High Availability Explained

By
Reyaz Shaik

High Availability is in the Eye of the beholder

- CEO: we don't loose sales
- Sales: we can extend our offer basing on HA level
- Accounts managers: we don't upset our customers (that often)
- Developers: we can be proud – our services are working ;)
- System engineers: we can sleep well (and fsck, we

So How many 9's?

Monthly: 1 hour of outage means 100% - 0.13888 ~= **99.86112** of availability

Yearly: 1 hour of outage means 100% - 0.01142 ~= **99.98858** of availability

Availability	**Downtime (year)**	**Downtime (month)**
90% (“one nine”)	36.5 days	72 hours
95%	18.25 days	36 hours
97%	10.96 days	21.6 hours
98%	7.30 days	14.4 hours
99% (“two nines”)	3.65 days	7.2 hours
99.5%	1.83 days	3.6 hours
99.8%	17.52 hours	86.23 minutes
99.9% (“three nines”)	4.38 hours	21.56 minutes
99.99 (“four nines”)	52.56 minutes	4.32 minutes
99.999 (“five nines”)	**5.26 minutes**	**25.9 seconds**

High Availability

- In computing, the term availability is used to describe the **period of time** when a **service is available**, as well as the time required by a system to respond to a request made by a user.
- In information technology, **high availability** refers to a system or component that is continuously operational for a desirably long length of time. **Availability** can be measured relative to "**100% operational**" or "never failing."

Fault Tolerance

- A good way to think of it is that you have **two separate machines** that are **mirrored**. In the event that the main system has a **hardware failure**, the secondary system takes over and there is **zero downtime**.

How High Availability Works?

To create a highly available system, three characteristics should be present:

- Redundancy
- Monitoring
- Failover

How High Availability Works?

- **Redundancy**

In computing, *redundancy* means that there are **multiple components** that can perform the **same task**. This eliminates the single point of failure problem by allowing a second server to take over a task if the first one goes down or becomes disabled.

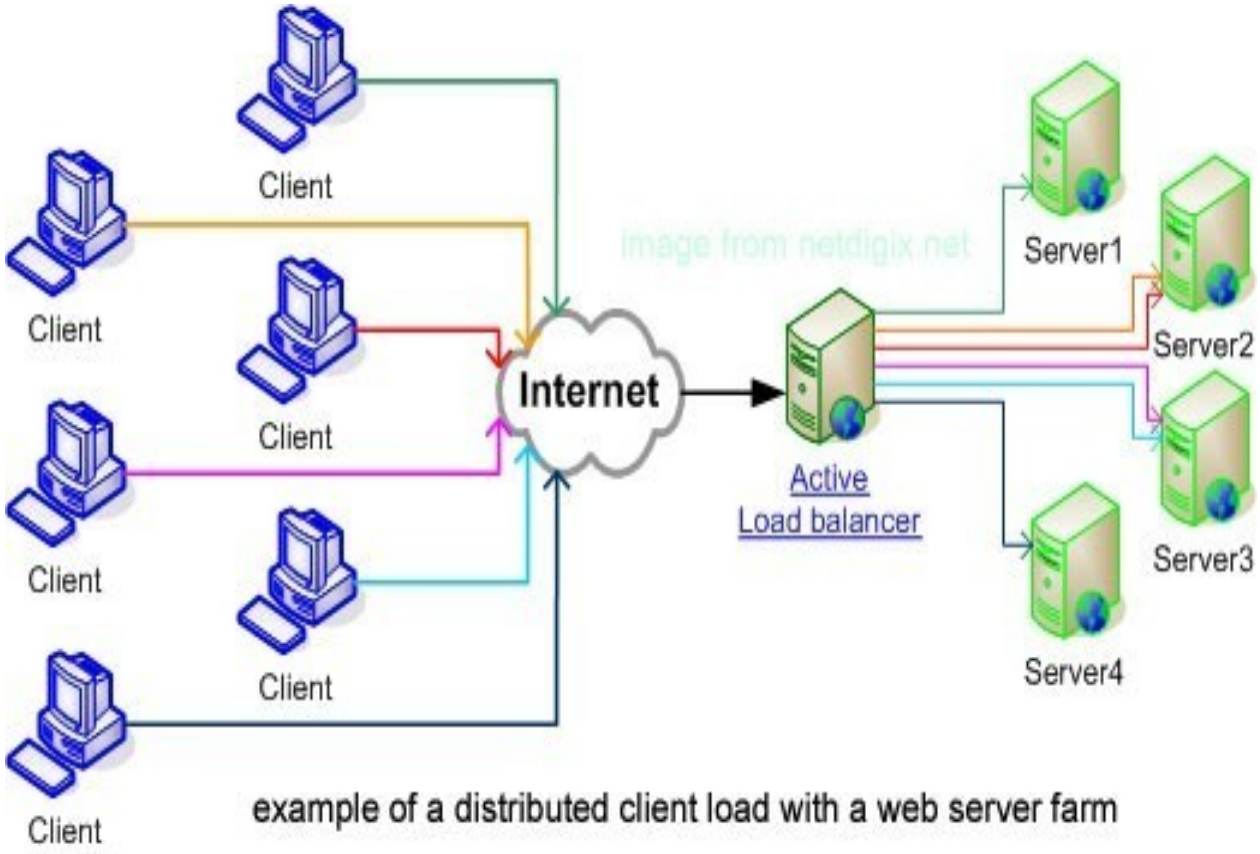
- **Monitoring**

In a highly available setup, the system needs to be able to monitor itself for failure. This means that there are regular checks to ensure that all components are working properly

