Cloud Computing UNIT-5

Cloud computing (Jawaharlal Nehru Technological University, Kakinada)

UNIT V

Cloud Application Development: Amazon Web Services : EC2 – instances, connecting clients, security rules, launching, usage of S3 in Java, Cloud based simulation of a Distributed trust algorithm, Cloud service for adaptive data streaming , Google: Google App Engine, Google Web Toolkit, Microsoft: Azure Services Platform, Windows live, Exchange Online, Share Point Services, Microsoft Dynamics CRM.

Cloud Application Development:

Cloud application development is the process through which a Cloud-based app is built. It involves different stages of software development in cloud. Three cloud delivery models, SaaS, PaaS, and IaaS; the level of difficulty increases as we move toward the base of the cloud service pyramid.
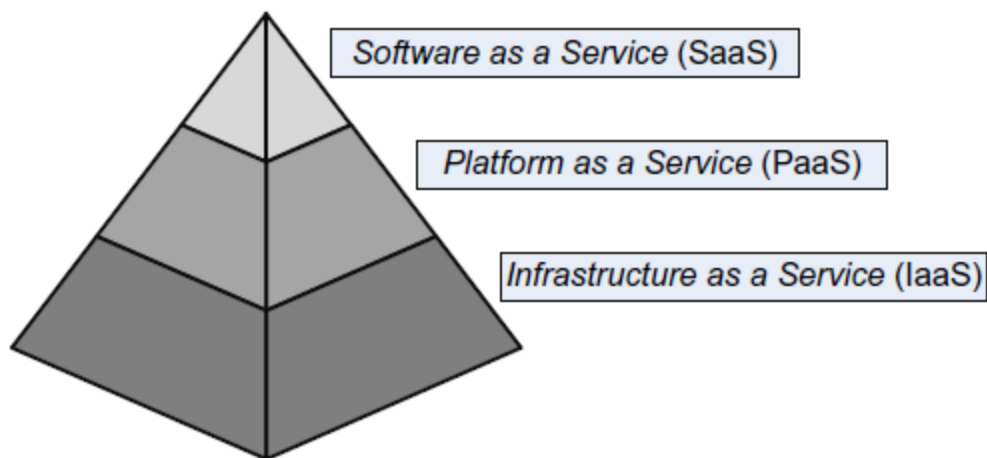
Amazon Web Services: EC2 instances

An EC2 instance is nothing but a virtual server in Amazon Web services terminology. It stands for Elastic Compute Cloud. It is a web service where an AWS subscriber can request and provision a compute server in AWS cloud.

- Amazon Machine Images (**AMIs**) are the basic building blocks of Amazon EC2

- An AMI is a template that contains a software configuration (operating system, application server and applications) that can run on Amazon's computing environment

- AMIs can be used to launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud.

An on-demand EC2 instance is an offering from AWS where the subscriber/user can rent the virtual server per hour and use it to deploy his/her own applications.

Amazon Management Console (AMC) window listing the Amazon Web Services offered at the time of this writing. The services are grouped into several categories: computing and networking, storage and content delivery, deployment and management, databases, and application services. AWS EC2 instance is a virtual server started in a region and the availability zone is selected by the user.

Amazon Web Services accessible from the *Amazon Management Console*.

Getting Started with Amazon EC2:

- Step 1: Sign up for Amazon EC2

- Step 2: Create a key pair

- Step 3: Launch an Amazon EC2 instance

- Step 4: Connect to the instance

- Step 5: Customize the instance

- Step 6: Terminate instance and delete the volume created

Instances are grouped into a few classes, and each class has available to it a specific amount of resources, such as: CPU cycles, main memory, secondary storage, and communication and I/O bandwidth. Several operating systems are supported by AWS, including Amazon Linux, Red Hat Enterprise Linux, SUSE Linux Enterprise Server 11, Ubuntu Server 12.04.1, and several versions of Microsoft Windows.

The local instance store persists only for the duration of an instance; the data will persist if an instance is started using the Amazon Elastic Block Storage (EBS) and then the instance can be restarted at a later time.

Connecting clients to cloud instances through firewalls

A firewall is a software system based on a set of rules for filtering network traffic. Its function is to protect a computer in a local area network from unauthorized access.

Firewalls:

- First generation - operated below the transport layer, and discarded packets based on the information in the headers of physical, data link, and network layer protocols.

- Second generation - operate at the transport layer and maintain the state of all connections passing through them and opened the possibility of denial of service attacks.

- Third generation - understand widely-used application layer protocols such as FTP, HTTP, TELNET, SSH, and DNS. These firewalls examine the header of application layer protocols and support intrusion detection systems (IDS).

Firewalls screen incoming traffic and sometimes filter outgoing traffic as well. A first filter encountered by the incoming traffic in a typical network is a firewall provided by the operating system of the router; the second filter is a firewall provided by the operating system running on the local computer
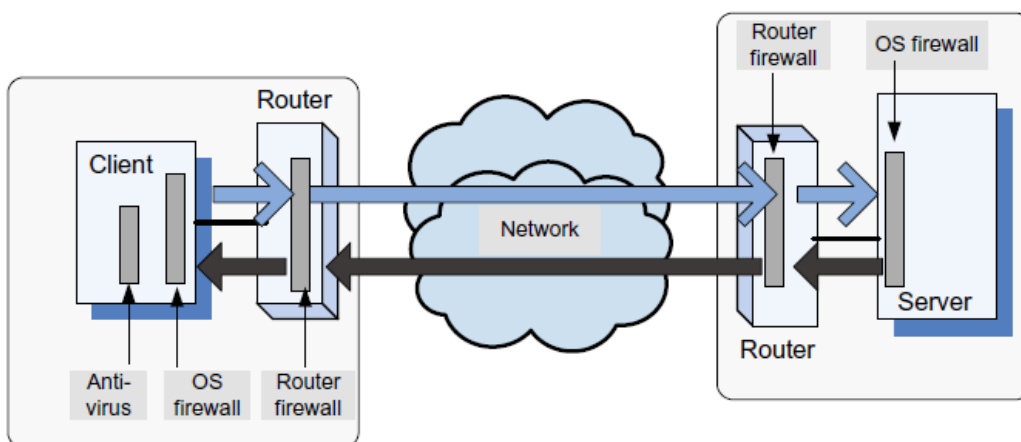
Typically, the local area network (LAN) of an organization is connected to the Internet via a router. A router firewall often hides the true address of hosts in the local network using the Network Address Translation (NAT) mechanism.

A rule specifies a filtering option at (i) the network layer, when filtering is based on the destination/ source IP address; (ii) the transport layer, when filtering is based on destination/source port number; or (iii) the MAC layer, when filtering is based on the destination/source MAC address.

To get the status of the firewall, specify the L (List) action of the *iptables* command:

```
sudo iptables -L
```

As a result of this command the status of the *INPUT, FORWARD,* and *OUTPUT* chains will be displayed.
To change the default behavior for the entire chain, specify the action P (Policy), the chain name, and the target name; e.g., to allow all outgoing traffic to pass unfiltered, use

```
sudo iptables -P OUTPUT ACCEPT s
```

To add a new security rule, specify: the action, A (add), the chain, the transport protocol, TCP or UDP, and the target ports, as in:

```
sudo iptables -A INPUT -p -tcp -dport ssh -j ACCEPT
sudo iptables -A OUTPUT -p -udp -dport 4321 -j ACCEPT
sudo iptables -A FORWARD -p -tcp -dport 80 -j DROP
```

Security rules for application and transport layer protocols in *EC2*

A client must know the IP address of a virtual machine in the cloud to be able to connect to it. The Domain Name Service (DNS) is used to map human-friendly names of computer systems to IP addresses in the Internet or in private networks.

For several reasons, including security and the ability of the infrastructure to scale up, the IP addresses of instances visible to the outside world are mapped internally to private IP addresses. A virtual machine running under Amazon's EC2 has several IP addresses:

1. EC2 Private IP Address. The internal address of an instance; it is only used for routing within the EC2 cloud.

2. EC2 Public IP Address. Network traffic originating outside the AWS network must use either the public IP address or the elastic IP address of the instance. The public IP address is translated using Network Address Translation (NAT) to the private IP address when an instance is launched and it is valid until the instance is terminated. Traffic to the public address is forwarded to the private IP address of the instance.

3. EC2 Elastic IP Address. The IP address allocated to an AWS account and used by traffic originated outside AWS. NAT is used to map an elastic IP address to the private IP address.

Elastic IP addresses allow the cloud user to mask instance or availability zone failures by programmatically remapping public IP addresses to any instance associated with the user's account. This allows fast recovery after a system failure.

The following steps allow the user to add a security rule:

1. Sign in to the AWS Management Console at http://aws.amazon.com using your email address and password and select EC2 service.

2. Use the EC2 Request Instance Wizard to specify the instance type, whether it should be monitored, and specify a key/value pair for the instance to help organize and search.

3. Provide a name for the key pair. Then on the left-side panel, choose Security Groups under Network & Security, select the desired security group, and click on the inbound tab to enter the desired rule.

