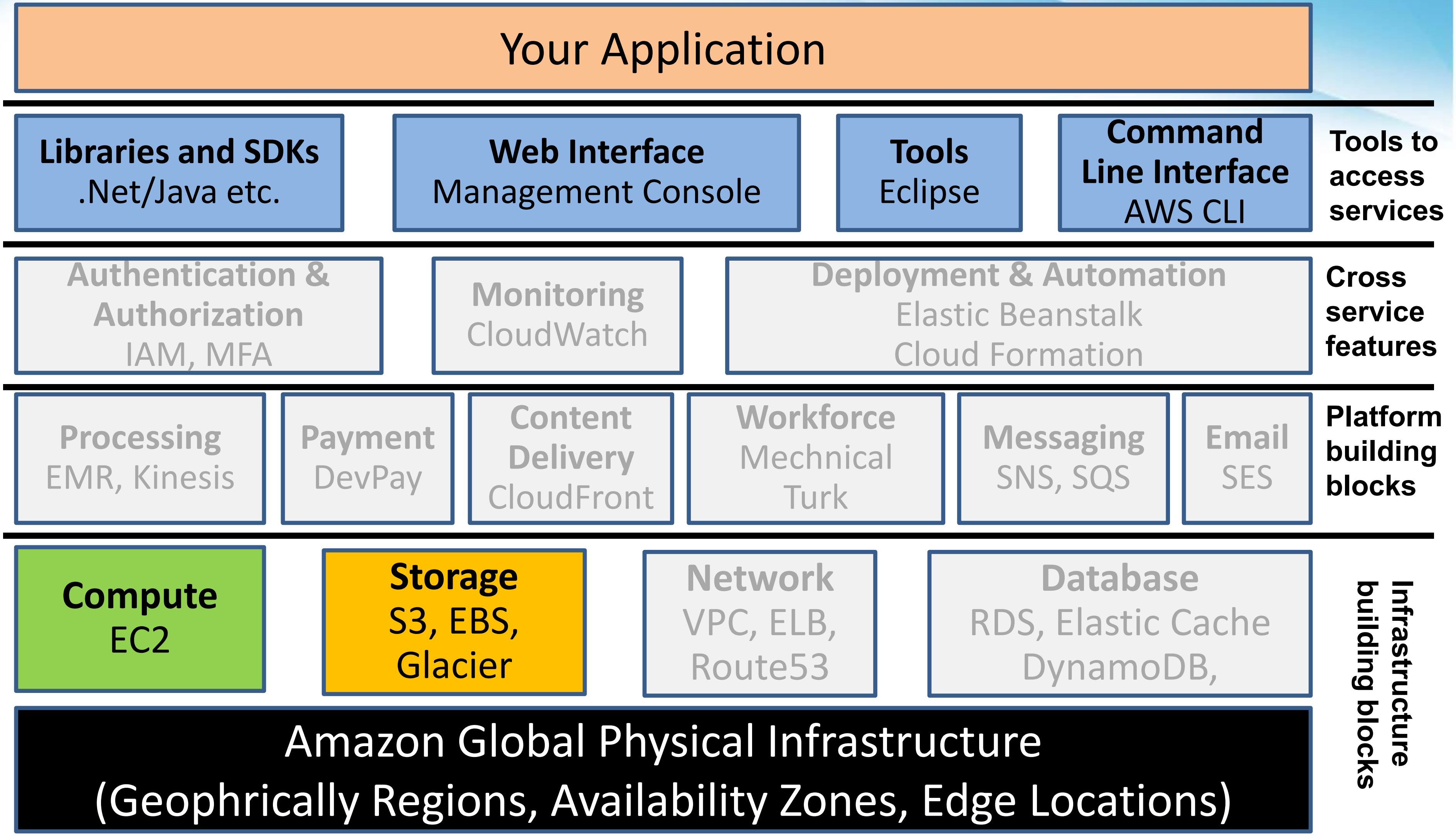
Automated Provisioning

Keep your Auto Scaling group healthy and balanced, whether you need one instance or 1,000.

Adjustable Capacity

Maintain a fixed group size or adjust dynamically based on Amazon CloudWatch metrics.

AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

Network
VPC, ELB, Route53

Database
RDS, Elastic Cache
DynamoDB,

Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

Infrastructure building blocks

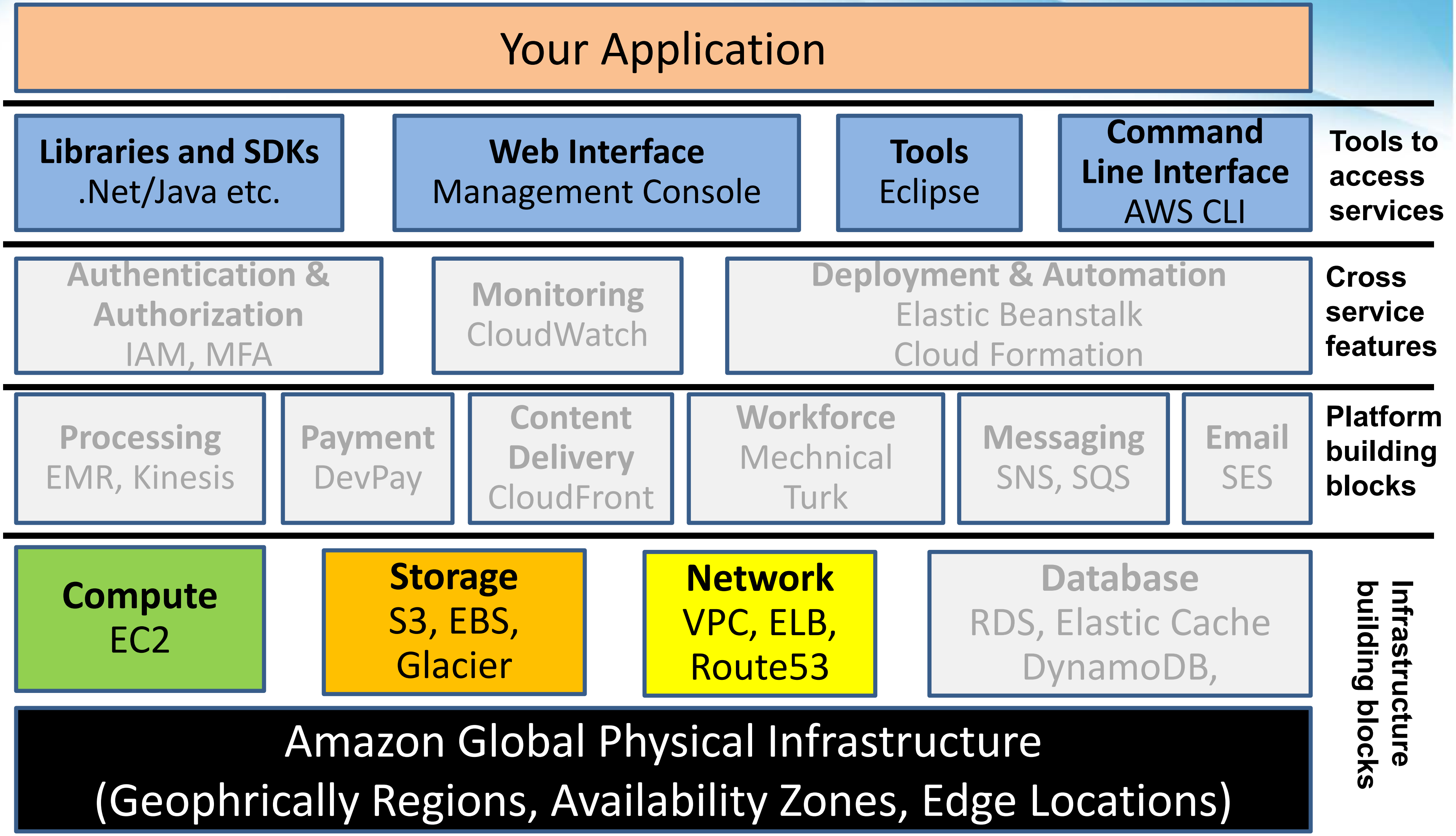
Storage Services

	Elastic Block Storage (EBS)	S3	Glacier
Storage type	**Block storage** (raw volume) unformatted	**File storage** (but only support upload and download)	**Archive storage**
Purpose	Used as a **hard drive** on the EC2 instance. It can be re-attached and elastically scaled.	very versatile and used for many different purposes, especially for **data collection and sharing**.	Dedicated for data **backup**. More suitable for data retrieval times of several hours.
Price	High	Medium	Low
IOPS	High (similar to local disk)	Medium (similar to remote disk)	Low (similar to tape)
Ref.	http://aws.amazon.com/ebs/	http://aws.amazon.com/s3/	http://aws.amazon.com/glacier/

