

Unit 5

By Tamal Dey,
MCA, PESU

Topics

- **IoT on Cloud**
 - IoT Core
 - IoT Analytics
- Machine Learning on cloud
 - QuickSight
 - SageMaker
- Blockchain on Cloud

IOT with AWS

- Rahul C Shekhar (PES1201802486)
- Rexibond Sohkhlet (PES1201702090)

What is IoT ?

- The internet of things, or IoT, is a system of interrelated computing devices that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Examples of IoT

- Smart House
- Agriculture
- Smart Cars
- Wearables
- Smart City

Why AWS for IoT ?

- High IoT Security Standards.
- Serverless Architecture.
- Powerful AWS IoT Analytics Paired With AI and Machine Learning.
- AWS Has a Strong Partner Network of IoT Device Manufacturers.
- Integration Across a Sheer Number of AWS Products and Services.

Hardware and Software Requirements

- Raspberry Pi 3 Model B
- DHT11 - Temperature and Humidity Sensor
- Connecting wires
- WinSCP

AWS Services for IoT Setup

1. IoT Core
2. IoT Analytics
3. Amazon SageMaker
4. QuickSight

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.
- With AWS IoT Core, your applications can keep track of and communicate with all your devices, all the time, even when they aren't connected.
- AWS IoT Core provides authentication and end-to-end encryption throughout all points of connection, so that data is never exchanged between devices and AWS IoT Core without proven identity.

AWS IoT Core

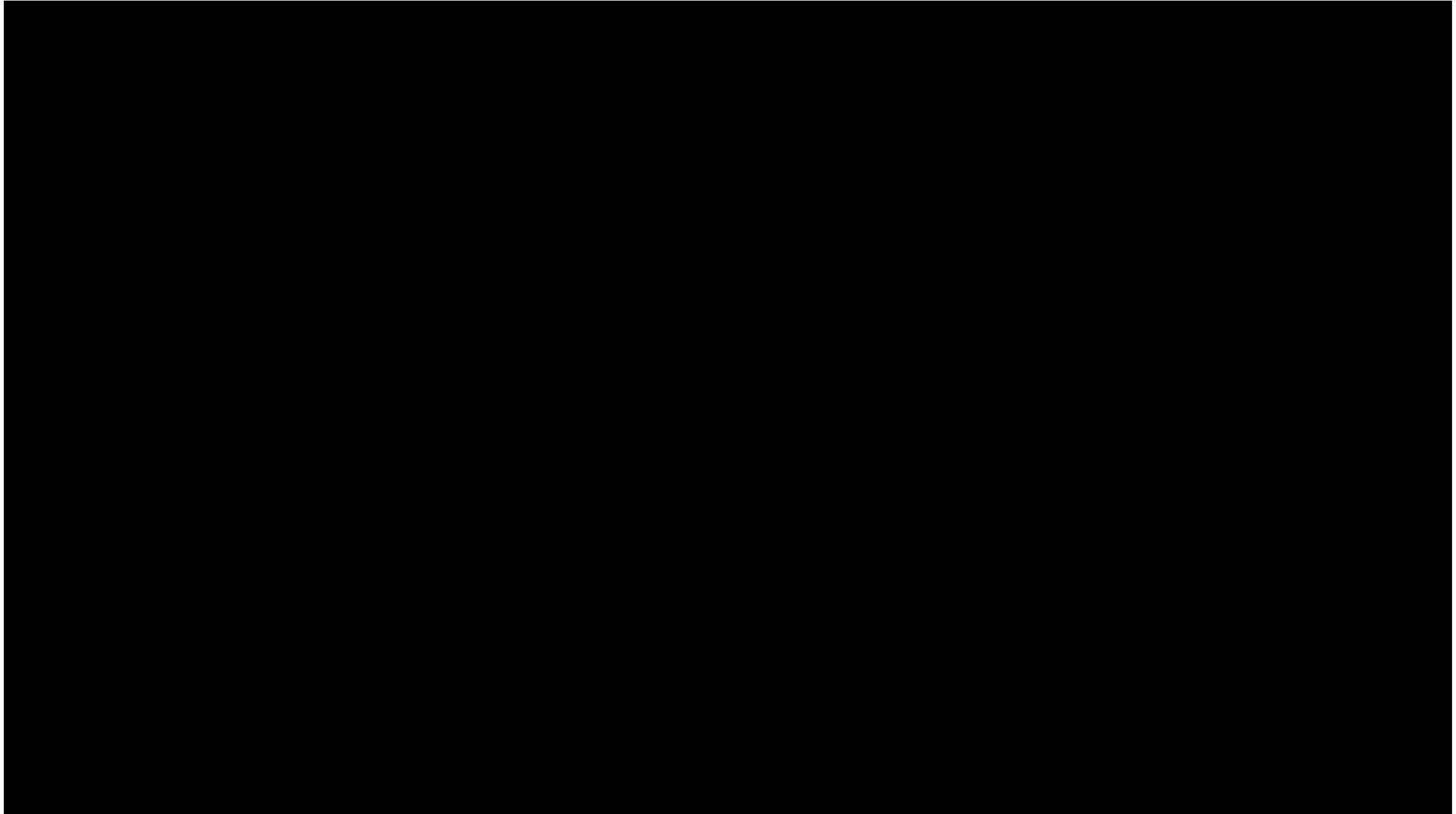


By: Tamal Dey, MCA, PESU

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
- It is the easiest way to run analytics on IoT data and get insights to make better and more accurate decisions for IoT applications and machine learning use cases.

IoT Analytics Video



By: Tamal Dey, MCA, PESU

Amazon SageMaker

- Amazon SageMaker is a fully-managed service that enables data scientists and developers to quickly and easily build, train, and deploy machine learning models at any scale.
- Amazon SageMaker includes modules that can be used together or independently to build, train, and deploy your machine learning models.
- Amazon SageMaker manages all of the underlying infrastructure for you and can easily scale to train models at petabyte scale.