AWS security. Choose Security Groups under Network & Security, select the desired security group, and click on the Inbound tab to enter the desired rule:
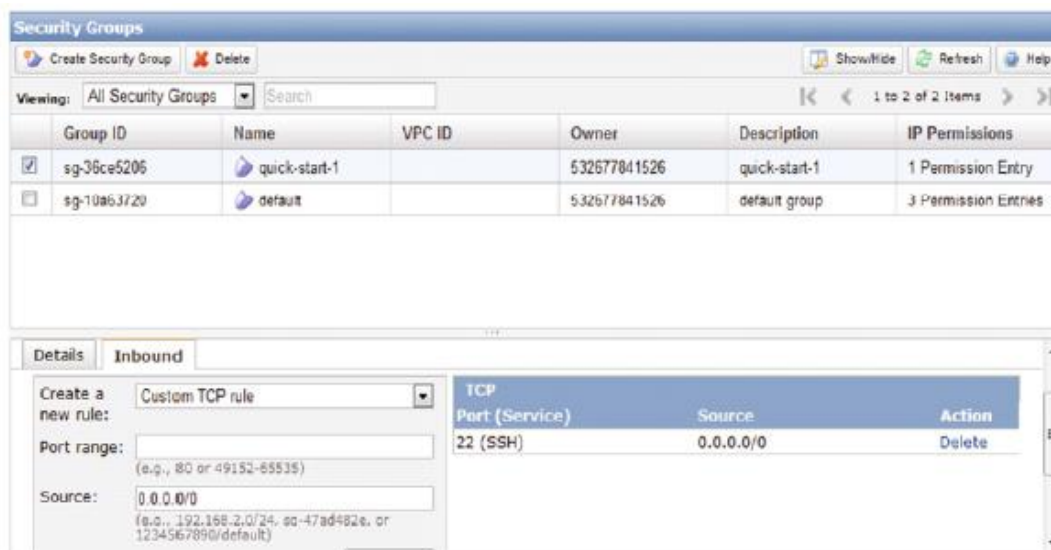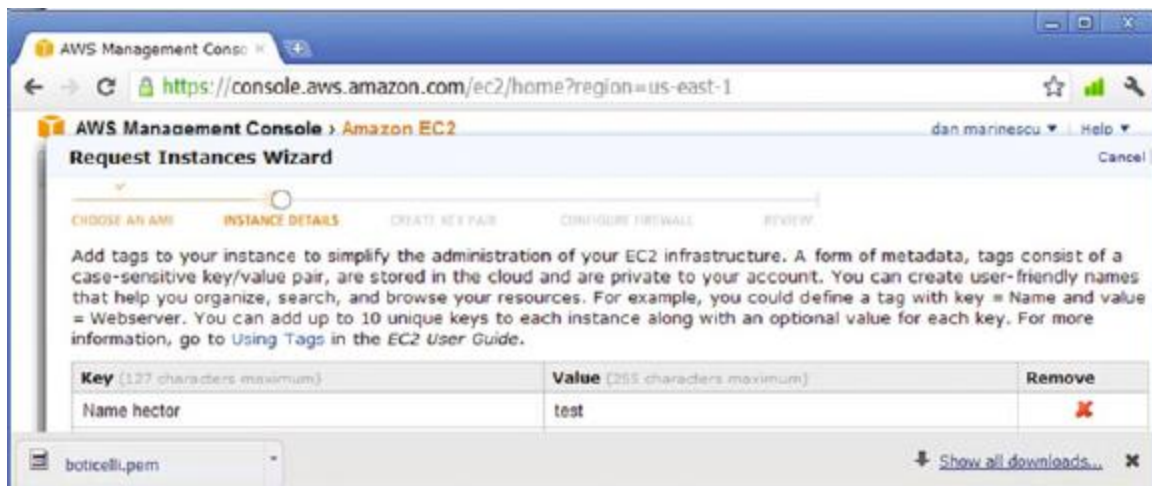

EC2 Request Instance Wizard is used to (a) specify the number and type of instances and the zone:
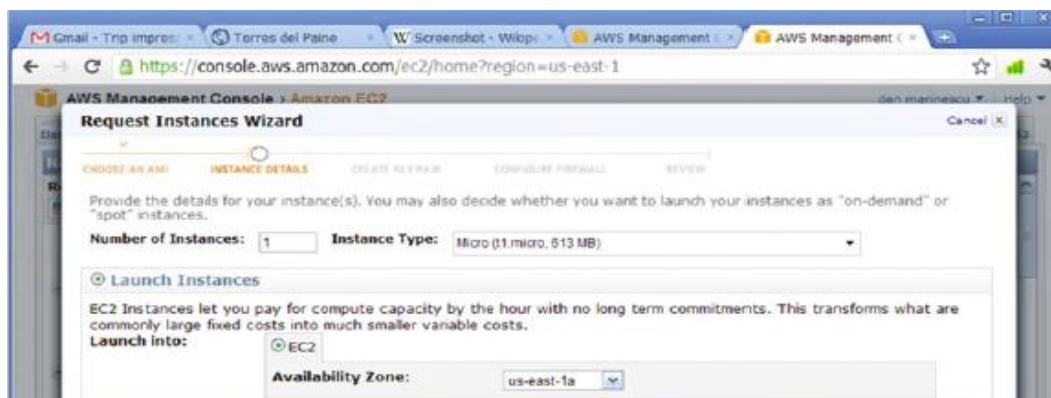
(b) specify the kernelId and the RAM diskId and enable the CloudWatch service to monitor the EC2 instance

(c) Add tags to the instance. A tag is stored in the cloud and consists of a case-sensitive key/value pair private to the account.

On Linux or Unix systems the port numbers below 1,024 can only be assigned by the root. The plain ASCII file called services maps friendly textual names for Internet services to their assigned port numbers and protocol types, as in the following example:

netstat 15/tcp
ftp 21/udp
ssh 22/tcp
telnet 23/tcp
http 80/tcp

How to launch an EC2 Linux instance and connect to it

To launch an *EC2 Linux* instance from a *Linux* platform step-by-step process:

A. Launch an instance

1. From the AWS Management Console, select EC2 and, once signed in, go to Launch Instance Tab.

2. To determine the processor architecture when you want to match the instance with the hardware, enter the command

   - uname -m

   and choose an appropriate Amazon Linux AMI by pressing Select.

3. Choose Instance Details to control the number, size, and other settings for instances.

4. To learn how the system works, press Continue to select the default settings.

5. Define the instance's security, as discussed in Section 11.3: In the Create Key Pair page enter a name for the pair and then press Create and Download Key Pair.

6. The key-pair file downloaded in the previous step is a .pem file, and it must be hidden to prevent unauthorized access. If the file is in the directory awcdir/dada.pem enter the commands

   - cd awcdir

   - chmod 400 dada.pem

7. Configure the firewall. Go to the page Configure firewall, select the option Create a New Security Group, and provide a Group Name. Normally we use ssh to communicate with the instance; the default port for communication is port 8080, and we can change the port and other rules by creating a new rule.

8. Press Continue and examine the review page, which gives a summary of the instance.

9. Press Launch and examine the confirmation page, then press Close to end the examination of the confirmation page.

10. Press the Instances tab on the navigation panel to view the instance.

11. Look for your Public DNS name. Because by default some details of the instance are hidden, click on the Show/Hide tab on the top of the console and select Public DNS.

12. Record the Public DNS as Public DNS name; it is needed to connect to the instance from the Linux terminal.

13. Use the ElasticIP panel to assign an Elastic IP address if a permanent IP address is required.

B. Connect to the instance using ssh and the TCP transport protocol.

1. Add a rule to the iptables to allow ssh traffic using the TCP protocol. Without this step, either an access denied or permission denied error message appears when you're trying to connect to the instance.

❖ sudo iptables -A iptables -p -tcp -dport ssh -j ACCEPT

2. Enter the Linux command:

❖ ssh -i abc.pem ec2-user@PublicDNSname

If you get the prompt You want to continue connecting? respond Yes. A warning that the DNS name was added to the list of known hosts will appear.

3. An icon of the Amazon Linux AMI will be displayed.

C. Gain root access to the instance

By default the user does not have root access to the instance; thus, the user cannot install any software. Once connected to the EC2 instance, use the following command to gain root privileges:

sudo -i

Then use yum install commands to install software, e.g., gcc to compile C programs on the cloud.

D. Run the service ServiceName

If the instance runs under Linux or Unix, the service is terminated when the ssh connection is closed. To avoid the early termination, use the command

nohup ServiceName

To run the service in the background and redirect stdout and stderr to files p.out and p.err, respectively,

execute the command

nohup ServiceName > p.out 2 > p.err &

How to use *S3* in Java

The Java API for Amazon Web Services is provided by the *AWS* SDK

Create an S3 client. S3 access is handled by the class AmazonS3Client instantiated with the account credentials of the AWS user:

```
AmazonS3Client s3 = new AmazonS3Client(
new BasicAWSCredentials("your_access_key", "your_secret_key"));
```

Buckets. An *S3 bucket* is analogous to a file folder or directory, and it is used to store *S3 objects*. Bucket names must be *globally unique*; hence, it is advisable to check first to see whether the name exists:

```
s3.doesBucketExist("bucket_name");
```

This function returns "true" if the name exists and "false" otherwise. Buckets can be created and deleted either directly from the AWS Management Console or programmatically as follows:

```
s3.createBucket("bucket_name");
s3.deleteBucket("bucket_name");
```

S3 objects. An S3 object stores the actual data and it is indexed by a key string. A single key points to only one S3 object in one bucket. Key names do not have to be globally unique, but if an existing key is assigned to a new object, the original object indexed by that key is lost.

```
File f = new File("local_file_name");
s3.putObject("bucket_name", "key", f);
```

To access this object with key key from the bucket bucket_name use:

```
S3Object myFile = s3.getObject("bucket_name", "key");
```

To read this file, you must use the S3Object's InputStream:

```
InputStream in = myFile.getObjectContent();
```

The InputStream can be accessed using Scanner, BufferedReader, or any other supported method. Amazon recommends closing the stream as early as possible, since the content is not buffered and it is streamed directly from the S3. An open InputStream means an open connection to S3. For example, the following code will read an entire object and print the contents to the screen:

```
AmazonS3Client s3 = new AmazonS3Client(
    new BasicAWSCredentials("access_key", "secret_key"));
    InputStream input = s3.getObject("bucket_name", "key")
      .getObjectContent();
    Scanner in = new Scanner(input);
    while (in.hasNextLine())

        {
          System.out.println(in.nextLine());
        }
 in.close();
 input.close();
```

Batch upload/download. Batch upload requires repeated calls of s3.putObject() while iterating over local files.

