# Cloud Programming

By

Dr. P. Sreedhar

# 1.3 GETTING STARTED WITH AWS, AMAZON CLOUDWATCH

# *AWS*

- What is AWS?
- Sign Up for AWS
- Create User and Groups
- Create Key Pairs
- Install AWS CLI
- Download SSH Client
- Setup Eclipse

# *AWS*

## What is AWS?

- Amazon Web Services is a cloud service provider that allows a user to consume different services to ease the application management
- AWS provides infrastructure, security, storage, networking, and other services that are useful for application life cycle.
- It provides services as an on-demand basis. In other words, users can enroll for services according to their requirements or demands for the applications and terminate the services when they are no longer needed.
- AWS is located at various geographical locations, also known as ***regions***. Users have the choice to select from any of these available geographical locations that they feel will be the most useful in terms of serving the application latency time.
- As AWS is ***an on-demand service***, it uses pay-as-you-go pricing. As a result, you will be charged only for the services you use. AWS also provides detailed monthly billing that helps users to know which services they are consuming the most.
- **Example AWS services are:**
- **EC2** (Elastic Compute Cloud), **SQS**, (Simple Queue Service), **S3**, (Simple Storage Service)
- **Kinesis**, **RDS**, (Relational Data Storage), EBS (Elastic Block Storage)

# *AWS*

## What is AWS?

- Amazon Web Services offers a broad set of global cloud-based products including compute, storage, databases, analytics, networking, mobile, developer tools, management tools, IoT, security, and enterprise applications: on-demand, available in seconds, with pay-as-you-go pricing.
- From data warehousing to deployment tools, directories to content delivery, **over 200 AWS services are available.**
- New services can be provisioned quickly, without the upfront fixed expense.
- This allows enterprises, start-ups, small and medium-sized businesses, and customers in the public sector to access the building blocks they need to respond quickly to changing business requirements.
- In 2006, Amazon Web Services (AWS) began offering IT infrastructure services to businesses as web services—now commonly known as cloud computing.
- Today, AWS provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers hundreds of thousands of businesses in **190 countries around the world.**
- To access the services, you can use the AWS Management Console, the AWS CLI, or Software Development Kits (SDKs).

# *AWS Management Console*

- Access and manage Amazon Web Services through the AWS Management Console, a simple and intuitive user interface

# *AWS CLI*

- **AWS CLI:** Is a Tool for managing AWS services.
- The AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell.
- The services include computing, storage, database, and application synchronization (messaging and queuing).
- With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.
- 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.

- Install AWS CLI V2 from https://aws.amazon.com/cli/

https://awscli.amazonaws.com/AWSCLIV2.msi

# *AWS SDK*

- Software Development Kits (SDKs) simplify using AWS services in your applications with an Application Program Interface (API) tailored to your programming language or platform

# *AWS Services*

## Analytics

- **Amazon Athena** is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL.
- **Amazon CloudSearch** is a managed service in the AWS Cloud that makes it simple and cost-effective to set up, manage, and scale a search solution for your website or application.
- **Amazon Kinesis** makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information.
- **Amazon OpenSearch:** Amazon OpenSearch Service (OpenSearch Service) makes it easy to deploy, secure, operate, and scale OpenSearch to search, analyze, and visualize data in real-time.

# *AWS Services*

## Application Integration

- **Amazon AppFlow**: Amazon AppFlow is a fully managed integration service that enables you to securely transfer data between Software-as-a-Service (SaaS) applications like Salesforce, Zendesk, Slack, and ServiceNow, and AWS services like Amazon S3 and Amazon Redshift, in just a few clicks
- **Amazon SNS (Simple Notification Service):** Amazon Simple Notification Service (Amazon SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications.
- **Amazon SQS (Amazon Simple Queue Service):** Amazon Simple Queue Service (Amazon SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.

## AR and VR

**Amazon Sumerian:** lets you create and run virtual reality (VR), augmented reality (AR), and 3D applications quickly and easily without requiring any specialized programming or 3D graphics expertise

## Blockchain

**Amazon Managed Blockchain** is a fully managed service that makes it easy to create and manage scalable blockchain networks using the popular open source frameworks Hyperledger Fabric and Ethereum.

# *AWS Services*

## Business Applications

- **Alexa for Business:** makes it efficient to voice-enable your products and services, thus providing context-aware voice experiences for your customers.
- **Amazon Chime** is a communications service that transforms online meetings with a secure, easy-to-use application that you can trust. Amazon Chime works seamlessly across your devices so that you can stay connected
- **Amazon Simple Email Service (Amazon SES)** is a cost-effective, flexible, and scalable email service that enables developers to send mail from within any application

## Cloud Financial Management

- **AWS Application Cost Profiler** provides you the ability to track the consumption of shared AWS resources used by software applications
- **AWS Cost Explorer** has an easy-to-use interface that lets you visualize, understand, and manage your AWS costs and usage over time
- **AWS Budgets** gives you the ability to set custom budgets that alert you when your costs or usage exceed (or are forecasted to exceed) your budgeted amount.