S3

- A distributed store
  - High available **key-value** structured storage system.
  - Based on **distributed hash table.**
  - Support **data replication** and **versioning**.
  - Use **sloppy Quorum** to provide data consistency.
  - It is the storage backend of many web apps, including dropbox.
  - Free window client for S3: http://s3browser.com/
  - Publication: "Dynamo". *Proceedings of ACM SOSP '07*. p. 205.
- **Buckets** and **Objects** are the resources of S3
  - **Bucket** is created on a selected region, but its name must be globally unique.
  - **Object** refers to a file, and stored under a bucket.
  - **Folder** can be created under buckets, but it only exist logically in namespace. S3 uses the full path as the object key.

AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

Network
VPC, ELB, Route53

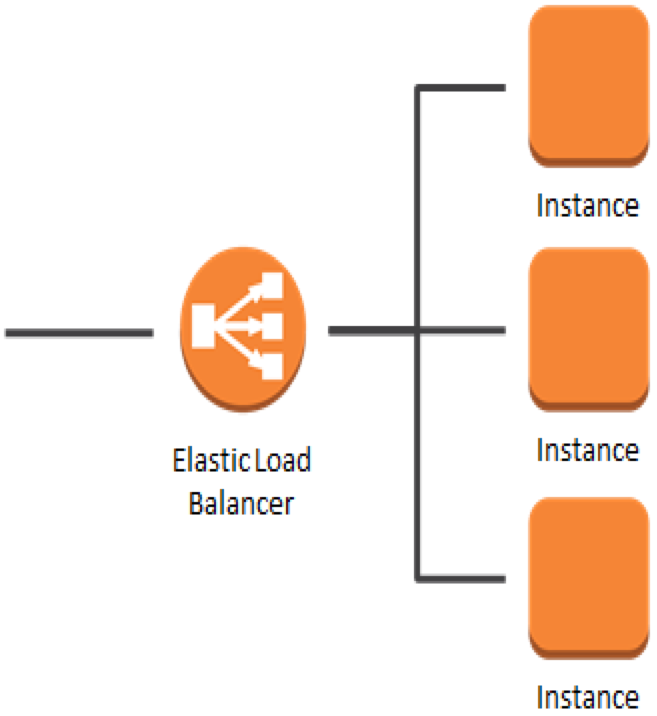
Database
RDS, Elastic Cache
DynamoDB,

Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

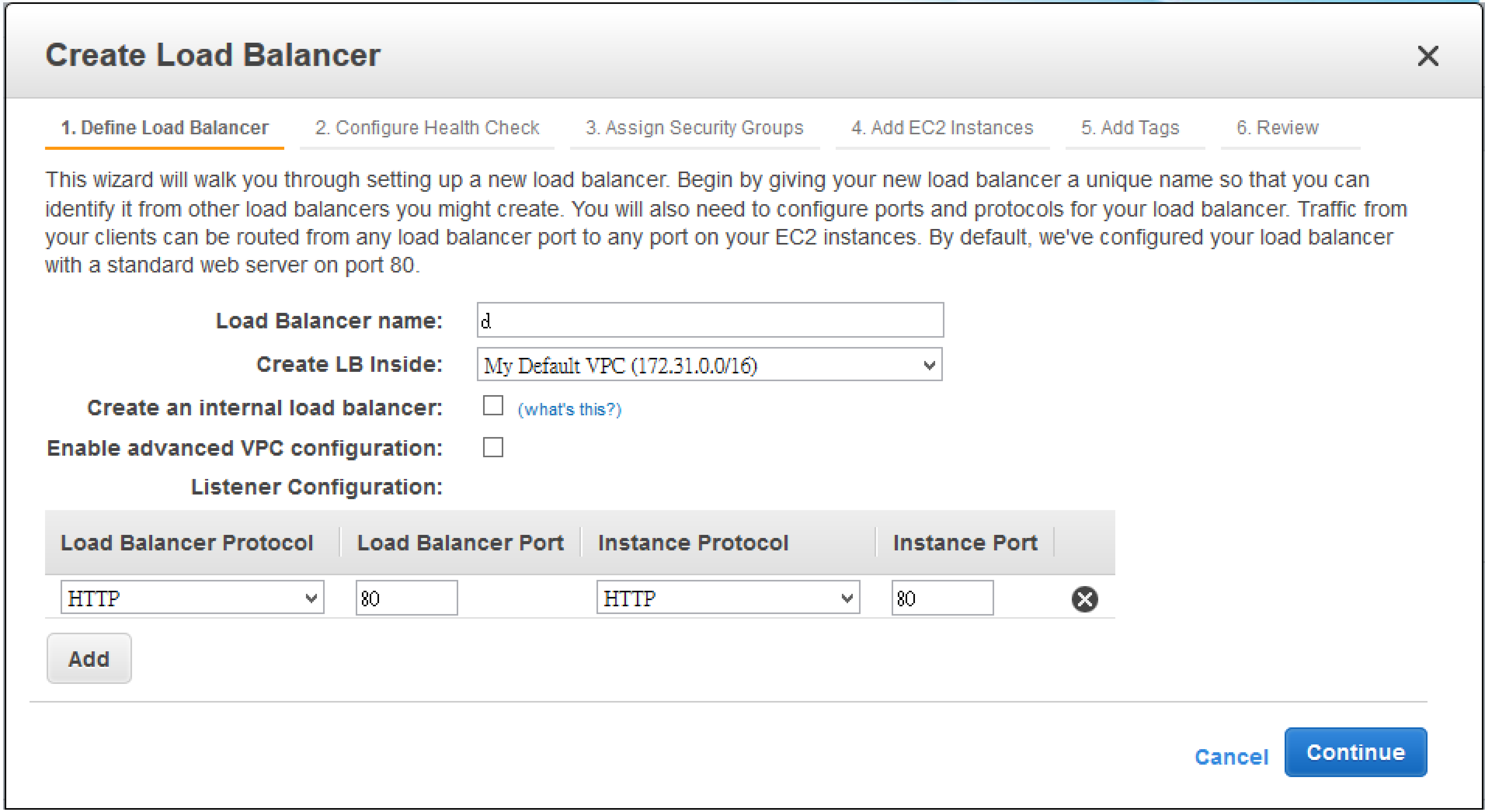
Infrastructure building blocks

Networking Services: ELB

- Elastic Load Balancing (ELB)
  - automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud.
- Benefits
  - **Available**: automatically route traffic across multiple instances and multiple Availability Zones. It ensures only healthy Amazon EC2 instances receive traffic.
  - **Elastic**: automatically scales its request handling capacity to meet the demands of application traffic.