To view the keys of all objects in a specific bucket, use

```
ObjectListing listing = s3.listObjects("bucket_name");
```

ObjectListing supports several useful methods, including getObjectSummaries(). S3ObjectSummary encapsulates most of an S3 object properties (excluding the actual data), including the key to access the object directly,

```
List<S3ObjectSummary> summaries = listing.getObjectSummaries();
```

For example, the following code will create a list of all keys used in a particular bucket and all of the keys will be available in string form in *List < String >allKeys*:

```
    AmazonS3Client s3 = new AmazonS3Client(
      new BasicAWSCredentials("access_key", "secret_key"));
      List<String> allKeys = new ArrayList<String>();
      ObjectListing listing = s3.listObjects("bucket_name");
      for (S3ObjectSummary summary:listing.getObjectSummaries())
       {
         allKeys.add(summary.getKey());
       }
```

Note that if the bucket contains a very large number of objects, then *s3.listObjects()* will return a truncated list. To test if the list is truncated, we could use *listing.isTruncated()*; to get the next batch of objects, use

```
    s3.listNextBatchOfObjects(listing)};
```

Cloud-based simulation of a distributed trust algorithm

Mobile wireless applications are likely to benefit from cloud computing.

- The convenience of data access from any site connected to the Internet.
- The data transfer rates of wireless networks are increasing; the time to transfer data to and from cloud is no longer a limiting factor.
- Mobile devices have limited resources; whereas new generations of smartphones and tablet computers are likely to use multicore processors and have a fair amount of memory, power consumptionis, and will continue to be, a major concern in the near future.

Cognitive Radio Networks (CRNs):

- Cognitive Radio Networks (CRNs) are emerging as a solution to increase the spectrum utilization by using unused or less used spectrum in radio environments.

- CRNs are often based on a cooperative spectrum-sensing strategy. In this mode of operation, each node determines the occupancy of the spectrum based on its own measurements, combined with information from its neighbors, and then shares its own spectrum occupancy assessment with its neighbors

- Information sharing is necessary because a node alone cannot determine the true spectrum occupancy. Indeed, a secondary node has a limited transmission and reception range; node mobility combined with typical wireless channel impairments, such as multipath fading, shadowing, and noise, add to the difficulty of gathering accurate information by a single node.

- The secondary nodes of an ad hoc CRN compete for free channels, and the information one node may provide to its neighbors could be deliberately distorted. Malicious nodes will send false information to the fusion center in a centralized CRN.
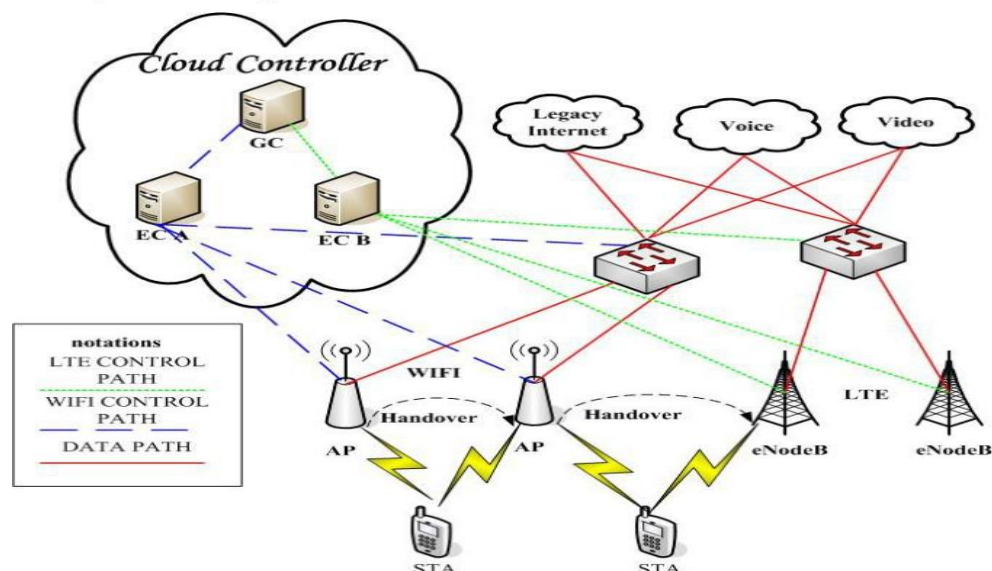
Trust:

The actual meaning of trust is domain and context specific. Consider, for example, networking; at the MAC layer the multiple-access protocols assume that all senders follow the channel access policy, e.g., in Carrier Sense Multiple Access with Collision Detection (CSMA-CD) a sender senses the channel and then attempts to transmit if no one else does. In a store-and-forward network, trust assumes that all routers follow a best-effort policy to forward packets toward their destination.

A Distributed Algorithm for Trust Management in Cognitive Radio:

The algorithm computes the trust of node $1 \leq i \leq n$ in each node in its vicinity, $j \in V_i$, and requires several preliminary steps. The basic steps executed by a node i at time t are:

1. Determine node i's version of the occupancy report for each one of the K channels:

$$S_i(t) = \{s_{i,1}(t), s_{i,2}(t), \ldots, s_{i,K}(t)\}$$

In this step node i measures the power received on each of the K channels.

2. Determine the set $V_i(t)$ of the nodes in the vicinity of node i . Node i broadcasts a message and individual nodes in its vicinity respond with their NodeId.

3. Determine the distance to each node $j \in V_i(t)$ using the algorithm.

4. Infer the power as measured by each node $j \in V_i(t)$ on each channel $k \in K$.

5. Use the location and power information determined in the previous two steps to infer the status of each channel:

$$s_{i,k,j}^{infer}(t), \quad 1 \leqslant k \leqslant K, \quad j \in V_i(t).$$

6. Receive the information provided by neighbor

$$j \in V_i(t), S_{i,k,j}^{recv}(t)$$

7. Compare the information provided by neighbor

$$j \in V_i(t)$$

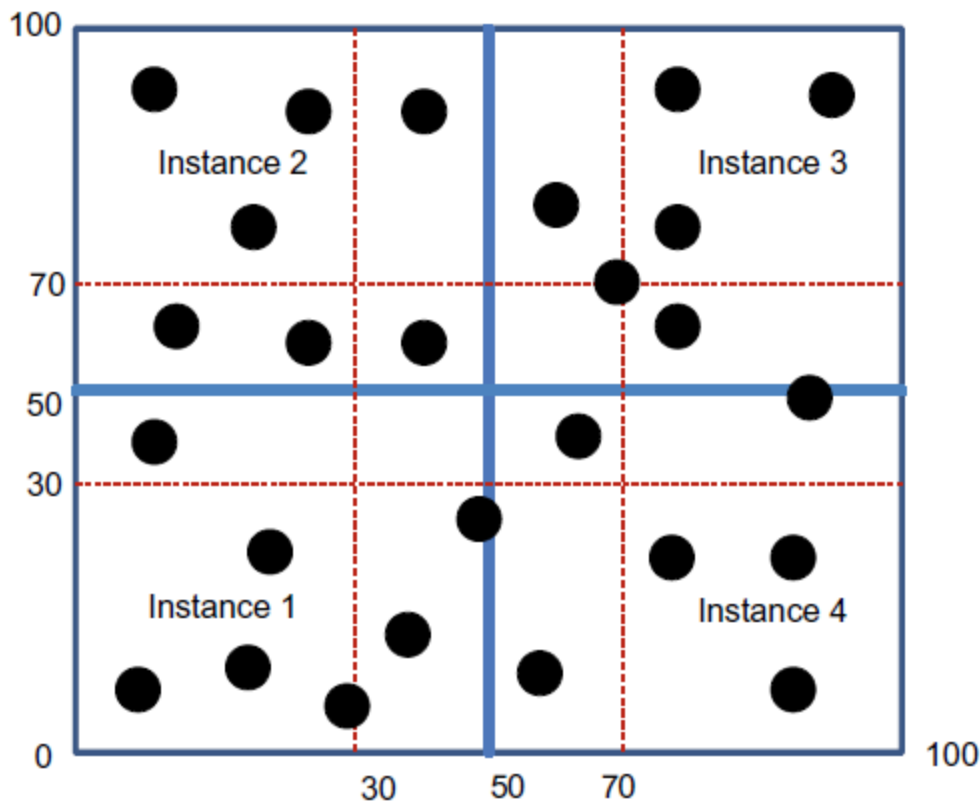$$S_{i,k,j}^{recv}(t) = \left\{ s_{i,1,j}^{recv}(t),\ s_{i,2,j}^{recv}(t), \ldots,\ s_{i,K,j}^{recv}(t) \right\}$$

8. Compute the number of matches, mismatches, and cases when no inference is possible, respectively,

$$\alpha_{i,j}(t) = \mathcal{M}\left[S_{i,k,j}^{infer}(t), \quad S_{i,k,j}^{recv}(t)\right]$$

$$\beta_{i,j}(t) = \mathcal{N}\left[S_{i,k,j}^{infer}(t), \quad S_{i,k,j}^{recv}(t)\right]$$

9. Use the quantities $\alpha_{i,j}(t)$, $\beta_{i,j}(t)$, and $X_{i,j}(t)$ to assess the trust in node j. For example, compute the trust of node i in node j at time t as

$$\zeta_{i,j}(t) = \left[1 + X_{i,j}(t)\right]\frac{\alpha_{i,j}(t)}{\alpha_{i,j}(t) + \beta_{i,j}(t)}.$$

Data partitioning for the simulation of a trust algorithm. The area covered is of size 100×100 units. The nodes in the four subareas of size 70×70 units are processed by an instance of the cloud application. The subareas allocated to an instance overlap to allow an instance to have all the information about a node in its coverage area.

