Indian Institute of Information Technology, Vadodara

Project Report

CS421 : Cloud Computing

Group : K

NOORUL HASAN ALI (201851078)
PARMANAND PATIDAR (201851080,**Team Lead**)
PATEL DARSHAN ARUNBHAI (201851081)

Table of Contents

Executive Summary:

Be it the life of a student or a professional, sports is an inseparable part of people's life. Sports helps us in building discipline, better overall health and so many other qualities like teamwork. Here, we are building an application which can be used by all people who are associated with sports in the city of Vadodara. Whether it be a player, a team or just a viewer of sport events, the application will be beneficial to all of them. The application would provide a User-friendly Interface where Viewers and Players/teams can interact among themselves through sharing schedules, photos, videos, event details and results.

The project will target populations in sets associated with different sports. The different sports, the stats and the basic features supported by the interface are mentioned in the details Market Analysis section. How the things would work behind the scenes, the server architecture and the cloud storage are mentioned in detail in Architecture and Monthly Storage sections. The requirements of memory would vary from day to day, from week to week and so will the number of users, so monthly computations are required. As we are planning to use the Amazon AWS platform, the complete computation requirements are mentioned in the requirements section based on the Amazon AWS cloud storage platform. Furthermore, the traffic on the interface and simultaneous upload or download of data would require a good Network to analyse and maintain user satisfaction. Uploading different kinds of data would have different network requirements which is discussed statistically and calculatively in the Networks Requirements section.

Then, comes one of the major parts which plays a vital role in maintenance as well as well execution and success of each and every project, that is, the Security of the data, supply chain and other Security required in the Project. A very fine detailed analysis of Security and Security costs is mentioned in the Monthly Analysis Section. Moving further, the overall financial analysis and growth of the project matters as it must earn the project owners some profit in order to stay financially stable in the market. Financial considerations and analysis hence, are very important to mention in order for the project to be competitive among similar products.

Market Analysis :

In Market Analysis, our aim is to estimate the no. of the potential users/teams of each sport. Here we are finding that “How many people will use this app?” To find this we have to find how many people are interested in playing sports, how many people are following sports and how many people are watching sports. On the basis of these we can make assumptions about our potential users which can use our app. So first of all we have to break down all sports by popularity. So we can estimate the number of users by each sport. For the initial phase the target of our app is Vadodara city.

Total population of Gujarat :- **62,700,000**

Total population of Vadodara :- **2,180,000**

No. of people associated with sports in Gujarat = 15% of total population of gujarat

= **9,405,000**

Now , to find now of people associated in sports in vadodara, we are applying unary method :

No. of people associated with sports in Vadodara =

((Population of Vadodara)*(No. of people associated with sports in Gujarat)) / (Population of Gujarat)

= (2,180,000*9,405,000) / 62,700,000

= **327,000**

No. of people associated with sports in Vadodara = **327,000**

Here we are assuming that many of people don’t have mobile phone or there are not using our app or any other case, So we assume that,

