Amazon Web Services

objective: complement/reinforce lectures with hands-on using Amazon Cloud

handouts: overview + step-by-step lab exercises

- AWS01: Amazon Web Services
- AWS02: AWS Core Services
- AWS03: Cloud Service Models: IaaS, PaaS, SaaS, FaaS
- AWS04: AWS MapReduce

2 Lab assignments (20%)

AWS01: Amazon Web Services

Outline

- Regions and Availability Zones
- AWS Login and Identity and Access Management (IAM)
- Access and Manage AWS Services
  - AWS Management Console
  - AWS Command Line Interface (CLI)
- Summary
- Videos & References

Amazon Web Services

“Amazon Web Services (**AWS**), a subsidiary of Amazon.com, offers a suite of cloud-computing services that make up an on-demand computing platform. These services operate from multiple geographical regions across the world.”

Amazon Web Services (AWS)

- a collection of remote computing services (> 175 services) that together make up a cloud computing platform

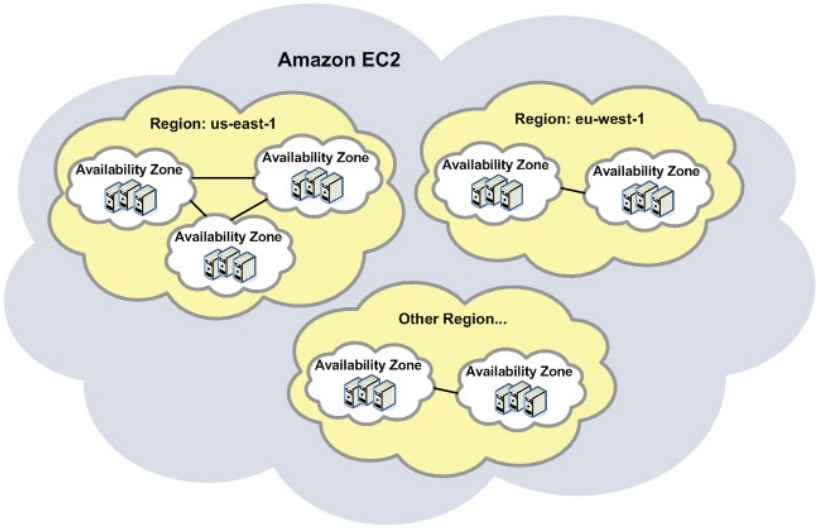
- Some basic services: [covered in AWS02]
  - Compute: EC2 (Elastic Computing Cloud)
  - Storage: S3 (Simple Storage Service)
  - Database: RDS (Relational Database Service)
  - Network: VPC (Virtual Private Cloud)

  - Scaling: Auto-scaling
  - Load balancing: Elastic Load Balancing
  - Monitoring services: Cloudwatch

How Services are Organized?

Regions and Availability Zones

- offers cloud services through a network of datacenters organized as:
  - **Regions:** distributed across different geographical areas and communicate through **Internet** but **do not share resources**
    - application can be closer to specific customers or to meet (data) legal requirements



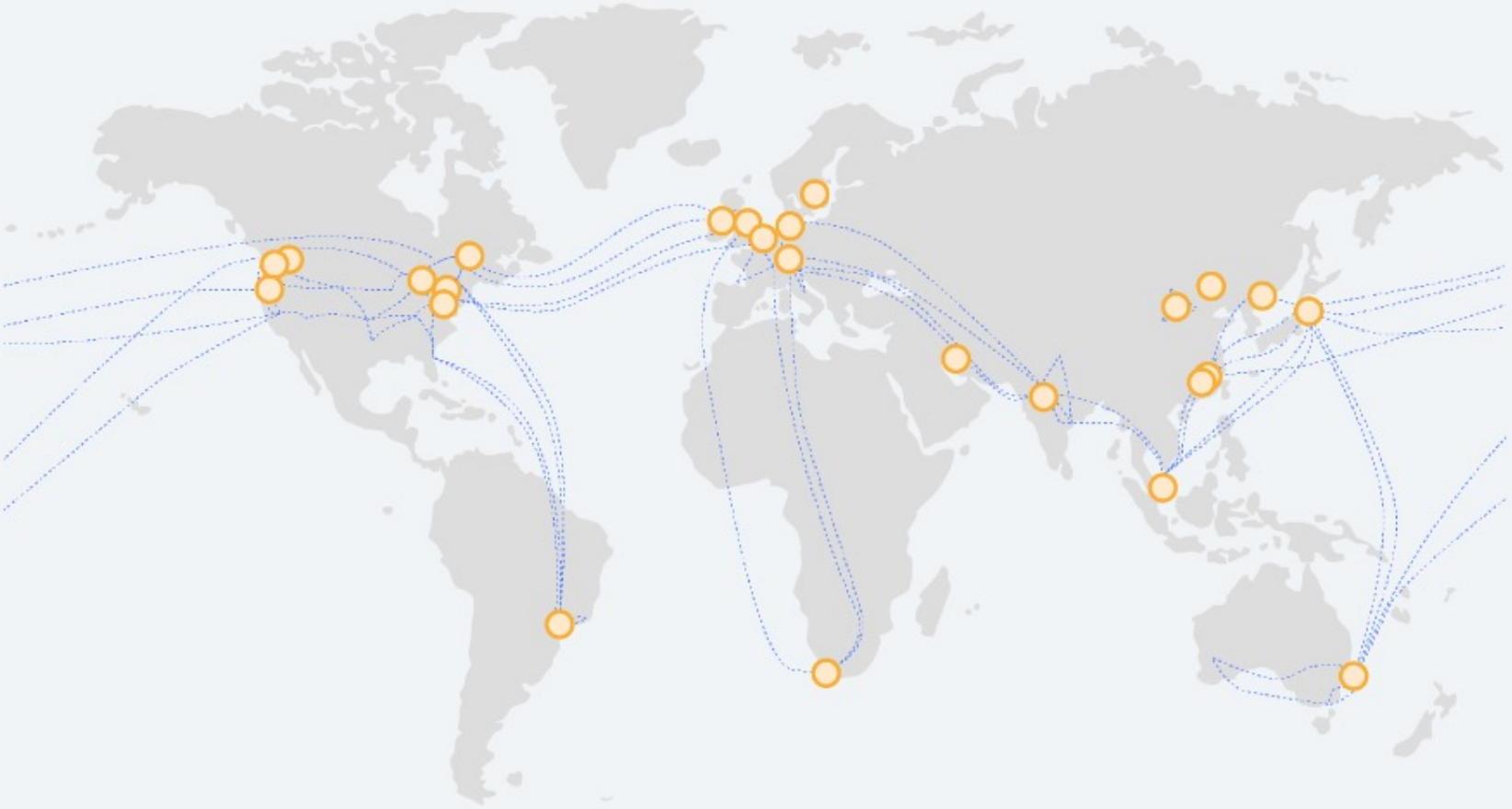
  - **Availability zones:** each region is divided into availability zones interconnected by **high-speed networks**
    - a data center consisting of a large number of servers
    - distinct locations within a region
    - protect applications from failure of a single location