A cloud service for adaptive data streaming

Data streaming is the name given to the transfer of data at a high rate with real-time constraints. Multimedia applications such as music and video streaming, high-definition television (HDTV), scientific applications that process a continuous stream of data collected by sensors, the continuous backup copying to a storage medium of the data flow within a computer, and many other applications require the transfer of real-time data at a high rate.

Data streaming involves three entities:
1. The sender.
2. A communication network.
3. A receiver.

The resources necessary to guarantee the timing constraints include CPU cycles and buffer space at the sender and the receiver, as well as network bandwidth. Adaptive data streaming determines the data rate based on the available resources. Lower data rates imply lower quality, but they reduce the demands for system resources.

Accommodating dynamic data flows with timing constraints is nontrivial; only about 18% of the top 100 global video Web sites use adaptive bit rate (ABR) technologies for streaming.

Design decision is how the two services should interact to optimize performance. Two alternatives come to mind:

1. The audio service running on the EC2 platform requests the data file from the S3, converts it, and, eventually, sends it back. The solution involves multiple delays and it is far from optimal.
2. Mount the S3 bucket as an EC2 drive. This solution reduces considerably the start-up time for audio streaming.

The results of our measurements when the instance is the t1.micro server exhibit a wide range of conversion times, 13–80 seconds, for the large audio file of about 6.7 MB when we convert from 320 to 192 Kbps.

Conversion time in seconds on a *EC2 t1.micro* server platform. The source file is of high audio quality, 192 Kbps. The individual conversions are labeled $C1$ to $C10$; $\overline{T}_c$ is the mean conversion time.											
Bit Rate (Kbps)	Audio File Size (MB)	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	$\overline{T}_c$
128	4.467982	14	15	13	13	73	75	56	59	72	14	40.4
64	2.234304	9	9	9	32	44	9	23	9	45	10	19.9
32	1.117152	6	6	6	6	6	6	20	6	6	6	7.4
16	0.558720	6	6	6	6	6	6	20	6	6	6	5.1
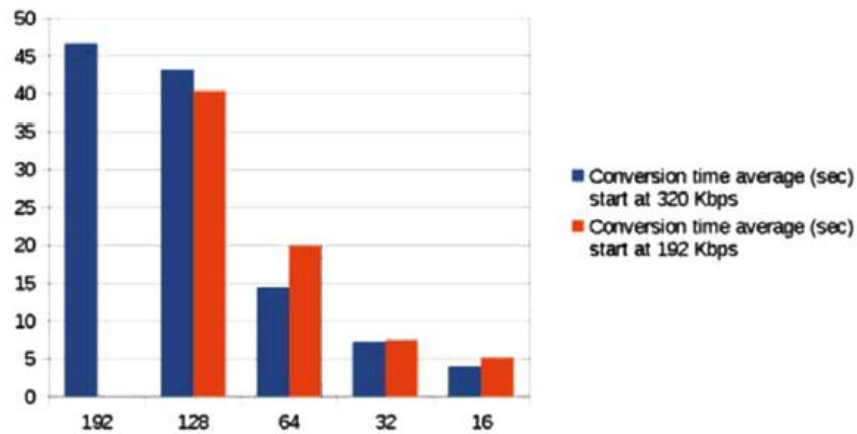
The average conversion time on a *EC2 t1.micro* platform. The bars at left and right correspond to the original file at the highest resolution (320 Kbps data rate) and next highest resolution (192 Kbps data rate), respectively.

GOOGLE APPLICATION ENGINE (GAE)

❖ Google App Engine is a PaaS cloud that provides a complete Web service environment(Platform)

❖ GAE provides Web application development platform for users.

❖ All required hardware, operating systems and software are provided to clients.

❖ Clients can develop their own applications, while App Engine runs the applications on Google's servers.

❖ GAE helps to easily develop an Web Application

❖ App Engine only supports the Java and Python programming languages.

❖ The Google App Engine (GAE) provides a powerful distributed data storage service.

Google App Engine Environments:

Google Cloud provides two environments:

1) Standard Environment with constrained environments and support for languages such as Python, Go, node.js

**Features of Standard Environment:**

- Persistent storage with queries, sorting, and transactions.
- Auto-scaling and load balancing.
- Asynchronous task queues for performing work.
- Scheduled tasks for triggering events at regular or specific time intervals.

- Integration with other GCP services and APIs.

2) Flexible Environment where developers have more flexibility such as running custom runtimes using Docker, longer request & response timeout, ability to install custom dependencies/software and SSH into the virtual machine.
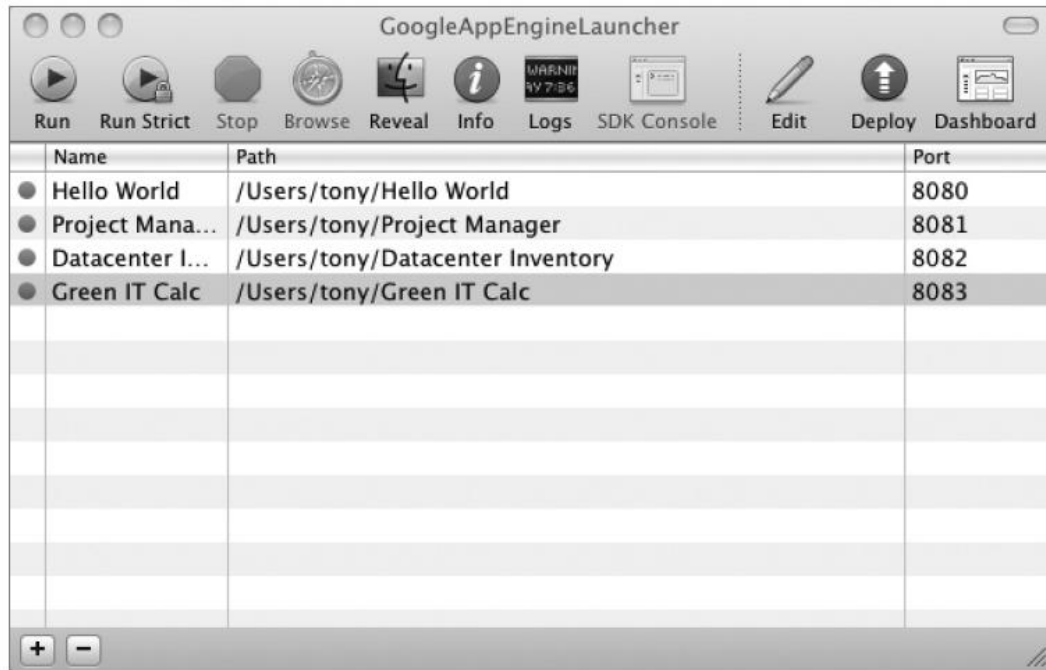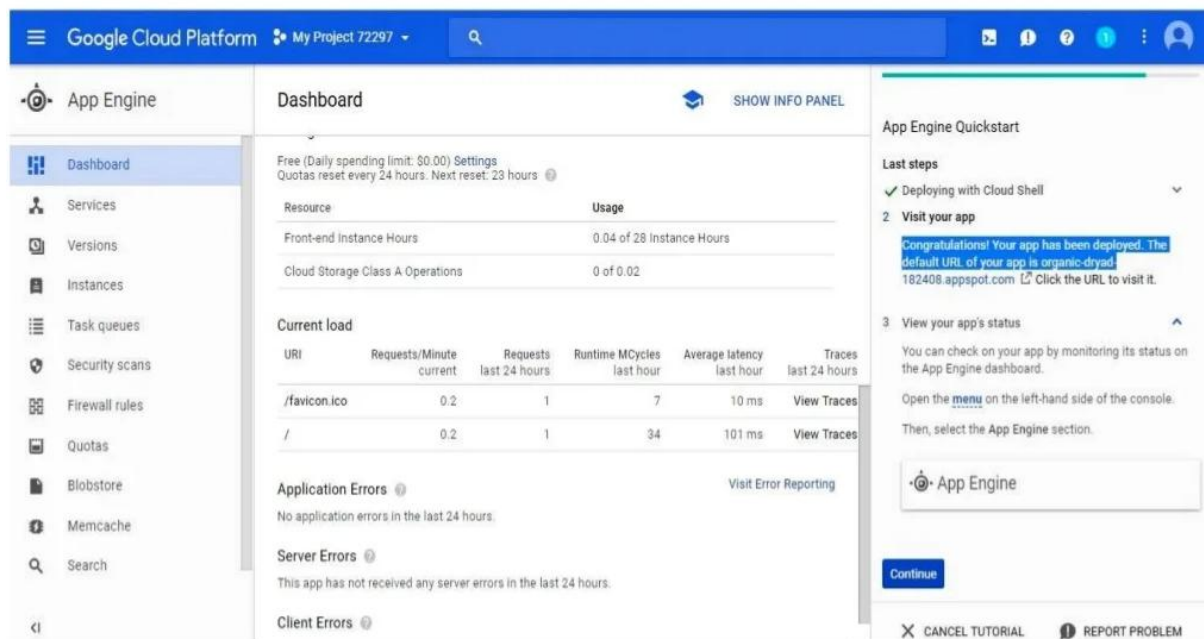