Potential users of application = **300,000**

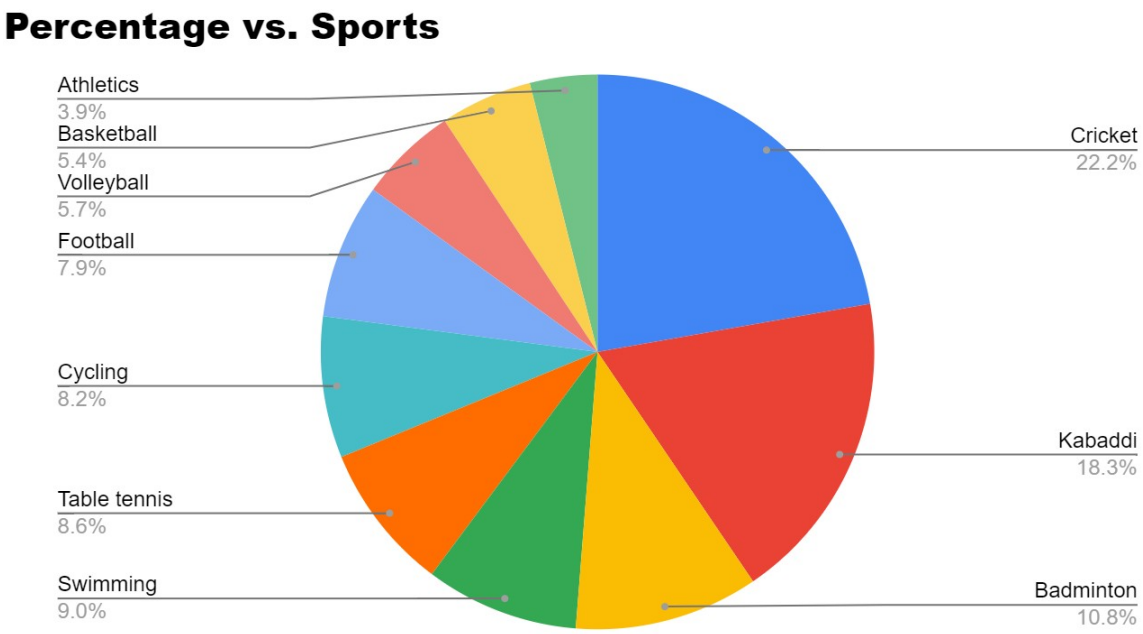
Here are the top 10 sports to be included in Phase 1 :

- Cricket
- Kabaddi
- Badminton
- Swimming
- Table tennis
- Cycling
- Football
- Volleyball
- Basketball
- Athletics

Approx users distribution for different sports :

Sports	Percentage	Users (approx.)
Cricket	62	186000
Kabaddi	51	153000
Badminton	30	90000
Swimming	25	75000
Table tennis	24	72000
Cycling	23	69000
Football	22	66000
Volleyball	16	48000
Basketball	15	45000
Athletics	11	33000

Each user can follow multiple sports.It is possible that a user is interested in more than 1 sports and may follow multiple sports.



Content types to be collected for different sports :

- Images
- Text
- Videos

Details to be collected from user :

User Personal Data

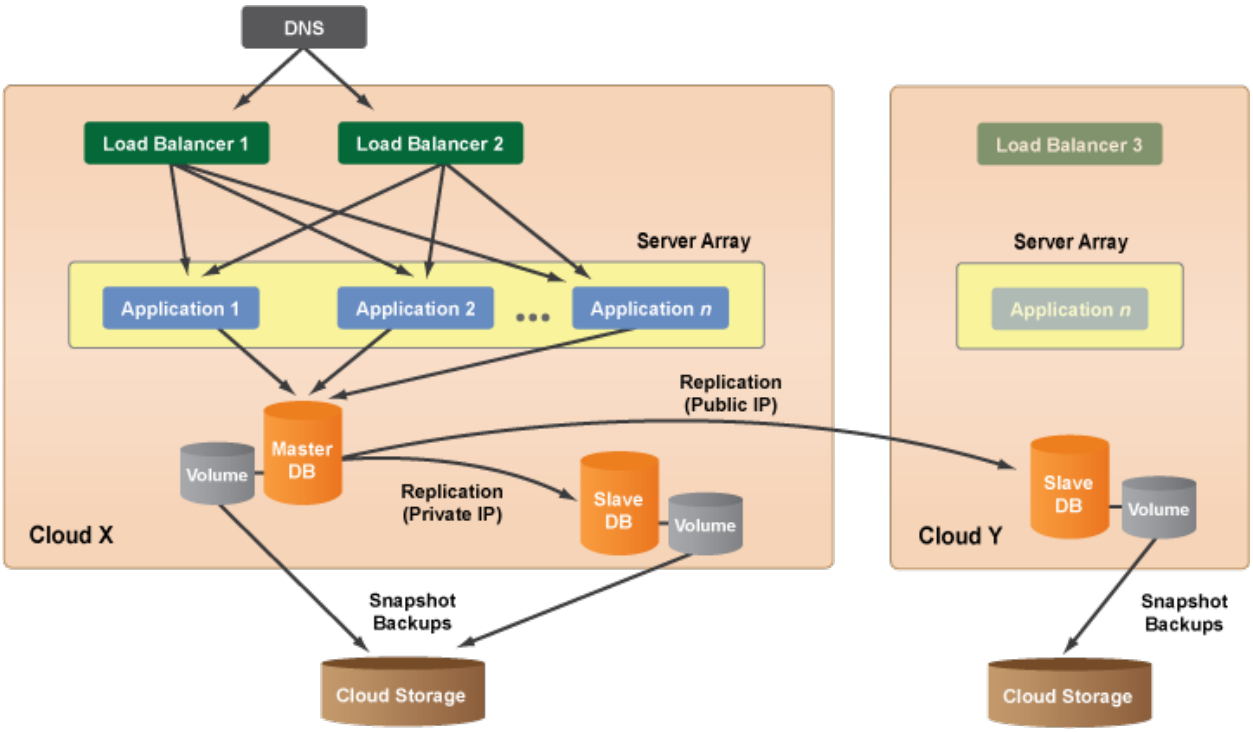
- First Name
- Last Name
- Profile Photo
- Gender
- DOB
- Email ID
- Password
- Address
- Interested Sports