Example: Global Network of AWS Regions

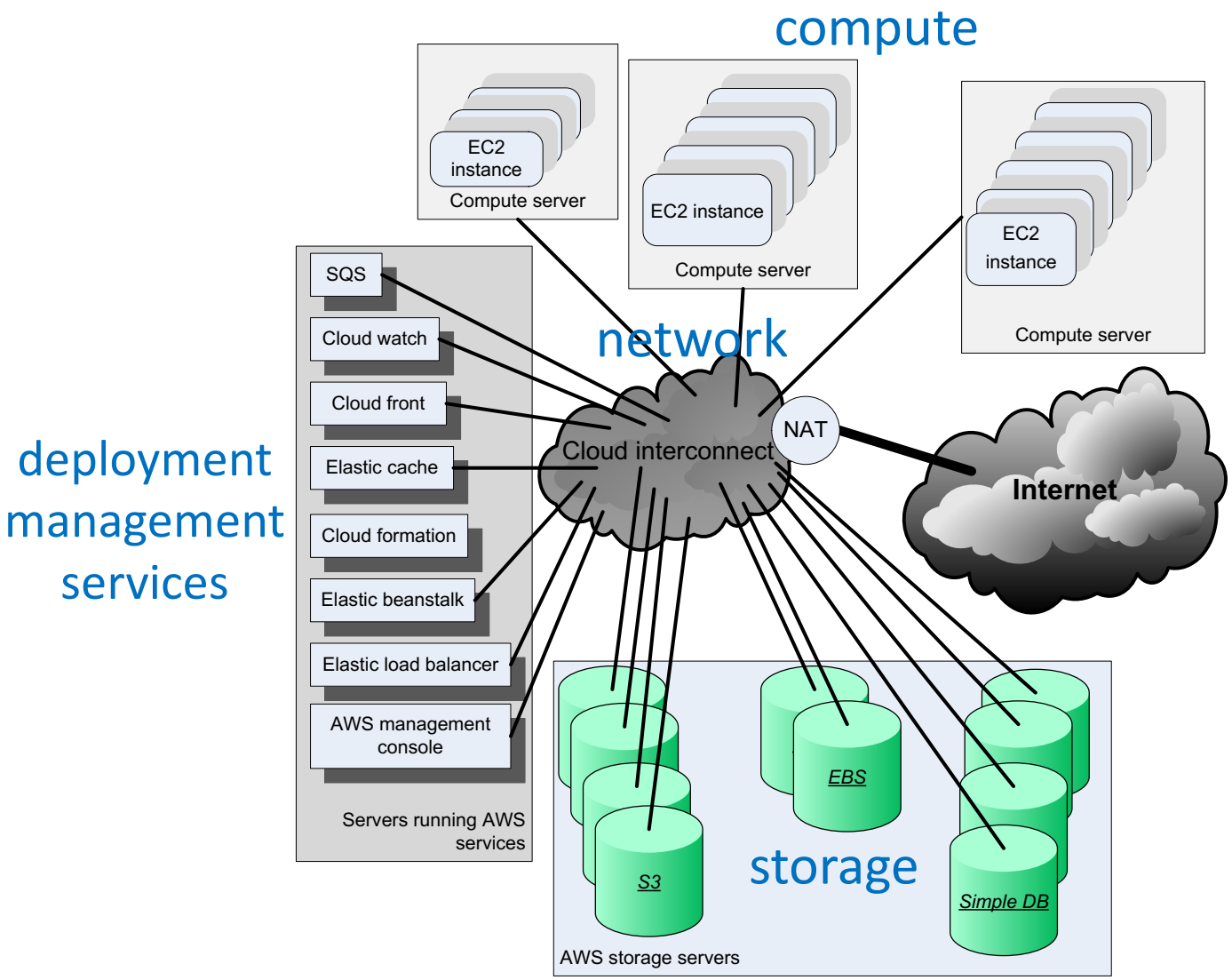
The AWS Cloud spans 84 Availability Zones within 26 geographic regions around the world, with announced plans for 24 more Availability Zones and 8 more AWS Regions in Australia, Canada, India, Israel, New Zealand, Spain, Switzerland, and United Arab Emirates (UAE).

Example: Global Network of AWS Regions

Every data center, AZ, and AWS Region is interconnected via a purpose-built, highly available, and low-latency private global network infrastructure. The network is built on a global, fully redundant, parallel 100 GbE metro fiber network that is linked via trans-oceanic cables across the Atlantic, Pacific, and Indian Oceans, as well as the Mediterranean, Red Sea, and South China Seas.



Example: An Availability Zone Configuration



Amazon Web Services


Analytics


Application Integration


AWS Cost Management

Blockchain


Business Applications


Compute


Containers


Customer Engagement


Database


Developer Tools


End User Computing


Front-End Web & Mobile


Game Tech


Internet of Things


Machine Learning


Management & Governance


Media Services


Migration & Transfer


Networking & Content Delivery


Quantum Technologies


Robotics


Satellite

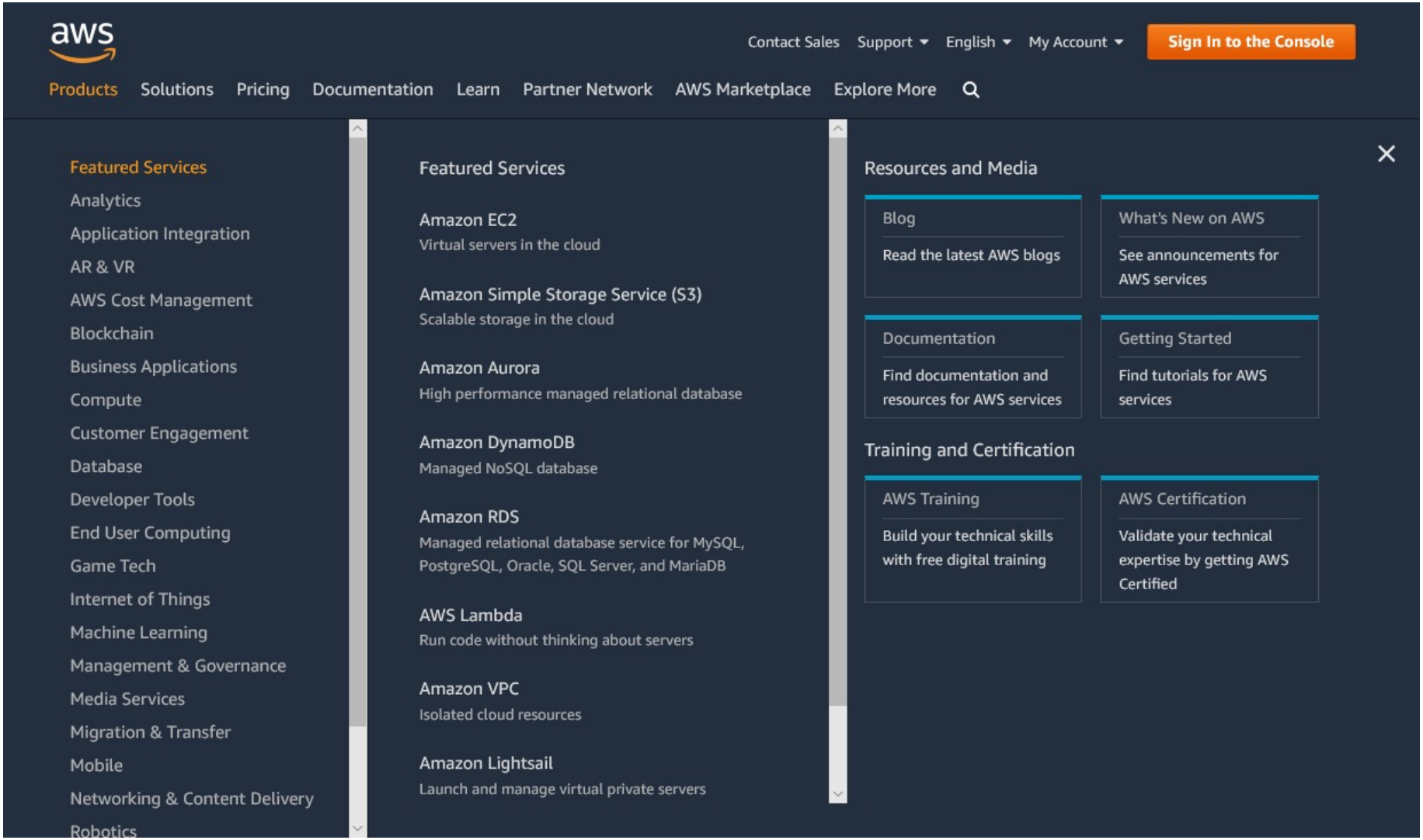

Security, Identity & Compliance


Serverless


Storage


VR & AR

AWS Catalog


aws
Contact Sales
Support
English
My Account
Sign In to the Console
Products
Solutions
Pricing
Documentation
Learn
Partner Network
AWS Marketplace
Explore More
Featured Services
Analytics
Application Integration
AR & VR
AWS Cost Management
Blockchain
Business Applications
Compute
Customer Engagement
Database
Developer Tools
End User Computing
Game Tech
Internet of Things
Machine Learning
Management & Governance
Media Services
Migration & Transfer
Mobile
Networking & Content Delivery
Robotics
Featured Services
Amazon EC2
Virtual servers in the cloud
Amazon Simple Storage Service (S3)
Scalable storage in the cloud
Amazon Aurora
High performance managed relational database
Amazon DynamoDB
Managed NoSQL database
Amazon RDS
Managed relational database service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB
AWS Lambda
Run code without thinking about servers
Amazon VPC
Isolated cloud resources
Amazon Lightsail
Launch and manage virtual private servers
Resources and Media
Blog
Read the latest AWS blogs
What's New on AWS
See announcements for AWS services
Documentation
Find documentation and resources for AWS services
Getting Started
Find tutorials for AWS services
Training and Certification
AWS Training
Build your technical skills with free digital training
AWS Certification
Validate your technical expertise by getting AWS Certified

Target Cloud Consumers

- Application Developers
  - Supports multiple languages
  - Auto-managed: version control, flexible capacity
- Businesses
  - Ease of deployment -> faster time-to-market
  - Lesser need for technical know-how
- Users
  - Needs are addressed fast
  - Feature updates made sooner
    - No need for special “going live mode” to update/release feature

Companies using AWS

Expedia is all in on AWS, with plans to migrate 80 percent of its mission-critical applications.

Using Amazon SageMaker, Intuit cuts the time to deploy machine-learning models by 90 percent.

General Electric (GE) is migrating more than 9,000 workloads to AWS.

Brooks Brothers runs key business-critical SAP applications on AWS.

Airbnb benefits from the scalability, agility, and reliability provided by AWS.

Mapbox can collect 100 million miles of telemetry data every day using AWS.