Networking Services: ELB



Create Load Balancer

1. Define Load Balancer 2. Configure Health Check 3. Assign Security Groups 4. Add EC2 Instances 5. Add Tags 6. Review

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name:	d
Create LB Inside:	My Default VPC (172.31.0.0/16)
Create an internal load balancer:	□ (what's this?)
Enable advanced VPC configuration:	□
Listener Configuration:	

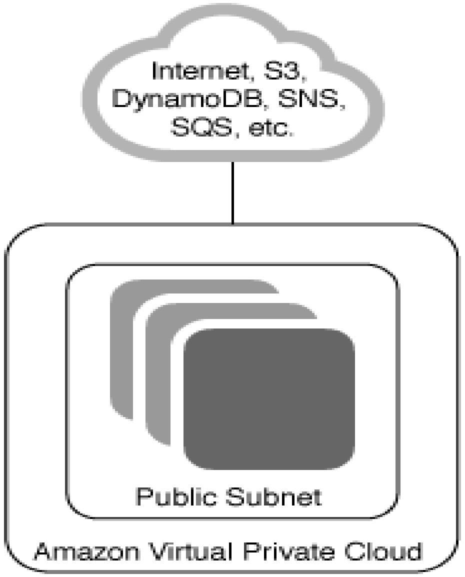
Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	
HTTP	80	HTTP	80	⊗

Networking Services: Route 53

- A highly available and scalable cloud Domain Name System (**DNS**) web service
- Effectively **connects user requests to infrastructure running in AWS** – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets
- You can use Amazon Route 53 to configure DNS health checks to **route traffic to healthy endpoints** or to independently monitor the health of your application and its endpoints.

Networking Services: VPC

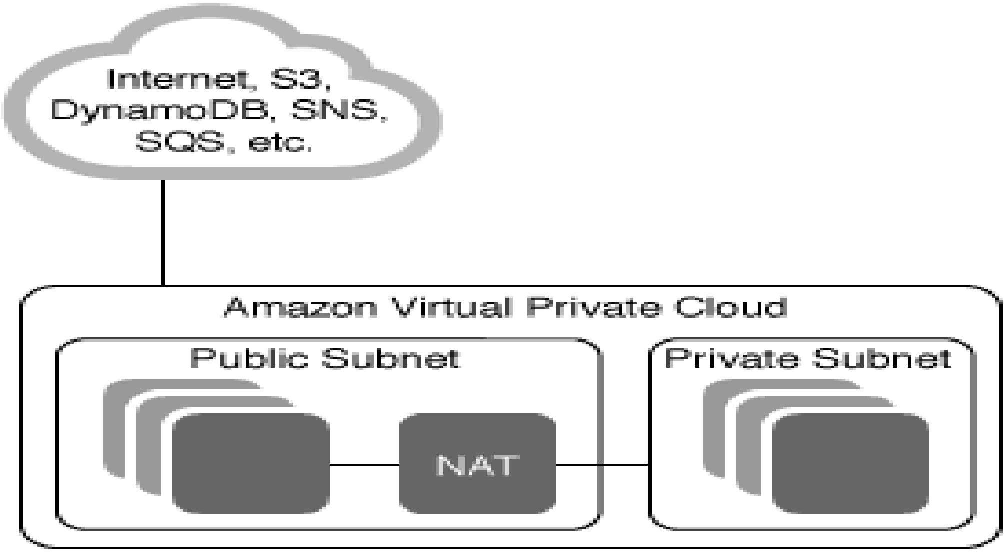
- Amazon Virtual Private Cloud(VPC) lets you provision a **logically isolated** section of the AWS cloud where you can launch AWS resources **in a virtual network that you define**.
- Four VPC configuration options:
  1) VPC with a Single Public Subnet:
     - **Host a basic one-tier web application**
     - Allow the webserver to respond to inbound requests while simultaneously prohibiting the webserver from initiating outbound connections.


Internet, S3, DynamoDB, SNS, SQS, etc.
Public Subnet
Amazon Virtual Private Cloud

Networking Services: VPC

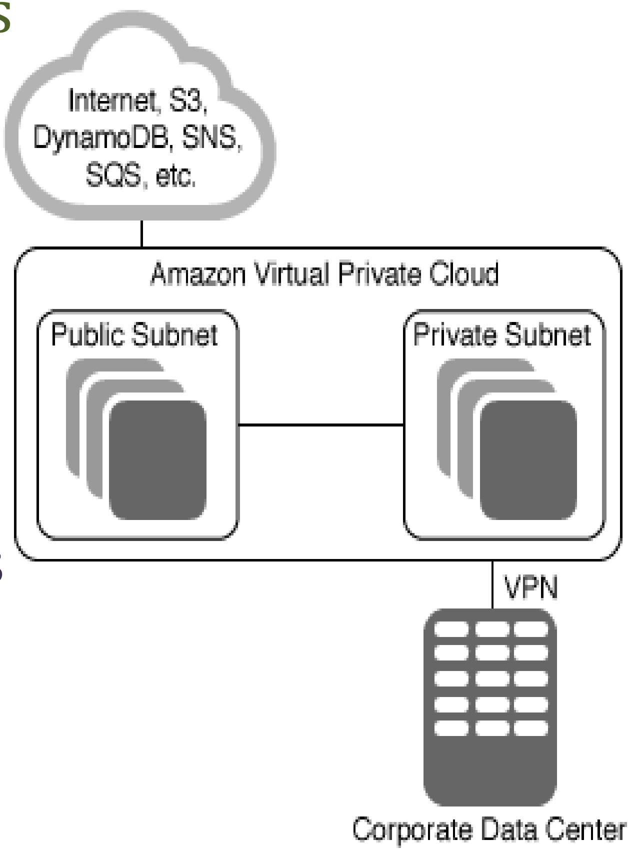
2) VPC with Public and Private Subnets:

- **Host multi-tier web applications**
- Launch webservers in a publicly accessible subnet and launch application servers and databases in non-publically accessible subnets.


Internet, S3, DynamoDB, SNS, SQS, etc.
Amazon Virtual Private Cloud
Public Subnet
NAT
Private Subnet

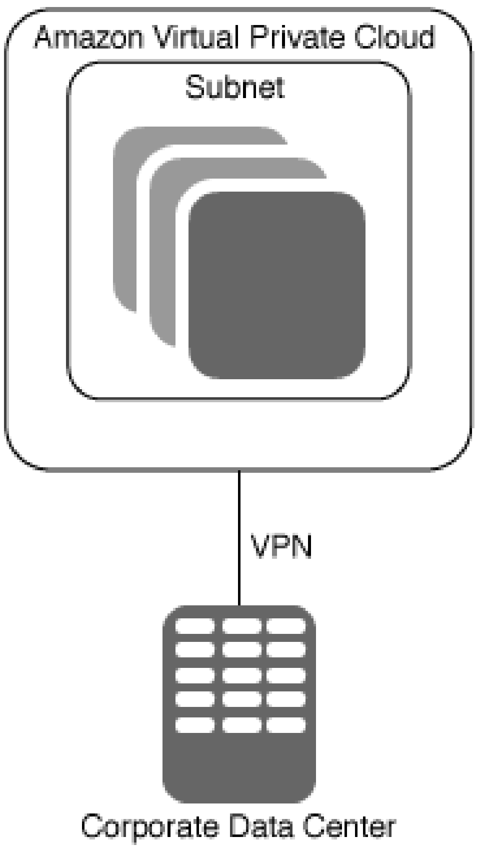
Networking Services: VPC

3) VPC with Public and Private Subnets and Hardware VPN Access
   - **Host scalable web applications in the AWS cloud that are connected to your datacenter.**
   - An IPsec VPN connection between your VPC and your corporate network
   - Webservers and application servers in your VPC can leverage Amazon EC2 elasticity and Auto Scaling features to grow and shrink as needed.

