Web Component Development with Servlet & JSP Technologies (EE 6)

Module-13: *Deploying J2EE Application to Cloud*

Team Emertxe

Objectives

Upon completion of this module, you should be able to:

- What is Cloud?
- Types of Cloud.
- Cloud Sevice Models.
- Advantages of Cloud-Computing.
- What is Web Service?
- Types of Web Services.
- Building Web services with JAX-WS
- Deploy JAX-WS web services on Tomcat
- AWS (Amazon Web Service)
- AWS and Normal Web Hosting Service
- AWS Architecture

Relevance

Discussion –

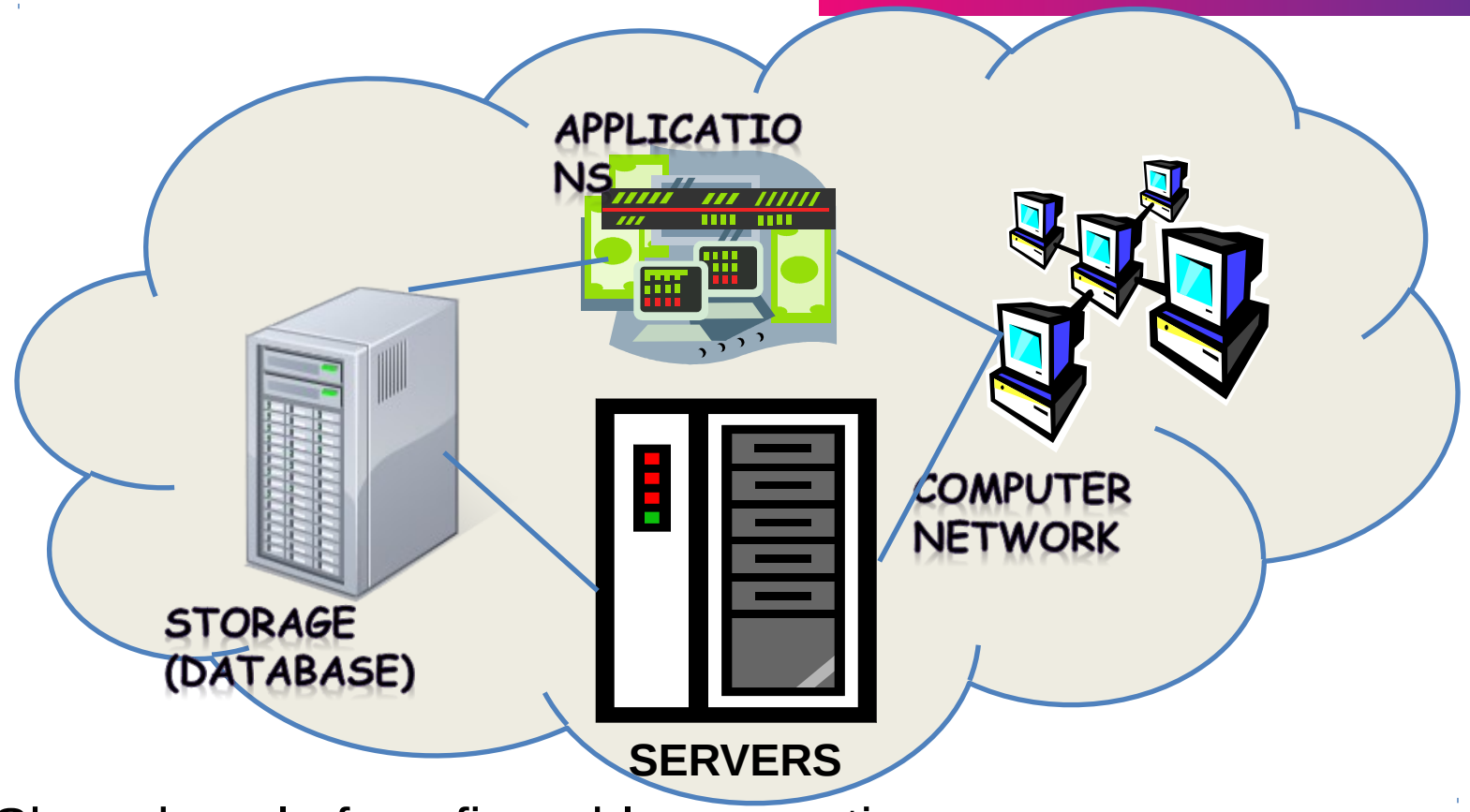
What is Cloud?