Lyft is going all in on AWS to deliver better customer experiences and focus on innovation.

Nextdoor has relied on AWS to support every component of its platform.

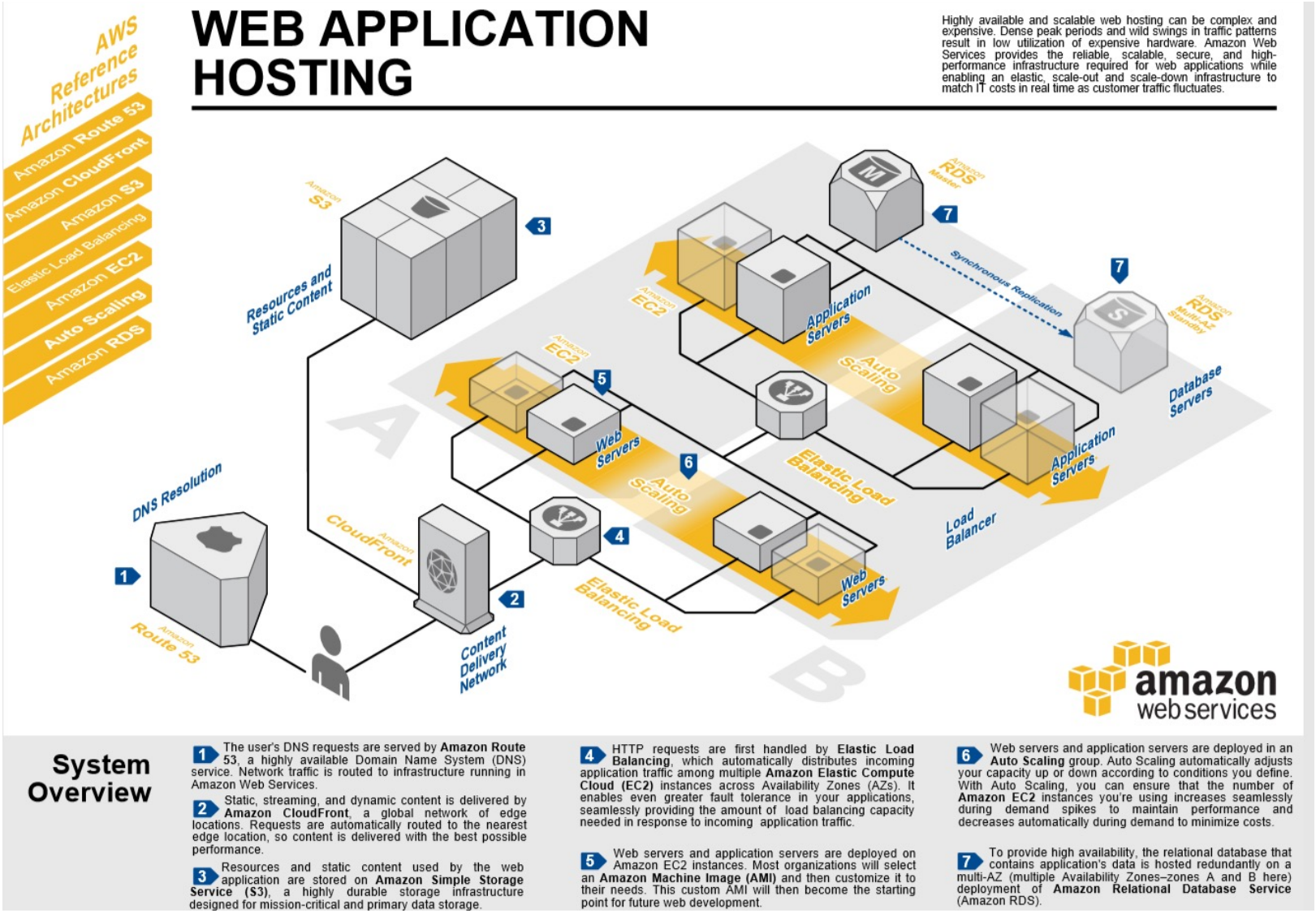
FINRA collects and analyzes billions of brokerage transactions daily using AWS.

By using AWS, Coursera can handle half a petabyte of traffic each month.

DigitalGlobe is all in on AWS and uses Amazon SageMaker to handle machine learning at scale.

The Food and Drug Administration (FDA) uses AWS to deliver cost-effective, innovative new programs.

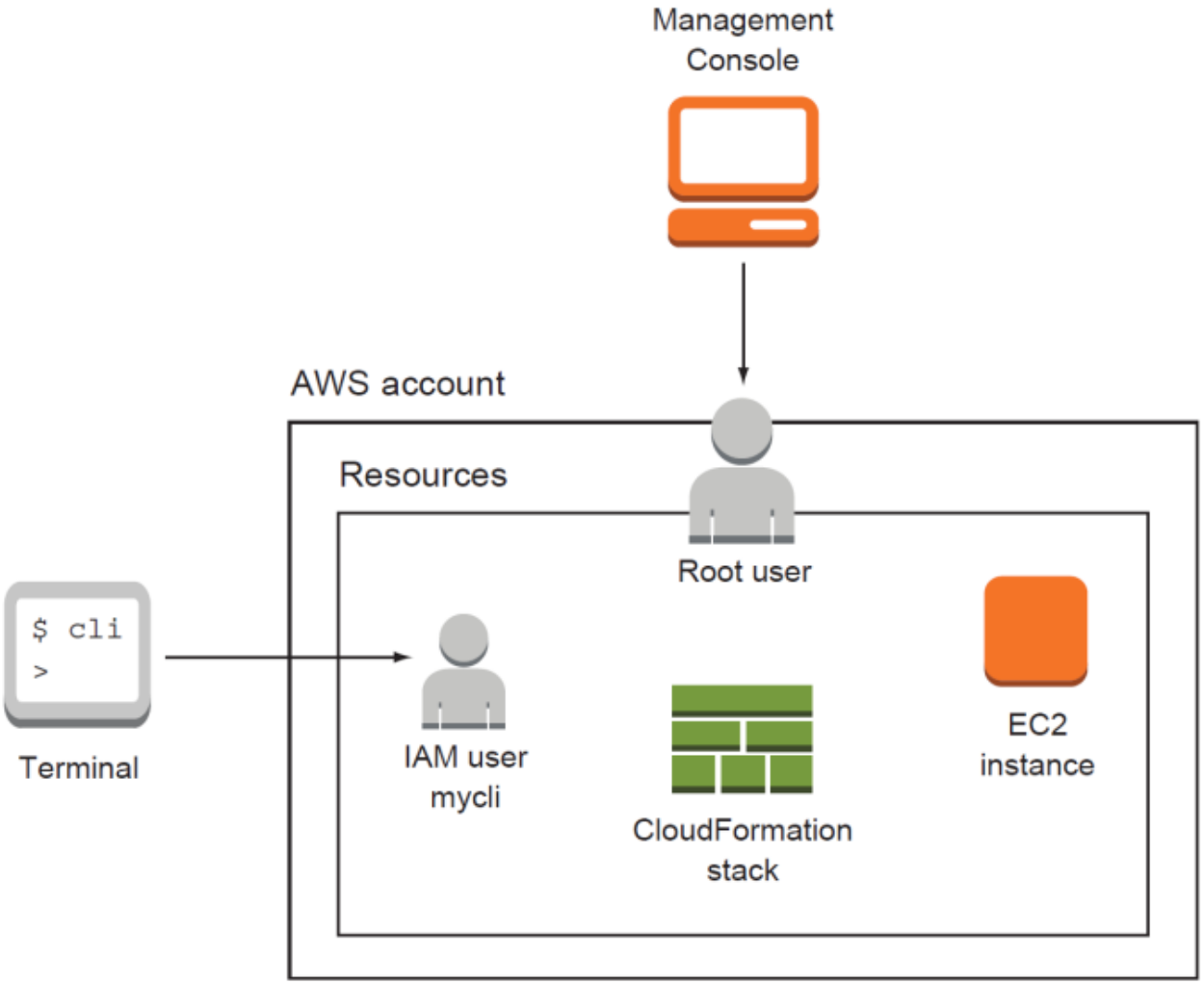
AWS Architecture on Web Application Hosting



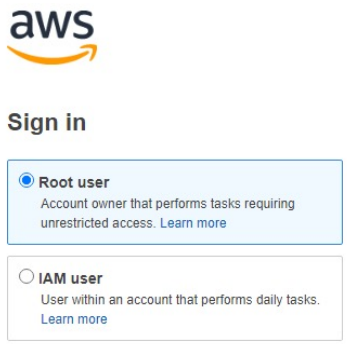
System Overview

1. The user's DNS requests are served by **Amazon Route 53**, a highly available Domain Name System (DNS) service. Network traffic is routed to infrastructure running in Amazon Web Services.
2. Static, streaming, and dynamic content is delivered by **Amazon CloudFront**, a global network of edge locations. Requests are automatically routed to the nearest edge location, so content is delivered with the best possible performance.
3. Resources and static content used by the web application are stored on **Amazon Simple Storage Service (S3)**, a highly durable storage infrastructure designed for mission-critical and primary data storage.
4. HTTP requests are first handled by **Elastic Load Balancing**, which automatically distributes incoming application traffic among multiple **Amazon Elastic Compute Cloud (EC2)** instances across Availability Zones (AZs). It enables even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic.
5. Web servers and application servers are deployed on Amazon EC2 instances. Most organizations will select an **Amazon Machine Image (AMI)** and then customize it to their needs. This custom AMI will then become the starting point for future web development.
6. Web servers and application servers are deployed in an **Auto Scaling** group. Auto Scaling automatically adjusts your capacity up or down according to conditions you define. With Auto Scaling, you can ensure that the number of **Amazon EC2** instances you're using increases seamlessly during demand spikes to maintain performance and decreases automatically during demand to minimize costs.
7. To provide high availability, the relational database that contains application's data is hosted redundantly on a multi-AZ (multiple Availability Zones–zones A and B here) deployment of **Amazon Relational Database Service** (Amazon RDS).