Amazon SageMaker



By: Tamal Dey, MCA, PESU

Amazon QuickSight

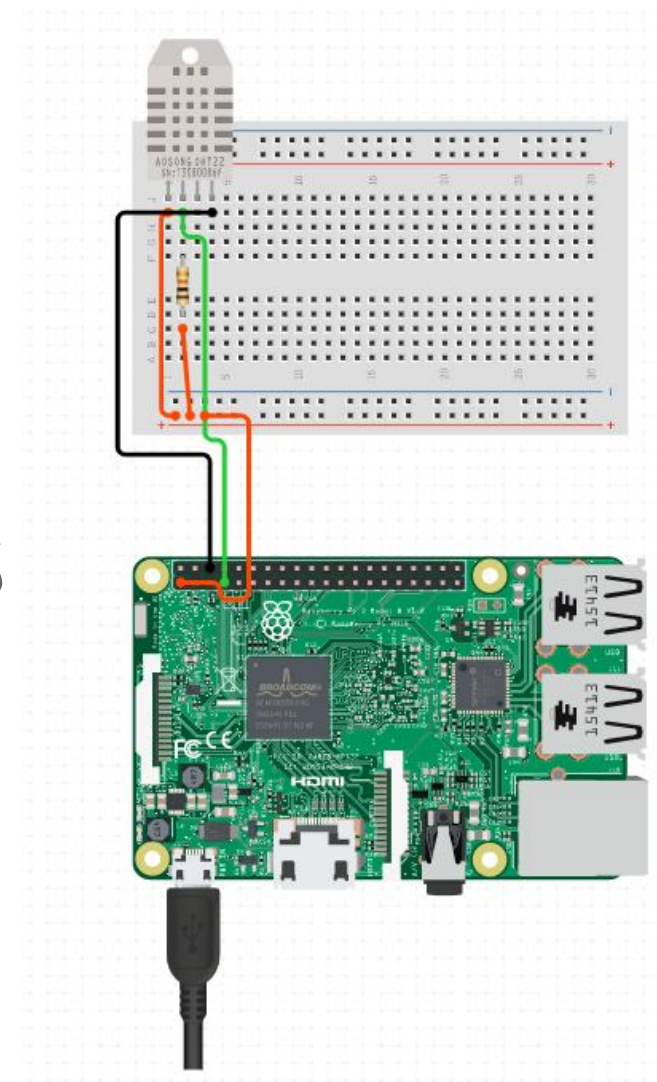
- Amazon QuickSight is a fast, cloud-powered business intelligence service that makes it easy to deliver insights to everyone in your organization.
- QuickSight lets you easily create and publish interactive dashboards that include ML Insights. Dashboards can then be accessed from any device, and embedded into your applications, portals, and websites.

Amazon QuickSight



By: Tamal Dey, MCA, PESU

Connections



By: Tamal Dey, MCA, PESU

Setup the Thing for IoT

Setup the Thing

- Step 1 : Search for IoT Core in AWS Services.
- Step 2 : Click on manage.
- Step 3 : Click on Register a thing.
- Step 4 : Click on Create a single thing.
- Step 5 : Enter Name : DHT11 and click on Next.
- Step 6 : Click on Create certificate of One-click certificate creation(recommended) option.
- Step 7 : After the certificate creation page gets redirected.

Setup the Thing

- Step 8 : Download certificate.pem, private.key and root CA for AWS.
- Step 9 : Click on Activate.
- Step 10 : Click on Done.
- Step 11 : Click on Secure —> Policies —> Create a policy.
- Step 12 : Enter Name : DHT11, Action : `iot:*`, Resource ARN : `*`, Effect : **Allow**.
- Step 13 : Click on Create.
- Step 14 : Click on Certificates —> Options.

Setup the Thing

- Step 15 : Click on Attach a policy.
- Step 16 : Click on Act —> Create.
- Step 17 : Enter Name : DHT11, Rule query Statement : `SELECT * FROM 'DHT11/data'`.
- Step 18 : Click on Add action.
- Step 19 : Select Send a message to IoT Analytics and click on Configure action.
- Step 20 : Select Quick create IoT Analytics resources and enter Resource Prefix : DHT11 and click on Quick create.
- Step 21 : Click on Add action and Create rule.

Setup the Raspberry Pi for IoT

Setup the Raspberry Pi

- Step 1 : To install the SDK on your Pi, open the terminal and type the following command :

>git clone <https://github.com/aws/aws-iot-device-sdk-python.git>

- Step 2 : This should have installed the *aws* directory on your Pi, now navigate into it using the following command :

>cd aws-iot-device-sdk-python

Step 3 : Inside the directory install the setup file using the line below :

>python setup.py install

Step 4 : Create a folder "DHT11" in your home directory.

- Step 5 : Using WinSCP connect to your Raspberry Pi and place the certificate.pem.crt, private.pem.key, rootCA.pem, aws.py and place them inside your DHT11 folder.

Setup the Raspberry Pi

- Step 6 : Edit the aws.py file and update the endpoint configuration which can be found when you click on
DHT11 —> Interact —> HTTPS
- Step 7 : Save the aws.py after making the changes.
- Step 8 : To run the program, open the terminal and type the following code :
 >python aws.py
- Step 9 : Once the program starts executing, click on AWS IoT —> Test —> Subscribe to a topic.
- Step 10 : In Subscription topic enter DHT11/data and click on subscribe to topic

Setup the Amazon SageMaker for IoT

Setup the Amazon SageMaker

- Step 1 : Search for Amazon SageMaker in AWS Services.
- Step 2 : Click on Create notebook instance.
- Step 3 : Enter Notebook instance name : DHT11 and click on Create notebook instance.
- Step 4 : Once your newly created notebook instance is ready click on Open Jupyter.
- Step 5 : Click on New and select conda_python3.

Setup the QuickSight for IoT

Setup the QuickSight

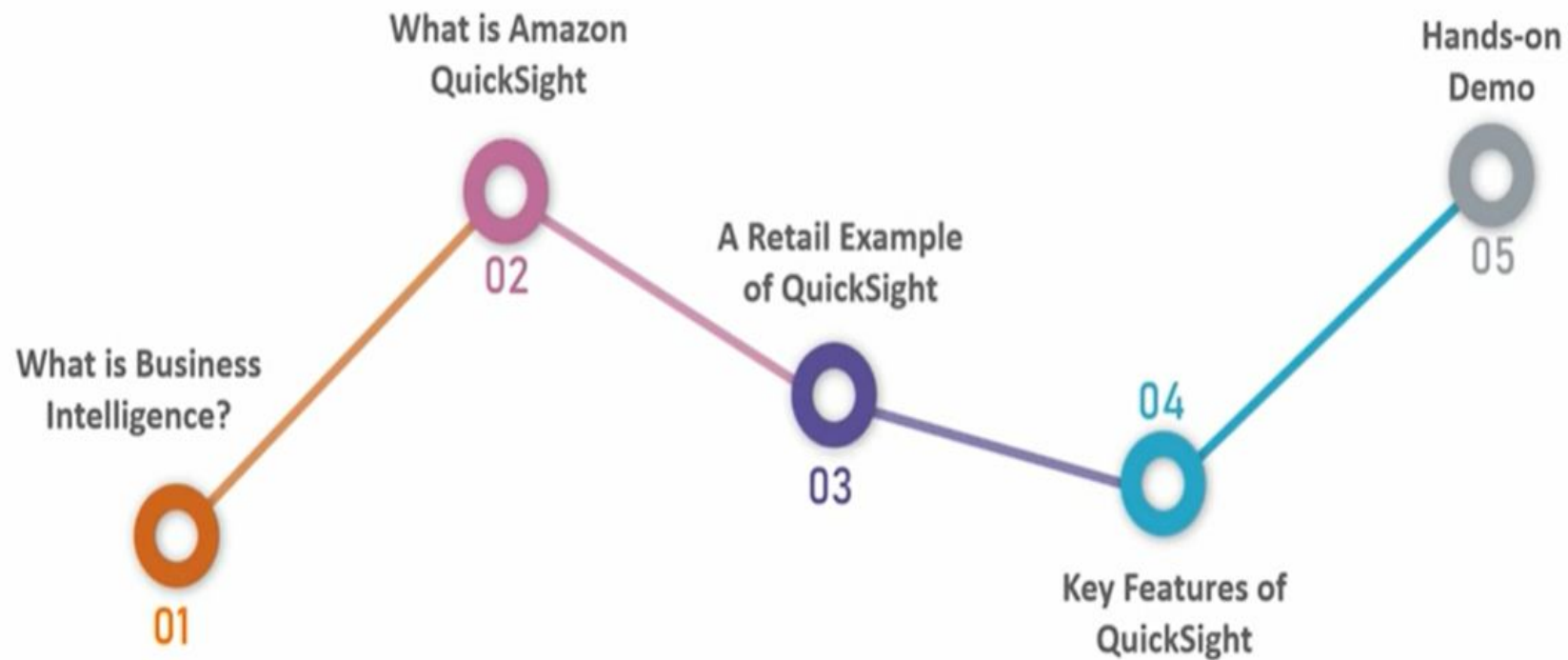
- Step 1 : Search for QuickSight in AWS Services.
- Step 2 : Click on New analysis.
- Step 3 : Select your dht11_dataset created in IoT Analytics.
- Step 4 : Click on Create analysis.