GAE Components:
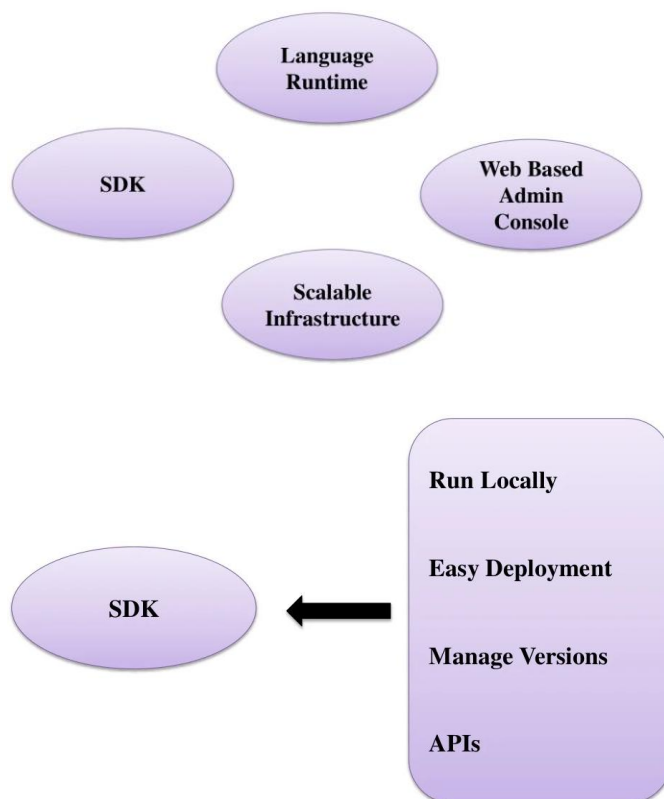
Language Run Time:
- JVM
  - ➤ Scala, Groovy, Jruby, Clojure
  - ➤ Servlets & JSPs
    - ❖ GWT
    - ❖ JSF
    - ❖ Struts
    - ❖ Wicket
- Python
  - ➤ Built in framework
  - ➤ Django
- **PHP**

Web Based Admin Console

GAE ARCHITECTURE:

App Engine is created under Google Cloud Platform project when an application resource is created. The Application part of GAE is a top-level container that includes the service, version and instance-resources that make up the app.
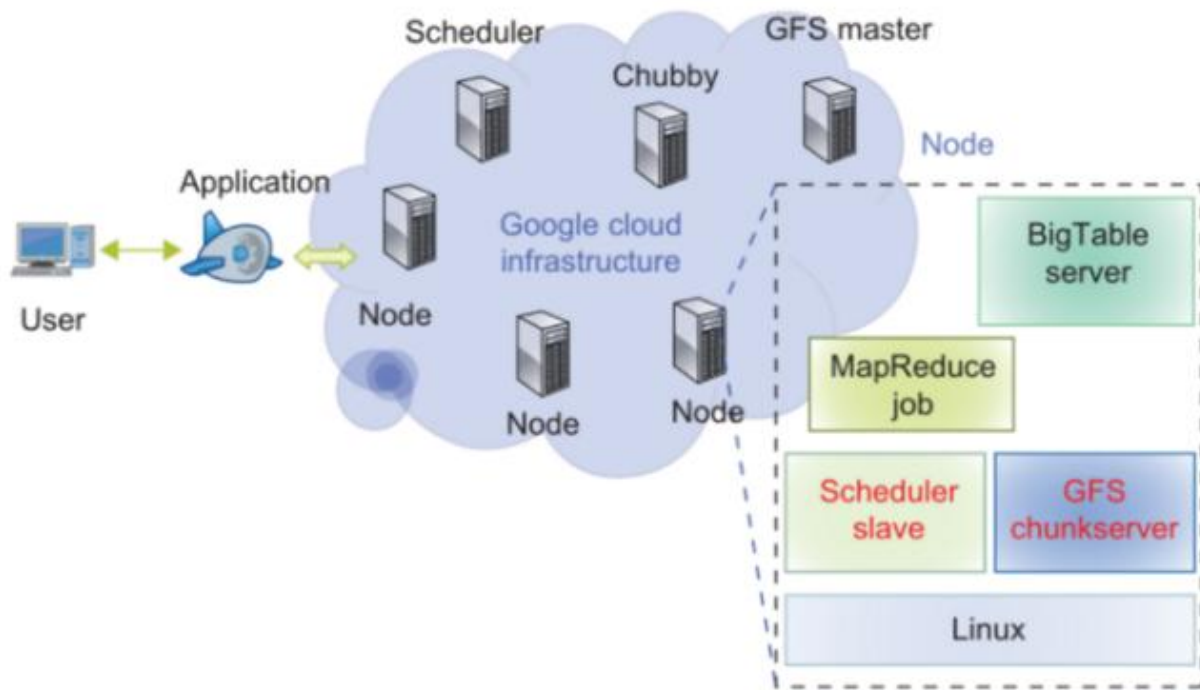
Google's cloud computing application includes the Google File System, the MapReduce programming framework, and Big Table. With these building blocks, Google has built many cloud applications.

- ➢ Google File System(GFS) ->for storing large amounts of data.

- ➢ MapReduce->for application program development.

- ➢ Chubby-> for distributed application lock services.

- ➢ BigTable-> offers a storage service.

- ➢ Third-party application providers can use GAE to build cloud applications for providing services.

- ➢ Inside each data center, there are thousands of servers forming different clusters.

- ➢ GAE runs the user program on Google's infrastructure.

- ➢ Application developers now do not need to worry about the maintenance of servers.

- ➢ GAE can be thought of as the combination of several software components.

- ➢ GAE supports Python and Java programming environments.

- ➢ The frontend is an application framework which is similar to other web application frameworks such as ASP, J2EE, and JSP. At the time of this writing, GAE supports Python and Java programming environments.

GAE APPLICATIONS

Well-known GAE applications

- ➢ Google Search Engine
- ➢ Google Docs
- ➢ Google Earth
- ➢ Gmail

    These applications can support large numbers of users simultaneously.

- ➢ Users can interact with Google applications via the web interface provided by each application.
- ➢ Applications run in the Google data centers.
- ➢ Inside each data center, there might be thousands of server nodes to form different clusters.
- ➢ Each cluster can run multipurpose servers.

Google Web Toolkit (GWT)

Google Web Toolkit (GWT) is a development toolkit for building and optimizing complex browser-based applications. Google Web Toolkit (GWT) is a development toolkit to create RICH Internet Applications (RIA).

- o Developer need not to be expert at browser incompatibilities as GWT handles through different binding.
- o GWT is a single code base for client and server.
- o GWT has chosen Java with which development becomes more effective as it provide code navigation, refactoring etc.
- o GWT MVP (Model View Presenter) allows shared working, faster testing with JUnit. The event bus allows many updates in client side application by placing event on the event bus.
- o It allows different Java tools to integrate with it such as hibernate through gilead.

Components of GWT:

- o **GWT Java-to-JavaScript Compiler:** It translates all the Java written code into JavaScript.
- o **GWT Development Mode:** It allows the developers to run application in development mode i.e. app run in Java without compiling JavaScript. Development mode allows native mode plug-in called Google Web Toolkit Developer Plug-in.

- o **JRE emulation library:** GWT includes a library that follows the Java runtime library i.e. java.lang, java.lang.annotation, java.math, java.io, java.sql, java.util etc.
- o **GWT Web UI class library:** For creating a widget GWT consists of set of interface and classes.
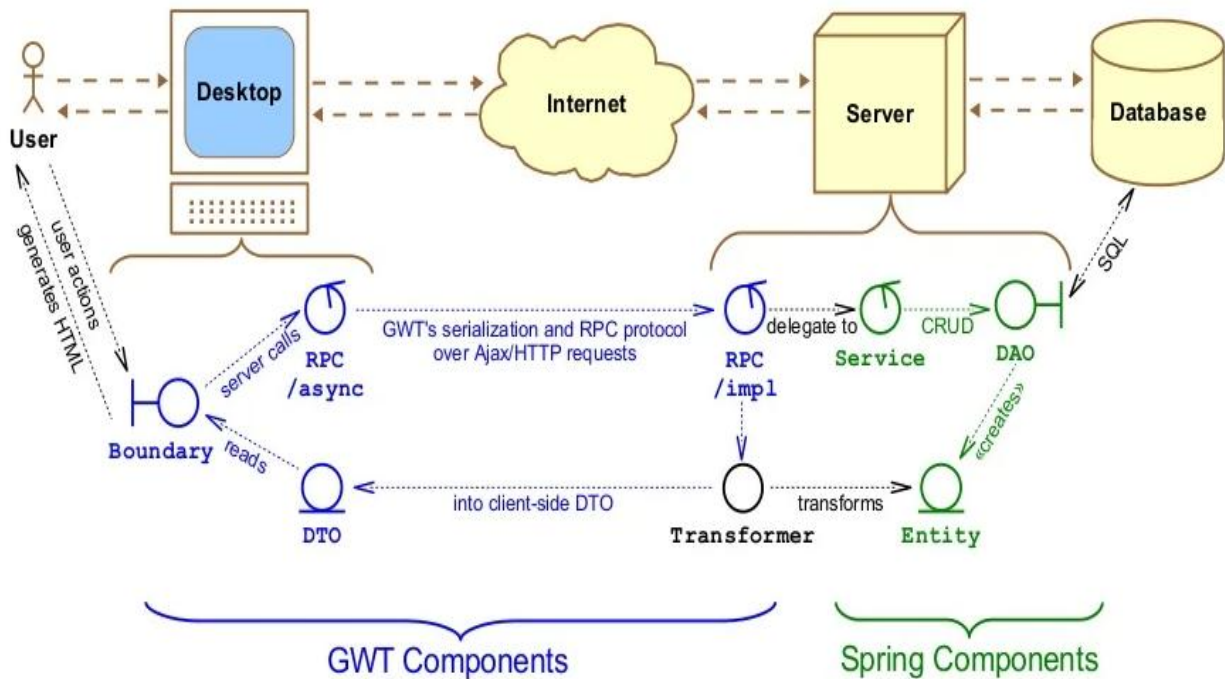
GWT provides two modes:

1. **Development Mode:** It provides the facility to debug Java code of the application directly with your Java debugger.
2. **Web Mode:** This mode enables the application to be translated into HTML, CSS and JavaScript code which can be directly deployed to a web server.
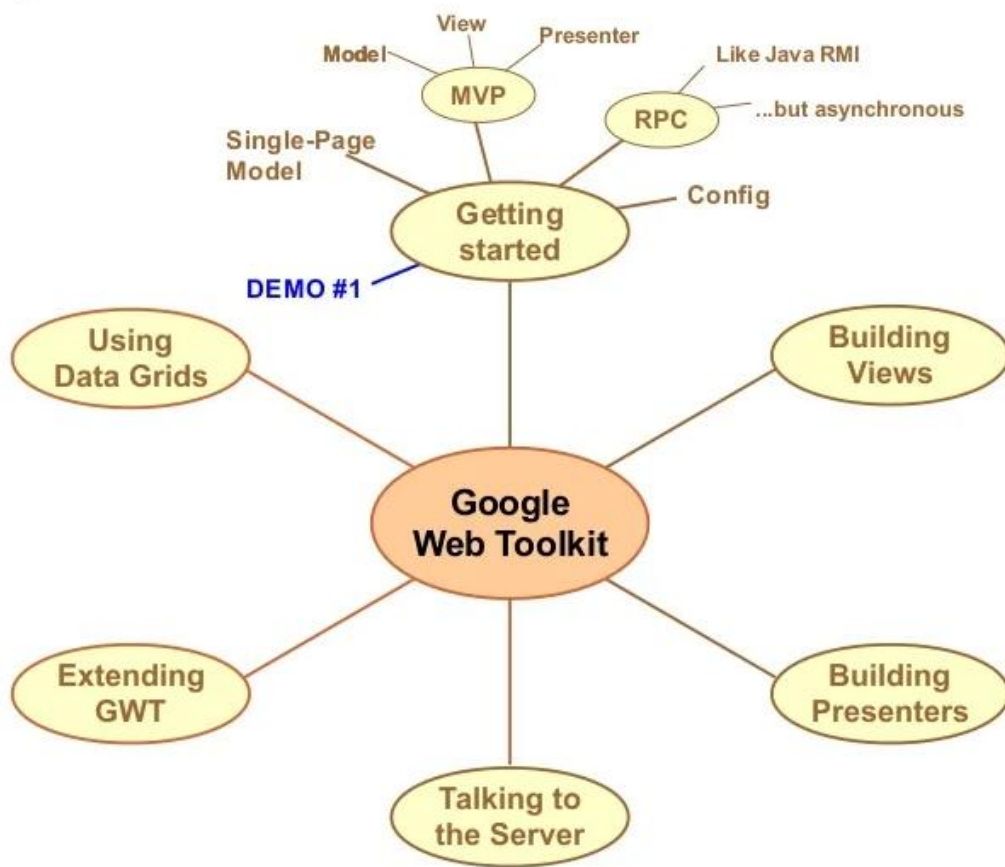
GWT Application:

To build a GWT application we have four sections, in this last section is optional.

1) Module Descriptors: This section helps in configuration of GWT application. To configure we write the configuration file into XML

Syntax: name.gwt.xml

Here, "**name**" is the name of application. All configuration files are in project root directory.

2) UI design: It consists of HTML, CSS or Images which used to design GWT application. We can configured its location by using **<public path = "location address" />** tag. We can find the configuration files in module configuration.

3) Client Side Code: In this section conversion of all the codes and business logic of application is converted into JavaScript using GWT compiler. We can find the location of resource using **<source path = "path" />** tag. This code consists of Entry Point code which can be written without the parameter. When GWT application module is loaded then **EntryPoint.onModuleLoad()** method is called each time.

4) Server Side Code: In this section we can execute Server side code. If our application does not have any backend (server side script or database) then this section is optional.

Microsoft: Azure Services Platform

Windows Azure service platform is a cloud **Platform as a Service (PaaS)** by Microsoft. It enables the development and hosting of applications on Microsoft's managed data center. Microsoft Azure is a cloud computing platform that provides a wide variety of services that we can use without purchasing and arranging our hardware.

Azure Services like compute, storage, network, and application services allow us to put our effort into building great solutions without worrying about the assembly of physical infrastructure.
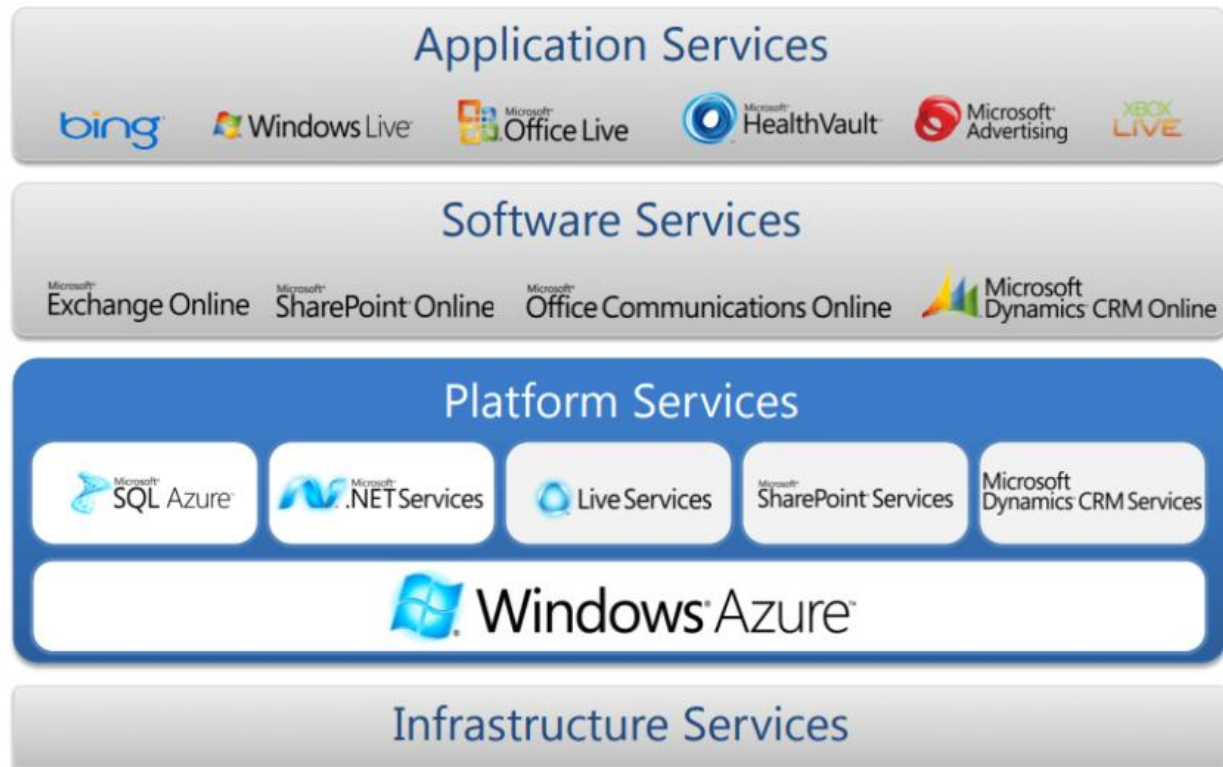
The components in the Azure Service platform includes **Live Services, SQL Azure for targeted cloud database management, SharePoint Services, Dynamic Customer Relationship Management (CRM) Services, and AppFabric** providing different sets of application-centric services.

Azure Services:

o **Compute services:** It includes the Microsoft Azure Cloud Services, Azure Virtual Machines, Azure Website, and Azure Mobile Services, which processes the data on the cloud with the help of powerful processors.

- o **Data services:** This service is used to store data over the cloud that can be scaled according to the requirements. It includes Microsoft Azure Storage (Blob, Queue Table, and Azure File services), Azure SQL Database, and the Redis Cache.

- o **Application services:** It includes services, which help us to build and operate our application, like the Azure Active Directory, Service Bus for connecting distributed systems, HDInsight for processing big data, the Azure Scheduler, and the Azure Media Services.