# *AWS Services*

## Compute Services:

- **Amazon EC2 (Elastic Compute Cloud):** Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.
- **AWS App Runner:** App Runner automatically builds and deploys the web application. AWS App Runner is a fully managed service that makes it easy for developers to quickly deploy containerized web applications and APIs, at scale and with no prior infrastructure experience required.
- **AWS Batch** — Fully managed batch processing at any scale

## Contact Center

- **Amazon Connect:** Amazon Connect is a self-service, omnichannel cloud contact center service that makes it easy for any business to deliver better customer service at lower cost

## Containers:

- **Amazon ECS (Amazon Elastic Container Service):** Highly secure, reliable, and scalable way to run containers.

# *AWS Services*

## Databases:

- **Amazon DynamoDB:** Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale.

- **Amazon RDS:** Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud.
- **Amazon Aurora:** Amazon Aurora is a MySQL and PostgreSQL compatible relational database engine that combines the speed and availability of high-end commercial databases with the simplicity and cost-effectiveness of open source databases.

# *AWS Services*

## Developer Tools:

- **Amazon Corretto** is a no-cost, multiplatform, production-ready distribution of the Open Java Development Kit (OpenJDK).
  - With Corretto, you can develop and run Java applications on popular operating systems, including Amazon Linux 2, Windows, and macOS.
- **Amazon Cloud9:** AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal
- **AWS CloudShell** is a browser-based shell that makes it easy to securely manage, explore, and interact with your AWS resources.
- **AWS CodeBuild** is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy.
- AWS CodeDeploy is a service that automates code deployments to any instance, including EC2 instances and instances running on premises

## End User Computing

- **Amazon AppStream 2.0** is a fully managed application streaming service. You centrally manage your desktop applications on AppStream 2.0 and securely deliver them to any computer

# *AWS Services*

- **Front-End Web & Mobile Services**
  - **Amazon Location Service** makes it easy for developers to add location functionality to applications without compromising data security and user privacy.
  - **Amazon PinPoint:** Amazon Pinpoint makes it easy to send targeted messages to your customers through multiple engagement channels. Examples of targeted campaigns are promotional alerts and customer retention campaigns, and transactional messages are messages such as order confirmations and password reset.
  - **AWS Amplify:** makes it easy to create, configure, and implement scalable mobile applications powered by AWS. Amplify seamlessly provisions and manages your mobile backend and provides a simple framework to easily integrate your backend with your iOS, Android, Web, and React Native frontends messages.
  - **AWS Device Farm:** AWS Device Farm is an app testing service that lets you test and interact with your Android, iOS, and web apps on many devices at once, or reproduce issues on a device in real time. View video, screenshots, logs, and performance data to pinpoint and fix issues before shipping your app.

# *AWS Services*

- **Game Tech**

- **Amazon GameLift:** Amazon GameLift is a managed service for deploying, operating, and scaling dedicated game servers for session-based multiplayer games.
- **Amazon Lumberyard** is a free, cross-platform, 3D game engine for you to create the highest-quality games, connect your games to the vast compute and storage of the AWS Cloud,

- **Internet of Things (IoT)**

- **AWS IoT Analytics:** AWS IoT Analytics is a fully-managed service that makes it easy to run and operationalize sophisticated analytics on massive volumes of IoT data without having to worry about the cost and complexity typically required to build an IoT analytics platform.
- **AWS IoT Core:** AWS IoT Core is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices. AWS IoT Core can support billions of devices and trillions of messages, and can process and route those messages to AWS endpoints and to other devices reliably and securely.

# *AWS Services*

## Internet of Things (IoT)

- **AWS IoT Device Management:** As many IoT deployments consist of hundreds of thousands to millions of devices, it is essential to track,monitor, and manage connected device fleets
- **AWS IoT Events:** AWS IoT Events is a fully managed IoT service that makes it easy to detect and respond to events from IoT sensors and applications.
- **AWS IoT Things Graph:** AWS IoT Things Graph provides a visual drag-and-drop interface for connecting and coordinating devices and web services, so you can build IoT applications quickly.

## Machine Learning:

- **Amazon CodeGuru** is a developer tool that provides intelligent recommendations to improve code quality and identify an application's most expensive lines of code.
- **Amazon Comprehend** uses machine learning to help you uncover the insights and relationships in your unstructured data. The service identifies the language of the text; extracts key phrases, places, people, brands, or events; understands how positive or negative the text is; analyzes text using tokenization and parts of speech;
- **Amazon Forecast:** Amazon Forecast is a fully managed service that uses machine learning to deliver highly accurate forecasts.
- **Amazon SageMaker:** Amazon SageMaker is a fully-managed service that enables ML models at any scale.

# *AWS Services*

- **Management and Governance:**
- **CloudWatch** collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications and services that run on AWS, and on-premises servers.
- **AWS Auto Scaling:** AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost.
- **AWS Compute Optimizer**: AWS Compute Optimizer recommends optimal AWS resources for your workloads to reduce costs and improve performance by using machine learning to analyze historical utilization metrics.
- **Media Services**
- **Amazon Elastic Transcoder:** Amazon Elastic Transcoder is media transcoding in the cloud. It is designed to be a highly scalable, easyto-use, and cost-effective way for developers and businesses to convert (or transcode) media files from their source format into versions that will play back on devices like smartphones, tablets, and PCs.
- **Amazon Interactive Video Service:** Send your live streams to Amazon IVS using streaming software and the service does everything you need to make low-latency live video available to any viewer around the world

# *AWS Services*

- **Migration and Transfer**

- **AWS Application Migration Service:** AWS Application Migration Service (AWS MGN) allows you to quickly realize the benefits of migrating applications to the cloud without changes and with minimal downtime.
- **AWS Database Migration Service:** AWS Database Migration Service helps you migrate databases to AWS easily and securely.
- **AWS Server Migration Service:** AWS Server Migration Service (SMS) is an agentless service which makes it easier and faster for you to migrate thousands of on-premises workloads to AWS
- **AWS Snowmobile:** Snowmobile makes it easy to move massive volumes of data
- to the cloud, including video libraries, image repositories, or even a complete data center migration. Transferring data with Snowmobile is secure, fast, and cost effective.