Cloud Analytics

QuickSight

By: Tamal Dey,
MCA, PESU

Agenda



What is Business Intelligence Tools?

- How Easier or Difficult to Find your Books in PESU Library?
 - a) Easy
 - b) Difficult
 - c) Need Assistance (Librarian)
- Solution
 - Categories of Books
 - Label the Books
 - Use Color code / Sequence Numbering

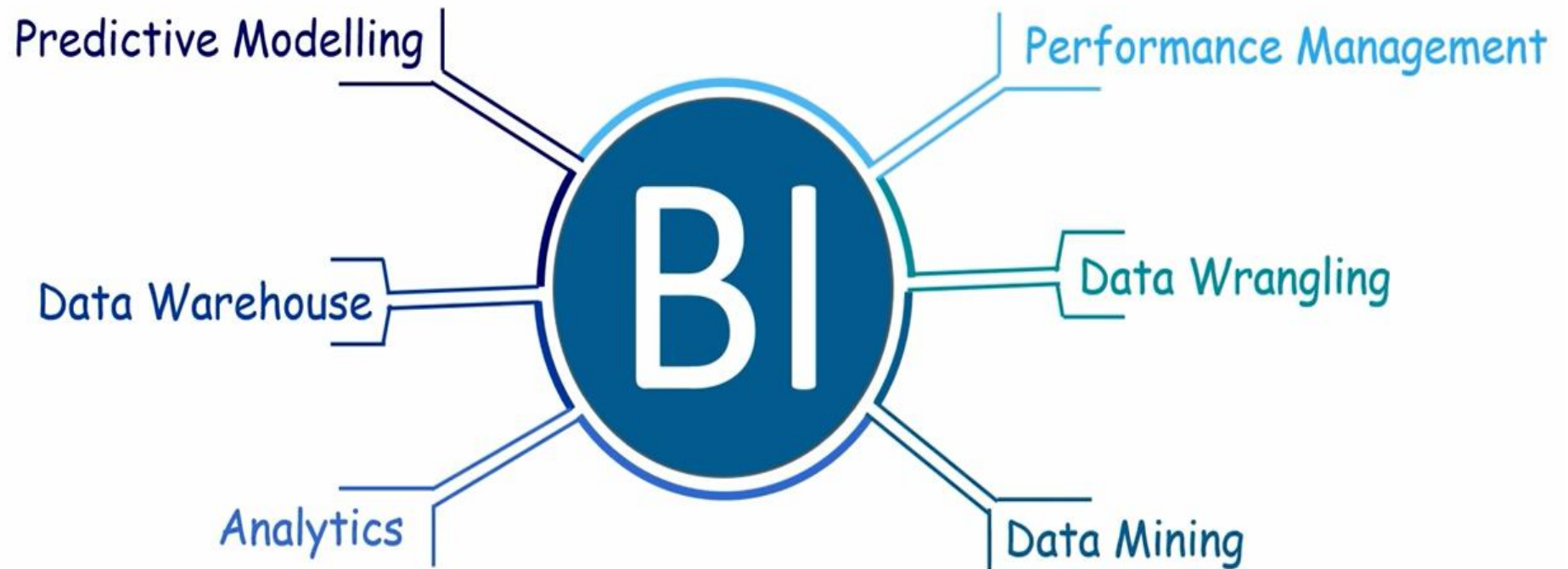
Business Intelligence



BI

Delivering Relevant & Reliable
information, to the right people, at the
right time with the goal of achieving
better decisions faster

Business Intelligence Buzzwords



Business Intelligence

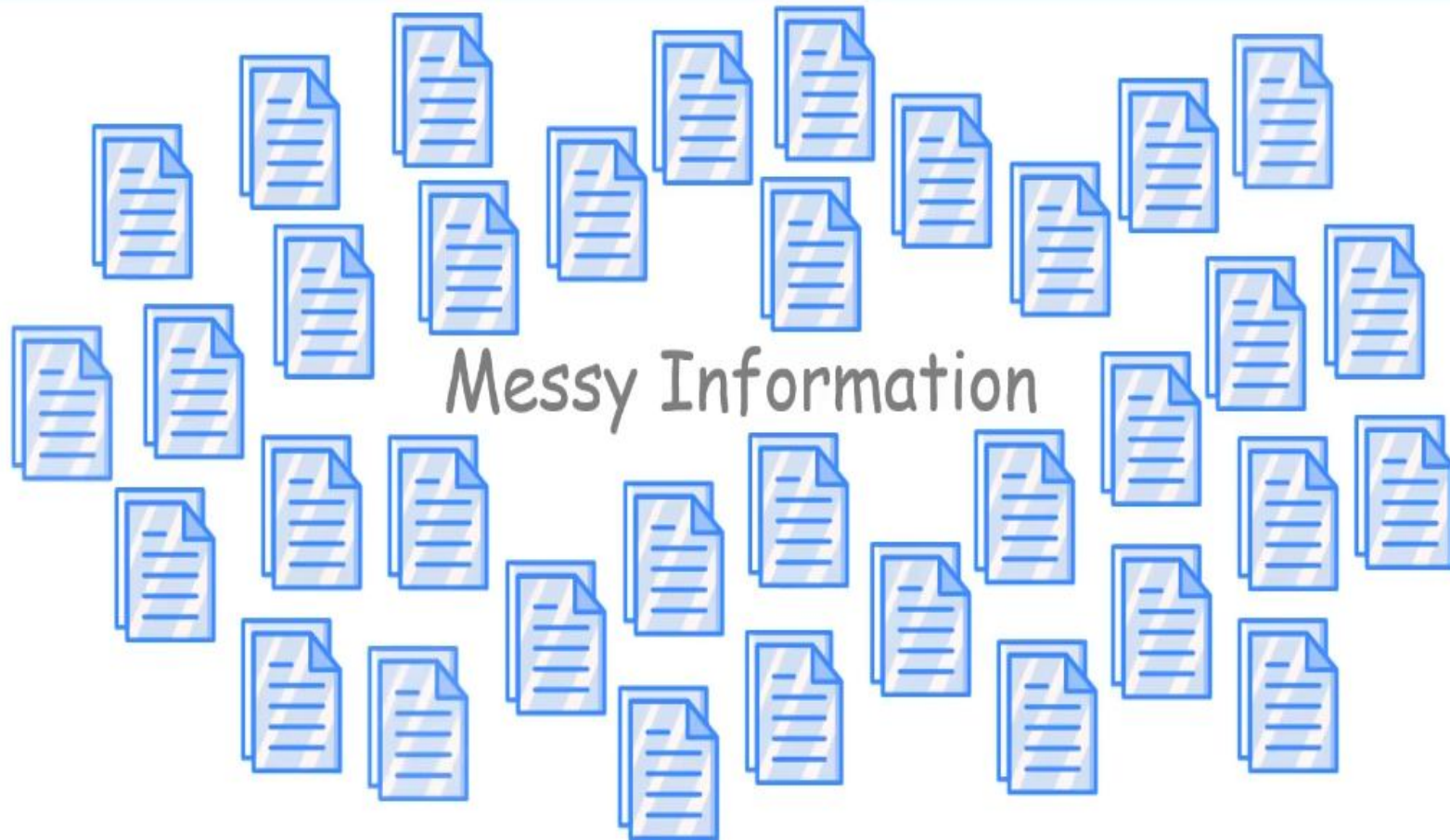
- Performance Management. For-Org., Dept., Emp.
- **Data Wrangling-** Collect from data sources and structure data
 - (Data acquisition, combining and Data cleansing)
- **Data mining-** Find hidden pattern and predict steps
- **Data Analytics-** Predict more selling brand and type of shoes
- **Data warehouse-** store past old record (10 years +)
- **Predictive modelling-** Find the key factor and collect information and make statistical model on linear data or analytics
 - Use Data Mining and Probability to get result.