- Cloud Computing is a general term used to describe a new class of network based computing that takes place over the Internet ,

  - a collection/group of integrated and networked hardware, software and Internet Infrastructure (called a plateform).
  - Using the internet for communication and transport provides hardware ,software and networking services to clients.

- These services hide the compexity and details of the underlying infrastructure from users and applications by providing Application Programming Interface.

What is Cloud?



Shared pool of configurable computing resources

- On-demand network access
- Provisioned by the Service Provider

EMERTXE

Types of Cloud

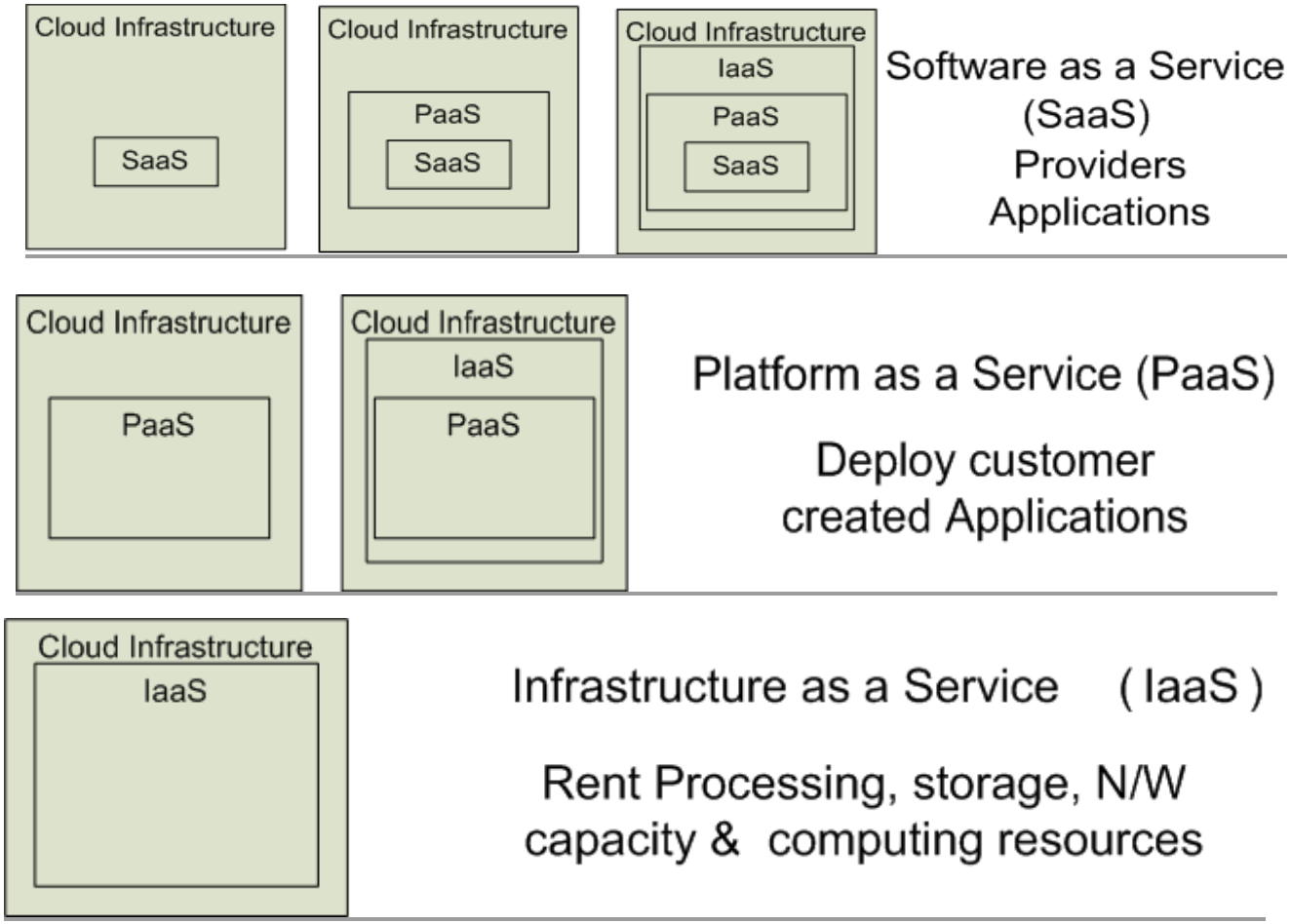
- Public Cloud :- Public cloud allows the accessibility of systems and services easily to general public. Eg. Amazon , IBM , Microsoft ,Google etc.
- Private Cloud :- Private cloud allows the accessibility of systems and services within organization.
- Hybrid Cloud :- Hybrid Cloud is the mixture of public and private cloud. Non critical activities are performed by public cloud and critical activities are performed by private cloud.