# *AWS Services*

## Networking and Content Delivery

- **Amazon CloudFront:** Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.
- **Quantum Technologies**

## Robotics:

- **AWS RoboMaker** is a service that makes it easy to develop, test, and deploy intelligent robotics applications at scale. oboMaker extends the most widely used open-source robotics software framework, Robot Operating System (ROS), with connectivity to cloud services.

## Satellite

- **AWS Ground Station:** AWS Ground Station is a fully managed service that lets you control satellite communications, downlink and process satellite data, and scale your satellite operations quickly, easily and cost-effectively without having to worry about building or managing your own ground station infrastructure.

# *AWS Services*

## Security, Identity, and Compliance

- **Cloud Directory:** Amazon Cloud Directory enables you to build flexible, cloud-native directories for organizing hierarchies of data along multiple dimensions. With Cloud Directory, you can create directories for a variety of use cases, such as organizational charts, course catalogs, and device registries.
- **Amazon Detective:** Amazon Detective makes it easy to analyze, investigate, and quickly identify the root cause of potential security issues or suspicious activities. Amazon Detective automatically collects log data from your AWS resources and uses machine learning, statistical analysis, and graph theory to build a linked set of data that enables you to easily conduct faster and more efficient security investigations.
- **Amazon GuardDuty:** Amazon GuardDuty is a threat detection service that continuously monitors for malicious or unauthorized behavior to help you protect your AWS accounts and workloads.
- **AWS Certificate Manager:** AWS Certificate Manager is a service that lets you easily provision, manage, and deploy Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources.
- **AWS Identity and Access Management:** AWS (IAM) enables you to securely control access to AWS services and resources for your users. Using IAM, you can

# *AWS Services*

## Storage:

- **Amazon Elastic Block Store (Amazon EBS)** provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud.
- **Amazon Elastic File System (Amazon EFS)** provides a simple, scalable, elastic file system for Linux-based workloads for use with AWS Cloud services and on-premises resources.
- **Amazon Simple Storage Service:** Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.

# *Creating Users & Groups*

- AWS IAM (Identity and Access Management) helps to control the number of users under this account that will consume the AWS resources.
- You can provide authentication and authorization for new users, which allows them to access AWS limited resources
- Search for IAM service in Root Login to Admin Console.
- In the left pane of IAM Dashboard, menu items exist for creating Users, User Groups, Roles, Policies, Identity Providers etc…
- Create User, User Group, Roles, Policies