Amazon QuickSight

Amazon QuickSight is a fast, cloud-powered BI service that makes it easy to build visualizations, perform ad-hoc analysis, and quickly get business insights from your data.



Amazon QuickSight A Retail Business Example



Amazon QuickSight



Amazon QuickSight



Smart Visualizations

Lets take a look at retail example



Stores

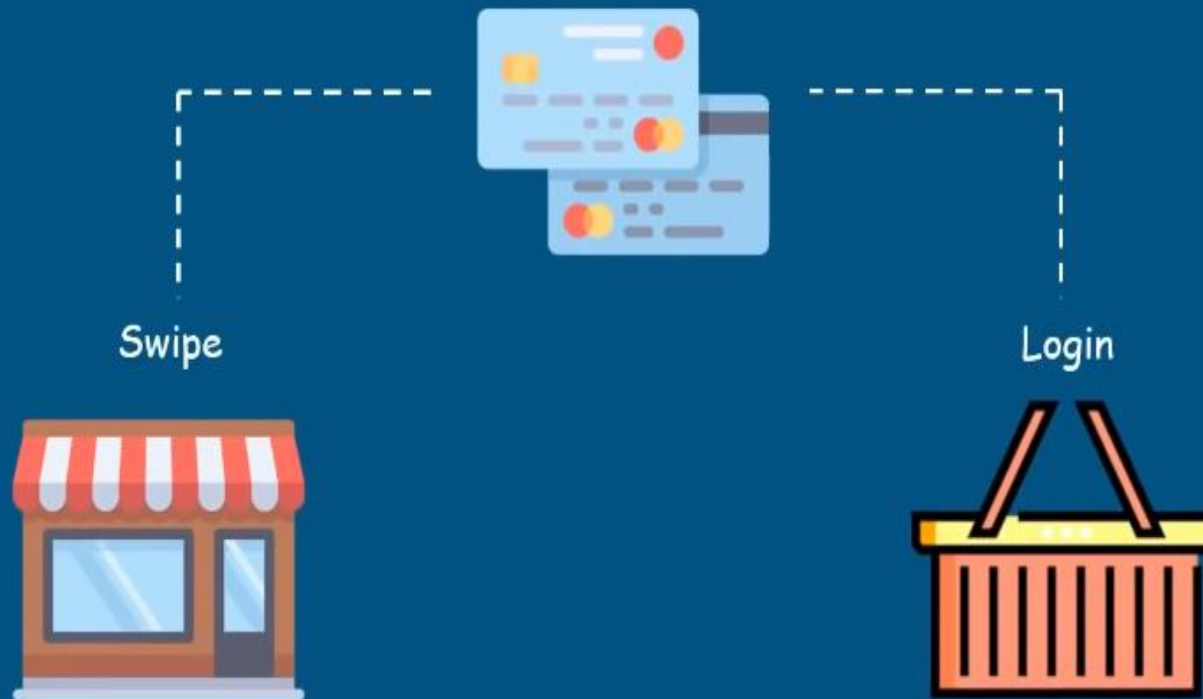


Online Shops



Manager

Lets take a look at retail example



Lets take a look at retail example



Associates everything a customer buys from your company in-store or online with his unique account number into your organisation database

Lets take a look at retail example

Structures and Models the Data



Data Base



Amazon QuickSight



If a customer is being loyal to
your brand or not?
What products customers buy
and how frequently?
If customer prefers visiting
store or shop online?

Features Of QuickSight



Option to choose
data source of
your choice

SPICE - Superfast,
Parallel, In-memory
engine



Smart Visualizations
which you can
share.

Option to view stats
and graphs
anywhere, anytime.



Features of Quick Sight

- **Data store-** MySQL, S3, RedShift or on premises connectivity
- **Spice-**Unique columnar storage and innovative data compression technique .
 - Its very sleek and fast.
 - Allow to perform interactive query on large data set rapidly with replication facility
- Visualize and Scalable in different devices

Quick Sight-Key Concepts

- Quick sight is a fully managed , serverless cloud business intelligence system
- No infrastructure is required. Fully managed by cloud.
- Data import, data preparation and build dashboards to visualize the data
- Connect to your data wherever it is .
- It can integrate with AWS Data sources, as well as on-premise and hosted databases and third party business applications
- It continue to expand to include Data sources from any source

Demo

- Sign up with your AWS account
 - purchase **standard user subscriptions** to get discounted pricing on Amazon QuickSight. When you create an Amazon QuickSight account, you automatically get one free user account. When you invite additional users to Amazon QuickSight, you are charged for those user accounts on a month-by-month basis.
- View sample charts
- Create new analysis
 - Visit **data.gov**
 - (Choose one . E.g. **Seattle Crime Stats by Police Precinct**)
 - Visualize and use filter
 - Share and Domain

Reading resources

- Quicksight-
<https://www.youtube.com/watch?v=WaOrQtxLXfs>
- <https://www.youtube.com/watch?v=CFurUjiJaQM&t=941s>
- <https://www.youtube.com/watch?v=dprtSTSbCEE&t=4s>
- <https://www.youtube.com/watch?v=hh6ltJbCUQ0>
- <https://catalog.data.gov/dataset/police-incidents-01012005-to-current>

Machine Learning On AWS Platform (SageMaker & Jupyter Notebook)

By Subham Singh
MCA (PES University)

Machine Learning On AWS (with categories)

- ML Service → **Build train** and **deploy** ML Fast
- AI Service → Easily **add Intelligence** to your applications
- Frameworks → **Choice and flexibility** with broadest framework support
- Compute → **Fastest and Lowest Cost Compute Option**
- Analytics and Security → **Comprehensive capabilities**, no compromise
- Learning Tools → Get deep on ML with **AWS DeepRacer and DeepLens**

What is Machine Learning ?