Design/Architecture :

Here, I am using Failover MultiCloud. Generally, in a web application scenario, we run multiple copies of our app simultaneously to cover the volume of our customer traffic. These multiple copies of our application will be hosted on identical EC2 instances (cloud servers), each handling customer requests.

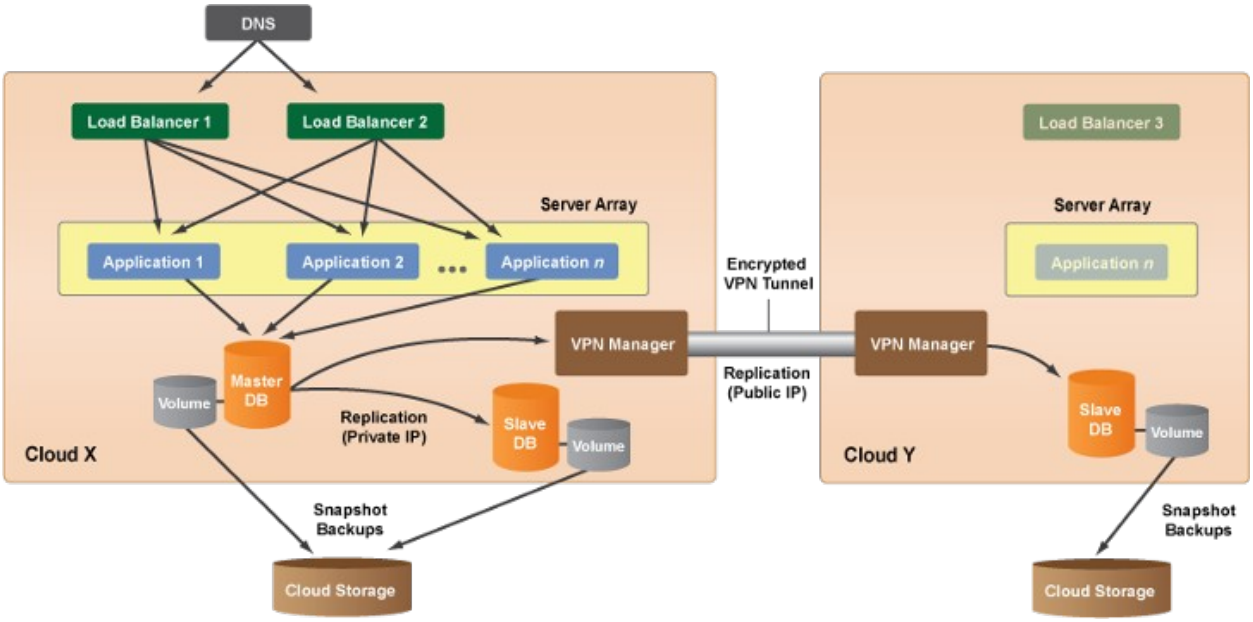
Adding Amazon EC2 Auto Scaling to our application architecture is one way to maximize the benefits of the AWS Cloud. When we use Amazon EC2 Auto Scaling, our application gains the following benefits,better fault tolerance, cost management and availability.

