# Introduction to Ec2 and S3

# What's in it for you

| Introduction to Cloud Computing | | |
|---|---|---|
| **S.NO.** | **AGENDA** | **TIME SLOT** |
| 1 | Introduction to AWS EC2 | 15 mins |
| 2 | Introduction to AWS S3 | 15 mins |
| 3 | Demo | 30 mins |
| 4 | Queries | - |


# Speakers

## Sanchit Jain

Lead Architect - AWS at Quantiphi
AWS APN Ambassador
2015 Batch Alumni

# Introduction to AWS EC2

01

# Introduction

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud.

- Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster
- Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic
- Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use
- Benefits
    - Elastic Web-Scale Computing
    - Complete Controlled
    - Flexible Cloud Hosting Service
    - Integrated
    - Reliable & Secure
    - Less Expensive

# Instance Types

- **General-Purpose :** This family instances provide a balance of CPU, memory, and network resources making them a good choice for many applications.
- **Compute-Optimized :** This family instances are geared towards applications that benefit from high compute power.
- **Memory-Optimized :** This family instances are designed for memory-intensive applications
- **Storage-Optimized :** This family instances provides you with direct-attached storage options optimized for applications with specific disk I/O and storage capacity requirements
- **GPU Instance :** This family instances allows you to take advantage of the parallel performance of NVidia Tesla GPUs using the CUDA or OpenCL programming models for GPGPU computing

# Instance Types

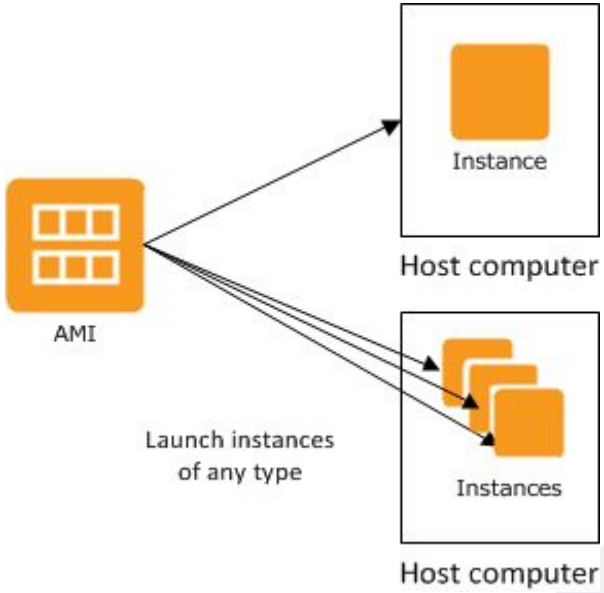
| Instance Family | Current Generation Instance Types |
|---|---|
| General purpose | `t2.nano` \| `t2.micro` \| `t2.small` \| `t2.medium` \| `t2.large` \| `m4.large` \| `m4.xlarge` \| `m4.2xlarge` \| `m4.4xlarge` \| `m4.10xlarge` \| `m3.medium` \| `m3.large` \| `m3.xlarge` \| `m3.2xlarge` |
| Compute optimized | `c4.large` \| `c4.xlarge` \| `c4.2xlarge` \| `c4.4xlarge` \| `c4.8xlarge` \| `c3.large` \| `c3.xlarge` \| `c3.2xlarge` \| `c3.4xlarge` \| `c3.8xlarge` |
| Memory optimized | `r3.large` \| `r3.xlarge` \| `r3.2xlarge` \| `r3.4xlarge` \| `r3.8xlarge` |
| Storage optimized | `i2.xlarge` \| `i2.2xlarge` \| `i2.4xlarge` \| `i2.8xlarge` \| `d2.xlarge` \| `d2.2xlarge` \| `d2.4xlarge` \| `d2.8xlarge` |
| GPU instances | `g2.2xlarge` \| `g2.8xlarge` |

# Amazon EC2 Pricing Factor

1. **On Demand Pricing**
   - With On-Demand instances, you pay for compute capacity by per hour or per second depending on instances type and no longer-term commitments or upfront payments are needed to be given
2. **Reserved Instances**
   - Reserved Instances provide you with a significant discount (up to 75%) compared to On-Demand instance pricing.
3. **Spot Instances**
   - Spot Instances are available at up to a 90% discount compared to On-Demand prices
4. **Dedicated Host**
   - A Dedicated Host is a physical EC2 server dedicated for a users.
   - Dedicated Hosts can help you address compliance requirements and reduce costs by allowing you to use your existing server-bound software licenses

# Instances AMIs

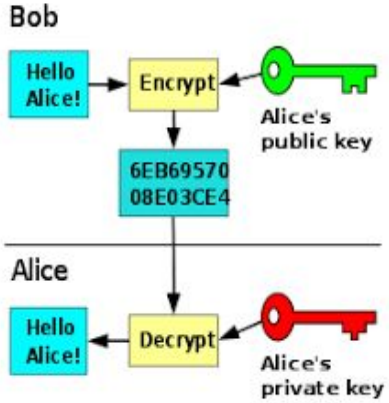
- An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications)
- From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud
- You can launch multiple instances of an AMI, as shown in the following figure