Machine Learning is the idea

where there are some **generic algorithm** that can tell you something interesting about a set of data without you having write any custom code for specific to the problem

Instead of writing code, you feed data to the **generic algorithm** and it builds its own logic based on the data.

(taken from [Blog](#) on Medium)

Machine Learning On AWS Platform

AWS provides a diverse platform for the machine learning and its computation :

Amazon Sagemaker Steps

Step 1: Create an Amazon S3 Bucket

Step 2: Create an Amazon SageMaker Notebook Instance

Step 3: Create a Jupyter Notebook

Step 4: Download, Explore, and Transform the Training Data

Step 5: Train a Model

Step 6: Deploy the Model to Amazon SageMaker

Amazon Sagemaker Steps

Step 7: Validate the Model

Step 8: Clean Up

Step 9: Integrating Amazon SageMaker Endpoints into Internet-facing Applications

Step 1: Create an Amazon S3 Bucket

Download two dataset from below link

1. <https://s3.amazonaws.com/aml-sample-data/banking.csv> (training data)
2. <https://s3.amazonaws.com/aml-sample-data/banking-batch.csv>. (testing data)

Open your dataset banking.csv (mostly open in Excel)

You can see the **header row** contains the attribute names for each column

Here attribute means → property to describe particular characteristics of each customer.(banking dataset)

upload the **banking.csv** and **banking-batch.csv** files to Amazon S3

Step 2: Create an Amazon SageMaker Notebook Instance

1. Open the Amazon SageMaker console at <https://console.aws.amazon.com/sagemaker/>.
2. Choose **Notebook instances**, then choose **Create notebook instance**.
3. On the **Create notebook instance** page, provide the following information (if a field is not mentioned, leave the default values):
 - a. For **Notebook instance name**, type a name for your notebook instance.
 - b. For **Instance type**, choose ml.t2.medium. This is the least expensive instance type that notebook instances support, and it suffices for this exercise.
 - c. For **IAM role**, choose **Create a new role**, then choose **Create role**.
 - d. Choose **Create notebook instance**.

Step 3: Create a Jupyter Notebook

Open the Notebook Instances,

and then open the notebook instance you created by choosing

either **Open Jupyter** for classic Jupyter view

or **Open JupyterLab** for JupyterLab view

next to the name of the notebook instance.

Step 3: Create a Jupyter Notebook

Click on **Open Jupyter** Under **Action Column**.

Jupyter Notebook windows will Open

If you opened the notebook in Jupyter classic view, on the **Files** tab, choose **New**, and **conda_python3**.

This preinstalled environment includes the default Anaconda installation and Python 3.



Following steps will be explained in Jupyter Notebook

Create (S3)
Hoteltips.csv



meal_amt	tips
34	5
108	17
64	11
88	8
153	14
51	5
100	15
205	10
55	5
60	5
75	10
90	10
150	20
200	10
70	5
80	10

Code #import libraries to execute

#import libraries to execute

Draw the graph	import matplotlib.pyplot as plt
Calculation	import numpy as np
Implementing Technique	from sklearn import linear_model
import dataset in notebook	import pandas as pd
Source	from sklearn.metrics
Evaluating the model	import mean_squared_error
Find SQRT of the Number	from math import sqrt

jupyter MLAWSPESU Last Checkpoint: 3 hours ago (autosaved)



File Edit View Insert Cell Kernel Widgets Help Not Connected Trusted conda_python3

⏏ + 🔍 📄 ⬆ ⬇ ⬆ Run ⏏ ↺ Code ⏏ nbdiff

```
In [1]: #import libraries to execute
import matplotlib.pyplot as plt
import numpy as np
from sklearn import linear_model
import pandas as pd
from sklearn.metrics import mean_squared_error
from math import sqrt
```

#Reading CSV from S3 Bucket

- #Reading CSV from S3 Bucket

```
df = pd.read_csv('https://mysmpesubbucket.s3.ap-south-1.amazonaws.com/ Hoteltips.csv', sep=',')
```

```
df.head()
```

or

```
df.tail()
```

Out[9]:

	meal_amt	tips
11	90	10
12	150	20
13	200	10
14	70	5
15	80	10

#Selecting features as X and Y

- #Selecting features as X and Y

```
X= df['meal_amt']
```

```
Y = df['tips']
```

```
print("Datatype of X: ", type(X))
```

```
print("Datatype of Y: ", type(Y))
```

```
#Selecting and reshaping the X and Y
```

```
X = X.values
```

```
X = X.reshape(len(X),1)
```

```
Y = Y.values
```

```
Y = Y.reshape(len(Y),1)
```

```
Datatype of X: <class 'pandas.core.series.Series'>
```

```
Datatype of Y: <class 'pandas.core.series.Series'>
```

#Split the data into training/testing sets

- #Split the data into training/testing sets

```
X_train = X[:-4]
```

```
X_test = X[-4:]
```

Split the targets into training/testing sets

```
Y_train = Y[:-4]
```

```
Y_test = Y[-4:]
```

#plot train Data

- #plot train Data

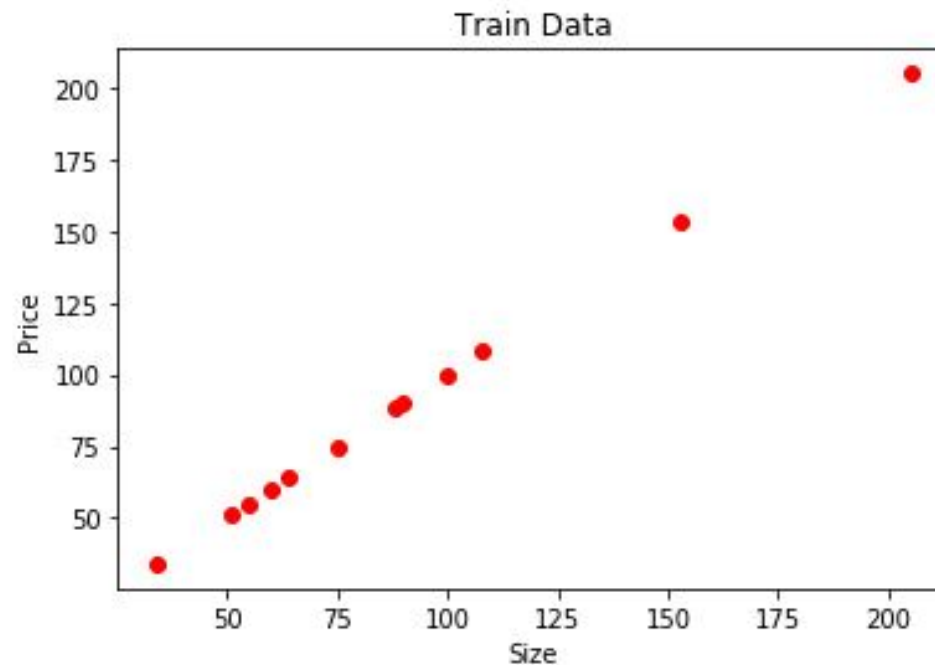
```
plt.scatter(X_train,Y_train, color='red')
```

```
plt.title('Train Data')
```

```
plt.xlabel('Size')
```

```
plt.ylabel('Price')
```

```
Text(0, 0.5, 'Price')
```