Also we can use load balancer for load balancing. Load balance can be software based or hardware based.



In the above diagram, there is a running Slave-DB server that's serving as a warm backup, but it's replicating data with the Master-DB across the public, not private IP address. Only servers within the same cloud infrastructure can communicate over a private IP address. However, if there is ever a problem or failure that would require you to switch clouds, a MultiCloud Architecture would allow you to easily migrate

your site/application. Notice that the other tiers of the reference architecture have already been configured and are ready to be launched if you need to migrate your production environment from Cloud X to Cloud Y.



If we want to send/receive data in a secure manner between servers in two different clouds, we can use data encryption or a VPN wrapped around the public IP address since any data transmitted between different cloud infrastructures (except if they're both private clouds) is sent over the public IP. In the diagram below, data replication across the public Internet is sent between the servers in two different clouds over a secure VPN tunnel.

Also Since Public clouds are scalable but provide no restrictions on the data whereas private clouds provide control, We can use Hybrid cloud to put restrictions on the data as well as provide scalability to the data. Therefore we shall have an autoscaling model which uses Hybrid cloud deployment to provide restrictions.

Rollout Plan

Week 1: Creating the cloud resource.

- Creating VM's and setting zones.
- Creating Instance templates which will define the configuration of every VM in the cluster
- (disk, CPUs, memory, etc). Managed instance groups use the instance template to instantiate
- multiple VM instances.
- Creating Load Balancers and servers.

Week 2: Performing foundational infrastructure tasks

- Specifying access controls.
- Trying and testing to publish and consume messages with pull subscribers.
- Trying several other functions to perform activities on cloud.

Week 3: Setting and configuring the cloud.

- Setting up and implementing the master-slave database.
- Testing connectivity across VM instances and VPC networks.
- Establishing uptime check alerts.
- Managing deployments using kubernetes for future expansion of cloud usage and scaling purposes.

Week 4: Deploying and managing the cloud environment.

- Completing production of application environment
- Ensuring monitoring and alerts
- Testing deployed environment.

Storage Analysis :

User Personal Data

- First Name
- Last Name
- Profile Photo
- Gender
- DOB
- Email ID
- Password
- Address
- Interested Sports

1 MB for Text based Data & 5 MB for Profile Photo
Maximum Storage for storing the personal data of a user will be **6 MB**

Post by a User

- Post Title
- Post Description
- Post Attachment (Images/Videos/Any other Files)

1 MB for Text based Data & 10 MB for attachments

Maximum Storage for storing the Post Data will be **11 MB**

Assumption : On the first day, our app initially has 150 users and thereby gains 25% for 3 months and 100% a year.