# *On Premises VS AWS solution*

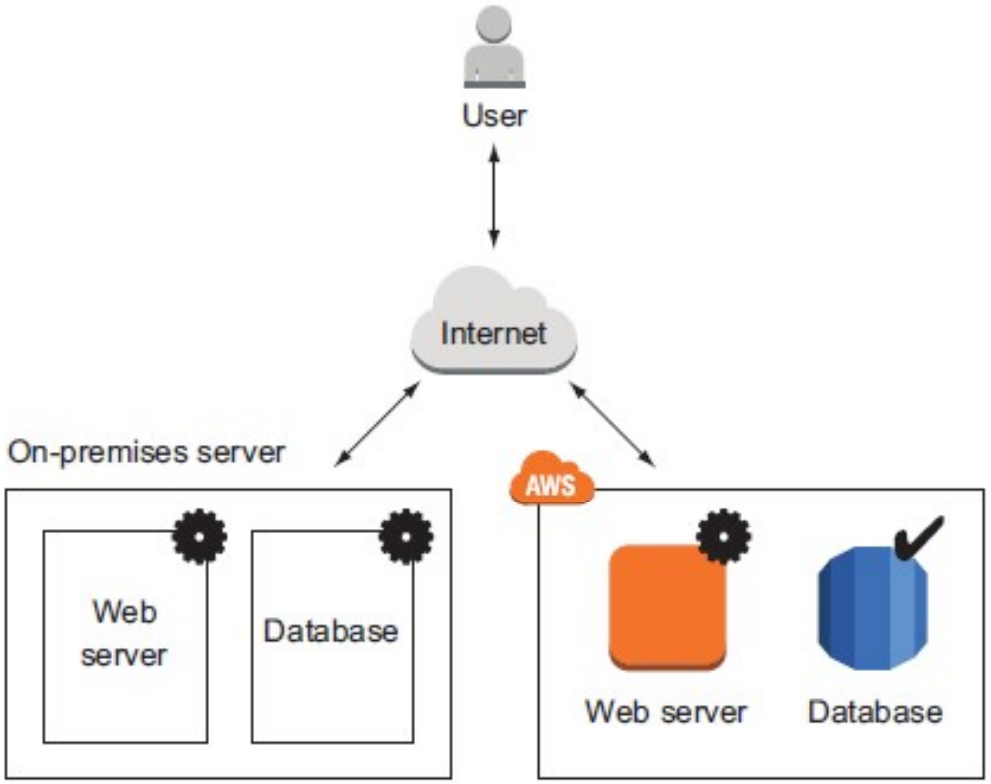


**Figure 1.2 Running a web shop on-premises vs. on AWS**

# *On Premises VS AWS solution*

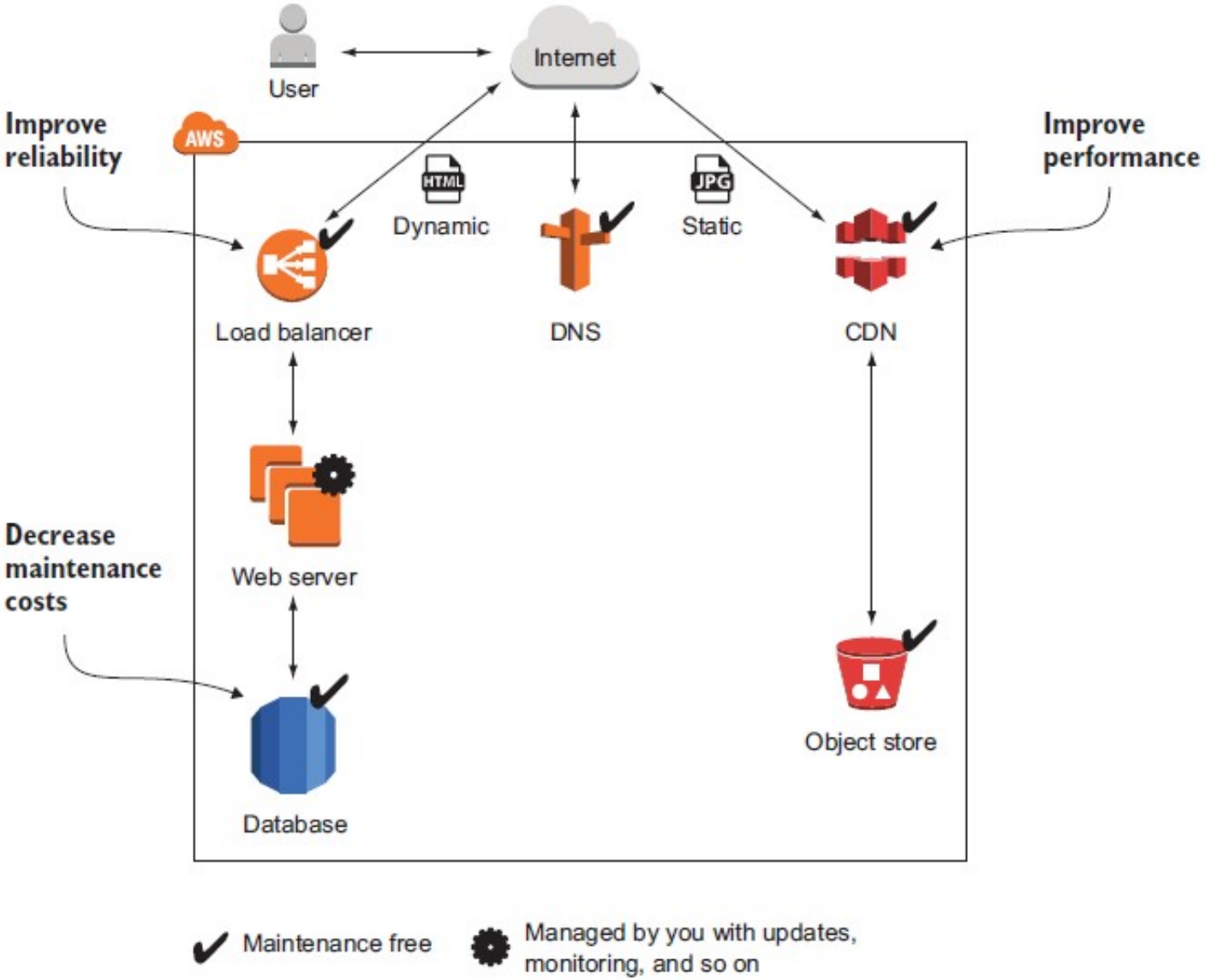


**Figure 1.3** **Running a web shop on AWS with CDN for better performance, a load balancer for high availability, and a managed database to decrease maintenance costs**

# Amazon EC2

# *Amazon Web Services (AWS) – EC2*

## AWS EC2 (Elastic Compute Cloud):

**Topics:**

- Introduction to EC2
- Features of EC2
- EC2 Instance Types
- Managing EC2 Using Management Console
- Managing EC2 Using AWS CLI
- Managing EC2 Using AWS SDK (Java)
- Monitoring Using CloudWatch

✓

# *Amazon Web Services (AWS) – EC2*

## AWS EC2 (Elastic Compute Cloud):

- Amazon Elastic Compute Cloud (EC2) provides virtual computing. EC2 can be used for various purposes such as running custom applications, storing files, and so on.
- Users have a wide variety of operating systems to launch EC2 instances.
- Amazon EC2 is an elastic virtual server that resides under the AWS cloud environment.
- Users can deploy their software on the EC2 instance.
- Amazon has given control to the users to create, start, stop, and terminate the instance at their convenience.
- Amazon has a variety of Amazon Machine Images (AMIs) available that can be used to create new servers in the cloud. Users can even create their own AMIs from their EC2 instance.
- EC2 also has the capability to auto-scale the servers up and down based on the load of existing servers in just a few minutes.
- Amazon provides different regions to launch the different services. As a result, you can create instances to handle failures at different geographical regions or you can utilize multi region deployment strategies for your application.
- Amazon uses the pay-as-you-go concept, which means you only pay for the number of Amazon services you have used.