Create linear regression object

- # Create linear regression object

```
regr = linear_model.LinearRegression()
```

Train the model using the training sets

- # Train the model using the training sets

```
regr.fit(X_train, Y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,  
normalize=False)
```

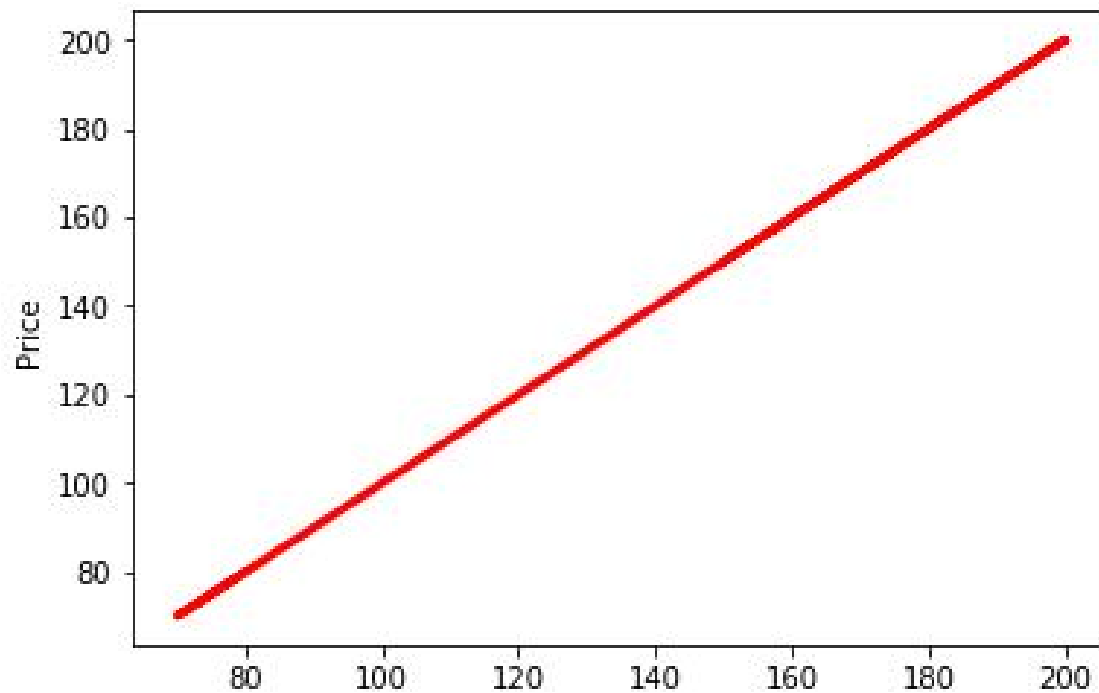
#Test the linear regression model on X_test Data

- #Test the linear regression model on X_test Data
`Y_pred=regr.predict(X_test)`

#Plot outputs->regression line is drawn

- #Plot outputs-> regression line is drawn
`plt.plot(X_test,Y_pred , color='red',linewidth=3)`
`plt.xlabel('Size')`
`plt.ylabel('Price')`


```
Text(0, 0.5, 'Price')
```



Root mean Squared Error

- `rms = sqrt(mean_squared_error(Y_test, Y_pred))`

```
print("RMS : ",rms)
```

RMS : 0.0

Reference

- <https://scikit-learn.org/>
- <https://docs.aws.amazon.com/sagemaker/index.html>
- <https://docs.aws.amazon.com/dlami/latest/devguide/setup-jupyter.html>

SaaS



By: Tamal Dey,
MCA, PESU

Introduction

- Amazon **Route 53** is a highly available and scalable cloud Domain Name System (DNS) web service.
- It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like **www.example.com** into the numeric IP addresses like **192.0.2.1** that computers use to connect to each other.
- Amazon Route 53 is fully compliant with IPv6 as well.
- Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon **EC2 instances**, **Elastic Load Balancing** load balancers, or Amazon **S3 buckets** – and can also be used to route users to infrastructure outside of AWS

1. Domain Registration

- **domain name:** The name, such as example.com, that a user types in the address bar of a web browser to access a website or a web application. Start by registering a domain name.
- **domain registrar:** A company that is accredited by ICANN (Internet Corporation for Assigned Names and Numbers) to process domain registrations for specific top-level domains (TLDs).
- **domain registry:** A company that owns the right to sell domains that have a specific top-level domain. For **example**, VeriSign is the registry that owns the right to sell domains that have a .com TLD.

Concepts

- **geographic top-level domains:** These TLDs are associated with geographic areas such as countries or cities.
- **alias record:** A type of record that you can create with Amazon Route 53 to route traffic to AWS resources such as Amazon CloudFront distributions and Amazon S3 buckets.
- **name servers:** Servers in the Domain Name System (DNS) that help to translate domain names into the IP addresses that computers use to communicate with one another.
- **record (DNS record):** An object in a hosted zone that you use to define how you want to route traffic for the domain or a subdomain.

Types of routing policy

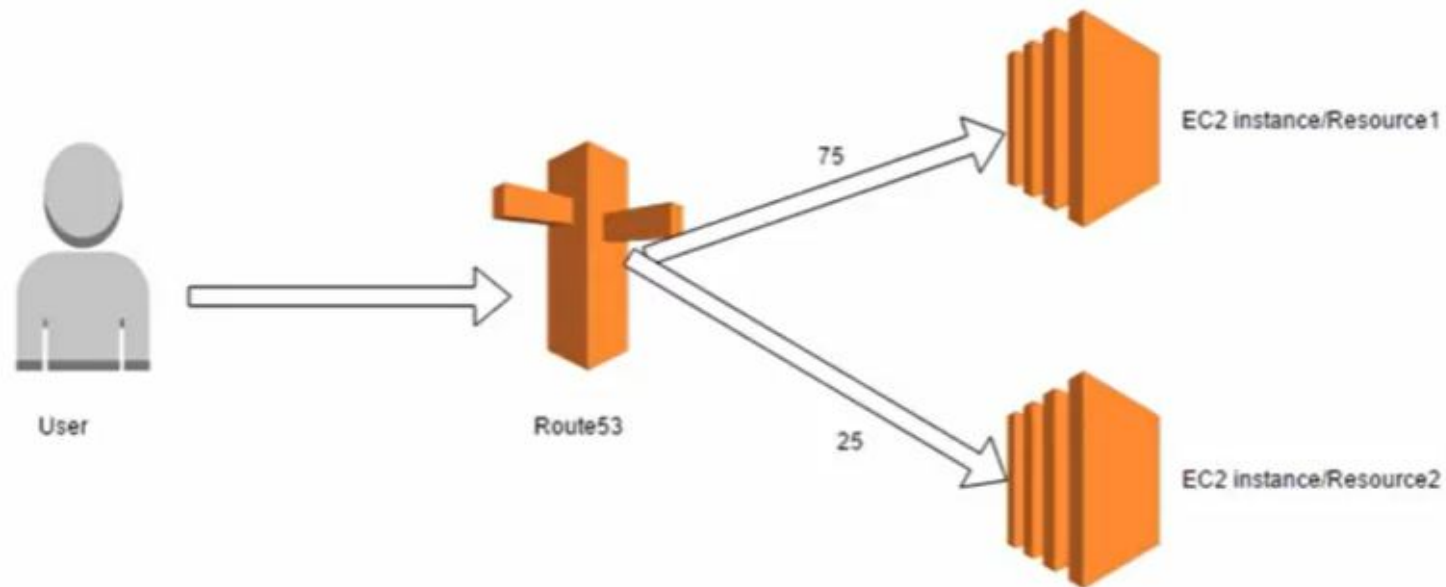
- **Simple routing policy** – Use to route internet traffic to a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
- **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.
- **Latency routing policy** – Use when you have resources in multiple locations and you want to route traffic to the resource that provides the best latency.
- **Failover routing policy** – Use when you want to configure active-passive failover.
- **Geolocation routing policy** – Use when you want to route internet traffic to your resources based on the location of your users.

