Big Data Analytics

**Lab Practical and date** – Practical 10, Monday 2nd November 2020

**Name and Roll Number**- Het Shah, 17BIT103

**Practical Objective**-Use following platforms for solving any big data analytic problem of your choice. (1) Amazon web services,(2) Micosoft Azure, (3)Google App engine

**Steps Involved-**

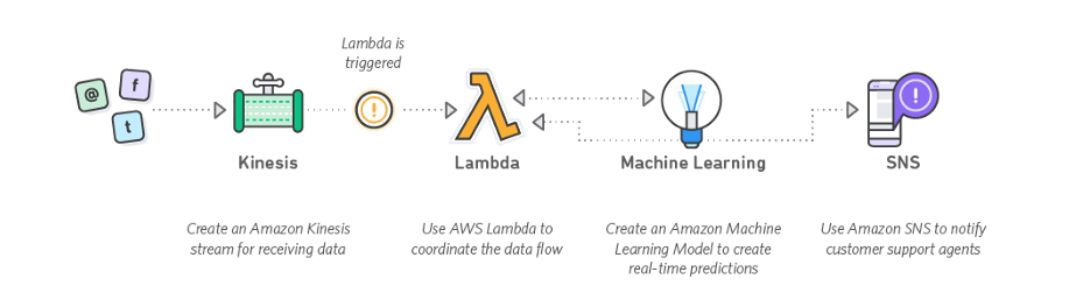
We are going to explore the Amazon Web Services platform for solving our big data analytics problem

**Background**

**AWS**

**Amazon Web Services** (**AWS**) is a subsidiary of [Amazon](https://en.wikipedia.org/wiki/Amazon.com) providing [on-demand](https://en.wikipedia.org/wiki/Software_as_a_service) [cloud computing](https://en.wikipedia.org/wiki/Cloud_computing) [platforms](https://en.wikipedia.org/wiki/Computing_platform) and [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) to individuals, companies, and governments, on a metered pay-as-you-go basis. These cloud computing [web services](https://en.wikipedia.org/wiki/Web_services) provide a variety of basic abstract technical infrastructure and [distributed computing](https://en.wikipedia.org/wiki/Distributed_computing) building blocks and tools. One of these services is [Amazon Elastic Compute Cloud](https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud) (EC2), which allows users to have at their disposal a [virtual](https://en.wikipedia.org/wiki/Virtualization) [cluster of computers](https://en.wikipedia.org/wiki/Computer_cluster), available all the time, through the Internet. AWS's version of virtual computers emulates most of the attributes of a real computer, including hardware [central processing units](https://en.wikipedia.org/wiki/Central_processing_unit) (CPUs) and [graphics processing units](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPUs) for processing; local/[RAM](https://en.wikipedia.org/wiki/Random-access_memory) memory; hard-disk/[SSD storage](https://en.wikipedia.org/wiki/Solid-state_drive); a choice of operating systems; networking; and pre-loaded application software such as [web servers](https://en.wikipedia.org/wiki/Web_server), [databases](https://en.wikipedia.org/wiki/Database), and [customer relationship management](https://en.wikipedia.org/wiki/Customer_relationship_management) (CRM).

Big Data for AWS



Big data can be described in terms of data management challenges that — due to increasing volume, velocity and variety of data — cannot be solved with traditional databases. While there are plenty of definitions for big data, most of them include the concept of what’s commonly known as “three V’s” of big data:

Volume: Ranges from terabytes to petabytes of data

Variety: Includes data from a wide range of sources and formats (e.g. web logs, social media interactions, ecommerce and online transactions, financial transactions, etc)

Velocity: Increasingly, businesses have stringent requirements from the time data is generated, to the time actionable insights are delivered to the users. Therefore, data needs to be collected, stored, processed, and analyzed within relatively short windows — ranging from daily to real-time

**Advantage of AWS**

* **Immediate Availability**

Most big data technologies require large clusters of servers resulting in long provisioning and setup cycles. With AWS you can deploy the infrastructure you need almost instantly. This means your teams can be more productive, it’s easier to try new things, and projects can roll out sooner.