# *Amazon Web Services (AWS) – EC2*

## EC2 Features:

- **Large number of OSs support**
  - Amazon EC2 provides a wide variety of operating systems that can be used to boot the server. Amazon Linux, Debian, SUSE, FreeBSD, CentOS, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, Ubuntu, Windows, and other Linux operating systems.
- **Elasticity**
  - Amazon EC2 infrastructure is elastic in nature. Elasticity means the system manages its resources by provisioning and de-provisioning them without manual intervention and, therefore, manages load dynamically.
  - You can configure the threshold limit when a new EC2 instance needs to be created based on the load on EC2 and terminate EC2 when the load on EC2 is below the threshold limit.
- **fault tolerance and latency**
- Amazon EC2 provides flexibility to auto-scale the servers on an as-needed basis. If one instance fails, another instance can be added to serve the purpose. Amazon also provides different availability zones where system architecture is designed in such a way that if a zone in a (region) is down, another zone in the same (region) can serve the request.
- **Pricing**
- Amazon charges only for the hours consumed. There may be different pricing options available for different capacity instances. Along with EC2, if we also opt for storage, pricing is based on storage type and the amount of data that is being stored in it.
- **Security**
- The security provided by EC2 is similar to the traditional firewall. It allows users to manage the accessibility of AWS resources to provide client information such as IP addresses or subnet groups. It is an effective way to manage security on EC2.
- **Service Commitment.**
- AWS provides the service commitment for its uptime of 99.95%. This means EC2 instances will be up and running for 99.95% of your monthly billing cycle. If this service commitment is not met, users will receive the service credit based on the criteria defined by AWS.

# *Amazon Web Services (AWS) – EC2*

**SSH (Secure Shell) Client:** Use **PuTTy** SSH Client to connect to the EC2 instances.

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

https://www.puttygen.com/download-putty

An SSH client is **a program that allows establishing a secure and authenticated SSH connections to SSH servers**.

- **PEM (Privacy Enhanced Mail)** is a base64 container format for encoding keys and certificates. .pem download from AWS when you created your key-pair. This is only a one time download and you cannot download it again.

- **PPK(Putty Private Key)** is a windows ssh client, it does not support .pem format. Hence you have to convert it to .ppk format using PuTTyGen.

- Convert EC2 **pem file to ppk file**. For this, we will use the puttygen.exe (because putty supports ppk only)

## EC2 Instance Types:

- AWS provides a long list of instance types that users can choose based on their needs.
- Users' needs vary based on their CPU, memory, storage, and networking capacity.
- Users choose a specific **instance type** that suits their application requirement.
- All instances have a specific **price tag** that users need to consider while selecting their instance type.

- **General purpose instances**
- **Compute Optimized instances**
- **Memory optimized instances**
- **Accelerated computing instances**
- **Storage optimized instances**

- **General purpose instances:** General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.
  - ✓ T4G instance, T3, T3a, T2, M6g, M4, A1, Mac

**A1 featres:**

Custom built AWS Graviton Processor with 64-bit Arm Neoverse cores

- Support for Enhanced Networking with Up to 10 Gbps of Network bandwidth
- EBS-optimized by default

Ex: A1.medium → (vCPU, MeM,,Storage, Network speed (Gbps) (1, 2GiB (gibi bytes), EBS-Only, upto 10 Gbps)

# *Amazon Web Services (AWS) – EC2*

## EC2 Instance Types:

✓ **Compute Optimized instances:**

Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this family are well suited for **batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.**

**Ex: C7g, C6g, C6gn, C6i, C5, C5a, C5n, C4 etc…**

✓ **Memory optimized instances**

- Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

**Ex: R6g, R6i, R5, R5a, R5b, R5n, R4, X1 etc…**

✓ **Accelerated computing instances**

- Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.

# *Amazon Web Services (AWS) – EC2*

## EC2 Instance Types:

- ✓ **Storage optimized instances:**

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random **I/O** operations **P**er **S**econd (IOPS) to applications.