Timeline	Users	Personal Detail Storage (MB)	Post Storage (MB)	Est.Total Storage (GB)	Est. Cost ($/ month)
1st day	150	900	1650	1.8	0
End of 1st week	160	960	1760	1.92	0.04416
End of 3rd week	178	1068	1958	2.136	0.049128
End of 1st month	188	1128	2068	2.256	0.051888
End of 2nd month	235	1410	2585	2.82	0.06486
End of 3rd month	294	1764	3234	3.528	0.081144
End of 4th month	588	3528	6468	7.056	0.162288
End of 5th month	1176	7056	12936	14.112	0.324576
End of 6th month	2352	14112	25872	28.224	0.649152
End of 7th month	4704	28224	51744	56.448	1.298304
End of 8th month	9408	56448	103488	112.896	2.596608
End of 1st Year	150528	903168	1655808	1806.336	41.545728
End of 2nd Year	301056	1806336	3311616	3612.672	83.091456

Computation and Network Analysis :

For networking we are assuming that our user base would not exceed its peak load for Vadodara City :

- Assuming maximum concurrent users to be **10% of total user base (30,000)**. All users might upload or view posts (download) at the same time.
- Hence **(max. Concurrent users)*11 MB** of data needs to be uploaded to or downloaded from the server in minimum **10 seconds**.
- Maximum data transfer per second = **(max. Concurrent users)*11/10 MB/s**
- Bandwidth required :- **Maximum data transfer per second * 8**

To handle peak load for our application we are using EC2 instances web server with following specifications:

- Operating system: Linux with SQL Server Web
- vCPUs : 8
- Memory (GiB): 16
- Network performance : Up to 5 Gigabit

Each of the instances could handle a peak load of 5 Gbps. EC2 provides an auto scaling feature based on the load on our application hence it would scale up and down accordingly.

Timeline	Users	Max. Concurrent Users	Peak data transfer (MB)	Bandwidth required (Gbps)	EC2 Instances needed	Cost
1st day	150	15	16.5	0.132	1	54
End of 1st week	160	16	17.6	0.1408	1	54
End of 3rd week	178	18	19.58	0.15664	1	54
End of 1st month	188	19	20.68	0.16544	1	54
End of 2nd month	235	24	25.85	0.2068	1	54
End of 3rd	294	29	32.34	0.25872	1	54

month						
End of 4th month	588	59	64.68	0.51744	1	54
End of 5th month	1176	118	129.36	1.03488	1	54
End of 6th month	2352	235	258.72	2.06976	1	54
End of 7th month	4704	470	517.44	4.13952	1	54
End of 8th month	9408	941	1034.88	8.27904	2	108
End of 1st Year	150528	15053	16558.08	132.46464	27	1458
End of 2nd Year	301056	30106	33116.16	264.92928	53	2862

Security Analysis :

We are considering following AWS services to provide security to our Cloud project :-

1. **AWS Firewall Manager :-**
   To create a particular set of firewall rules to:-
   - Limit the data uploading limit
   - Limiting the data types
   - Restricting the data content
   - Restricting the receiving data

2. **Amazon Cognito :-**
   Amazon cognito is a special service used to handle all the registrations and authentications. It automatically handles the registrations and signin and sign 16 outs. It has features like uploading to facebook, view on linkedin and sign in with Google. This makes scale authentication easy and feasible.

3. **Amazon Guard Duty :-**
   It is a service offered by AWS to detect any threats to our instances or data or accounts and take action immediately. It continuously handles all the security checks and takes action accordingly.
   It also :-
   - Helps in detecting fake accounts
   - Helps in deleting pirated media
   - Helps in preventing cyber attacks
   - Helps in preventing media leaking

AWS Guard duty :-

- First 500 million events / month $0.80 per 1 million events
- Next 4500 million events / month $0.40 per 1 million events
- Over 5000 million events / month $0.20 per 1 million events

This means that for the first 500 million events, we will be charged **0.80$ per 1 million events**. At the peak of the app, **3 lakh users** are expected to use the app, 17 it will trigger at most 3 lakh events per month. Under assumption, we will be charged **0.80$ for 10 lakh** events, which means **0.23$ per month**.

AWS Cognito :-

- The first 50,000 registrations are free, the next 1 lakh registrations will be charged 0.0055$ per registration.