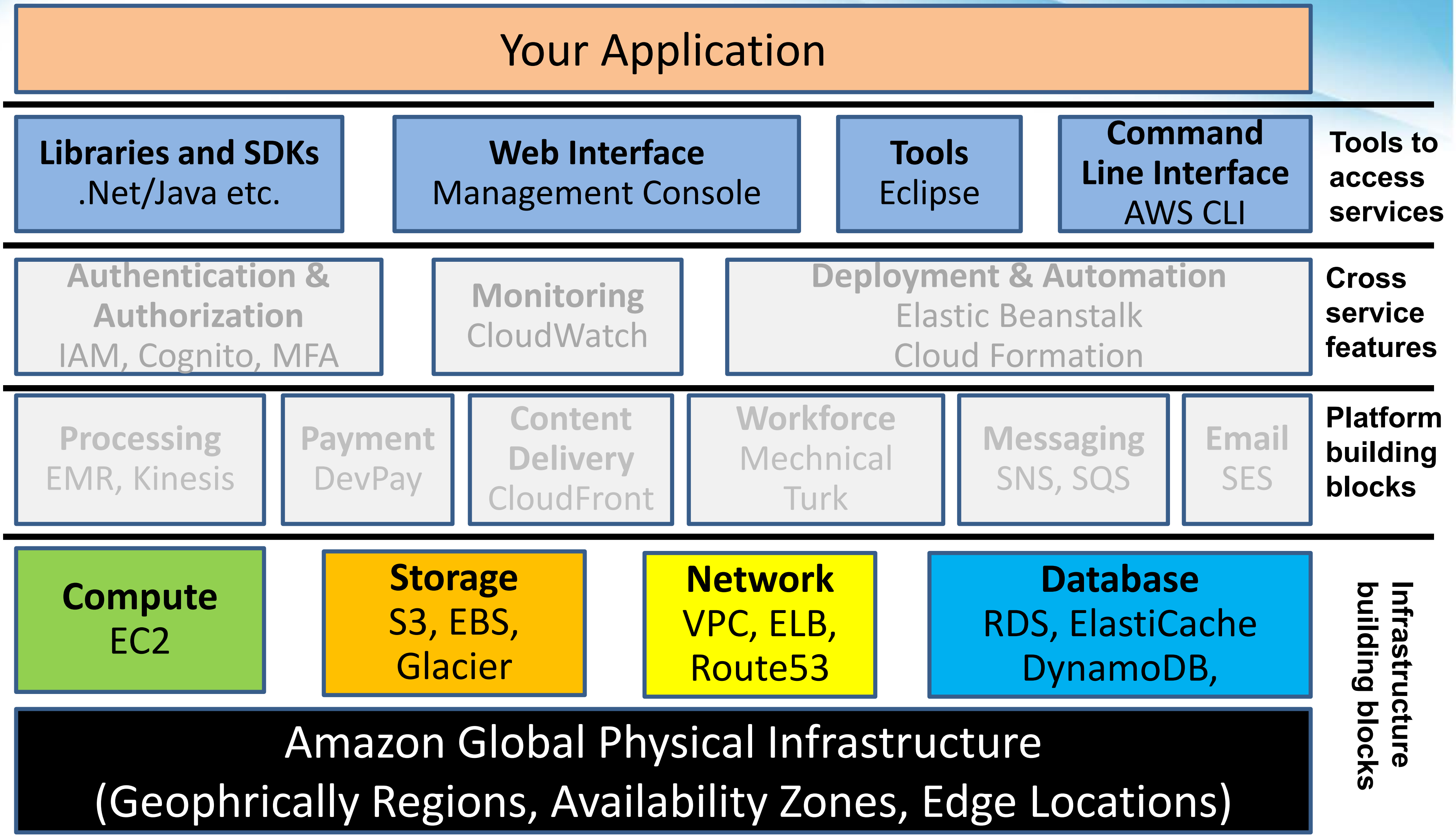

Internet, S3, DynamoDB, SNS, SQS, etc.
Amazon Virtual Private Cloud
Public Subnet
Private Subnet
VPN
Corporate Data Center

Networking Services: VPC

4) VPC with Private Subnets Only and Hardware VPN Access
   - **Extend your corporate network into the cloud.**
   - Add more compute capacity to your network by connecting your VPC to your corporate network.



AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, Cognito, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

Network
VPC, ELB, Route53

Database
RDS, ElastiCache
DynamoDB,

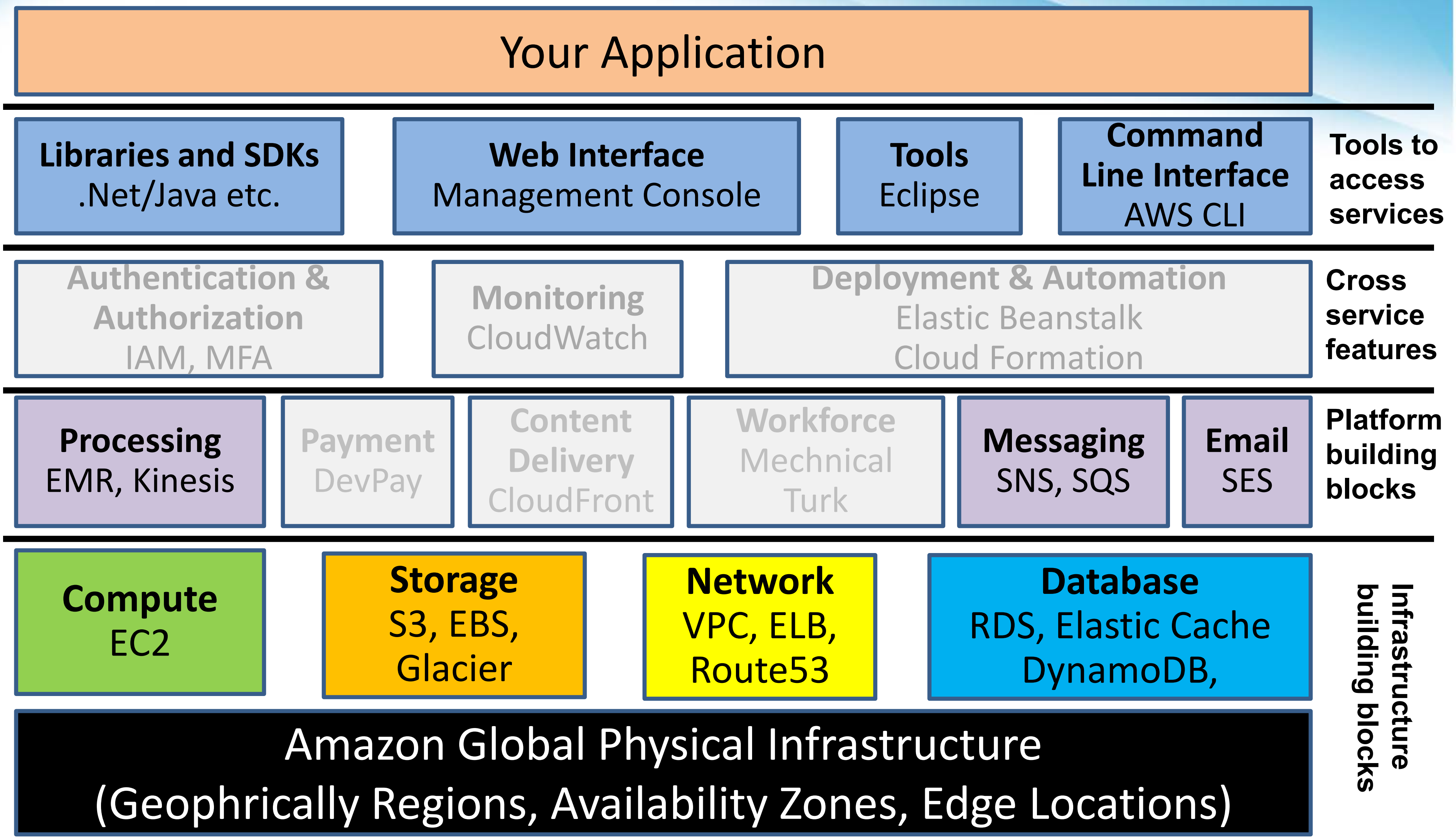
Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

Infrastructure building blocks

Database Services

- RDS:
  - **Traditional relational SQL database system.**
  - Table has a predefined schema.
  - All records stored in the table must have the same set of columns.
  - **Fast response time.**
- DynamoDB:
  - **NoSQL database system.**
  - Schema free for unstructured or semi-structured data.
  - Operate by its own defined API not SQL.
  - Better **scalability** than RDS.
- ElastiCache
  - **A in-memory cache web service.**
  - supports two open-source in-memory caching engines: **Memcached** & **Redis.**
  - Only support simple **key-value** store.

AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

Network
VPC, ELB, Route53

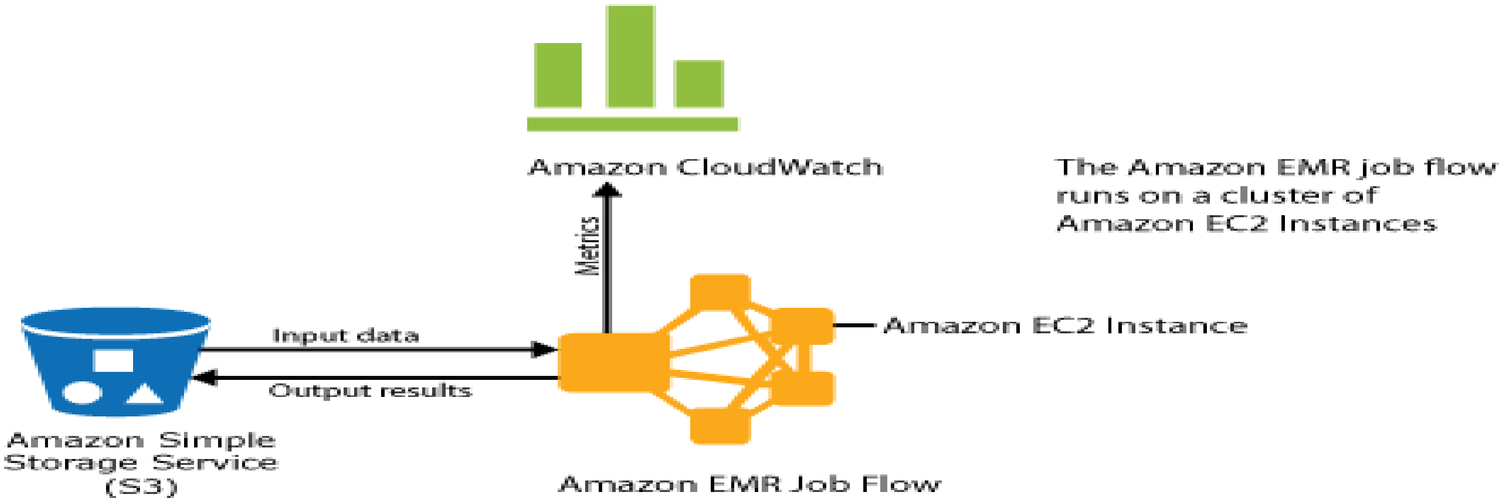
Database
RDS, Elastic Cache
DynamoDB,

Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

Infrastructure building blocks

EMR Service

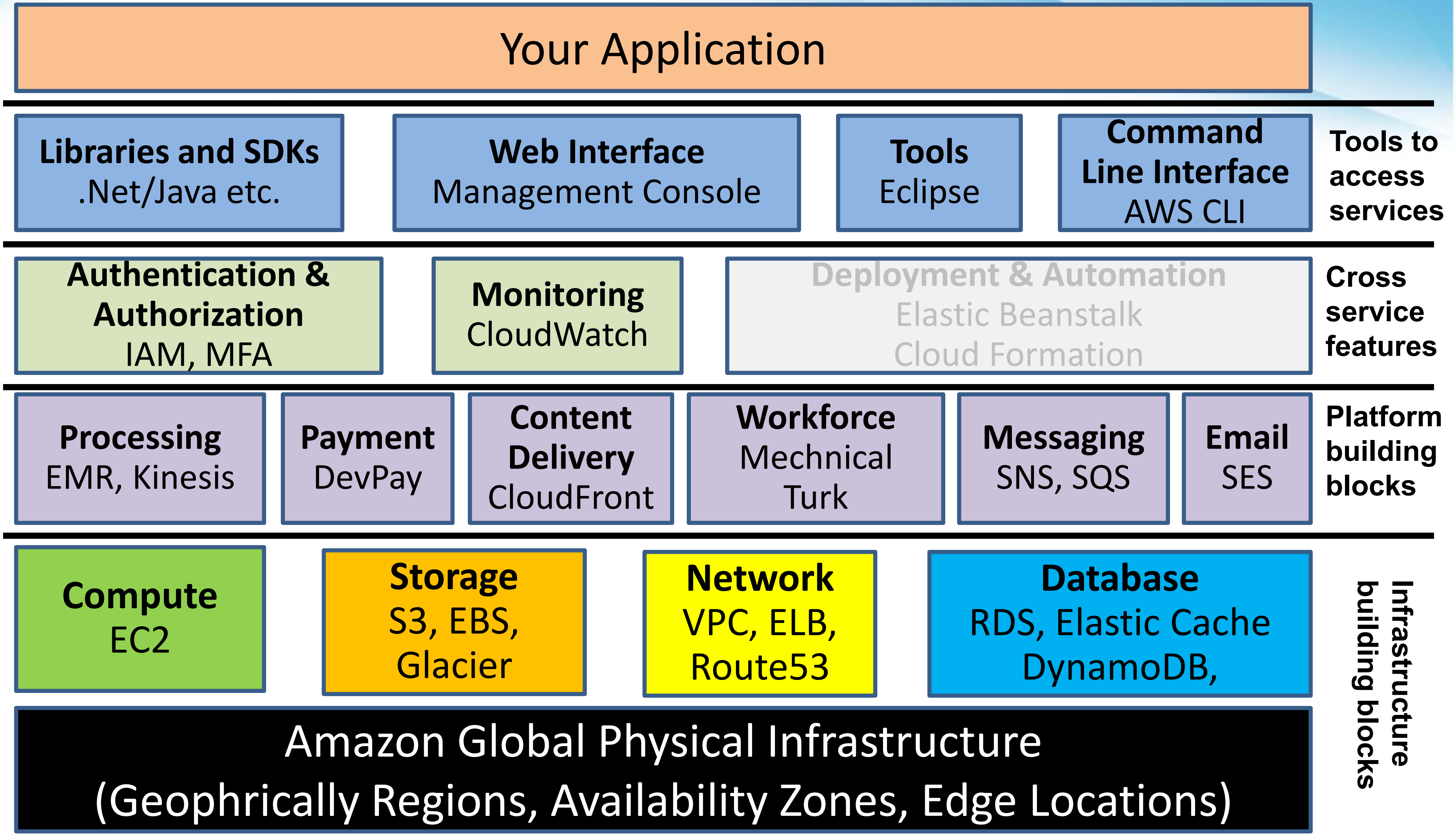
- Elastic MapReduce
  - A scalable **distributed parallel data processing engine.**
  - Use Hadoop, a open source framework of **Map Reduce**.
  - It can also run other distributed frameworks including **Spark.**
  - **Combined with auto-scaling** to improve cost effectiveness.
  - Allow data to be ingested from all Amazon storage services including S3, Glacier, DynamoDB, RDS, etc.


Amazon CloudWatch
Metrics
The Amazon EMR job flow runs on a cluster of Amazon EC2 Instances
Amazon EC2 Instance
Input data
Output results
Amazon Simple Storage Service (S3)
Amazon EMR Job Flow

Messaging & Email Services

- Simple Queue Service (SQS)
  - Transmit any volume of data, at any level of throughput, without losing messages or requiring other services to be always available.
  - Basic operations: create queue, send/recv message.
  - Can be used to connect service components, and construct more complex data processing workflow.
- Simple Notification Service (SNS)
  - Push messaging service.
  - Directly push notification to mobile devices, SMS text message, email or SQS queues.
  - To prevent messages from being lost, all messages published to SNS are stored redundantly **across multiple availability zones**.
- Simple Email Service (SES)
  - Email-sending service

AWS Cloud



Your Application

Libraries and SDKs
.Net/Java etc.