AWS Account


Management
Console
AWS account
Resources
Root user
$ cli
>
Terminal
IAM user
mycli
CloudFormation
stack
EC2
instance

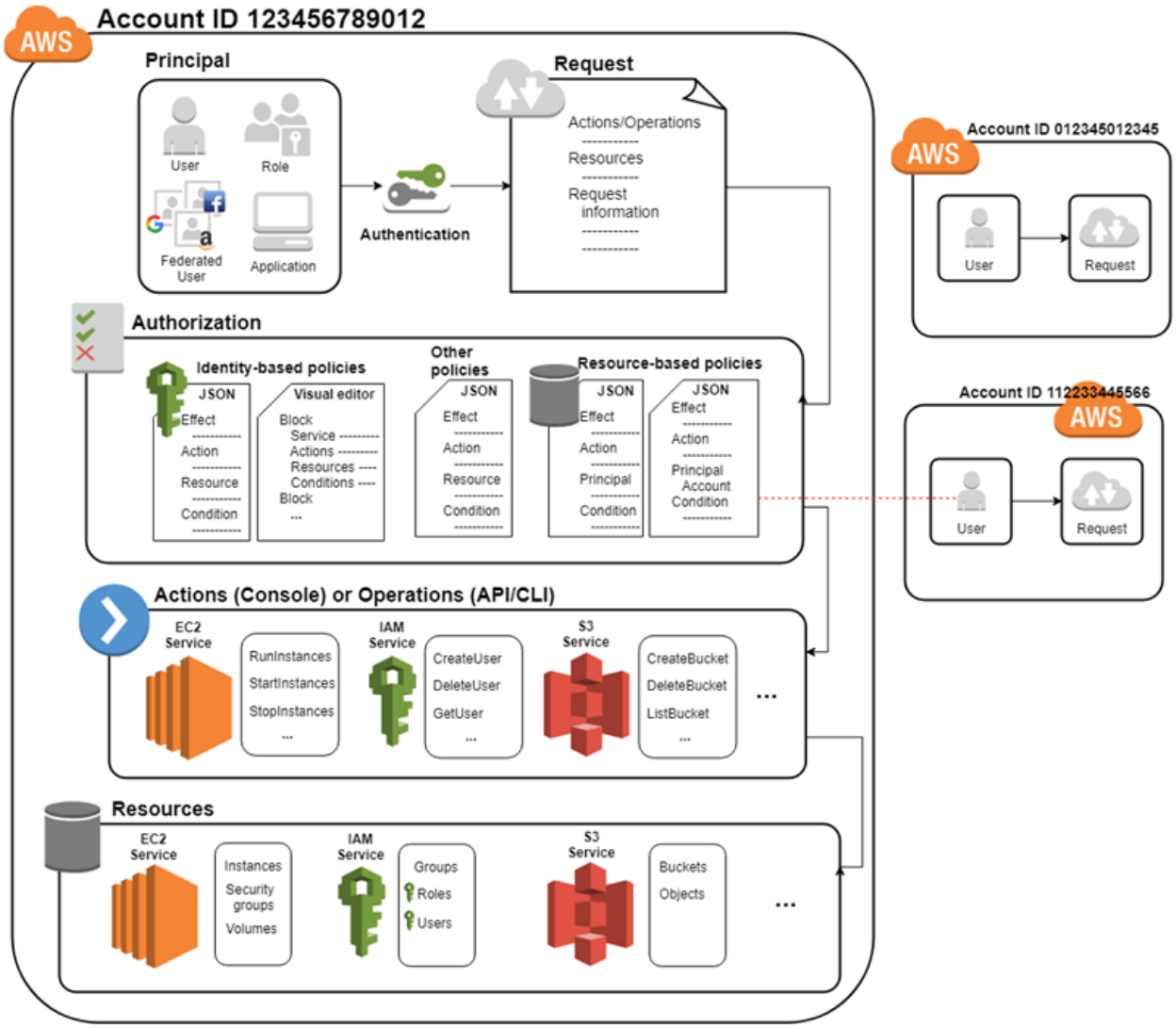
AWS Login



- Root user (AWS01)
  - account you first created on AWS
  - complete access to all AWS services and resources
  - similar to UNIX root or Windows Administrator account
  - sign in with email address and password used to create the account

- IAM (Identity & Access Management) user (AWS02-04)
  - how can an organization set up accounts for its employees and manage and control accesses to AWS resources?
  - purpose: IAM gives control to people/applications that can access your AWS account (AWS resources); example: control who is authenticated (signed in) and authorized (has permissions) to use resources.
  - how: IAM user is created using the IAM service, a web service to control access to AWS resources
  - example: an organization subscribes to a pool of AWS resources and uses IAM to control employee's access to the AWS resource pool (allows resource pooling as discussed in L03: Cloud Architecture)

How IAM Works



Amazon Web Services

objective: complement/reinforce lectures with hands-on using Amazon Cloud

handouts: overview + step-by-step lab exercises

- AWS01: Amazon Web Services
- AWS02: AWS Core Services
- AWS03: Cloud Service Models: IaaS, PaaS, SaaS, FaaS
- AWS04: AWS MapReduce

2 Lab assignments (20%)

AWS Accounts in this CS5224

1. Personal Account (12-month free tier)
   – you need a root account to carry out AWS IAM experiments (AWS01 - Lab1)

2. IAM account, not root
   CS5224 Classroom: IAM user accounts will be created using your NUS email addresses (you need to use this for Lab2, Lab3 and Lab4)

Access and Manage AWS Services

Two main interfaces:

1. AWS Management Console

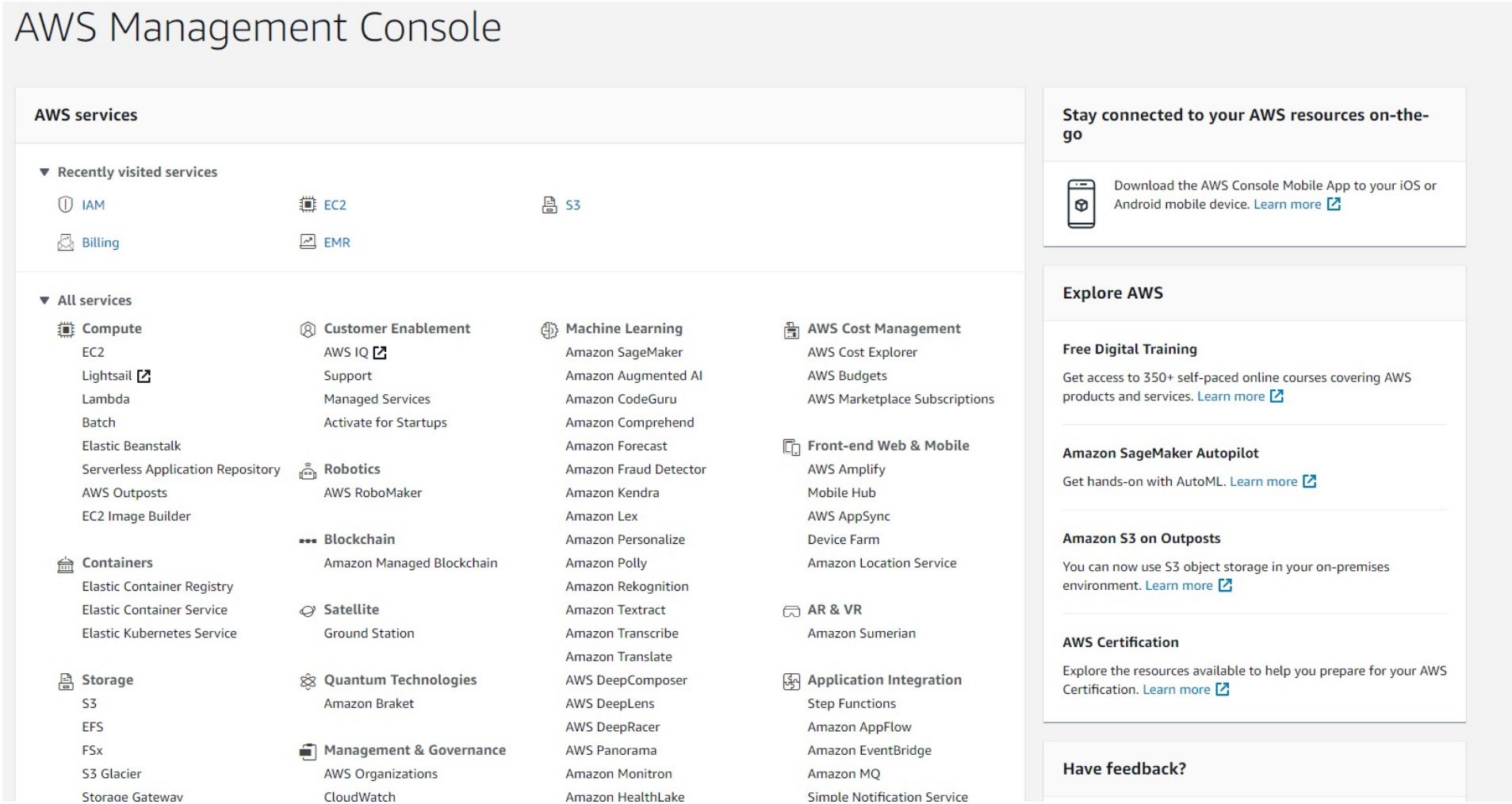
2. AWS Command Line Interface

AWS Management Console

- a web interface to access and manage the AWS cloud
- login to management console using your AWS or IAM account credentials

https://console.aws.amazon.com/console/home

AWS Management Console



AWS Command Line Interface (CLI)

- enables interaction with AWS services using commands in command-line shell
  - With minimal configuration, the AWS CLI enables you to start running commands that implement functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your terminal program
- All IaaS (infrastructure-as-a-service) AWS administration, management, and access functions in the AWS Management Console are available

Command Structure

AWS CLI command line:

$ **aws** <*command*> <*subcommand*>
[*options and parameters*]

- base call to AWS program
- <command> - typically corresponds to an AWS service
- <subcommand> specifies operation to perform
- [options or parameters] required by the specified operation

Example: lists all your Amazon S3 buckets

```
$ aws s3 ls
2018-12-11 17:08:50 my-bucket
2018-12-14 14:55:44 my-bucket2
```

Summary

- How are cloud services organized?
- Login and IAM Access Control
- How to access and manage services?

Videos & References

Videos

- What is AWS? [2019, 3 min]
- Introduction to AWS IAM - Identity and Access Management on AWS [2014, 2 min]
- Introduction to the AWS CLI v2 [2019, 46 min]
- Deep Dive: AWS Command Line Interface [2015, 64 min]

References

- https://aws.amazon.com/
- https://aws.amazon.com/getting-started/fundamentals-overview/
- https://aws.amazon.com/console/
- https://aws.amazon.com/iam/
- https://aws.amazon.com/cli/


HAPPY NEW YEAR
2022
YEAR OF THE TIGER
新
年
快
乐

AWS01: Amazon Web Services (Lab)

Outline

- Objectives
- Milestones
  - Create AWS IAM Group
  - Install AWS CLI
- Summary
- References

Objectives

- AWS Account
  - Create AWS account
- Management Console and IAM
  - Use AWS account (root user) to access to cloud services
  - Create IAM admin group to avoid using root user (for security)
- AWS CLI
  - Install CLI Environment

Outline

- Objective
- Milestone
  - Create AWS Account
  - Create AWS IAM Group
  - Install AWS CLI
- Summary
- Reference

Outline

- Objective
- Milestones
  - Create AWS Account
  - Create AWS IAM Group
  - Install AWS CLI
- Summary
- Reference

Create AWS IAM Group

Step 1: Login to AWS Management Console

Step 2: Enable IAM

Step 3: Add IAM User

Step 4: Create Group

Creating IAM Admin User and Groups

- As a best practice, do not use the AWS account root user for any task where it's not required. Instead, create a new IAM user for each person that requires administrator access.

- To try this, you have to apply for an AWS root account (Free but need to provide credit/debit card info)

Creating IAM Admin User and Groups

- Sign in to the IAM console as the account owner by choosing Root user and entering your AWS account email address.

  https://console.aws.amazon.com/iam/

  On the next page, enter your password.

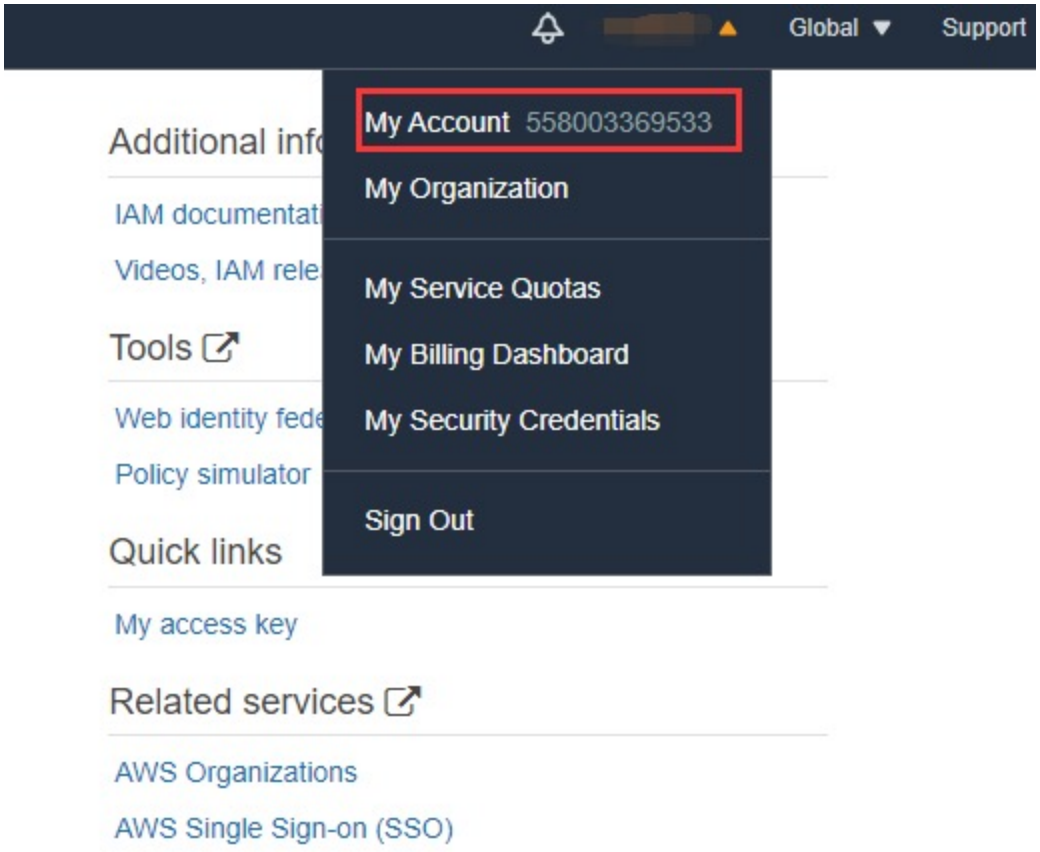
Step 1: Login to AWS Management Console (Personal account)

Enable access to billing data for the IAM admin user that you created as follows:

- On the navigation bar, choose your account name, and then choose **My Account**
- Next to **IAM User and Role Access to Billing Information**, choose **Edit**. You must be signed in as the **root user** for this section to be displayed on the account page
- Select the check box to **Activate IAM Access** and choose **Update**
- On the navigation bar, choose **Services** and then **IAM** to return to the IAM dashboard

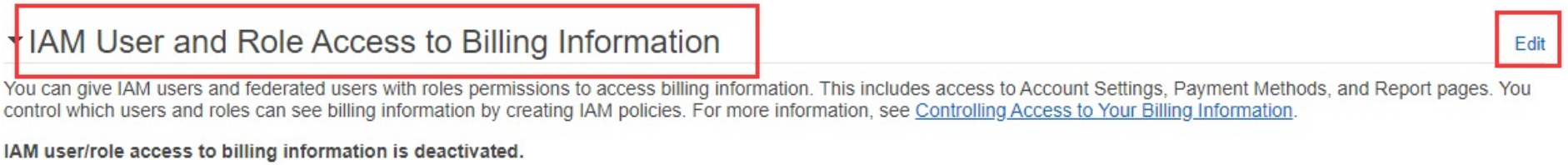
Step 1: Login to AWS Management Console

On the navigation bar, choose your account name, and then choose **My Account**



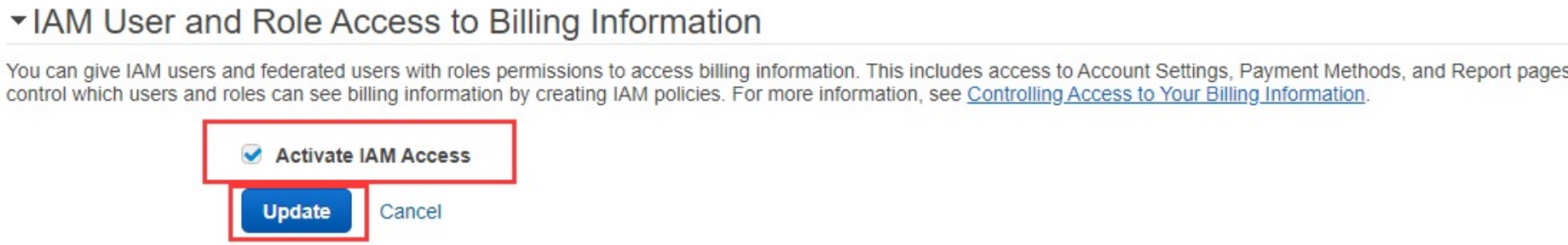
Step 2: Enable IAM

Next to **IAM User and Role Access to Billing Information**, choose **Edit**. You must be signed in as the **root user** for this section to be displayed on the account page.