- o **Network services:** It helps you to connect with the cloud and on-premises infrastructure, which includes Virtual Networks, Azure Content Delivery Network, and the Azure Traffic Manager.

Azure Services Platform Components:

1. Windows Azure
2. SQL Services
3. .NET Services
4. Live Services

1. Windows Azure:

➢ Windows Azure is a Cloud-based Operating System.

➢ Provides the Development, hosting, and service management environment to Azure Services Platform.

➢ Supports existing standards like SOAP, REST, and XML.

➢ Developers can use the Visual Studio skills to build applications and services

Uses of Windows Azure:

- • Widows Azure can be used to

    1. Add Web Service applications to existing applications
    2. Build and Modify applications and then move them onto the Web
    3. Make, test, debug, and distribute web services efficiently and inexpensively
    4. Reduce the cost of IT Management

2. SQL **Services**

➢ Microsoft SQL Services extend SQL Server capabilities to the cloud as web-based services.

- ➢ Allows Storage of Varieties of data

    - o Structured Data

    - o Semistructured Data

    - o Unstructured Data

- ➢ Delivers a set of Integrated Services that allow

    1. Relational Queries

    2. Search

    3. Reporting

    4. Data Analytics

    5. Integration and Synchronization of data
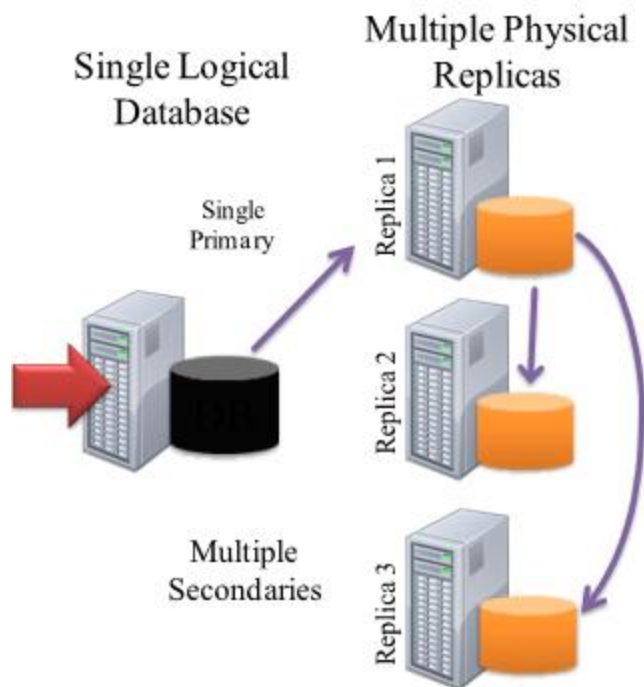

3. .NET Services

- ➢ Microsoft .NET Services are a Set of Microsoft hosted, developer oriented services.

- ➢ Provides components and high-level class libraries.

- ➢ Similar to .NET Framework that help developers to focus more on their product development.

- ➢ .NET services are also available to other development technologies through the use of industry standard protocols like REST, SOAP, and HTTP.

4. Live Services

- ➢ Live Services is a development center and supplier of software development kits for Windows Live and Azure Services Platforms
- ➢ **Windows Live** is a discontinued brand name for a set of web services and software products from Microsoft as part of their software plus services platform.
- ➢ The Windows Live brand was phased out during August 2012, when Windows 8 was release to manufacturing

Windows Live

Windows Live is the former collective brand name for a set of services and software products from Microsoft; part of their software plus services platform. A majority of these services are Web applications, accessible from a browser, but there are also client-side binary applications that require installation. There are three ways in which Windows Live services are offered: Windows Essentials applications, web services, and mobile services.
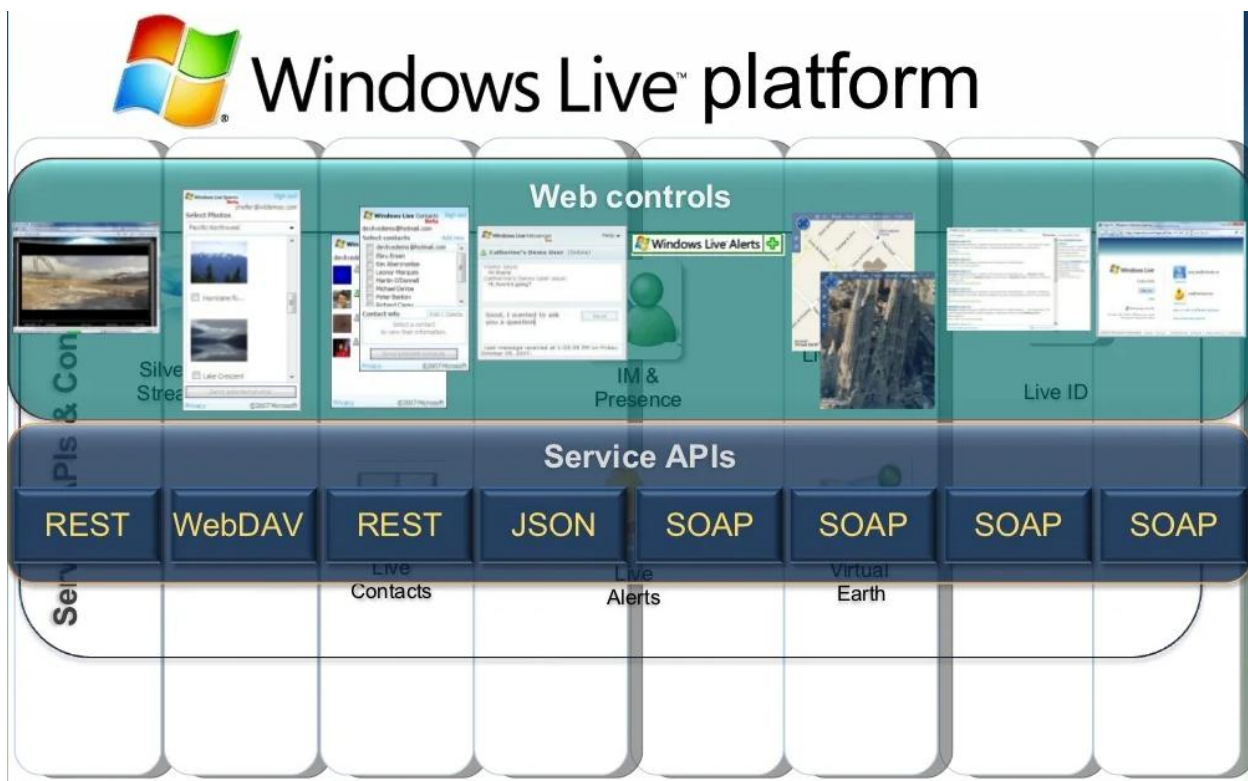
Windows Live is an integrated set of online services that makes it easier and more fun for consumers to communicate and share with others. The new generation of Windows Live includes updated experiences for photo sharing, email, and instant messaging, as well as integration with multiple third-party sites.

Communicating and Collaborating:

➢ Windows Live provides social features available to all customers, including an updated profile, a "what's new" feed of activities across the network, and web, photo sharing, and on-the-go access from virtually any device with Windows Live SkyDrive. Online storage is increasing from 5GB to 25GB.

➢ Windows Live Messenger includes more personalization, a "what's new" feed with updates from contacts across the Web, drag-and-drop photo sharing in the conversation window

➢ Windows Live Mail: POP3 email client that easily integrates with non-Microsoft email services

➢ Windows Live SkyDrive: Facilitates online Microsoft Office collaboration and provides free cloud storage for documents and photos.

➢ Windows Live Messenger Companion: Internet Explorer add-in for live collaboration