# Broad & Deep Capabilities

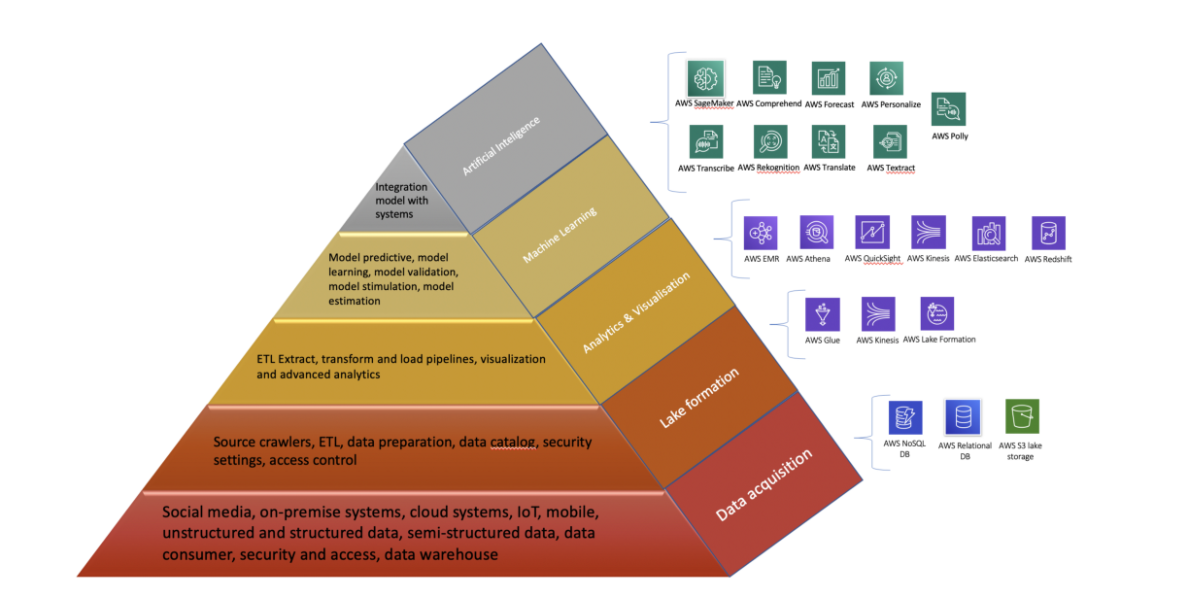
Big data workloads are as varied as the data assets they intend to analyze. A broad and deep platform means you can build virtually any big data application and support any workload regardless of volume, velocity, and variety of data. With 50+ services and hundreds of features added every year, AWS provides everything you need to collect, store, process, analyze, and visualize big data on the cloud.

# Trusted & Secure

Big data is sensitive data. Therefore, securing your data assets and protecting your infrastructure without losing agility is critical. AWS provides capabilities across facilities, network, software, and business processes to meet the strictest requirements. Environments are continuously audited for certifications such as ISO 27001, FedRAMP, [DoD SRG](https://aws.amazon.com/compliance/dod/), and PCI DSS. Assurance programs help you prove compliance with 20+ standards, including HIPAA, NCSC, and more. Visit the Cloud Security Center to learn more.

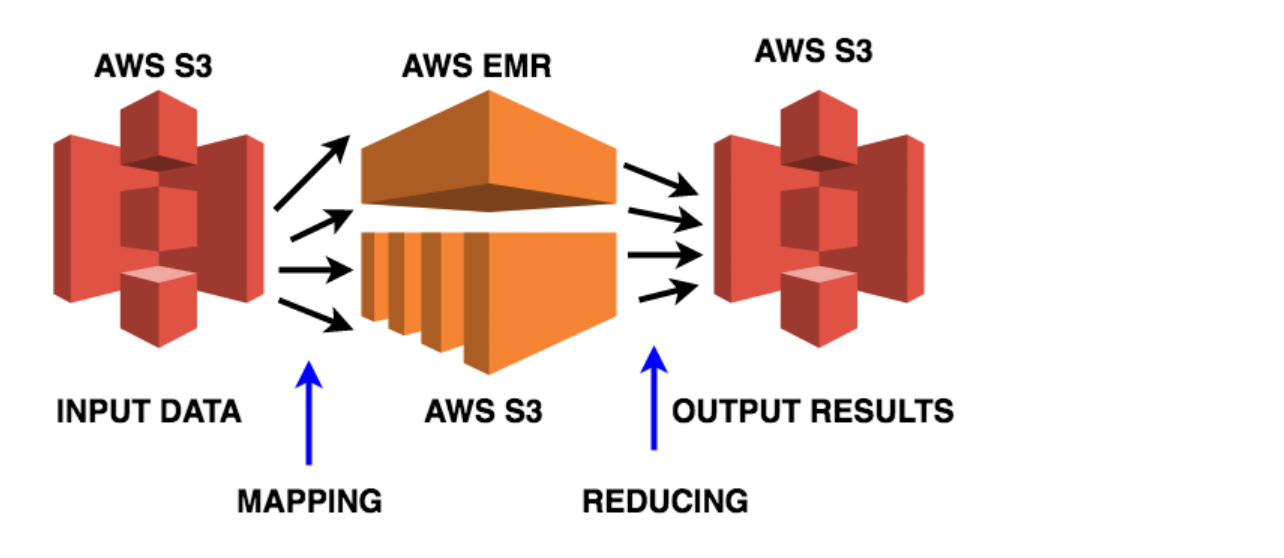
# Hundreds of Partners & Solutions

A large partner ecosystem can help you bridge the skills gap and get started with big data even faster. Visit the AWS Patner Networkto get help from a consulting partner or choose from many tools and applications across the entire data management stack.



