



Building Web Application Using Cloud Computing

Maha A. Sayal¹, Ehsan Ali Al-Zubaidi²

¹Computer Science Department, College of Computer Science & Mathematics, University of Thi-Qar
Thi-Qar, IRAQ

²Computer and Internet Department, Faculty of Physical Planning, University of Kufa, Najaf,IRAQ

Abstract: Web applications evolved from sites of web or Web systems, This application builds on and extend a Web system to add business functionality . There are challenges that include scalability and high availability, Cloud data stores provide high availability and scalability properties for web applications [1]. Cloud Computing offers the vision of a virtually infinite pool of storage, computing and networking resources where applications can be scalable deployed [2]. This paper demonstrates how to build any web application in the cloud, provide multiple interfaces for it , and how to choose appropriate service from Amazon Web Services for an application.
Index Terms: Scalability, Web applications, cloud computing, Web system, Amazon Web Services.

I. Introduction

Cloud computing enables companies and customer to compute resources as a utility to maintain and build computing infrastructures in-house. Cloud computing promises several attractive benefits for businesses and users. The main benefits of cloud computing include:

1. Elasticity: Companies can scale up and down of the recourses.
2. Self-service provisioning: users can spin up computing resources for almost any type of workload on-demand.
3. Pay per use: Computing resources are measured at a particular level, allowing customers to pay only for the resources and workloads they use[3].

Cloud computing is a technology which uses one remote server and the internet to maintain various data and applications. Cloud computing provides important cost effective IT resources as a cost on demand IT based on the actual usage of the customer[4]. Because of rapid growth, many companies are unable to handle their IT requirement even after having an in-house data center [5]. Cloud services are used to improve IT capabilities without investing large amounts in new datacenters. This technology helps companies with much more efficient computing by centralizing processing, storage, memory and bandwidth[6]. Cloud computing segments : Applications, Storage, and Connectivity.



Figure (1) illustrates the main benefits of cloud computing

II. Proposed Application

The traditional web application uses three layers: presentation, application, and database. When dealing with robust and modular web applications, however, you typically see the presentation layer split into two different layers: client and representation. Figure (2) shows the three-tier web pattern[7].



Figure(2) The three –tier web pattern

A. The- tier web pattern

The three-tier web pattern is designed to help developer extrapolate the major parts of developer application and enable it to be more modular and scalable .This architecture is designed to provide a public_ facing interface to it as a whole. By splitting an application into multiple tiers, he can add middleware, secure each individual layer and even offer up layers to untrusted third –party applications. The developer will also have the ability to replace a layer to adapt to his need.In this pattern, the presentation layer defines what the clients show on their system. The application has both the model and the control bits, where all logic actually happens in this layer. Figure(3) show an alternative version of the tiered web pattern that splits the presentation layer into two layers. By splitting the presentation layer into client and representation layer, the developer can offload the client layer to the actual client and simply provide one single representation for multiple different clients. This becomes increasingly important in today's world with mobile smartphones and devices that have less screen real estate and various interfaces, such as touch instead of a typical mouse [3].

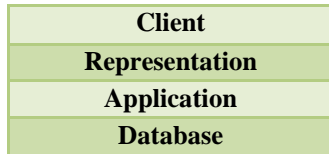


Figure (3) The four –tier web pattern

B. Reasons for Usage

The developer wants to use this pattern if any of following applies:

He needs to build a modular cloud application .To provide multiple interfaces to his application. And integrate multiple applications with each other (using Amazon cloud services).

C. Implementation

A typical implementation of n-tier architecture is built on of four –tier architecture. Most important is where each layer goes. Figure (4) shows how a developer can distribute the layers across provided by his cloud services.