Web Interface
Management Console

Tools
Eclipse

Command Line Interface
AWS CLI

Tools to access services

Authentication & Authorization
IAM, MFA

Monitoring
CloudWatch

Deployment & Automation
Elastic Beanstalk
Cloud Formation

Cross service features

Processing
EMR, Kinesis

Payment
DevPay

Content Delivery
CloudFront

Workforce
Mechnical Turk

Messaging
SNS, SQS

Email
SES

Platform building blocks

Compute
EC2

Storage
S3, EBS, Glacier

Network
VPC, ELB, Route53

Database
RDS, Elastic Cache DynamoDB,

Amazon Global Physical Infrastructure
(Geophrically Regions, Availability Zones, Edge Locations)

Infrastructure building blocks

Monitoring Service: CloudWatch

- Gain system-wide visibility into resource utilization, application performance, and operational health.

- Features & Benfits

1) Monitor AWS resources
   - Resources including EC2 instance, SQS queues, DB table, etc.
   - View metrics for CPU utilization, data transfer, disk usage activity, queue length, etc.
   - Only charge money for higher resolution and metric aggregation.
2) Monitor custom metrics
   - Submit custom metrics generated by your own applications via a simple API request and have them monitored by Amazon CloudWatch.

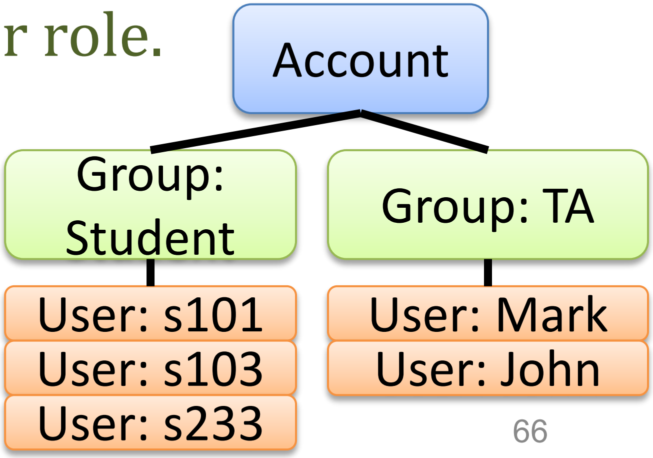
Monitoring Service: CloudWatch

3) Monitor and Store Logs
   - use CloudWatch Logs to monitor and troubleshoot your systems and applications using your existing system, application, and custom log files.
4) Set Alarms
   - Set alarms on any of your metrics to send you notifications or take other automated actions. For example, when a specific Amazon EC2 metric crosses your alarm threshold, you can use Auto Scaling to dynamically add or remove EC2 instances or send you a notification.
5) View Graphs and Statistics
   - View graphs and statistics for any of your metrics on the Amazon CloudWatch dashboard.

Administration Services: IAM

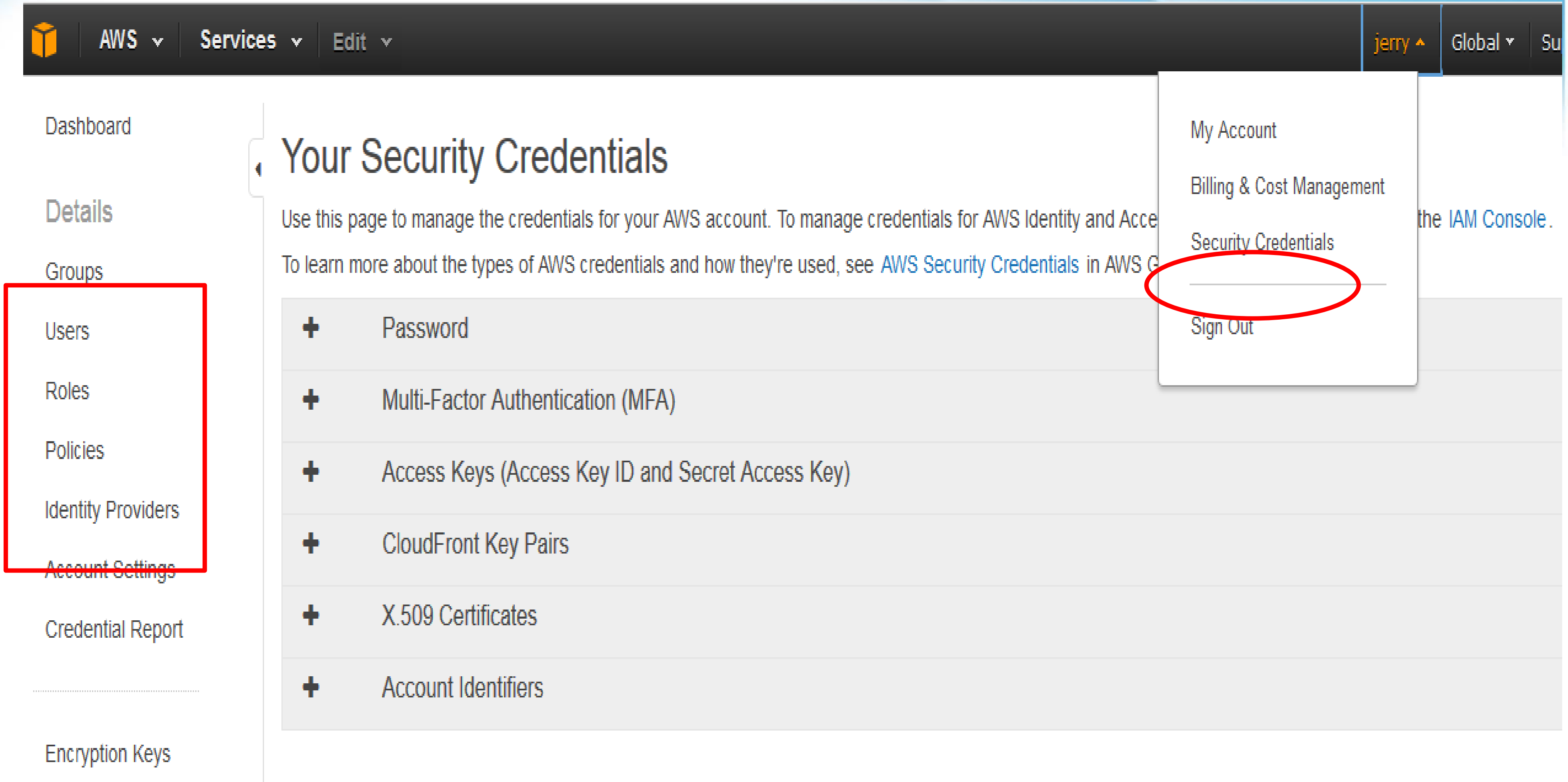
- AWS Identity and Access Management (IAM)
  - Enable **securely control access to AWS services and resources**.
- Basic term in IAM
  - **User:** an entity that you create to interact with AWS.
  - **Group:** a collection of IAM users for management.
  - **Role:** an entity that can be acquired by **temporary security credentials**.
  - **Policy:** define permissions on a resource for an AWS account (the "root" user), an IAM user, group, or role.

Roles



reference: http://aws.amazon.com/iam/

IAM Dashboard



Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Acce the IAM Console.