- **Failover**

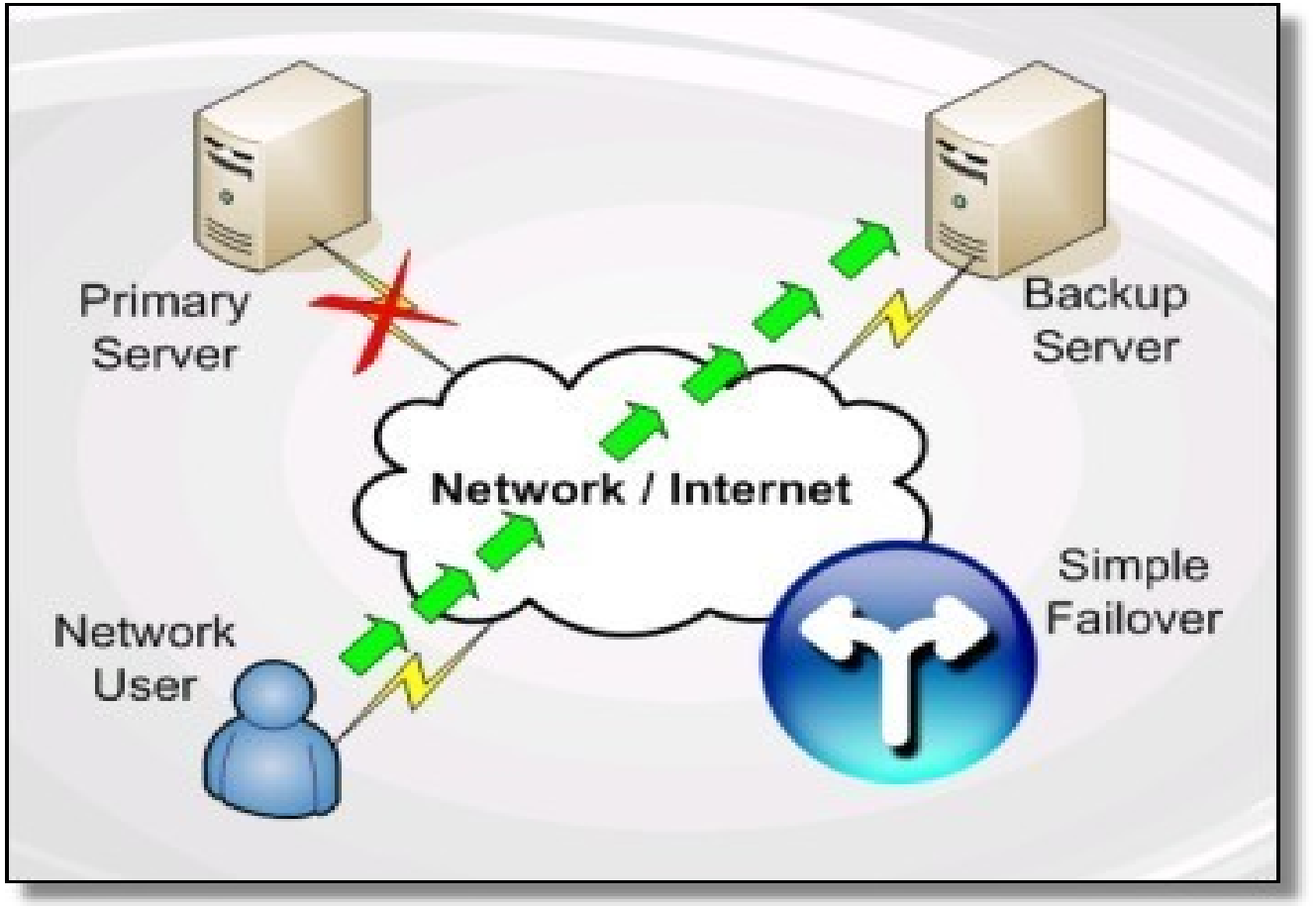
Failover is the process by which one node takes over the job of another in the event that one becomes disabled. This comes as a result of monitoring for failures by the system..

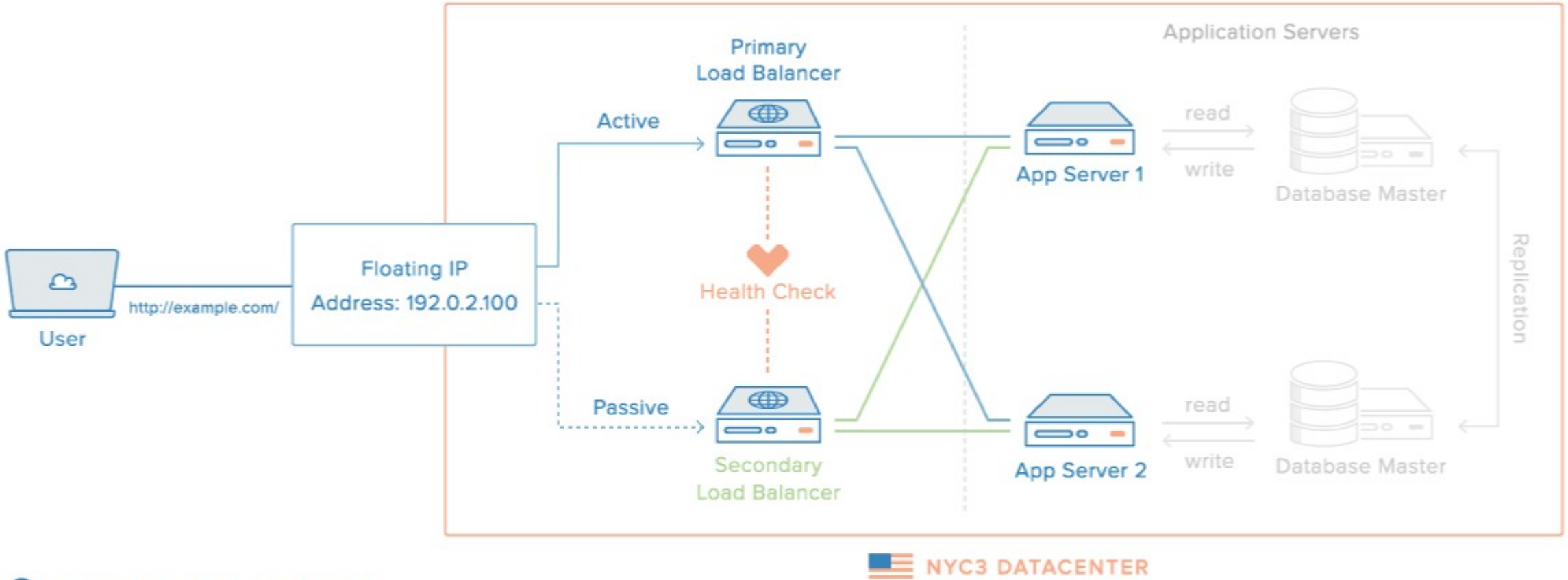
Load-balancing



example of a distributed client load with a web server farm

Failover:


Primary
Server
Backup
Server
Network / Internet
Simple
Failover
Network
User

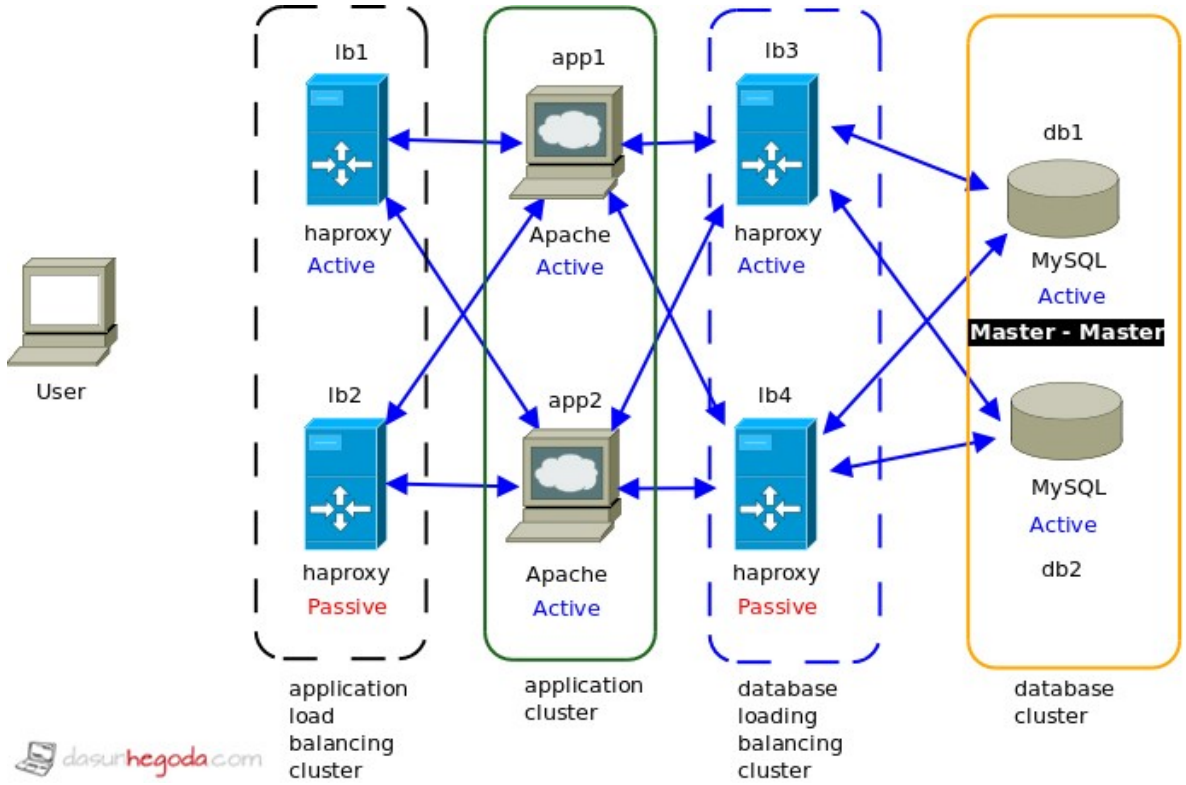


1. Active/Passive Cluster is healthy
2. Primary node fails
3. Floating IP is assigned to Secondary node

HA vs FT

	Windows Server Failover Clustering(HA)	Fault Tolerant Solution
Hardware Failure	P	P
OS Level Failure	P	
Application Failure	P	