Step 2: Enable IAM

Select the check box to **Activate IAM Access** and choose **Update**.

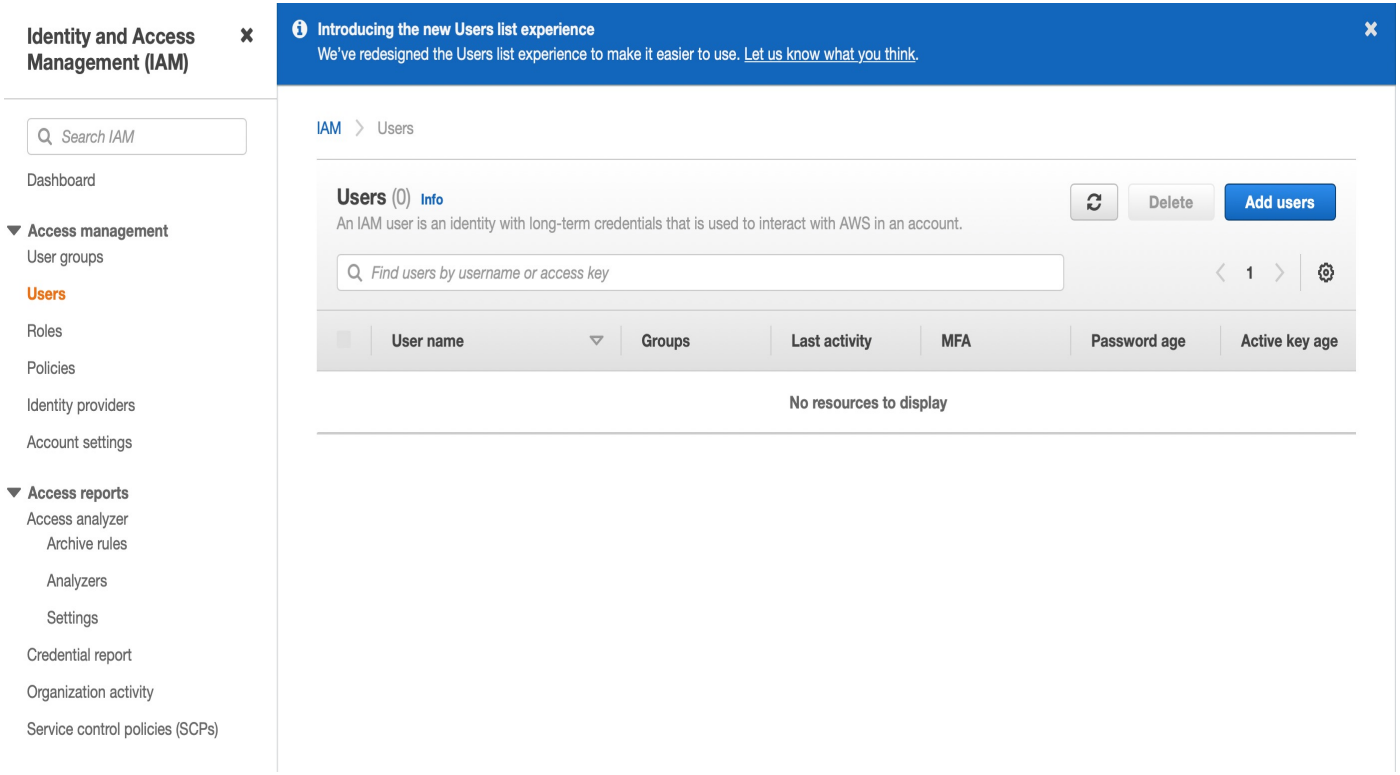


Step 3: Add IAM User

On the navigation bar, choose **Services** and then **IAM** to return to the IAM dashboard

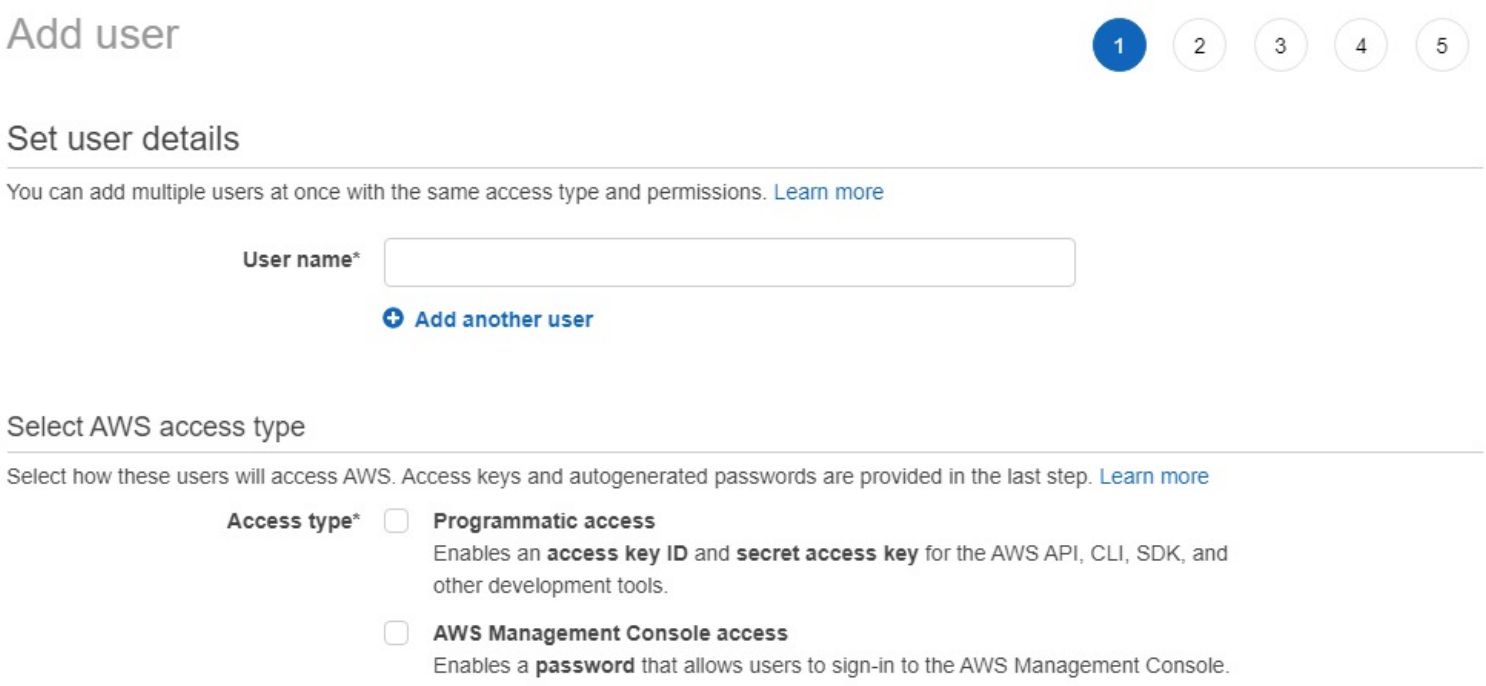
Step 3: Add IAM User

In the navigation pane, choose **Users** and then choose **Add users**



Step 3: Add IAM User

For username, type Administrator



Step 3: Add IAM User

- Select the check box for AWS Management Console access, select Custom password, and then type your new password in the text box.