Simple Routing Policy



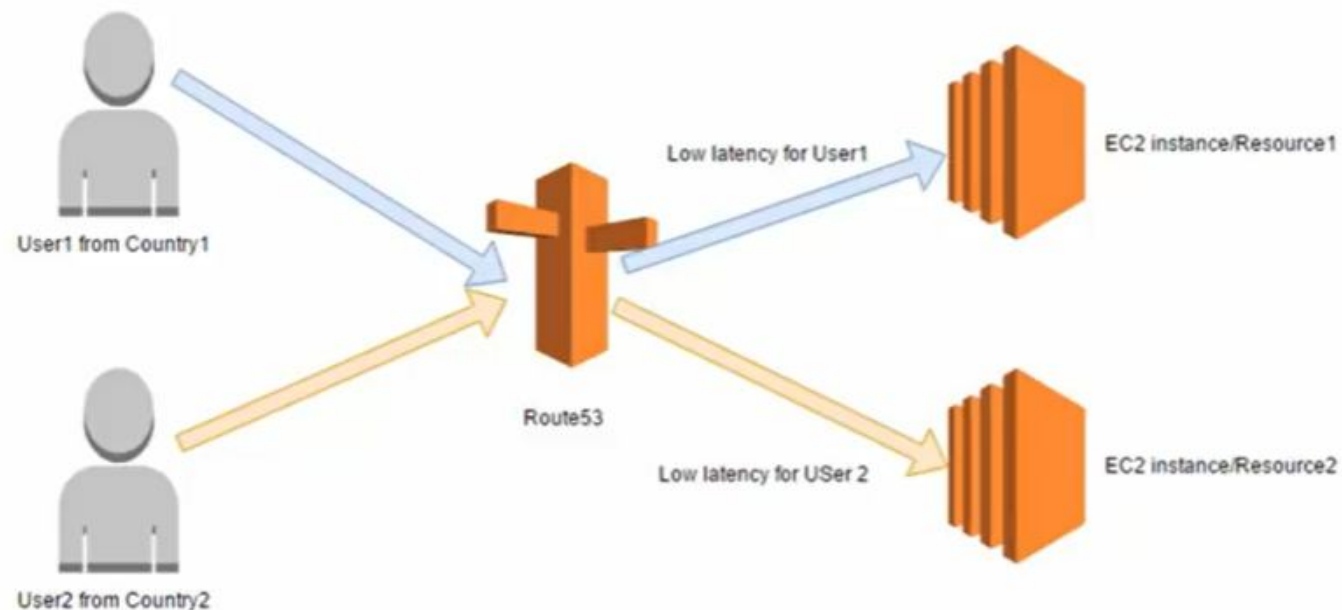
- Use to route internet traffic to a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.

Weighted Routing Policy



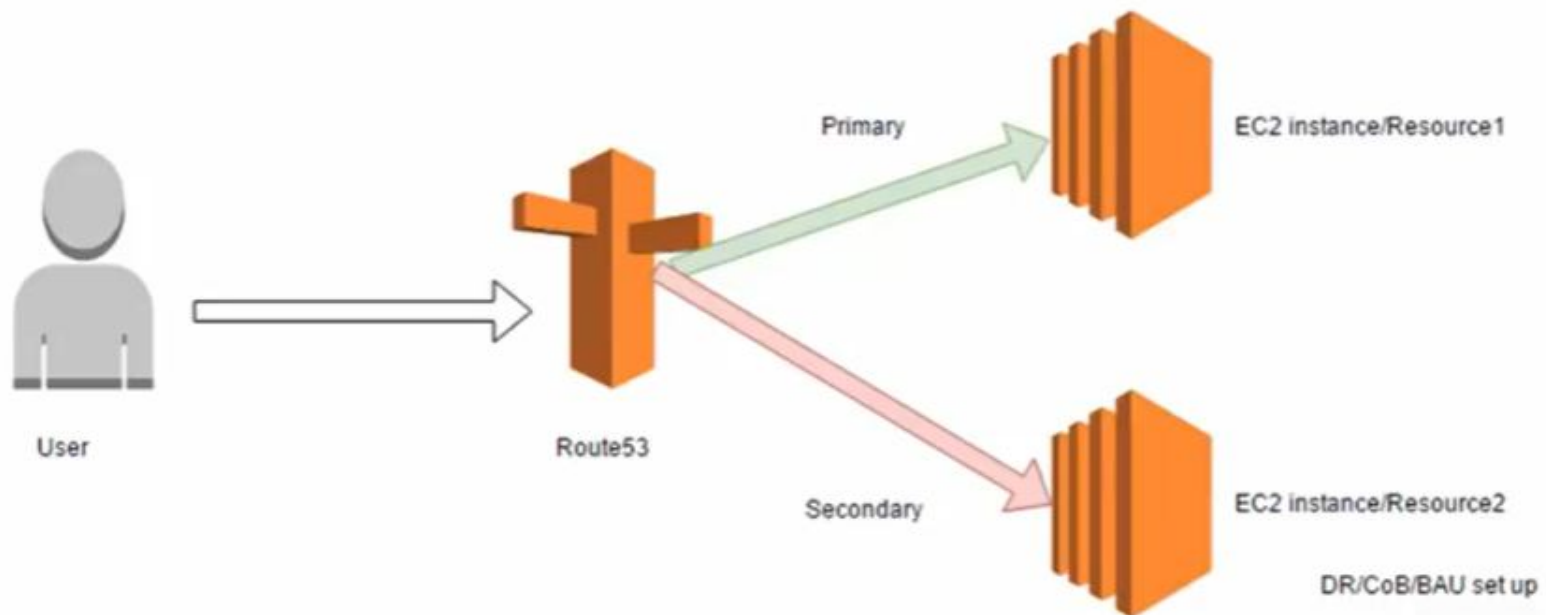
- Use to route traffic to multiple resources in proportions that you specify. E.g. 1 Quarter vs 3 Quarter to instance service