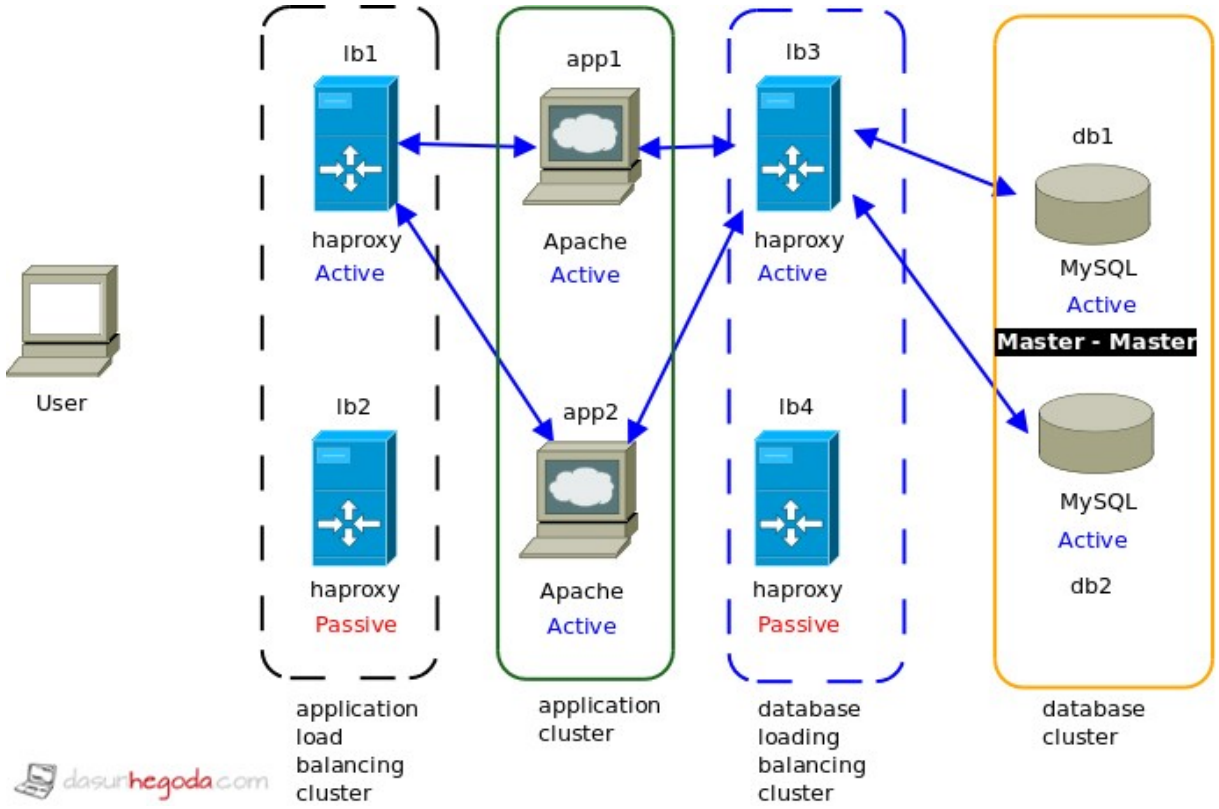
HA Example



Without beating around the bush let me elaborate this more.

- **lb1** & **lb2** are load balancers for the application servers. All together we call them application load balancing cluster.
- **app1** & **app2** are application servers. All together we call them application cluster.
- **lb3** & **lb4** are load balancers for the database servers. All together we call them database load balancing cluster.
- Then we have the database cluster at the end that are **db1** and **db2**. All together we call them database cluster.
- **Active** means particular component is accepting requests and **passive** means particular component is not accepting requests but when the active component is down passive component will take over and will start accepting requests.

HA Example

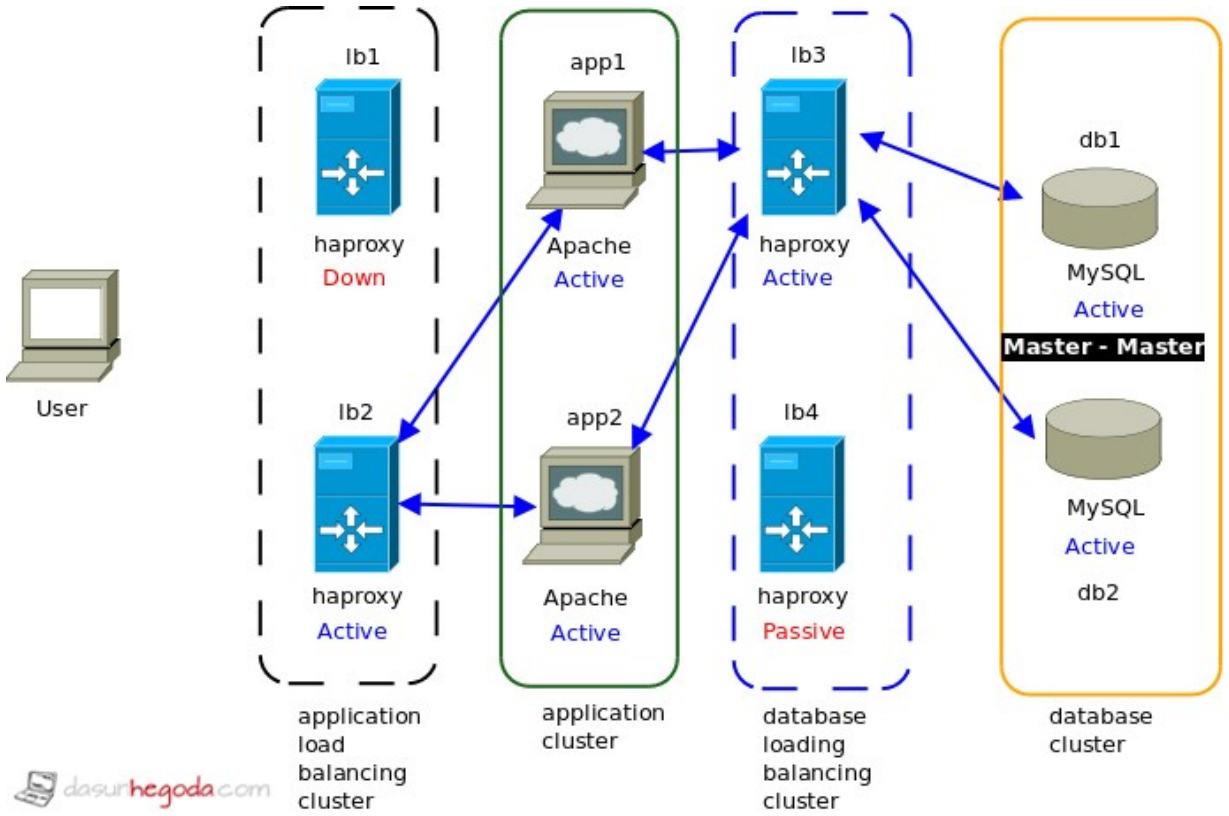


On a happy day user will make a request.

- The request will be accepted by lb1 and depending on the load balancing algorithm in lb1 request will be passed to the app1 or app2.
- From there the request will be handed over to the lb3 by app1 or app2. lb3 will communicate with the database.
- The response will follow the same path as the request's path.

Okay now let's get ready for some action because you can't expect a happy day everyday.