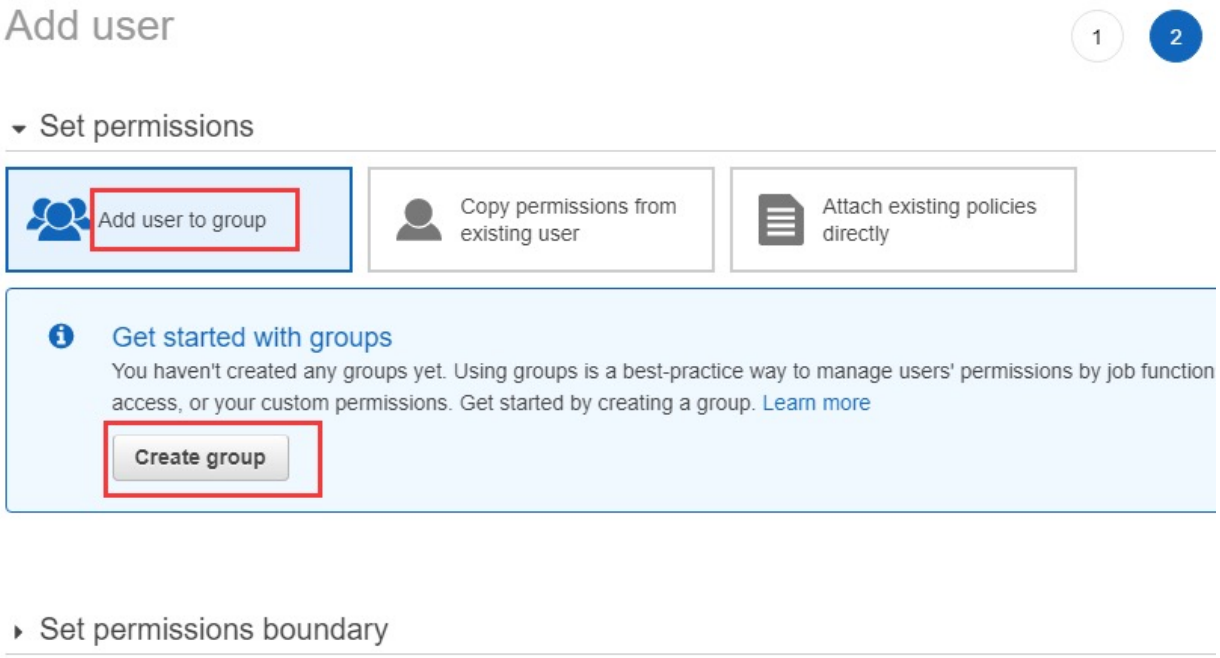
- By default, AWS forces the new user to create a new password when first signing in. You can optionally clear the check box next to User must create a new password at next sign-in to allow the new user to reset their password after they sign in.

- Choose Next: Permissions

Step 4: Create Group

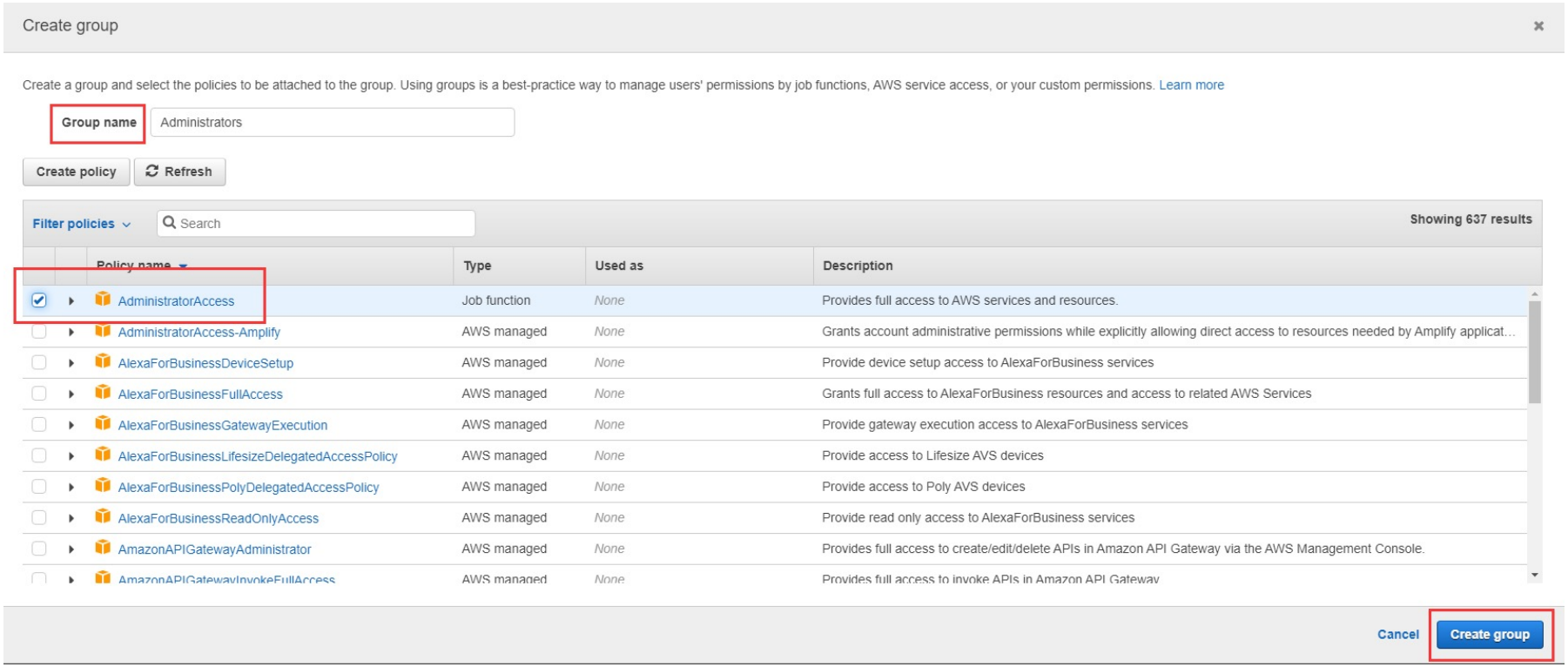
On the Permissions page, do the following:

- Choose **Add user to group**
- Choose **Create group**



Step 4: Create Group

- In the Create group dialog box, for Group name type Administrators
- Select the check box for the AdministratorAccess policy

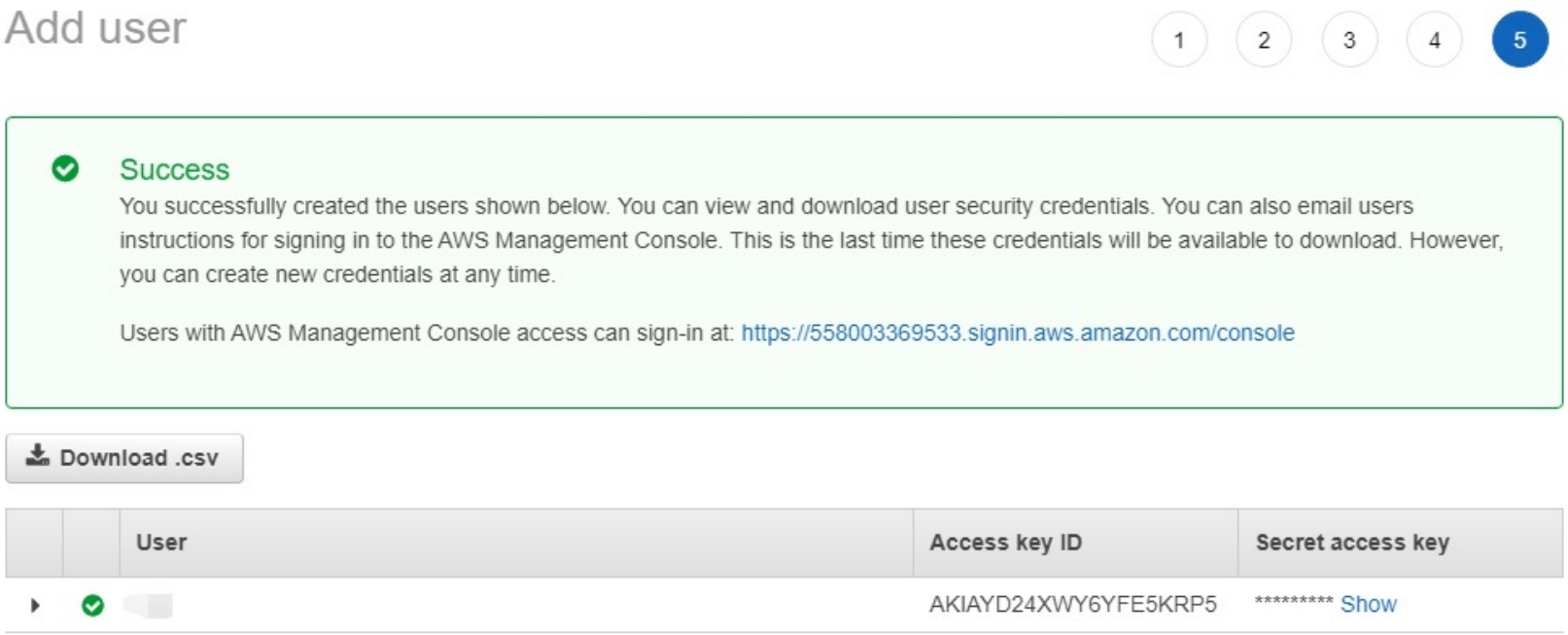


Step 4: Create Group

- Choose Create group
- Back on the page with the list of groups, select the check box for your new group. Choose Refresh if you don't see the new group in the list
- Choose Next: Tags
- Choose **Next: Review**
- Verify the group memberships to be added to the new user. When you are ready to proceed, choose **Create user**

Milestone: Create AWS IAM User and Group

If successful, you should see this on your screen. This is the last time these credentials will be availbale to download / view.