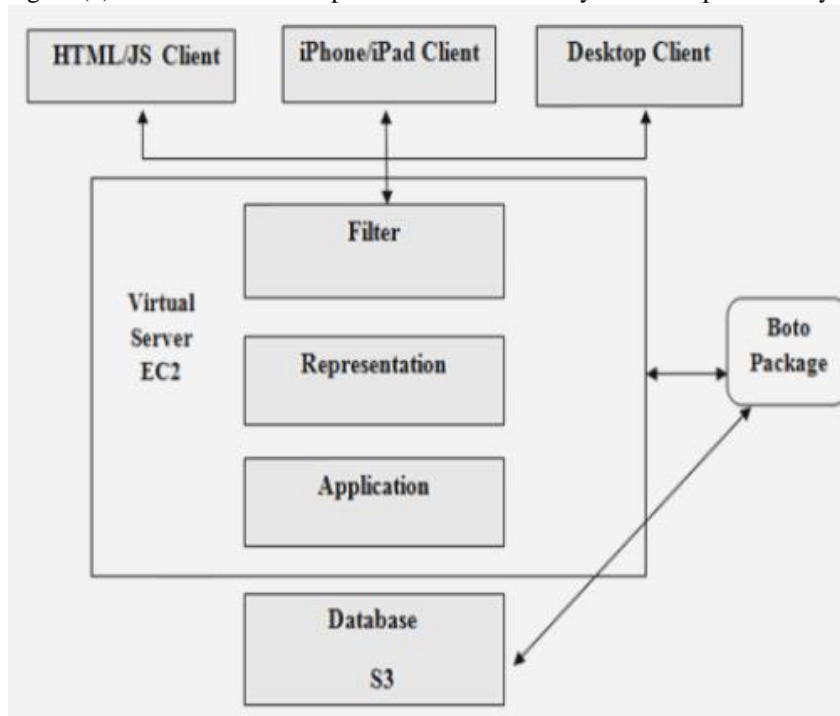


Figure (3) Deployment of the n-tier architecture

In this method of n-tier architecture, three different client layers are provided: one for basic HTML and JavaScript, another for iPhone and Ipad Clients, and a third for standard desktop clients .These three layers each can present the information to the user and set directly on the client's machine. A developer cannot trust anything coming from any of these clients, so a developer must perform all authentication and authorization below this layer. Directly under these client layers is the filter layer that enables a developer to abstract the authentication and authorization from the representation and application layers. It sits directly on top of representation layer, modifying any input and output that flows through it. These two layers must both speak the same communication protocol .

The representation layer sits directly below the filter layer and typically is on the same system as the application layer. This layer modifies any output from the application layer into a usable format for the filter and client layers. This typically transforms your internal representation, usually a raw object, into a standard such XML (a markup language that defines a group of rules for encoding documents in a format that is both machine-readable and human-readable)[8].

The application layer rests directly below the representation layer and often actually merged with the representation layer. This layer contains all business logic and communicates directly with the database layer, in this layer developer can use boto package from Amazon cloud, all programming perform on a virtual server (EC2) that is taken from Amazon cloud service.

The database layer rests outside of primary servers. In some cloud platforms, this will actually be provided for developer automatically and developer not needs to control it.

Database layers come in two main flavors: relational and nonrelational. Typical relational database layers include MySQL and Oracle, in this work S3 is used as a relational DB is taken from Amazon cloud services .some newer databases, including SimpleDB, are nonrelational and do not support complex queries or joining to provide better scalability.

III. Methodology:

A. Signing Up for Amazon Web Services

In the beginning, developer needs to sign up one for the services according to what developer needs in his web application; this work is needed the following:

- Elastic Compute Cloud (EC2)
- Simple Queue Service (SQS)
- Simple Storage Service (S3)

B. Installing Boto Package

A developer can install boto in many different ways, but the best way to make sure a developer is using the latest code is to download the source from GitHub at <http://github.com/boto/boto>. There are several various ways to download code, but the easiest is just to click the Downloads button and choose an appropriate version to download boto is an interface for AWS, boto library works with python library. A developer can use these instructions.

```
tar xzf http://boto.googlecode.com/files/boto-1.9b.tar.gz
cd boto-1.9b
Python setup.py install
```

to install boto writes instruction: `pip install -U boto`

C. Choosing of The Environment

Although there are several ways to choose your environment for boto, use the one that's also compatible with using the downloaded Amazon Tools, which user can find at <http://aws.amazon.com/developertools>.

D. Connecting with Amazon S3

To connect with S3, write these instructions .

```
import boto
s3 = boto.connect_s3()
bucket = s3.create_bucket('media.yourdomain.com') # bucket names must be unique
key = bucket.new_key('examples/first_file.csv')
key.set_contents_from_filename('/home/patrick/first_file.csv')
key.set_acl('public-read')
```

E. Working with Amazon SQS

SQS is a message-queuing service from Amazon that consider important part for building cloud applications. SQS allows developer to read and write messages. In addition, he can get an count of existing messages and deleted messages from this queue. so developer can store his message data on Amazon S3 and pass into the message the bucket and key to the actual message data, like so:

```
import boto
sqs = boto.connect_sqs()
q = sqs.create_queue('my_message_pump')
data1 = simplejson.dumps(<some data structure>)
```

```
s3 = boto.connect_s3()
bucket = s3.get_bucket('message_pump.yourdomain.com')
key1 = bucket.new_key1('2016-03-19/%s.json' % str(uuid.uuid4()))
key1.set_contents_from_string(data)
message = q.new_message(body=simplejson.dumps({'bucket': bucket.name, 'key': key1.name}))
q.write(message)
```