On estimation, total 3 lakh registrations will cost :-

50,000 * 0$ = **0$**

2,50,000 * 0.0055 = **1375$**

- Therefore it would cost **1375$** for the complete registration of all users.
- For the first 8 months the cognito cost would be 0 as users < 50000
- For the next 4 months, average cost = 1375$ /4 = **343.75 $**.

AWS Firewall manager :-

It charges 100 $ per policy per region per month. We create only one policy which limits data uploading capacity and type of data. Since we only use 1 region, we would be charged 100$ for the policy, per month.

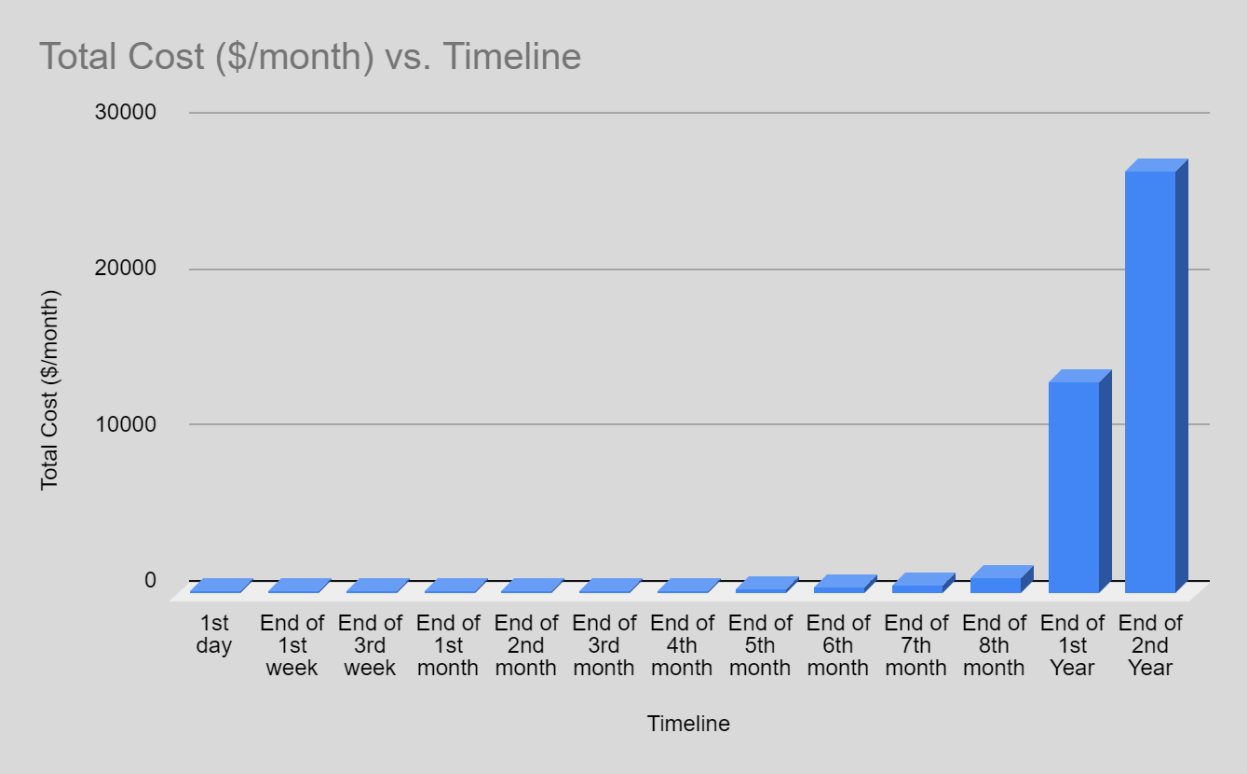
Complete Security cost :-

For the first 8 months The security would cost us :- 0.23 + 100 + 0 = **100.23 $ per month.**

For the first 4 months The security would cost us :- 0.23 + 100 + 343.75.5$ = **443.98 $ per month.**

Same follows for 2nd year.