What if the lb1 is down

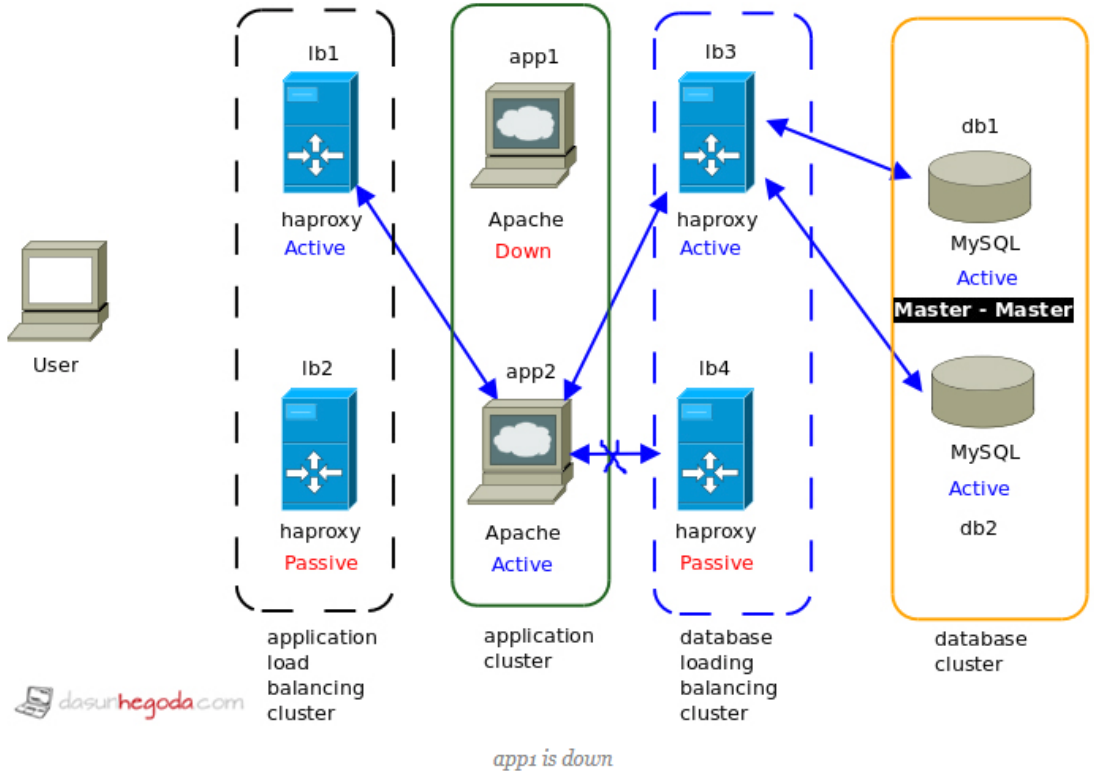


Here even though lb1 is down lb2 has taken over and now lb2 is in active status.

System is functional even though one load balancer has failed.

Best part is here users will not experience any downtime of the system due to the failure of lb1.

Let’s assume app1 is down

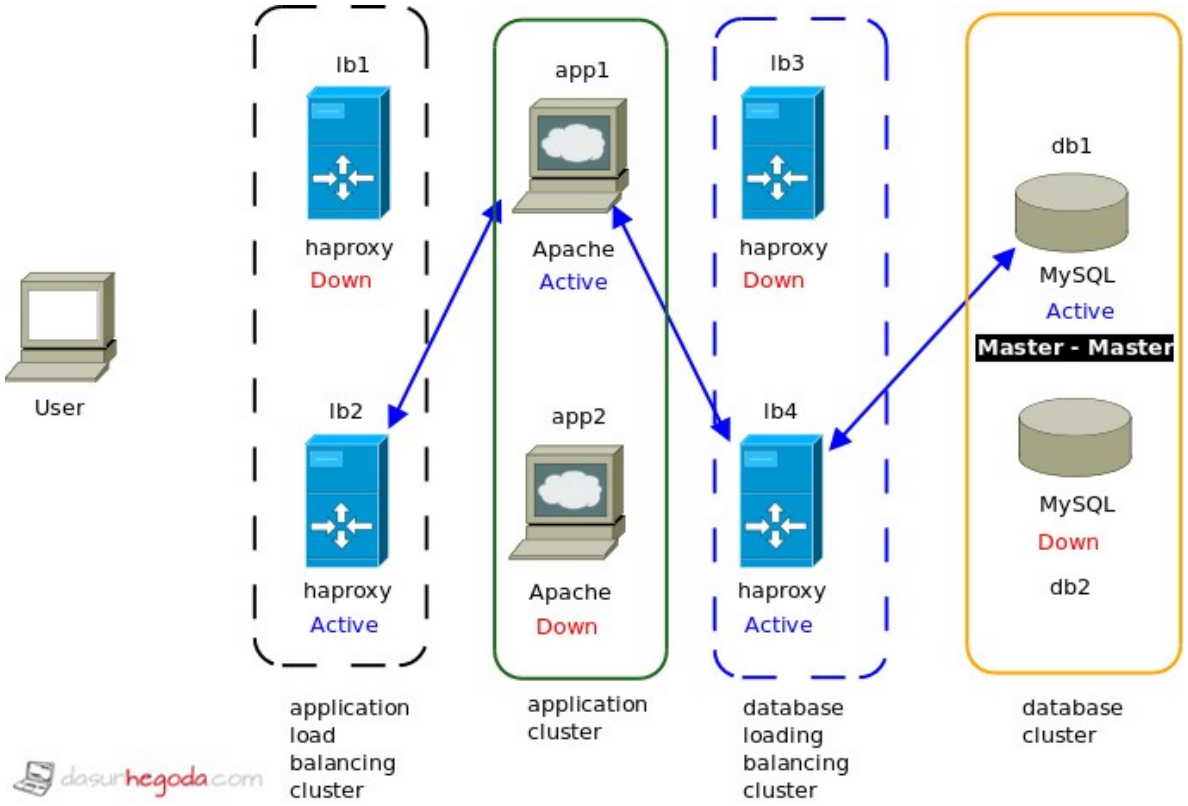


app1 is down

Now you can see even though one application sever went unavailable system is functional without an issue.