➢ Windows Live Family Safety: Extends parental controls in Windows 7 and Vista
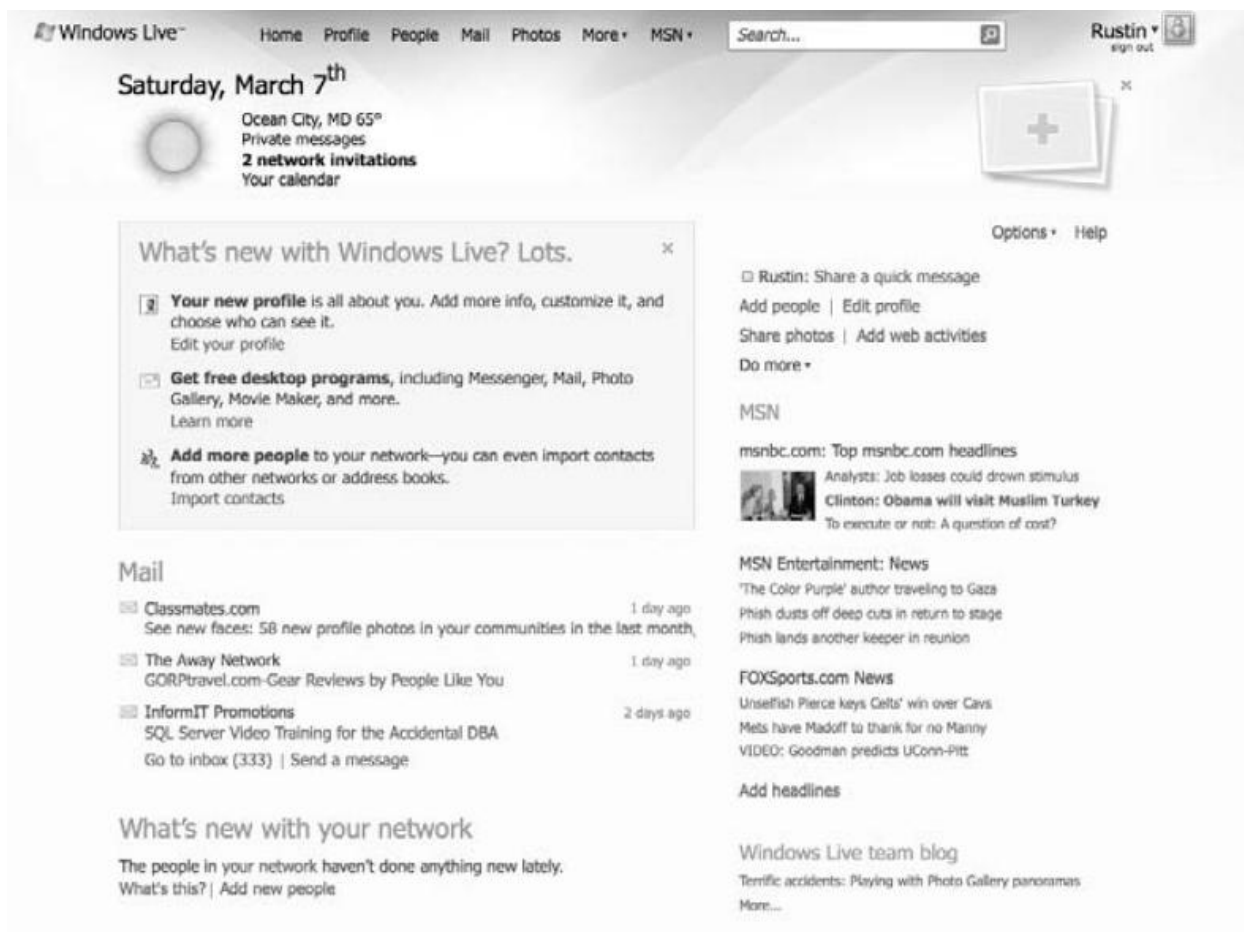
Microsoft Exchange Online

Microsoft Exchange Online is a hosted messaging solution that delivers email, calendar, contacts, and tasks from PCs, the web, and mobile devices. It integrates fully with Azure Active Directory, enabling administrators to use group policies, as well as other administration tools, to manage Exchange Online features across their environment.

Organizations that subscribe to Exchange Online retain control over the messaging services they offer to users. With the Exchange Online hosted plans described in this document, email is hosted on servers that support multiple customers simultaneously.

Exchange Online servers are geographically dispersed. The service is aimed at easing.IT's management duties by removing your need to deploy, configure, monitor, and upgrade on-site email solutions.

Customers using Active Directory can use a synchronization tool to keep the online and local Active Directories in sync.
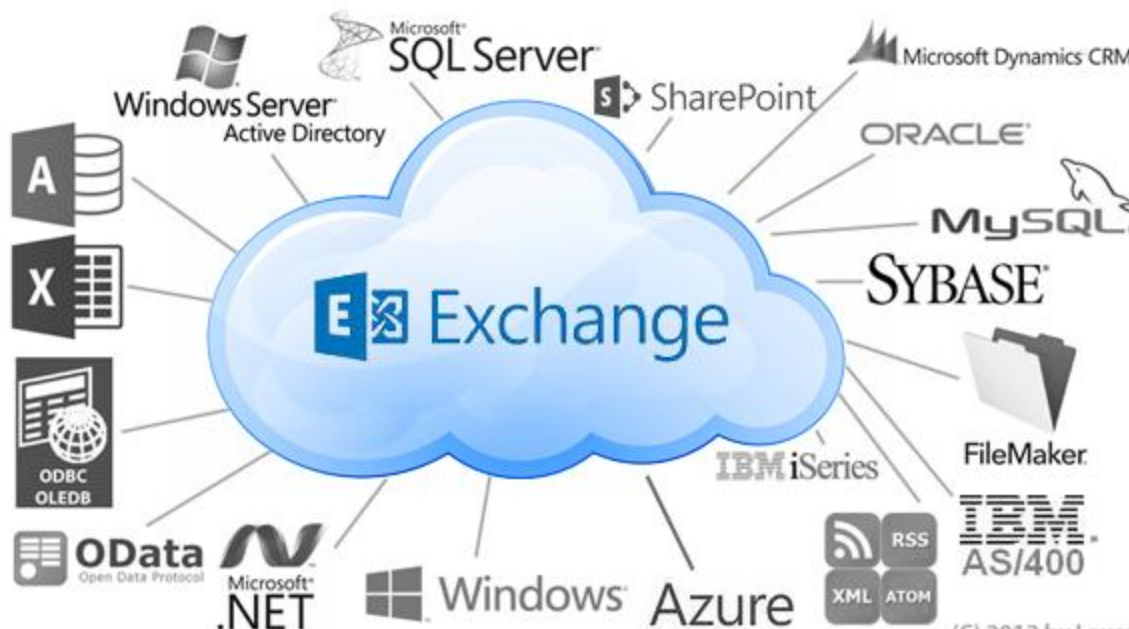
These are the key features of the online standard version of the solution:

➢ A 5GB mailbox (additional storage available for purchase—up to 25GB), shared calendar, contacts, tasks

➢ Outlook Client Connectivity including Outlook Anywhere

➢ Outlook Web Access

➢ Virus/spam filtering via Exchange Hosted Filtering

➢ Push email for Microsoft Windows Mobile 6.0/6.1 and Exchange ActiveSync 12 devices

---

- Email synchronization for Nokia E series and N series and iPhone 2.0 (no ActiveSync push)

- Built-in business continuity and disaster recovery capabilities

- Scheduled uptime of 99.9 percent with financially backed service level agreements

- Use of HTTPS to help keep Internet access secure

- Tier 2 support 24/7 (web form and phone based) for IT administrators

- Sign-In Tool for single sign-on capability

- Directory Synchronization Tool to help keep on-premise and online Active Directories in sync

- Coexistence, or the ability for some users to be on mail servers on premises and for some to be online.

- Migration Tools to help you move your current mailbox data into the online environment

Share Point Services

Windows Sharepoint Services (WSS) is a portal-based platform for creating, managing and sharing documents and customized Web services. Microsoft offers its SharePoint Services to aid collaboration efforts.

- SharePoint Services provides communities for team collaboration and makes it easy for users to work together on documents, tasks, contacts, events, and other information. Additionally, team and site managers can coordinate site contents and user activity.

- SharePoint sites are made up of Web Parts and Windows ASP.NET-based components. Web Parts are designed to be add-ons to web pages and configured by site administrators and users to create complete page-based applications.

- SharePoint sites are places where teams can participate in discussions, shared document collaboration, and surveys. Site content can be accessed from a web browser and through clients that support web services. Document collaboration controls allow you to check in, check out, and control document versioning.

- Microsoft Office System programs use SharePoint site content. A site's collaborative content—like documents, lists, events, and so forth—can be read and edited with Microsoft Office Word. Picture editing is also possible. Microsoft Office Outlook allows SharePoint site event calendars to be viewed side by side with personal calendars.

SharePoint services include consulting, development, migration, integration, support, and managed services.

SharePoint consulting team will help you with the following customized solutions:

➢ Third-party Collaboration tools.

➢ Business Workflows.

➢ Intranet/Extranet Portals.

➢ Business Intelligence Dashboards.

➢ Core Content Management Systems.

➢ CRM Tools.

➢ Data Collaboration Solutions.

➢ Excel Services.

➢ Integration Tools.

Microsoft Dynamics CRM (Customer relationship management)

Microsoft Dynamics CRM Online is an on-demand customer relationship management service hosted and managed by Microsoft. The Internet service delivers a full suite of marketing, sales, and service capabilities through a web browser or directly into Microsoft Office and Outlook. It provides "instant-on" access to businesses that want a full-featured CRM solution with no IT infrastructure investment or setup required.

"At Microsoft, we're revolutionizing how companies deploy marketing, sales, and service solutions to users within their organization," said Brad Wilson, general manager of Microsoft Dynamics CRM at Microsoft.

"Microsoft Dynamics CRM delivers the power of choice to customers, with a familiar and productive user experience and a multitenant platform that enables fast on-premise implementations or 'instant-on' deployments over the Internet."

Microsoft Dynamics CRM Online supplements Microsoft's software plus services strategy for delivering integrated business solutions over the Internet, and it is a part of Microsoft's multibillion-dollar investment in global datacenters.

In addition to full access through a zero-footprint browser client, the new service delivers marketing, sales, and service information within a native Microsoft Office experience, integrated with the desktop tools that employees already use every day, enabling businesses to ramp up end-user adoption and productivity rapidly.

Microsoft Dynamics CRM Online is initially packaged in two service offerings:

- Microsoft Dynamics CRM Online Professional delivers a full suite of CRM capabilities with extensive configurability and extensibility options. Businesses get 5GB of data storage, 100 configurable workflows, and 100 custom entities. The Professional edition is priced at US$44 per user per month, with an introductory offer of US$39 per user per month.

- Microsoft Dynamics CRM Online Professional Plus delivers all the capabilities of the Professional version plus offline data synchronization with expanded data storage, workflow, and customization options that give businesses 20GB of data storage, 200 configurable workflows, and 200 custom entities. The Professional Plus edition is priced at US$59 per user per month.

<u>**CRM system helps benefit your individual business teams:**</u>

- Marketing teams
- Sales teams
- Customer service teams
- Field service teams
- Project service automation teams