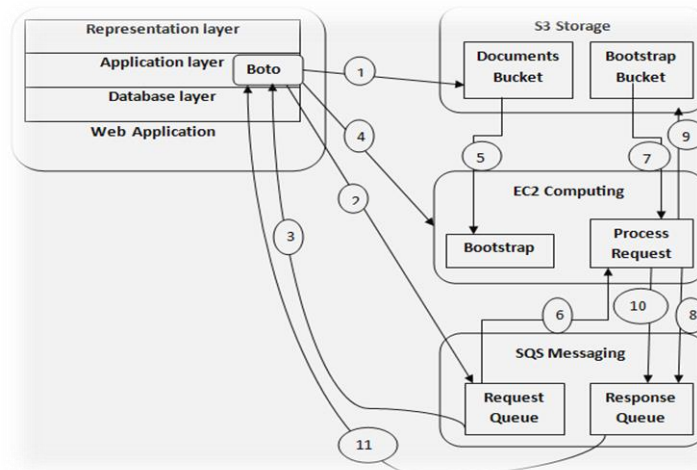
F. Processing with Amazon EC2

EC2 is the virtual server in cloud, When developer boot EC2 instance, he'll has it install what he wants installed easily, saving his time as well as getting services ready to run a script to process the queue without he having to log on:

```
reservation = ec2.run_instances(
    image_id = "ami-bb709dd2",
    key1_name = "ec2-sample-key",
    user1_data = ""#!/bin/bash
apt-get update
apt-get install -y imagemagick
""")
```

IV. Results

When developer gathers all this services and package boto, a simple application can be built, demonstrate how to gathered this software in a real application, take a look at this proposed application to perform Portable Document Format (PDF) documents into preview images to a possibility of displaying it on a Web site. This application that allows users to upload a PDF and a short time later have PNG previews of every page in that document.



STEPS

- 1- Upload PDF
- 2- Queue message to process PDF
- 3- Check count for messages
- 4- Boot up Node(s) to process messages
- 5- Download process script
- 6- Read Queue message and begin processing
- 7- Download PDF to process
- 8- Queue response message indicating processing started
- 9- Upload PNGs for each page of PDF
- 10- Queue response message indicating processing complete or exception
- 11- Process response messages and PDF update models

The PDF is transferred to Amazon S3 from the local Web server. Messages are put in a queue to process the uploaded PDFs; a node is booted up to response to the queue count; and the node bootstrap itself with software necessary to process the PDF. The node then writes a message of a result to a queue that a background task is checking on the Web server, which updates the database with the appropriate record.

V. Conclusion

This paper is considered a guide if developer wants to use Amazon cloud services, and he can easily build application using cloud , also the cost of cloud computing is little because all software and hardware that are needed are rent for use time as well as high performance that application acquired it from cloud services.

References

- [1] Yiduo Mei; Ling Liu ; Xing Pu and Sankaran Sivathanu," Performance Measurements and Analysis of Network I/O Applications in Virtualized Cloud", IEEE 3rd International Conference on Cloud Computing, pp.59-66, 2010.
- [2] Luit Infotech, "What is Cloud Computing ", IBM Journal of Research and Development, Vol.60, No.4, pp.41-44, 2012.
- [3] Christopher M. Moyer, "Building Applications in the Cloud", ISBN-13: 978-0- 321-72020-7, Addison Wesley, 2011.
- [4] Alex Huth and James Cebula, "The Basics of cloud computing", International Journal of Network Security & Its Applications (IJNSA), Vol.6, No.3, pp.72- 76, 2014.
- [5] Nancy Lynach and Seth Gilbert, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services", ACM Sigact Journal, Vol.33, No. 2, pp, 51-59, 2002.
- [6] Rajkumar Buyya and Karthik Sukumar, " Platforms for Building and Deploying Applications for Cloud Computing", CSI Communications, Vol.35, no.1 pp. 6-11, May 2011.
- [7] Matt Tavis and Philip Fitzsimons, " Web Application Hosting in the AWS Cloud", ISBN 978-0-9805768-3-2, Site Point, September 2012.
- [8] Alex Amies; Harm Sluiman; QiangGuo Tong and GuoNing Liu , "Developing and Hosting Applications on the Cloud", ISBN-13: 978-0-13-306684-5, IBM Press, July 2012.