**Ex: Im4gn, Is4gn, I4i, I3, D3, H1 etc…**

# Chapter 1.3
# Amazon EC2 using Management Console

# *Amazon Web Services (AWS) – EC2*

## Launching EC2 instance:

- **Before launching an EC2 instance, you should create a security group** that will permit network traffic that is appropriate to your application to connect to the instance.
- At a minimum, the security group should enable access on **port 22**, so that you can SSH into the EC2 instance.
- You may also want to create a **keypair**, although you can also create the keypair while going through the launch wizard. Finally, you should think about which instance type is appropriate to your application;
- AWS Free Tier includes **750 hours of Linux and Windows t2.micro instances**, ( t3.micro for the regions in which t2.micro is unavailable) each month for one year. **To stay within the Free Tier, use only EC2 Micro instances.**

- Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.
- **https://docs.aws.amazon.com/toolkit-for-eclipse/v1/user-guide/ami-launch.html**

# *Amazon Web Services (AWS) – EC2*

**Actions that can be performed using EC2 AWS Management Console:**

1. Use IAM to manage User Groups, Users, Roles, Policies
2. Create Key Pair
3. Create Security Group
4. Create EC2 instance
5. Connect to the EC2 instance
6. CloudWatch
7. CloudShel

Using CLI:

1. Working with IAM
2. Working with EC2 instance
3. Connecting to EC2 instances
4. Keypairs, Secgroups,
5. CloudWatch

**Creating IAM users  User groups**

# *Amazon Web Services (AWS) – EC2*

**Launching EC2 instance:**

Step 1: Choose AMI

Step 2: Choose Instance Type

Step 3: Configure Instance

Step 4: Add Storage

Step 5: Add Tags

Step 6: Configure Security Group

Step 7: Review and Launch Instance

# Chapter 1.3
# Amazon EC2 using CLI

# *EC2 using CLI*

- You can access the features of Amazon Elastic Compute Cloud (Amazon EC2) using the AWS Command Line Interface (AWS CLI).
- configure the settings that the AWS Command Line Interface (AWS CLI) uses to interact with AWS. These include your security credentials, the default output format, and the default AWS Region.
- **You can also use AWS Cloud Shell.**

**To get the Help:** aws ec2 help

**Configuring CLI : aws configure**

Get credential report from IAM dash board:
https://console.aws.amazon.com/iam/home#/credential_report

**Security Groups** → Create, Attach rules, List, Delete

**Key Pair** → Create, List, Delete, Import

**Ec2 instance:**

- Describe images
- Select image type
- Launch EC2
- List
- Get Status
- Assign tags
- Start, Stop and Terminate instance

AWS Cloud Shell → Is a web based shell for accessing AWS services.

# *EC2 using CLI*

- To work with a service from CLI, first configure aws using the following command:
- From the command prompt, give:
  - **aws configure** → (asks to enter Access key id, Secret Access Key, region, output format)

**To give Access Key ID, Secret Access Key in the above command,** use keys downloaded using option 'user menu → Security Credentials → 'Access keys'

# *IAM (Identity & Access Management)*

**User groups:**

- A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.

**Creation of User Group: Step 1:** Give name for the user group

**Step 2:** Add users to the group

**Step 3:** Attach permissions policies (max 10 policies)

**Users:** An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

**Creating Users:**

**Step 1:** Add User Name(s)

**Step 2:** Select AWS Access Type

**(Access Key – Programmatic Access,**

**Password – AWS Management Console Access)**

**Step 3:** Set Permissions

Add User to user group/ Copy permissions from an existing user / Attach existing policies directly

**Step 4:** Add Tags

# *IAM (Identity & Access Management)*

**Roles:** An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

**Creating roles:**

**Step 1:** Select trusted entity (AWS Service/AWS Account/Web Identity etc…), Select Usecase

**Step 2:** Add Permissions → Choose one or more permissions to add to your role.

**Step 3:** Name (add role name), Review & Create.

# *IAM (Identity & Access Management)*

## Policies:

- A policy is an object in AWS that defines permissions
- A policy is an entity that, when attached to an identity or resource, defines their permissions.
- You can use the AWS Management Console, AWS CLI, or AWS API to create customer managed policies in IAM.
- Customer managed policies are standalone policies that you administer in your own AWS account.
- You can then attach the policies to identities (users, groups, and roles) in your AWS account.

# *AWS CLOUD SHELL*

- **AWS CloudShell** is a browser-based, pre-authenticated shell that you can launch directly from the AWS Management Console. You can run AWS CLI commands against services using your preferred shell (Bash, PowerShell, or Z shell).
- $ aws ec2 describe-key-pairs --key-name keypair3

# Chapter 1.3
# using AWS CLI V2

# *AWS CLI*

**https://docs.aws.amazon.com/cli/latest/reference/iam/**

**Creating Users:** Creates a new user.

Ex: aws iam create-user --user-name bh1

Creates a user named bh1

**Creating User Group:**

Creates a new group.

**Ex:** aws iam create-group --group-name Admins

Creates a group named ‘Admins’

**Creating Role:** Creates a new role for your Amazon Web Services account.

aws iam create-role --role-name Test-Role --assume-role-policy-document file://Test-Role-Trust-Policy.json

**get-user:** Retrieves information about the specified IAM user, including the user's creation date, path, unique ID, and ARN.

**Ex:** aws iam get-user --user-name sreedhar5 → creates a user ‘bhaskar8’

# *AWS CLI*

**get-group:** Returns a list of IAM users that are in the specified IAM group.

**Ex:** get-group –group-name <grpname>

**list-users:** Lists the IAM users that have the specified path prefix. If no path prefix is specified, the operation returns all users in the Amazon Web Services account. If there are none, the operation returns an empty list.