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials in AWS G

- ✚ Password
- ✚ Multi-Factor Authentication (MFA)
- ✚ Access Keys (Access Key ID and Secret Access Key)
- ✚ CloudFront Key Pairs
- ✚ X.509 Certificates
- ✚ Account Identifiers

How and when to use those credentials:
http://docs.aws.amazon.com/AWSSecurityCredentials/1.0/AboutAWSCredentials.html

IAM: Account, User, Group

- Account:
  - The root and default administrator with all permissions.
  - **Billing is charged for the whole account**.
  - Has an unique 12-digit AWS Account ID
- User:
  - **Access key pair** (Access Key ID, Secret Access Key): for identification when access any AWS services or resources. Must download in creation.
  - **Password**: use an email address and password to sign in AWS console webpage
  - Root account has a few other credentials. E.g.: X.509 certification, CloudFront key pair, etc.
- Group:
  - Control a set of users together in group.
  - A user can belong to multiple groups.

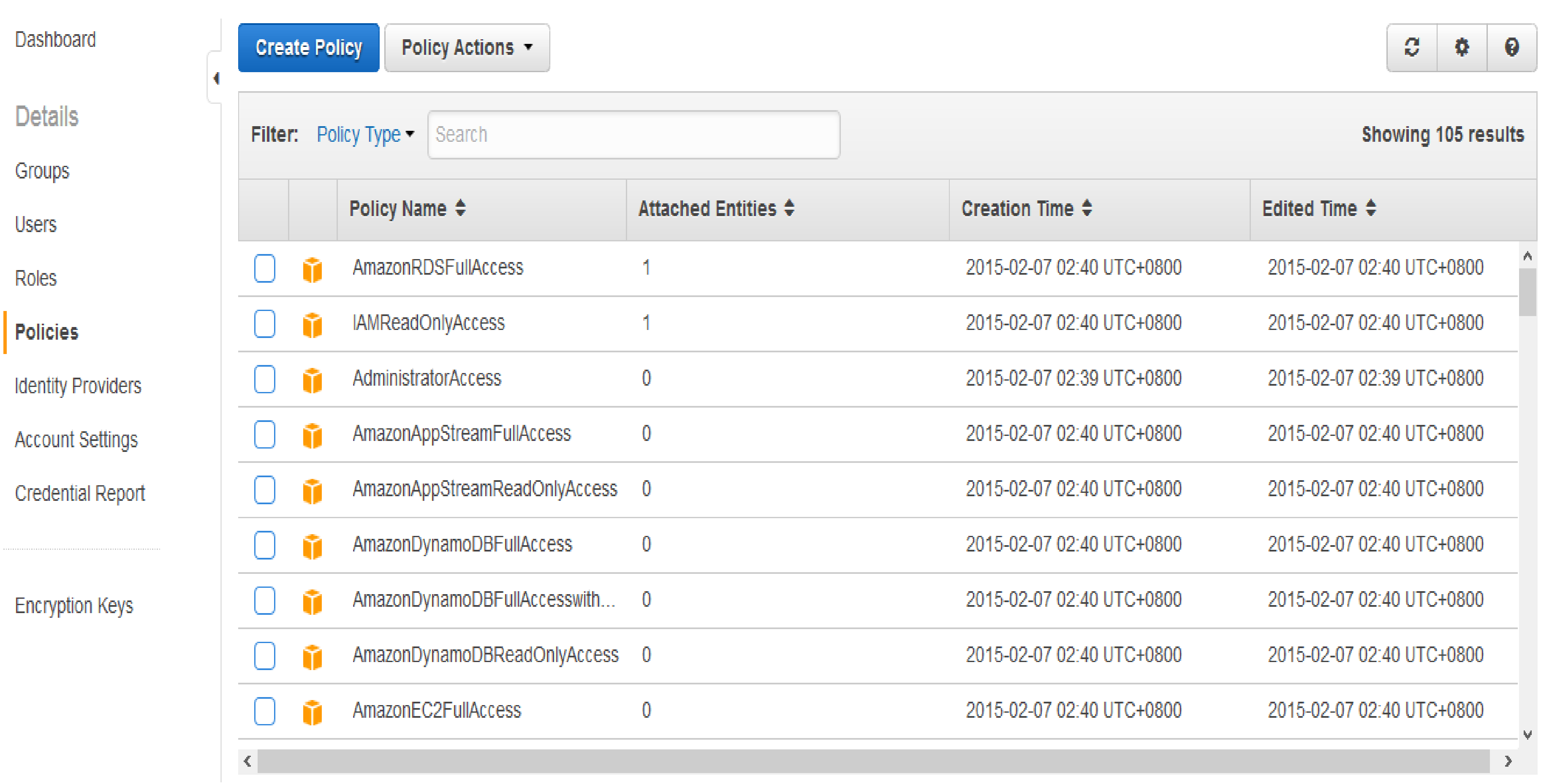
IAM: Policies & Permissions

- Use policy to assign permissions to a **user, group or role**:
  - **Action**: Which AWS actions you allow. E.g.: allow a user to call the Amazon S3 ListBucket action.
  - **Resources**: Which AWS resources you allow the action on. E.g.: which buckets will you allow to perform the action.
  - **Effect**: Whether to **allow or deny** access.
  - **Conditions**: Which conditions must be present for the policy to take effect. E.g.: user must connecting from a specific IP range
- Policy is specified in JSON format:

```
{"Version":"2012-10-17",
 "Statement":[{
    "Effect":"Allow",
    "Action":"s3:ListBucket",
    "Resource":"arn:aws:s3:::example_bucket"
   }]
}
```

IAM: Policies & Permissions

- A set of policies has been **pre-defined** by Amazon.



Filter: Policy Type ▾ Search

Showing 105 results

		Policy Name	Attached Entities	Creation Time	Edited Time
□		AmazonRDSFullAccess	1	2015-02-07 02:40 UTC+0800	2015-02-07 02:40 UTC+0800
□		IAMReadOnlyAccess	1	2015-02-07 02:40 UTC+0800	2015-02-07 02:40 UTC+0800
□		AdministratorAccess	0	2015-02-07 02:39 UTC+0800	2015-02-07 02:39 UTC+0800
□		AmazonAppStreamFullAccess	0	2015-02-07 02:40 UTC+0800	2015-02-07 02:40 UTC+0800
□		AmazonAppStreamReadOnlyAccess	0	2015-02-07 02:40 UTC+0800	2015-02-07 02:40 UTC+0800
□		AmazonDynamoDBFullAccess	0	2015-02-07 02:40 UTC+0800	2015-02-07 02:40 UTC+0800
□		AmazonDynamoDBFullAccesswith...	0	2015-02-07 02:40 UTC+0800	2015-02-07 02:40 UTC+0800
□		AmazonDynamoDBReadOnlyAccess	0	2015-02-07 02:40 UTC+0800	2015-02-07 02:40 UTC+0800
□		AmazonEC2FullAccess	0	2015-02-07 02:40 UTC+0800	2015-02-07 02:40 UTC+0800