Cloud Service Models

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)



EMERTXE

Advantages of Cloud

- **Lower Cost Computers for users -** In Cloud , we don't require a high-powered computer to run cloud computing's web based applications because applications run on cloud not on desktop PC or laptop.
- **Lower IT infrastructure cost -** By using cloud computing , we don't need to invest in larger numbers of more powerful servers ,not require IT staff also for handling such powerful servers.
- **Lower Software Cost -** It reduces the software cost because we don't need to purchase separate software packages fo each computer in the organization.

Advantages of Cloud

- **Instant Software updates –** Another software related advantage in cloud computing is that users don't need to face with the choice between obsolete software and high upgrade costs . If the app is web-based , updates happen automatically and are available next time when the user logs in to the cloud.
- **Increased Computing Power –** The execution capacity of cloud servers are very high. It processes the application very fast.
- **Unlimited storage capacity -** Cloud offers a huge amount of storage capacity like 2000GB or more than that if required.

Web Services

A Web Service can be defined in following ways :

- is a client server application or application component for communication.
- method of communication between two devices over network.
- is a software system for interoperable machine to machine communication.
- is a collection of standards or protocols for exchanging information between two devices or application.

Types of Web Services

There are two types of Web Services:

1) Soap Web Services

2) RESTful Web Services

Soap Web Services

Soap web services use XML messages that follow the Simple Object Access Protocol (SOAP) standard , an XML language defining a message architecture and message formats. Such system often contain a machine -readable description of the opeations offered by the service, written in the Web Services Description Language(WSDL) , an XML lanaguage for defining interface syntactically.

RESTful Web Services

In Java EE 6 , JAX-RS provides the functionality for Representational State Transfer(RESTful) web services.

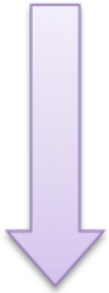
RESTful web services often better integrated with HTTP than SOAP-based services are , do not require XML messages or WSDL service -API definitions.

RESTful web services use existing W3C and internet Engineering Task Force (IETF) standards (HTTP , XML ,URI ,MIME) and have a lightweight infrastructure that allows services to be built with minimal tooling ,devloping RESTful services is inexpensive.

SOAP & RESTful Web Service

Web Services 1.0

- "SOAP Web Services"
- Using the **SOAP** protocol (Simple Object Access Protocol)
- **XML** (Extensible Markup Language)



Web Services 2.0

- "RESTful Web Services"
- Using the **REST** protocol (Representational State Transfer)
- Uses standard **HTTP** methods (GET, PUT, POST, DELETE) (HTTP: Hypertext Transfer Protocol)
- Uses **JSON** (JavaScript Object Notation) or XML

Building Web services with JAX-WS

JAX-WS allows developers to write message-oriented as well as Remote Procedure Call-oriented(RPC -oriented) web services.

The starting point for developing a JAX-WS web service is a java class annoted javax.jws.WebService annotation. The @ WebService annotation defines the web service endpoint.

A service endpoint interface or service endpoint Implementation (SEI) is a java class ,that declares the methods that a client can invoke on the service. An interface is not required when building a JAX-WS endpoint.

Deploy JAX-WS web services on Tomcat

Steps of a web service deployment

- Create a web service
- Create a sun-jaxws.xml , defines web service implementation class
- Create a standard web.xml ,defines WSServletContextLitener ,WSServlet and structure of a web project.
- Build tool to generate WAR file.
- Copy JAX-WS dependencies to “${Tomcat}/lib” folder.
- Copy WAR to “${Tomcat}/webapp” folder.
- Start it.

