**CLOUD COMPUTING**

**WHY CLOUD?**

Suppose if you want to host a website , these are the following things that you would need to do

>Buy stack of servers - paying lot of money

>High traffic

>Monitoring and maintaining servers

All the above defined things let to the certain disadvantages :

1. If you consider costs then this server setup is expensive
2. Troubleshooting problems can be more difficult and may conflict with your business.
3. Since the traffic is varying and your server will be idle most of the time.

Now-a-days we do everything online like we shop online , we buy food online and everything is online now-a-days.

**WHAT IS CLOUD?**

It makes sure that you are able to orchestrate your various functioning applications , managing your resources by combining all these data centres together , through a network and then providing you the control to use these resources and to manage them.

The cloud service providers will take care of all your issues like your security, your underlying infrastructure and so on.

**WHAT IS CLOUD COMPUTING?**

Cloud computing is :

Storing data / applications on remote servers.

Processing data / applications from servers

Accessing data / applications via the internet.

**SERVICE MODELS**

Cloud Computing provides different service models like

IAAS - Infrastructure - As - A - Service

PAAS - Platform - As - A - Service

SAAS - Software - As - A - Service

**SAAS - Software - As - A – Service**

Ø Cloud provider leases applications or softwares which are owned by them to its client. (eg : gmail)

**PAAS - Platform - As - A – Service**

Ø No control over the underlying architecture including OS , Storage , servers etc.

Example : Google app engine where you create your own application and you can put it on google app engine so that others can use it as well.

**IAAS - Infrastructure - As - A – Service**

Ø Provides Virtualization computing resources over the internet.

Ø No worries about underlying physical machine.

Ø Abstract the user from the physical machine.

**DEPLOYMENT MODELS**

Ø Public cloud

Ø Private cloud

Ø Hybrid cloud

**PUBLIC CLOUD**

- A service provider makes resources , such as applications and sstorage , available to the general public over the internet.

- Easy and inexpensive set – up because hardware , application and bandwidth , costs are covered by the provider.\

- No wasted resources because you pay for what you use.

**PUBLIC CLOUD**

- Offers hosted services to a limited number of people behind firewall , so it maintains the security concerns.

- Private cloud gives companies direct control over their data.

**HYBRID CLOUD**

- A cloud computing environment which uses a mix of on-premises , private cloud and third party , public cloud services.

- It helps to leverage the best of the both worlds.

**CLOUD PROVIDERS**

1) Amazon Web services

2) Google cloud platform

3) Digital ocean

4) IBM Cloud

5) Microsoft Azure

6) Terremark

**AMAZON WEB SERVICES**

**INTRODUCTION TO AWS**

Amazon Web Services (AWS) is a secure cloud services platform , offering compute power , database storage , content delivery and other functionality to help scale and grow.

**USES**

Takes care of all IT infrastructure and management.

Architecture consulting – the main concern is prototyping and rendering.

Rendering services

Media company

Large Enterprise – reaches globally (global architecture)

**ADVANTAGES**

Flexible

Cost effective

Scalability

Security

**AWS ARCHITECTURE**

Divided into two major types

- Regions

- Availability zones (Data centers)

These are nothing but different locations across the world , were they have various datacenters.

**DOMAINS IN AWS**

**- Compute Domain**

**1)EC2 (Elastic Compute Cloud)**

-It is a resizable compute capacity , a raw server to host a website in a clean slate.

**2)ELASTIC BEANSTALK**

-Where you deploy your various applications in AWS and one thing you need to know is you do not have to worry about the underlying architecture.

**3)MIGRATION**

- Where you can migrate your data from one location to another through the storage services.

**4)SECURITY AND COMPLIANCE**

- Based on IAM (IDENTIFICATION AND AUTHENTICATION MANAGEMENT TOOL)

**5)S3 STORAGE**

-These objects are to saved in foot files , which act as bucket basically , and then we have

>Cloudfront – content delivery network

>Glacier – place where you store archieves , because it is highly affordable.

**6)NETWORKING**

- Services like VPC (VIRTUAL PRIVATE CLOUD) , Direct connect to ROUTE 53 which is a DNS SERVER

**7)MESSAGING**

-AWS assures secured messaging some app will take care of these processes

Applications like – cloud trial , Opsworks

**8)DATABASES**

-Storage and databases are similar where store all your data and even the executable files.

DynamoDB Which is non relational DBMS – Dealing with unstructured data sources.

**9)MANAGEMENT TOOLS**

- Monitoring tools to set the alarms.

**INSTANCES**

Any instances is a virtual servers for running applications is Amazon’s EC2. It can be understood like a tiny part of a large computer, any tiny part which has its own Harddrive , network connection , os etc , But it is actually all virtual.

**WHAT IS EC2 ?**

Amazon Elastic compute cloud is a web server from Amazon that provides resizable compute services in the cloud.

**FEATURES :**

Scalable

Cost Efficient

Flexible

**TYPES OF INSTANCES :**

**1)GENRAL PURPOSE INSTANCE**

For applications that require a balance of performance and cost , where you need a prompt response , cost effectiveness , less processing.

