day 10

Web application

```
Emp getEmp() {
//make http call
  RestClient restclient = new RestClient ();
  JSONObject obj = restclient.get("https://localhost/api/users/"+empno")
} catch(){
//cached data
show message("try some")
just wait for some time
}
 obj.data.
}
```

Map -> Service1 - "https://localhost/api/users/"

Service2 - "https://localhost/api/invoices"

Integration /

->Flight -> Integration

->one-many

Jet,10 airline

30 sec -> 8 airline (data)

1 airline(timeout),

1 not reachable

> 70% or more

Google API

Cloud computing

Why Cloud?

Machine ->Server->Cluster

->LB

Client---------->Lb(IP)---->S1,S2,S3

Software/Hardware

Host

1. Purchasing all these machines -> Services

->1. Service - pay/Usage

->2. Cost

->3. Maintain -> Installation, Configuration, H/w,S/w

1. 100 requests -> 1 server

2. 10000 -> 5 server

Means

Instead of using our own data centers

---------->Use Services

Characteristics

On demand self service

Broad N/w access

Resource Pooling

Rapid elasticity

Measured services

Multi Tenacity

Amazon/Microsoft/Google

--->Huge HArdware-->Virtualization

--->different OS -> How much RAM

SComp --> 200 machines

Laptop----------> Regions

Laptop----------> Machine -> MySQL

IAAS,PAAS,SAAS

IAAS : Amazon Ec2 Instance

http://ec2-100-26-167-159.compute-1.amazonaws.com:8080/

Output: Hello World!

PAAS : Platform

------> OS,Installations,Configurations

AWS RDS,MySql Fiddle,Sololearn

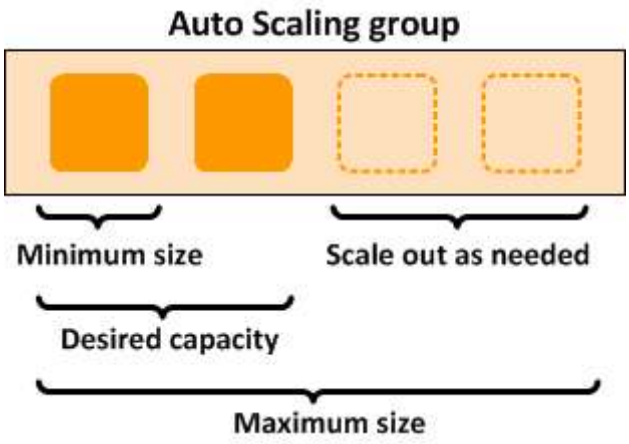
SAAS: IAAS+PAAS+Software-->just configure and use it--->

Google Drive -->Start using it.

SAAS: Software

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, the following Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.



Scaling options

Amazon EC2 Auto Scaling provides several ways for you to scale your Auto Scaling group.

Maintain current instance levels at all times

You can configure your Auto Scaling group to maintain a specified number of running instances at all times. To maintain the current instance levels, Amazon EC2 Auto Scaling performs a periodic health check on running instances within an Auto Scaling group. When Amazon EC2 Auto Scaling finds an unhealthy instance, it terminates that instance and launches a new one. For more information, see Maintaining a fixed number of instances in your Auto Scaling group.

Scale manually

Manual scaling is the most basic way to scale your resources, where you specify only the change in the maximum, minimum, or desired capacity of your Auto Scaling group. Amazon EC2 Auto Scaling manages the process of creating or terminating instances to maintain the updated capacity. For more information, see Manual scaling for Amazon EC2 Auto Scaling.

Scale based on a schedule

Scaling by schedule means that scaling actions are performed automatically as a function of time and date. This is useful when you know exactly when to increase or decrease the number of instances in your group, simply because the need arises on a predictable schedule. For more information, see Scheduled scaling for Amazon EC2 Auto Scaling.

Scale based on demand

A more advanced way to scale your resources, using scaling policies, lets you define parameters that control the scaling process. For example, let's say that you have a web application that currently runs on two instances and you want the CPU utilization of the Auto Scaling group to stay at around 50 percent when the load on the application changes. This method is useful for scaling in response to changing conditions, when you don't know when those conditions will change. You can set up Amazon EC2 Auto Scaling to respond for you. For more information, see Dynamic scaling for Amazon EC2 Auto Scaling.

Use predictive scaling

You can also use Amazon EC2 Auto Scaling in combination with AWS Auto Scaling to scale resources across multiple services. AWS Auto Scaling can help you maintain optimal availability and performance by combining predictive scaling and dynamic scaling (proactive and reactive approaches, respectively) to scale your Amazon EC2 capacity faster.

Load balancer

A **load balancer** accepts incoming traffic from clients and routes requests to its registered targets (such as **EC2** instances) in one or more Availability Zones. The **load balancer** also monitors the health of its registered targets and ensures that it routes traffic only to healthy targets.