Ex: aws iam list-users

**list-groups:** Lists the IAM groups that have the specified path prefix.

**Ex:** aws iam list-groups

**list-roles:** Lists the IAM roles that have the specified path prefix.

**Ex:** aws iam list-roles

**delete-user:** Deletes the specified IAM user.

aws iam delete-user --user-name bhaskar1

# *AWS CLI*

**delete-group:** Deletes the specified IAM group. The group must not contain any users or have any attached policies.

**Ex:** aws iam delete-group --group-name MyTestGroup

**get-user:** Retrieves information about the specified IAM user, including the user's creation date, path, unique ID, and ARN.

**Ex:** aws iam get-user --user-name Paulo

**get-group:** Returns a list of IAM users that are in the specified IAM group.

**Ex:** aws iam get-group --group-name Admins

# *AWS CLI*

**Key Pair** → Create, List, Delete, Import

**Security Groups** → Create, Attach rules, List, Delete

**Keypairs:**

**Creation of Keypair:**

aws ec2 **create-key-pair** --key-name <keypairName>

**List Keypairs:**

aws ec2 **describe-key-pairs** --key-name <keypairName(s)>

**Delete keypair:**

aws ec2 delete-key-pair --key-name <KeypairName>

**Import Keypair:**

aws ec2 import-key-pair --key-name "AWS-CLI-IMPORTED-KEY" --public-keymaterial file://EC2CreationViaAWSCLI.pub

# *AWS CLI*

**Security Groups** → Create, Attach rules, List, Delete

**Security Groups:**

**Creating Security Group:**

aws ec2 create-security-group --group-name <securityGroupName>

--description “security group description“

**Attach an Inbound Rule to Security Group:**

Once the security group is created, we will attach the CIDR and port to this newly created security group. In our case, we will attach SSH port (22) and the IP address as 0.0.0.0/0, which means you can access it from anywhere.

**Ex:**

aws ec2 authorize-security-group-ingress --group-name EC2CreationViaAWSCLI

--protocol tcp --port 22 --cidr 0.0.0.0/0

**List the Security Group:**

aws ec2 describe-security-groups --group-names <SecGroupName>

**To List All Security Groups:**

aws ec2 describe-security-groups

# *AWS CLI*

**To delete Security Group:**

aws ec2 delete-security-group --group-name <security Group Name>

| Amazon Machine Image Type | UserName |
|---|---|
| Amazon Linux | ec2-user |
| RHEL5 | root or ec2-user |
| Ubuntu | ubuntu |
| Fedora | fedora or ec2-user |
| SUSE Linux | root or ec2-user |

# *AWS CLI*

## Connecting to EC2 instance: From CLI/AWS

1. Select your instance and make sure your key (pem file) matches your key pair name

2. Use the chmod command (in bold below) to make sure your private key file isn't publicly viewable.

**chmod 400 /path_to_key/my_key.pem**

3. Finally, SSH to your EC2 instance

**ssh -i /path_to_key/my_key.pem ec2-user@public_dns_name**

**Public dns:**

ec2-13-235-104-29.ap-south-1.compute.amazonaws.com

**To run unix commands on windows machine download: (Gnu on Windows):**

https://github.com/bmatzelle/gow/releases/download/v0.8.0/Gow-0.8.0.exe

**Tutorial:**

https://www.clickittech.com/aws/connect-ec2-instance-using-ssh/

# EC2 - Connect

## Connecting to EC2 instance using SSH from Windows

**Step 1:** Open **PuttyGen** on your PC

Make sure that 'RSA' is selected in 'type of key to generate' (Bottom of dialog) in Parameters.

**Step 2:** click 'Load' and select the '.pem' file of keypair. Click ÓK'for the notification.

Loads a private key.

**Step 3:** Click on 'save private key'. Saves the loaded file as .ppk file

**Step 4:** open Putty. In Left side pane → SSH → Auth → select .ppk file selected.

**Step 5:** go back and select 'Session'. Specify in the 'hosname IP address of AWS EC2 instance.

**Step 6:** give username (in this case ec2-user)

# AWS SDK

# *AWS SDK*

- • AWS SDK for .NET
- • AWS SDK for C++
- • AWS SDK for Go
- • AWS SDK for Java V2
- • AWS SDK for JavaScript
- • AWS SDK for PHP V3
- • AWS SDK for Python
- • AWS SDK for Ruby V3

# Chapter 1.3
# Amazon EC2 using AWS SDK (Java)

# *EC2 using AWS SDK*

**Installing AWS SDK for Eclipse:**

**Prerequisites:**

1) AWS Subscription
2) Java 1.8 or later
3) Eclipse IDE for Java EE developers 4.2 or later

**To install the AWS Toolkit for Eclipse:**

1. Within Eclipse, click Help and then click ***'Install New Software'***.

2. In the Work with box, type ***https://aws.amazon.com/eclipse***; and then press Enter.

3. Choose the components of the Toolkit for Eclipse that you want to install. Click Select All to install all components at once. (AWS Toolkit for Eclipse Core (in the AWS Core Management Tools section) is required; all other components are optional.)