Just like the previous example.

Now let’s take the worst case scenario where 4 components are down

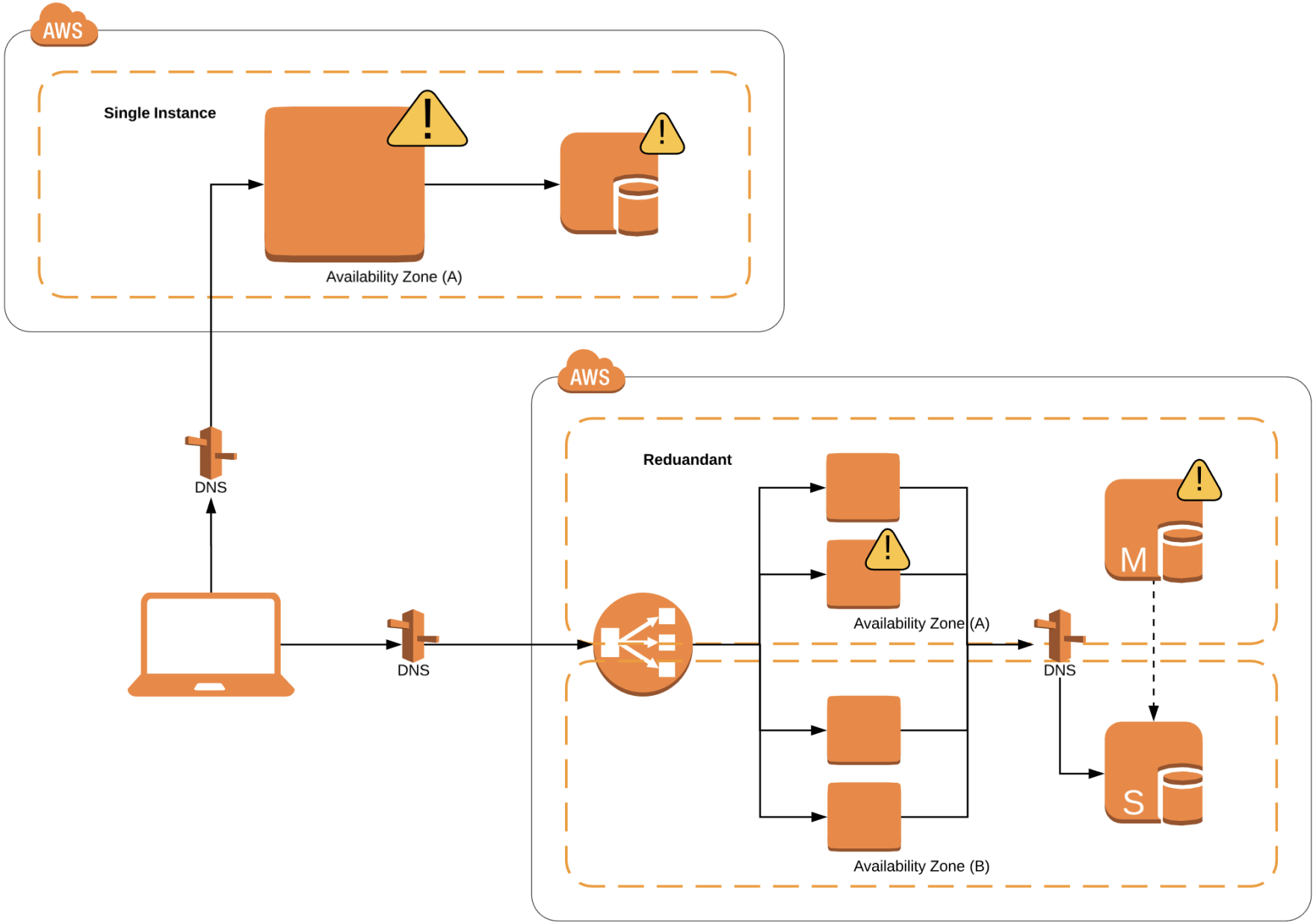


Oh! Nooohhhh!!!

Here a single component is unavailable from each cluster that means four components are not unavailable altogether.