Creating Web Service

File : HelloWeb.java

```
package com.emertxe.ws;
import javax.jws.WebMethod;
import javax.jws.WebService;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.Style;

//Service Endpoint Interface
@WebService
@SOAPBinding(style = Style.RPC)
public interface HelloWeb{

    @WebMethod String getHelloWebAsString();

}
```

EMERTXE

Creating Web Service

File : HelloWebImpl.java

```
package com.emertxe.ws;

import javax.jws.WebService;

//Service Implementation Bean

@WebService(endpointInterface = "com.emertxe.ws.HelloWeb")
public class HelloWebImpl implements HelloWeb{

	@Override
	public String getHelloWebAsString() {
		return "Hello Web  JAX-WS";
	}
}
```

Create a web service deployment descriptor

File : sun-jaxws.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<endpoints
  xmlns="http://java.sun.com/xml/ns/jax-ws/ri/runtime"
  version="2.0">
  <endpoint
      name="HelloWeb"
      implementation="com.emertxe.ws.HelloWebImpl"
      url-pattern="/hello"/>
</endpoints>
```

EMERTXE

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems,
Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_3.dtd">

<web-app>
    <listener>
        <listener-class>

com.sun.xml.ws.transport.http.servlet.WSServletContextListener
        </listener-class>
    </listener>
```

web.xml

```
<servlet>
        <servlet-name>hello</servlet-name>
        <servlet-class>
            com.sun.xml.ws.transport.http.servlet.WSServlet
        </servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
<servlet-mapping>
        <servlet-name>hello</servlet-name>
        <url-pattern>/hello</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>120</session-timeout>
    </session-config>
</web-app>
```

ΣMERTXE

WAR Content

WEB-INF/classes/com/emertxe/ws/HelloWeb.class

WEB-INF/classes/com/emertxe/ws/HelloWebImpl.class

WEB-INF/web.xml

WEB-INF/sun-jaxws.xml

EMERTXE

JAX-WS Dependencies

Go here http://jax-ws.java.net/.

copy following JAX-WS dependencies to Tomcat library folder “{$TOMCAT}/lib“.

jaxb-impl.jar
jaxws-api.jar
jaxws-rt.jar
gmbal-api-only.jar
management-api.jar
stax-ex.jar
streambuffer.jar
policy.jar

Deployment

Copy the generated WAR file to {$TOMCAT}/webapps/ folder and start the Tomcat server.

For testing, access this URL :
http://localhost:8080/HelloWeb/hello

AWS (Amazon Web Service)

- AWS is a subsidiary of Amazon.com ,offers a suite of cloud computing services that make up an on-demand computing plateform.
- The most central and best-known of these services include Amazon Elastic Compute Cloud , also known as “EC2” and Amazon Simple Storage Service, also known as “S3”.

AWS (Amazon Web Service)

- Amazon Web Services offers a broad set of global cloud-based products including storage , database ,analytics, networking ,mobile, developer tools ,management tools, security, compute and enterprise applications.

- These services help organizations move faster , lower IT costs and scale .

- AWS is trusted by the largest enterprises and starts-ups to power a wide variety of workloads including : web and mobile applications ,game development ,data processing and warehousing ,storage ,archieve and many others.

Normal Web Hosting Service

- Shared :- A physical server that is shared by many different customers. User account is restricted to certain files , and very limited access. Usually this web server runs one Web Server (usually Apache).

- Virtual Private :- Many virtual server are stored on one physical server. Each Customer has their own private virtual server.

- Dedicated : A physical server that is leased to a single customer.