Outline

- Objective
- Milestones
  - Create AWS Educate Account
  - Create AWS IAM Group
  - Install AWS CLI
- Summary
- Reference

AWS CLI Installation

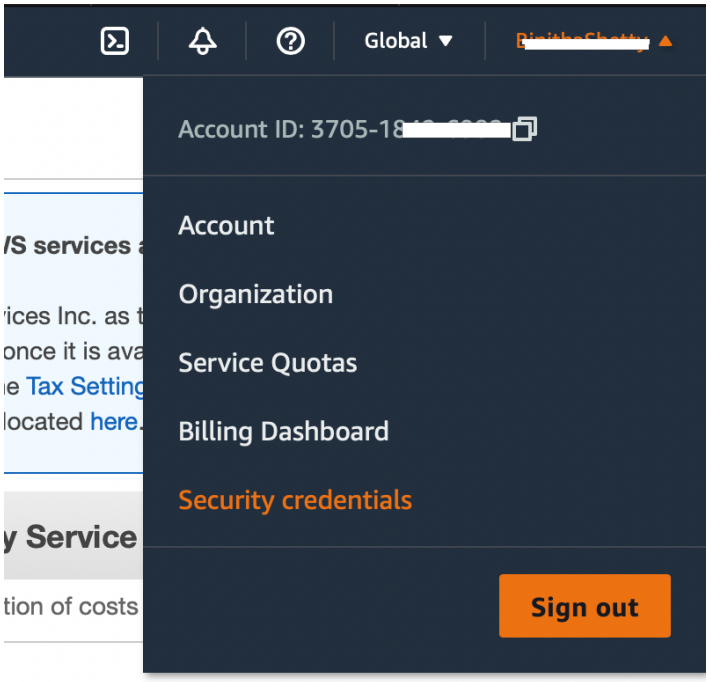
CLI depends on the OS you are using, please refer to the following document for CLI installation wrt to your OS:

https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html

We will leave this as an exercise for students that like to use CLI to access AWS services

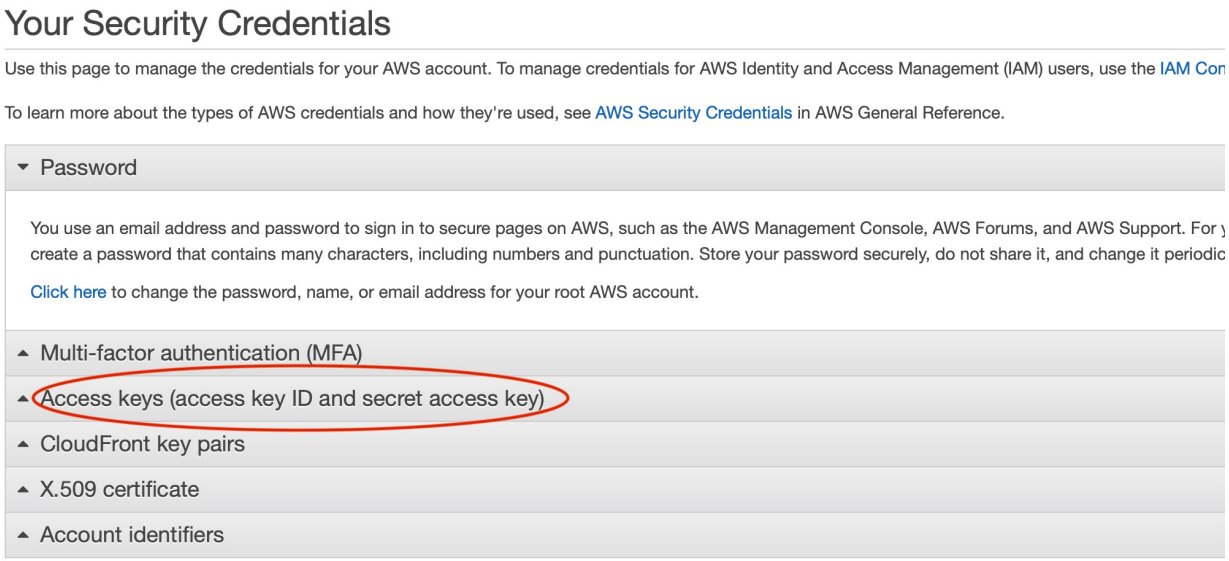
Step 1: Login to AWS Management Console (personal account)

On the navigation bar, choose your account name, and then choose **Security Credentials**



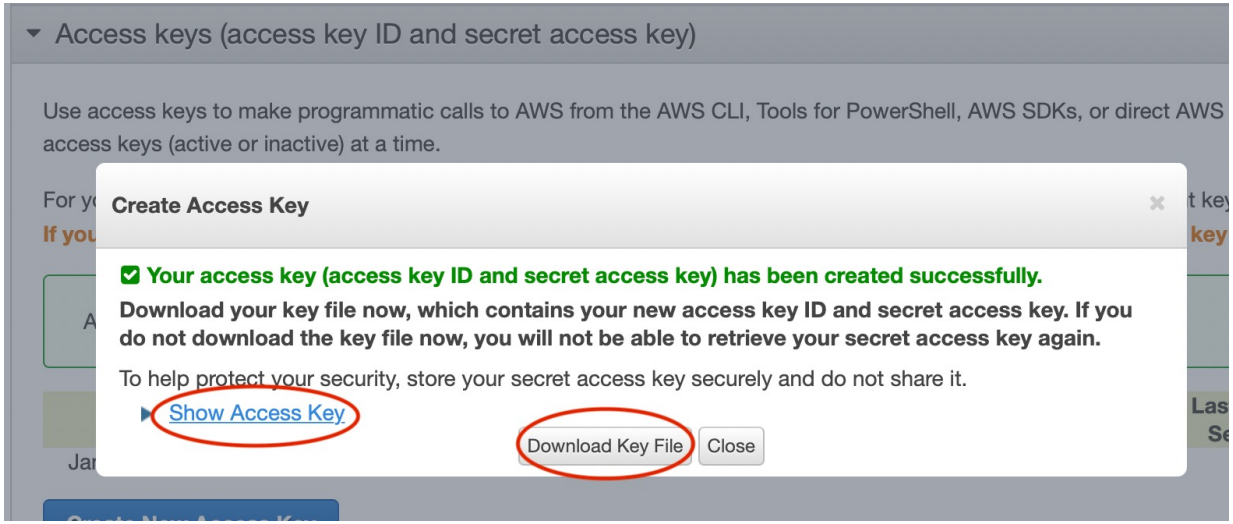
Get AWS credentials

- Choose the **Access keys** section



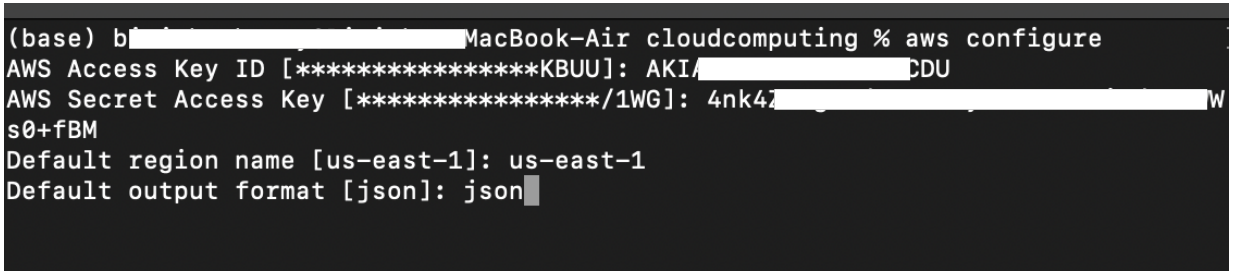
Get AWS credentials

- Choose **Create New Access Key** and retrieve the access key and secret key either by clicking on the **Show Access Key** or **Download Key File**.



Get AWS credentials

- Use '**aws configure**' command to set up your AWS CLI installation. Once you enter the command, type in the credentials (retrieved in the previous step) and other info.



Summary

- Management Console and IAM
  - Create AWS account (root user) to access to cloud services
  - Create IAM admin group to avoid using root user account
- AWS CLI
  - Install CLI Environment

References

- https://aws.amazon.com/education/awseducate/apply/
- https://docs.aws.amazon.com/IAM/latest/UserGuide/getting-started_create-admin-group.html
- https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html