# Security

## EC2 Key Pairs

- When you launch an instance, you specify the key pair which you require to use.
- At the boot time, the public key content is placed on the instance in an entry within **~/.ssh/authorized_keys**
- To log in to your instance, you must specify the private key when you connect to the instance




## Security groups

- Security groups are virtual firewall that controls the traffic for an instance/RDS.
- When you launch an instance, you can specify one or more security groups
- You can add rules to each security group that allow traffic to or from its associated instances (Inbound & Outbound Rules)

# Introduction to AWS S3

# 02

# Overview

- Amazon Simple Storage Service (Amazon S3) is storage for the Internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web.

- Amazon S3 stores data as objects within buckets. An object consists of a file and optionally any metadata that describes that file

- Types of S3 Storage:

    - S3 Standard.
    - S3 Standard Intelligent Tiering.
    - S3 Standard IA.
    - S3 Standard one zone IA.
    - S3 Glacier.
    - S3 Deep Archive.

# Overview

| | S3 Standard | S3 Intelligent-Tiering* | S3 Standard-IA | S3 One Zone-IA† | S3 Glacier | S3 Glacier Deep Archive |
|---|---|---|---|---|---|---|
| Designed for durability | 99.999999999% (11 9's) | 99.999999999% (11 9's) | 99.999999999% (11 9's) | 99.999999999% (11 9's) | 99.999999999% (11 9's) | 99.999999999% (11 9's) |
| Designed for availability | 99.99% | 99.9% | 99.9% | 99.5% | 99.99% | 99.99% |
| Availability SLA | 99.9% | 99% | 99% | 99% | 99.9% | 99.9% |
| Availability Zones | ≥3 | ≥3 | ≥3 | 1 | ≥3 | ≥3 |
| Minimum capacity charge per object | N/A | N/A | 128KB | 128KB | 40KB | 40KB |
| Minimum storage duration charge | N/A | 30 days | 30 days | 30 days | 90 days | 180 days |

# S3 Features

- Simple to use from console/app. REST API’s and SDK’s from third party technology integration.
- 99.999999999% durability
- Scalable and elastic
- Security - automatic encryption once uploaded. Control access using IAM (Identity and Access Management)
- Availability is 99.99% over an year. Choice of region for lower latency.
- Integrated with other AWS services.
- Easy to manage.

# Demo - AWS EC2

# 03

# Demo - Host a static website

- In this exercise, you will do the following:
  - Hosting on Ec2
    - Launch an EC2 instance through the AWS console (i.e. build a “virtual laptop” to serve your website)
    - SSH into to the EC2 instance and install a web server (i.e. use the terminal to “log in” to the “virtual laptop” and interact with it)
    - Host a static webpage on the EC2 instance (i.e. “deploy” a static html file to the web)
  - Hosting on S3 bucket

# Demo | Launch an Ec2

- **Note**: If you haven't created an AWS account yet, do that first
- Sign in to the AWS console and search for "EC2". Navigate to the EC2 dashboard and click "Launch Instance"
    - Choose AMI
    - Choose Instance Type
    - Configure Instance
    - Add Storage
    - Add Tags
    - Configure Security Group
    - Review

# Demo | Configure

- **Note**: If you haven't install the Mobex Term utility (only for Windows users) yet, do that first
- Selecting the instance (click the button next to the instance) displays information about the instance below. In this area, you will see the IPv4 Public IP address of your instance. Copy it to your clipboard
  - Provide permissions on your key-pair file: `chmod 400 <path_to_key_pair_file>`
  - SSH into your new EC2 instance: `ssh -i <key_pair_file> ec2-user@<public_ip>`
  - Update all of the packages on the instance: `sudo yum update -y`
  - Install an apache webserver: `sudo yum install httpd -y`
  - Start the webserver: `service httpd start`
  - (Optional) Configure the web server to restart if it gets stopped: `chkconfig httpd on`

# Demo | Deploy the pages

- In this section you will create an index.html file to be served.
  - Navigate to the directory: `cd /var/www/html`
  - Manually create an index.html file in this directory: `vi index.html`
  - Add valid html to the file: `<html><body>My first EC2 instance!!</body></html>`
  - Exit and save. Make sure that the file has content: `cat index.html`
  - Start the webserver: `service httpd start`
  - Navigate back to the EC2 dashboard in the AWS console and copy the IPv4 Public IP address of your instance. Paste that address into your browser. If all went well, you will see the html that you just created!

# Demo - AWS S3

04

# Demo

- Create a Bucket in S3.
- Upload Files
- Retrieve Your Website URL

Queries ?

05

# THANK YOU