Eg: Email response systems

**2)COMPUTE INSTANCE**

For applications that require a lot of processing from the CPU.

Eg: Analysing streaming Data

**3)MEMORY INSTANCE**

For applications that are heavy in nature , therefore require a lot of RAM. Eg: Applications that need multi tasking

**4)STORAGE INSTANCE**

For applications that are huge in size or have a data set that occupies a lot of space.

Eg: Larged sized applications

**5)GPU INSTANCE**

For applications that require some heavy graphics rendering. Eg: 3D modelling.

**ELASTIC BEANSTALK**

**WHAT IS ELASTIC BEANSTALK ?**

AWS Elastic Beanstalk is an PAAS Service used for deploying and scaling web applications and services developed with JAVA, PHP , .NET ,NODE.JS etc . on familiar servers such as a Apache , Nginx ,Tomcat and IIS.

**PLATFORM – AS - A – SERVICE**

- Quicker performance

- Simplest Operations

- Cost effectiveness

- Multi-tenant Architecture

- Better user Experience

**WEB APPICATION HOSTING PLATFORMS**

Various application hosting platforms providing AWS:

Open shift – web hosting platform offered by RedHat

Google app engine

Scalingo – Platform – as – a – service

Python anywhere – It is an online integrated development platform and web hosting services, but based on python language

Elastic Beanstalk - AWS

Azure – Microsoft

**FEATURES**

1) Elastic Beanstalk is the fastest and simplest way to deploy your applications on AWS.

2) Enables you to focus on writing code rather than spending time, managing and configuring servers etc.

3) Automatically scales up your applications and scales down based on applications specific needs.

4) PAAS, it provides you options , where you can change the pre-assigned configurations.

**FUNDAMENTALS OF ELASTIC BEANSTALK**

**1)COMPONENTS**

An application is a collection of components including environments, versions and environment configuration , here environment is nothing but where you run your applications.

An application in Elastic Beanstalk is conceptually similar to a folder.

**2) APPLICATION VERSION**

An application version refers to a specific , labelled iteration of deployable code of a web application.

An application version points to an Amazon S3 object that contains the deployment code such as JAVA war file.

**3) ENVIRONMENT**

Environment within Elastic Beanstalk , Applications house which is a different running version of your applications

Each environment has only a single application version at a time.

An application undergoes all the stages given below:

Development Stage

Production Stage

Testing Stage

**4)ENVIRONMENT TIER**

There are two types of tier

>Web server tier

>Worker tier

Webserver tier servers HTTP requests

Work tier will do all the background processing and background task.

**5)ENVIRONMENT HEALTH**

Elastic Beanstalk reports the health of a web server Environment depending on how the application running in it responds to the health check.

Uses on the four colours to describe the status:

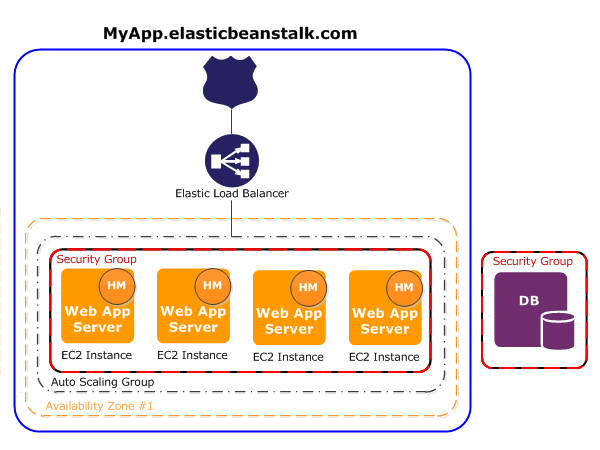
>Gray – Environment is being updated

>Green – Passed recent health checks

>Yellow – Failed one or more checks

>Red – Failed three or more checks

**ARCHITECTURE OF ELASTIC BEANSTALK**

**1)Beanstalk Environment**

**2)Elastic Load Balancer –** When you launch an environment or URL is created , and this URL in the form of c name made to point the elastic load balancer.

C name is nothing but the alternate name for your URL

So when your applications receives request all the request are forwarded to elastic load balancer and this load balancer distributes these request among EC2 instances of auto scaling group.

**3)AUTO SCALING GROUP –** where automatically deploy an instances for your applications whenever needed and also scales down when it is not in use.

**4)EC2 –** When you launch an elastic Beanstalk environment , Beanstalk will assign your application with suitable EC2 instances , but the software stack like the OS , the servers and different softwares which are supposed to be installed on your instance are decided by a device called container type.

**5)HOST MANAGER –** Responsible for various task like :

- Provide detailed reports of performance of your applications

- Provides level events

- Monitors application log files

- Monitors Application server

**6)SECURITY GROUP –** like firewall , allows port 80 by default.

**WORKER ENVIRONMENT TIER**

A worker Environment Tier is process that handles background task during resource intensive or time intensive operations.

It also do some other background task like

-Email Notifications

-Generates Reports

-Clean-up Databases