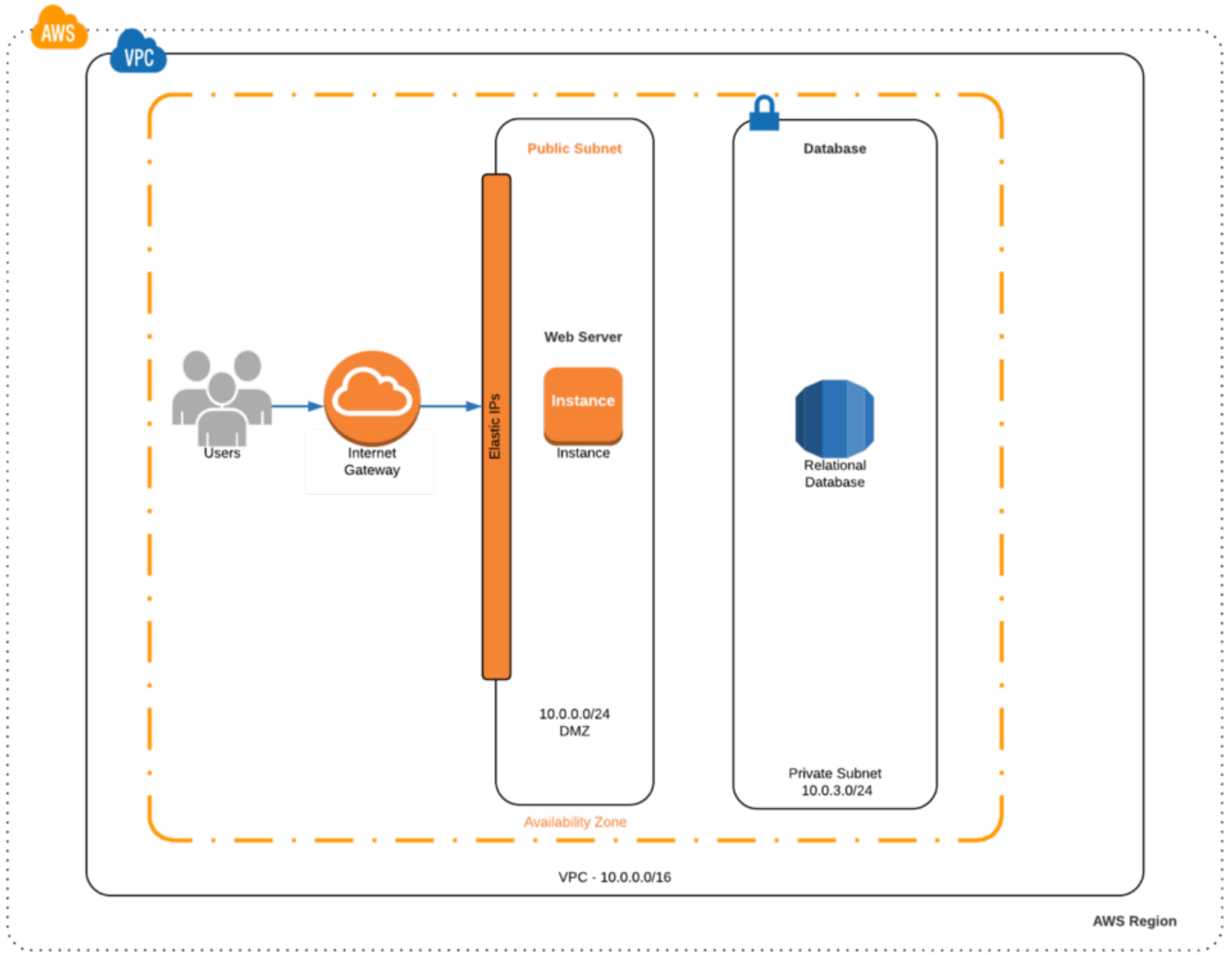
Will the application be able to function as on a happy day? Yes, a big yes.

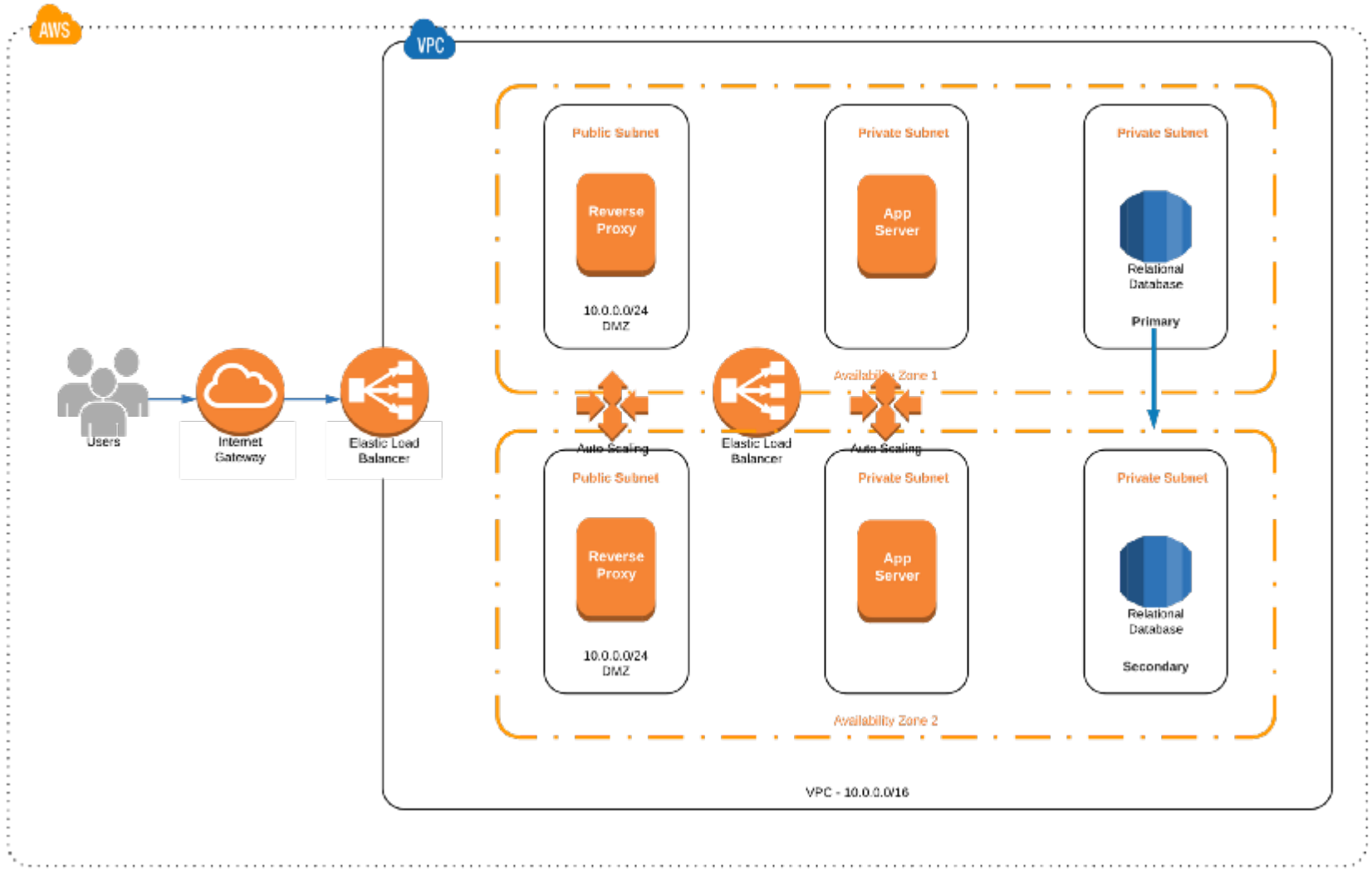
You can image the power of having high availability in your application. Even though a single component or multiple components are unavailable your application will be available for it’s intended parties without an issue. Perfect right!!!.