AWS for solving all Big Data steps

MapReduce on AWS



We will implement mapreduce on input data

Input data is stored in amazon S3 bucket and we solve it using python program and store the output in the S3 bucket as well.

Steps

1. **Input bucket**

Aws s3 –region ap-south-1 mb s3://emr-demo-het-wordcount

1. **Add 10 input txt files by using this command**

Aws s3 sync wordcount s3://emr-demo-het-wordcount/

Upload: wordcount/input/0011 to s3://emr-demo-het-wordcount/0011

1. **Upload the python file as well**

Upload: wordcount/wordsplitter.py to s3://emr-demo-het-wordcount/wordsplitter.py

1. **Create AWS EMR Roles**

aws emr create-default-roles

aws iam list-roles| grep -i emr

"RoleName": "AWSServiceRoleForEMRCleanup",

"Arn": "arn:aws:iam::nnnn:role/aws-service-role/elasticmapreduce.amazonaws.com/AWSServiceRoleForEMRCleanup",

"RoleName": "EMR\_DefaultRole",

"Arn": "arn:aws:iam::nnnn:role/EMR\_DefaultRole",

"RoleName": "EMR\_EC2\_DefaultRole",

"Arn": "arn:aws:iam::nnnn:role/EMR\_EC2\_DefaultRole",

1. **Create EMR Cluster**

aws --region ap-south-1 emr create-cluster --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.large InstanceGroupType=CORE,InstanceCount=2,InstanceType=m4.large --name "Test Cluster" --log-uri s3://emr-demo-sree/logs/ --enable-debugging --tags Name=emr \

--ec2-attributes '{"KeyName":"my-demo-key","InstanceProfile":"EMR\_EC2\_DefaultRole","SubnetId":"subnet-nnnn","EmrManagedSlaveSecurityGroup":"sg-nnnn","EmrManagedMasterSecurityGroup":"sg-nnnn"}' \

--release-label emr-5.13.0 \

--service-role EMR\_DefaultRole

{

"ClusterId": "j-LT8GPHMUIWL4"

}

cid=j-LT8GPHMUIWL4

aws --region ap-south-1 emr list-clusters --active

{

"Clusters": [

{

"Id": "j-LT8GPHMUIWL4",

"Name": "Test Cluster",

"Status": {

"State": "WAITING",

"StateChangeReason": {

"Message": "Cluster ready after last step completed."

},

"Timeline": {

"CreationDateTime": 1526017522.75,

"ReadyDateTime": 1526017740.954

}

},

"NormalizedInstanceHours": 0

}

]

}

1. **Add Processing Steps**

aws --region ap-south-1 emr add-steps --cluster-id $cid \

--steps Type=STREAMING,Name='Word Count',ActionOnFailure=CONTINUE,Args=--files,s3://emr-demo-sree/wordSplitter.py,-mapper,wordSplitter.py,-reducer,aggregate,-input,s3://emr-demo-sree/input,-output,s3://emr-demo-sree/output

{

"StepIds": [

"s-2M4DI1IS7MLCM"

]

}

sid=s-2M4DI1IS7MLCM

aws --region ap-south-1 emr describe-step --cluster-id $cid --step-id $sid \

--query "Step.Status.State"

"RUNNING"

aws --region ap-south-1 emr describe-step --cluster-id $cid --step-id $sid \

--query "Step.Status.State"

"COMPLETED"

1. **Output**

head wordcount/output/part-00000

a 14716

aa 52

aakar 3

aargau 3

abad 3

abandoned 46

abandonment 6

abate 9

abauj 3

abbassid 4

**PythonSpiltter.py**

#!/usr/bin/python

import sys

import re

def main(argv):

pattern = re.compile("[a-zA-Z][a-zA-Z0-9]\*")

for line in sys.stdin:

for word in pattern.findall(line):

print "LongValueSum:" + word.lower() + "\t" + "1"

if \_\_name\_\_ == "\_\_main\_\_":

main(sys.argv)

**Conclusion**

In this practical we learned about AWS and how to perform wordcount on AWS instance fetching the data from Amazon S3 Bucket and using EMR to perform the analytics