Financial Analysis :



Timeline	Users	Est. Storage Cost ($/month)	Network & Compute Cost ($/month)	Security cost ($/month)	Total Cost ($/month)
1st day	150	0	0	443	443
End of 1st week	160	12.8	54	443	509.8
End of 3rd week	178	14.24	54	443	511.24
End of 1st month	188	15.04	54	443	512.04
End of 2nd month	235	18.8	54	443	515.8
End of 3rd	294	23.52	54	443	520.52

month					
End of 4th month	588	47.04	54	443	544.04
End of 5th month	1176	94.08	54	443	591.08
End of 6th month	2352	188.16	54	443	685.16
End of 7th month	4704	376.32	54	443	873.32
End of 8th month	9408	752.64	108	443	1303.64
End of 1st Year	150528	12042.24	1458	443	13943.24
End of 2nd Year	301056	24084.48	2862	443	27389.48

Growth Consideration :

Cloud Networking and Security:

It gives users access to networking resources through a centralized third-party provider ting to Wide Area Network (WAN) or other internet-based technology and helps to distribute content quickly operating inter-connected servers. This involves connect and securely. For security purpose:

1. Strategy and policy for ownership and accountability (internal/external) of cloud security risks.
2. Leverage robust identity and management and authentication processes to ensure only authorized users have access to the cloud environment applications and data.
3. Never allow the use of shared passwords.
4. Continual security and user activity monitoring across the organization.
5. Be aware of data backup. Retention and recovery policies for cloud vendors.

Assumptions :

1. For now, we are only considering cricket and football
2. Area of Vadodara = 220.3km2, we will be working with 250 km2.
3. The population of Vadodara is 22 lakhs.
4. Going by the statistics on a national level, 65% of the population is less than the age of 35 and around 50% are below 25.
5. 41% of the population is less than the age of 18.

In the first few months the user count is very less so a single EC2 instance would be sufficient, e.g. t2.micro/t2.nano. The one instance would run the entire web stack, for example, web app, database, management, etc. This would be good until our traffic is less. Therefore here we can scale vertically by increasing the capacity of your EC2 instance to address the growing demands of the application when the users grow up to 100. This means that we will scale by adding more power (CPU, RAM) to an existing machine. As the users grow we will hit the maximum power limit of the machine, c3.8xlarge.
Application tiers are likely to have different resource needs and those needs might grow at different rates. We have to seperate the tiers, so that we can compose each tier using the most appropriate instance type based on different resource needs.

Now, we will design our application so it can function in a distributed fashion. For example, it should be able to handle a request using any web server and produce the same user experience. Store application state independently so that subsequent requests do not need to be handled by the same server. Once the servers are stateless, we can scale by adding more instances to a tier and load balance incoming requests across EC2 instances using Elastic Load Balancing (ELB) as mentioned in the cloud architecture diagram.

After the users start increasing at the rate of 100% we need to scale horizontally. For this we will be adding machines in the pool of existing resources. When our users grow more than 1000, vertical scaling can't handle requests and horizontal scaling is required. Horizontal scalability can be achieved with the help of clustering, distributed file system, and load balancing. For this we are having the S3 buckets and Amazon EBS system.
The first is to make our application stateless on the server side as much as possible. Any time our application has to rely on server-side tracking of what it's doing at a given moment, that user session is tied inextricably to that particular server. If, on the other hand, all session-related specifics are stored browser-side, that session can be passed seamlessly across literally hundreds of servers. The ability to hand a single session (or thousands or millions of single sessions) across servers interchangeably is the very epitome of horizontal scaling.
The second goal to keep square in our sights is to develop our app with a service-oriented architecture. The more our app consists of self-contained but interacting logical blocks, the more we'll be able to scale each of those blocks
independently as your user load demands. The app will be developed with independent web, application, caching and database tiers. This is critical for realizing cost savings.