Latency Routing Policy



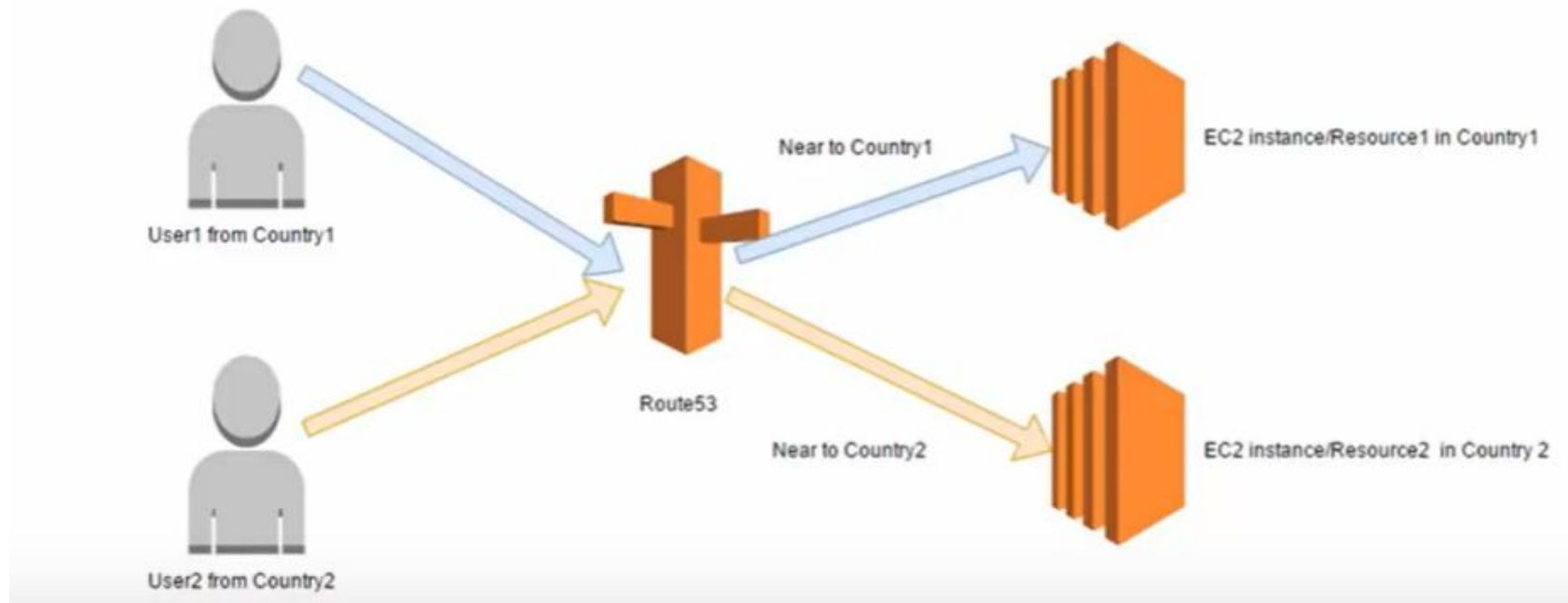
- Use when you have resources in multiple locations and you want to route traffic to the resource that provides the best latency.

Failover Routing Policy



- Use when you want to configure active-passive failover.

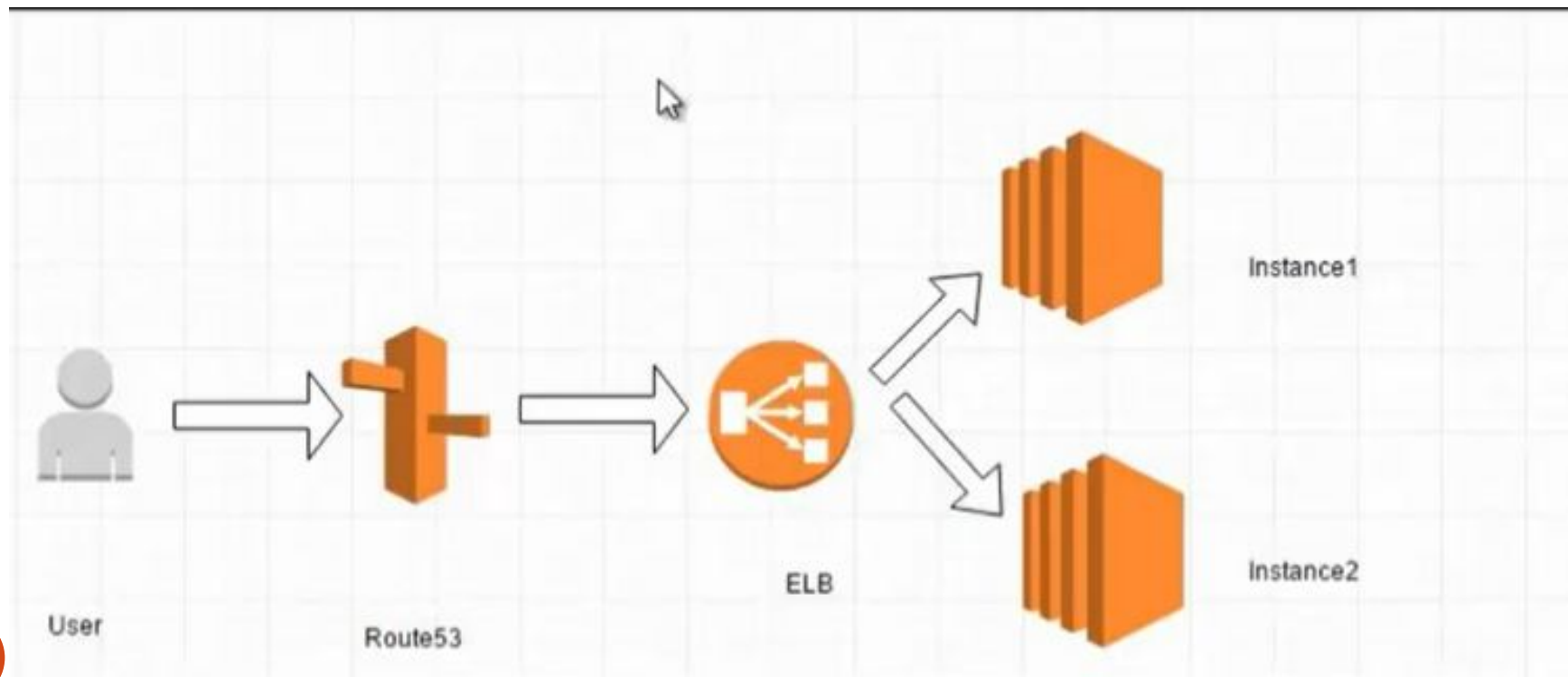
Geolocation Routing Policy



- Use when you want to route internet traffic to your resources based on the location of your users.

Steps of Route53

1. Register Domain Name
2. Configure Instances to host your Application /Service
3. DNS Service equipped with routing policies /protocols



Reference

- <https://docs.aws.amazon.com/route53/index.html>
- <https://www.youtube.com/watch?v=Kw0CA5Jlj2M>
- <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/Welcome.html>

For Blockchain refer additional
Unit-5 PDF

Thank You!