4. Once you have made your selections, click Next (or Finish) to complete installation

# *EC2 using AWS SDK*

**KeyPairs →** Create

List

Delete

Import

**Secrurity Groups →** Create

List

Delete

Attach an Inbound Rule

**EC2 Instance →** Get AMI ID

select instance type

~~launch EC2 instance~~

list

get status

start-stop-terminate instances

Instance tags

**AWSCredentials object →**

# *AWD SDK – AWS Credentials object*

- To connect with AWS for Java using SDK, you need to provide the AWS Credentials.
- There are various ways to provide AWS credentials:
  1. Default Credential Provider
  2. Specify Credential Provider or Provider Chain
  3. Provide Access Key, and Secret Key

**1. Default Credential Provider:**

- This is the default provider when we don't provide any parameters while creating an AWS service client. **Default AWS Credentials Provider Chain** is the class that will be used to create the AWS credentials

This class will find the credentials in the following order:

- **Environment Variables:** This technique uses the Environment Variable Credentials Provider class that fetches values for AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY.
- **System Properties:** This technique is used to find the values for aws.access KeyId and aws.secret Key in Java system properties.

# *AWD SDK – AWS Credentials object*

- **Credentials Property File:** A default credentials property file can be loaded to get these credentials. This property file resides under C:\Users\{USER}\.aws\credentials or ~/.aws/credentials. This path differs based on the operating system

- Use **Profile Credentials Provider** to get the credentials details from the credentials folder in c:\users\{Username}\.aws\credentials

**Ex:**

- AWS Credentials credentials = new Profile Credentials Provider("default").getCredentials();

**2. Specify Credential Provider or Provider Chain:** There are different providers you can use to create the AWS Credentials object.

You can even create your own class by doing the following:

- Implementing AWS Credentials Provider interface
- Sub-classing the AWS Credentials Provider Chain class

# *AWD SDK*

**3. Provide Access Key and Secret Key Explicitly:**

There may be cases when we need to explicitly pass the access key ID and secret access key to create an AWS Credentials object.

**To create an AWS Credentials object**, we will use the following:

```
AWS Credentials credentials =
new Basic AWS Credentials("access_key_id", "secret_access_key");
```

**Specifying default AWS region:**

- You should set a default AWS Region that will be used for accessing AWS services with the AWS SDK for Java. For the best network performance, choose a region that's geographically close to you (or to your customers).
- If you don't select a region, then **us-east-1** will be used by default.

1. **Specify in C:\Users\USERNAME\.aws\config file on Windows**

```
[default]
region = ap-south-1
```

# *AWD SDK – Working wth Security Groupd*

**Operations:**

- ✓ creating a security group,
- ✓ attaching an inbound rule to a security group,
- ✓ listing a security group, and
- ✓ deleting a security group

**https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/examples-ec2-instances.html**

# Chapter 1.3
# Amazon CloudWatch

# *CLOUD WATCH*

- Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), IT managers, and product owners.
- CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, and optimize resource utilization.
- CloudWatch collects monitoring and operational data in the form of **logs, metrics, and events.**
- You get a unified view of operational health and gain complete visibility of your AWS resources, applications, and services running on AWS and on-premises.
- You can use CloudWatch to detect anomalous behaviour in your environments, set alarms, visualize **logs and metrics** side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.
- You can create the CloudWatch alarm either using the EC2 Management Console or the CloudWatch Console
- Set alarms and automate actions based on predefined thresholds or on machine learning (ML) algorithms that identify anomalous behavior in your metrics. For example, you can start Amazon EC2 Auto Scaling automatically or stop an instance to reduce billing

# *CLOUD WATCH*

Amazon EC2 sends metrics to Amazon CloudWatch. You can use the AWS Management Console, the AWS CLI, or an API to list the metrics that Amazon EC2 sends to CloudWatch

**To enable detailed monitoring:**

Select the instance → click on 'Monitoring' → 'manage detailed monitoring' → 'Enable'

- ✓ Use CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud instances, AWS CloudTrail, or other sources.

**Create an Alarm Using the EC2 Management Console:**

1. Go to the EC2 Management Console and select "Instances" from the menu.
2. Select the instance for which you want to create alarm.
3. Click '+' button in the 'Alarm Status' column to add an alarm.

**For EC2 instance: monitoring metrics**

- ✓ CPU utilization
- ✓ Disk read/write bytes
- ✓ Disk read/write ops (operations)
- ✓ Network in/out
- ✓ Network packets in/out
- ✓ CPU credits usage/balance
- ✓ Status check failed: either/instance/system

# *SQS using AWS CLI & SDK*

- add-permission
- remove-permission
- List Dead Letter Source Queues
- Receive Message with Change Visibility
- Receive Message with Change Visibility Batch

# *EC2 using CLI*

- You can access the features of Amazon Elastic Compute Cloud (Amazon EC2) using the AWS Command Line Interface (AWS CLI).
- configure the settings that the AWS Command Line Interface (AWS CLI) uses to interact with AWS. These include your security credentials, the default output format, and the default AWS Region.

**To get the Help:** aws ec2 help

**Configuring CLI :** aws configure

Get credential report from IAM dash board:
https://console.aws.amazon.com/iam/home#/credential_report

**Security Groups** → Create, Attach rules, List, Delete

**Key Pair** → Create, List, Delete, Import

**Ec2 instance:**

- Describe images
- Select image type
- Launch EC2
- List
- Get Status
- Assign tags
- Start, Stop and Terminate instance