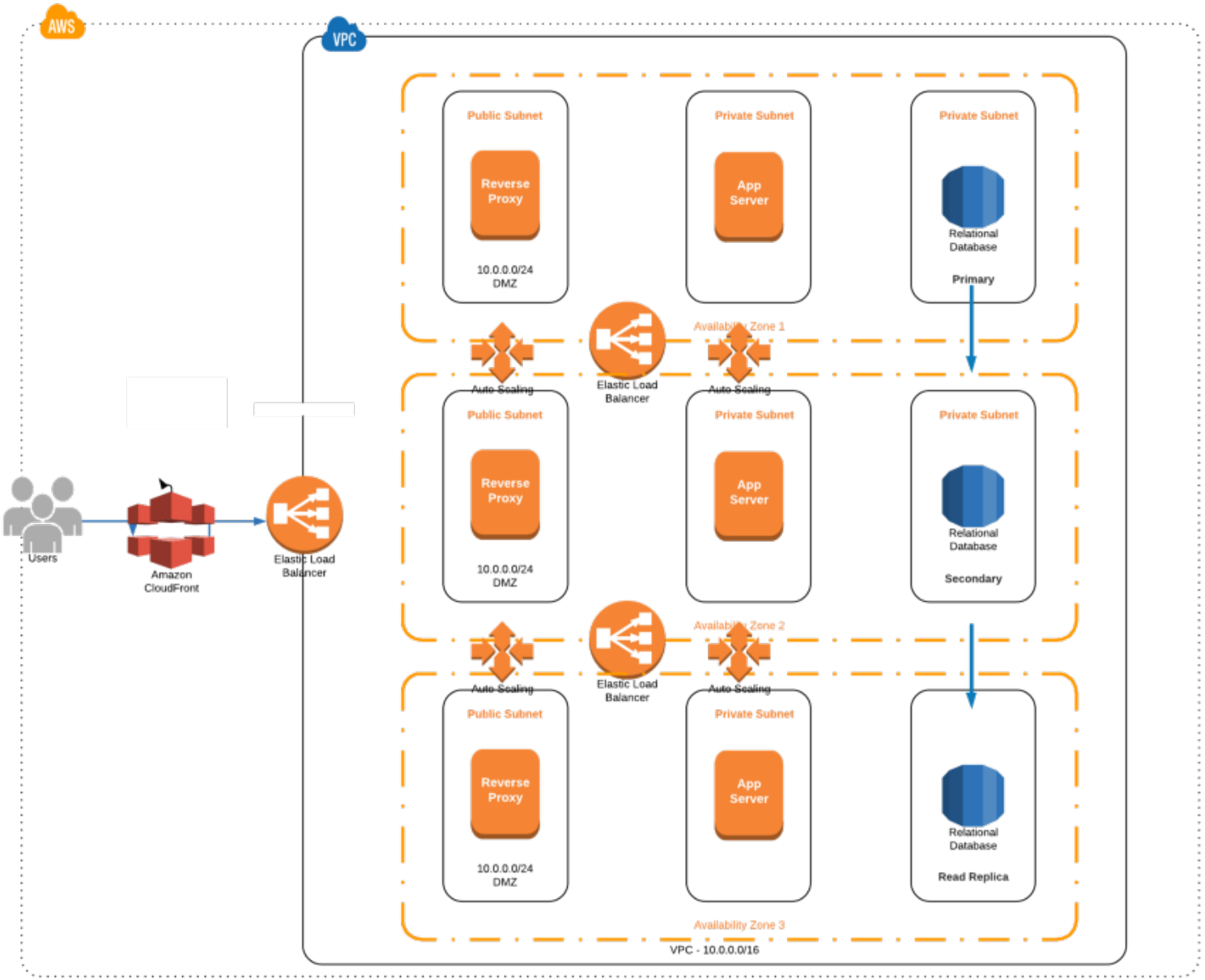

AWS
Single Instance
Availability Zone (A)
DNS
AWS
Reduandant
Availability Zone (A)
M
DNS
DNS
S
Availability Zone (B)

AWS 2 9S

Sathiya Shunmugasundaram | March 23, 2018




AWS
VPC
Public Subnet
Reverse Proxy
10.0.0.0/24
DMZ
Private Subnet
App Server
Private Subnet
Relational Database
Primary
Users
Internet Gateway
Elastic Load Balancer
Auto Scaling
Elastic Load Balancer
Auto Scaling
Availability Zone 1
Public Subnet
Reverse Proxy
10.0.0.0/24
DMZ
Private Subnet
App Server
Private Subnet
Relational Database
Secondary
Availability Zone 2
VPC - 10.0.0.0/16


AWS
VPC
Public Subnet
Reverse Proxy
10.0.0.0/24
DMZ
Private Subnet
App Server
Private Subnet
Relational Database
Primary
Availability Zone 1
Auto Scaling
Elastic Load Balancer
Auto Scaling
Public Subnet
Reverse Proxy
10.0.0.0/24
DMZ
Private Subnet
App Server
Private Subnet
Relational Database
Secondary
Users
Amazon CloudFront
Elastic Load Balancer
Availability Zone 2
Auto Scaling
Elastic Load Balancer
Auto Scaling
Public Subnet
Reverse Proxy
10.0.0.0/24
DMZ
Private Subnet
App Server
Relational Database
Read Replica
Availability Zone 3
VPC - 10.0.0.0/16


That's all Folks!