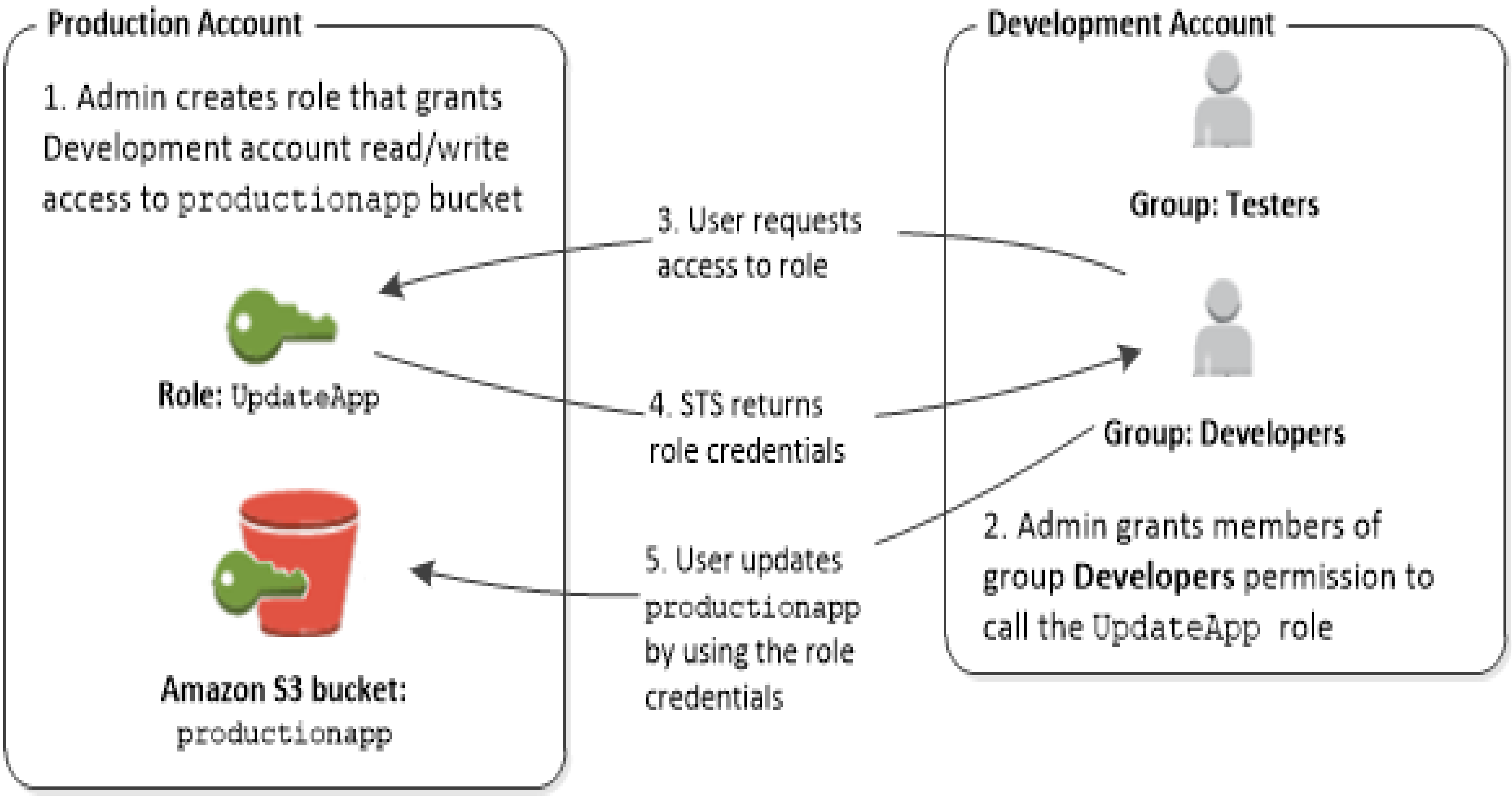
IAM: Roles

- Purpose: delegate access to users, applications, or services that don't normally have access to your AWS resources.
  - allow a mobile app to use AWS resources **without storing AWS keys within the app**.
  - give **users who already have identities outside of AWS**, such as through your corporate directory, access to AWS resources, that is, create **federated identities**.
  - **grant access to your account to a third party**, for example, so that they can perform an **audit** on your resources.

IAM: Roles

- There are two ways to use roles
  - In the IAM console: selects the Switch Role option in the Identity menu.
  - Programmatically in the AWS CLI or API: An application or **AWS service (like EC2)** can *assume* a role by requesting **temporary security credentials, and then** make programmatic requests to AWS.
- A role is created by specifying two separate policies:
  - The trust policy, which specifies who is allowed to assume the role.
  - The permissions policy, which defines what actions and resources the principal is allowed to use.

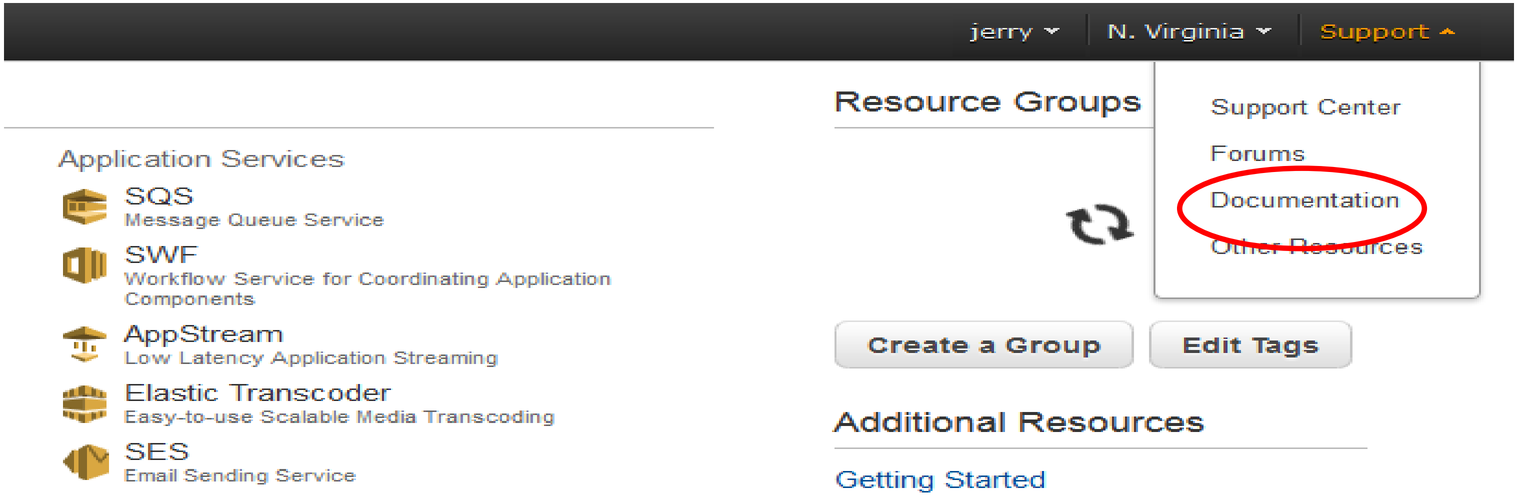
IAM: Roles



Production Account
1. Admin creates role that grants Development account read/write access to productionapp bucket
Role: UpdateApp
Amazon S3 bucket: productionapp
3. User requests access to role
4. STS returns role credentials
5. User updates productionapp by using the role credentials
Development Account
Group: Testers
Group: Developers
2. Admin grants members of group **Developers** permission to call the UpdateApp role

How to Self-Learn a Service?

- Documentation page: https://aws.amazon.com/documentation/
  - Getting start guide
  - Developer/User guide
  - API and CLI reference
- Product page: http://aws.amazon.com/products/
  - Basic introduction & pricing information.



Outline

- Introduction of Cloud Services
- Introduction of Amazon AWS
- AWS Global Infrastructure
- AWS Tools for Accessing Services
- AWS Services
- AWS Use Cases

Common Use Cases

- Web site hosting
- Application hosting/SaaS hosting
- Mobile and Social Applications
- Internal IT application hosting
- Content delivery and media distribution
- High performance computing, batch data processing and large scale analytics
- Storage, backup, and disaster recovery
- Development and test environments

NASA - Mission Data Processing

Challenge

Because of the latency of data transmission from and to Mars, during a 2 hour window, it took mission planners 90 minutes to process telemetry data from the Mars Rover, 20 minutes to decide where to move the Rover to, and 10 minutes to up load the data.

Solution

NASA-JPL, loading their custom software application on Amazon EC2, was able to horizontally scale the number of virtual machines supporting the data processing.

Benefit

- Reduced data processing time from 90 minutes to 15 minutes using parallel processing.
- Increased mission planning time, resulting in higher quality scientific observations.

(all data provided by NASA)

More AWS Customer Success Stories

htc	Expedia	Pinterest	FINRA
CONDÉ NAST	dropcam	airbnb	NOVARTIS
Bristol-Myers Squibb	FDA	CITRIX	
Adobe	DOW JONES	NETFLIX	SUNCORP

- http://aws.amazon.com/solutions/case-studies/