Amazon Web Service

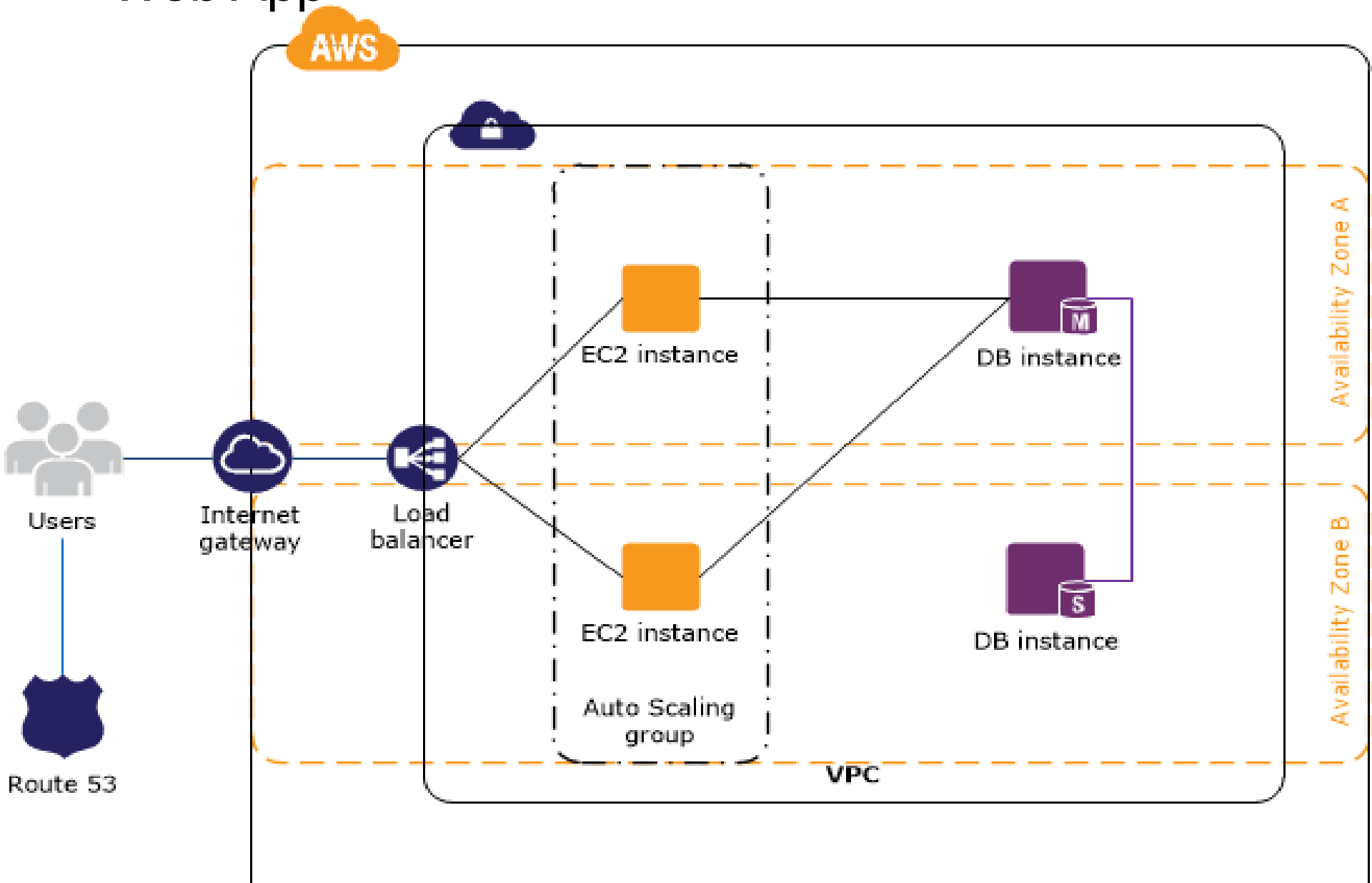
Standard :- AWS allows for dedicated root access to the server , which is a feature not available in most virtual private servers.

Dedicated :- Dedicated Amazon will provide a virtual server that is not on a shared server ,but its own private cloud . It is similar to a dedicated server , but with the flexibility of a virtual private server.

Amazon Web Service advantages over normal Web Hosting Service

- **High -availability** (Eliminating Single points of failure)
- **Distributed Infrastructure ,**reducing latency to all regions of the world.
- **Cost saving ,**scaling down on hardware being used,saving money in the long term.
- **On-demand infrastructure for scaling applications or tasks** (adding servers or “horizontal scaling “ to massively increase the hardware power available to the application)

AWS Architecture for a Web App



AWS Architecture for a Web App

- The Web Application tiers runs on EC2(Amazon Elastic Compute Cloud) instances in VPC.
- Access to the EC2 instances over SSH is controlled by a security group which acts as a firewall.
- The Autoscaling maintains a fleet of EC2.Auto Scaling group spans multiple availability Zones to protect against the potential failureof a single scaling group.
- When the Auto Scaling group launches or terminates instances based on the load ,the load balancer automatically adjusts accordingly.

AWS Architecture for a Web App

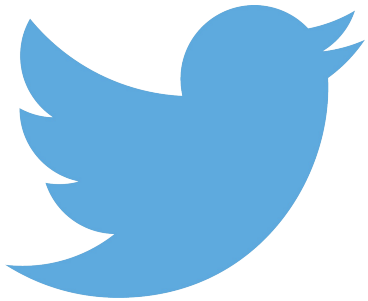
- The database tier consists of DB instances in VPC, including a master and a local slavelocated in multiple Availability Zones.
- Access to the DB instances from the EC2 instances is controlled by a security group.
- Amazon Route 53 provides secure and Reliable routing of the domain name to infrastructure hosted on AWS.

Stay connected

About us: Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,
No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046
T: +91 80 6562 9666
E: training@emertxe.com

https://www.facebook.com/Emertxe



https://twitter.com/EmertxeTweet

https://www.slideshare.net/EmertxeSlides

EMERTXE

Thank You