This is a high-level reference architecture for a web based services start-up company which hopes for a significant growth in near future. This proposed architecture is based on AWS solutions and inherently enables various enterprise capabilities to the overall solution. This architecture is designed to meet the client requirements provided in “AWS Architecture Design - Assignment #2” document.   The target audience for this document are the company CTO, Engineering Directors/Managers and any other authority who involves in technological purchase decisions.

`

Why AWS for a start-up company?

AWS Solutions elasticity nature enables the enterprise cloud capabilities to the solution.

Since growth of the company is not guaranteed and to keep the solution cost effective over the time, the solution should be easily re-sizeable to match the growth. AWS solutions readily brings these capabilities at every stage of the solution to the desk. AWS components are elastic in nature, that is they can auto re-size to a given load at run time. Say, using EC2 you can have more servers added to the solution if the load on servers increases and vice-versa. AWS pay-as-you pricing model keeps investments as well elastic. That is when you have more load, you use more resources then you pay for added resources and vice-versa. Choosing AWS enables the elasticity to the overall solution for any loads over any time.

AWS solution is a Scalability booster for handling the unpredicted traffic loads.

It’s very often heard that upcoming web based solutions suffer performance during peak loads. This is a result of resources not being scalable to meet the unexpected increase in load at real time. AWS auto scale provisioning of VM instances helps you add more web servers readily as needed to serve the increased loads. They can be taken offline or removed when the load returns to normal as needed. This capability boosts the scalability of overall solution to the ever changing traffic loads.

Easy manageability of the solution deployments and maintenance.

Being a start-up and web based solution, company may often need to deploy changes to the system. They would require a consistent close of production environment at every level of their Development, QA testing, Staging and Prod. Yet again the on-demand provisioning of resources in AWS improves turnaround time to deploy and test the changes at every stage to ensure stable release of product in the Production. This minimizes the hardware and maintenance costs involved at every stage of development cycle.

An AWS Cloud Architecture for a Web Based start-up company:

In this architecture, I am classifying the key components involved for easier manageability by the resources now and going forward. Below are the various layers at which a specific skillset is needed to manage and own the responsibility. Such a classification helps company to manage the access rights, security and ownership of the components by their staff as they grow organizationally.

Key Components Classification based on administration skill sets:

1. Network Tier:
   a. Manages external/internal network configurations and security via AWS Route 53, ELB, Multi-Availability Zones, Elastic IPs, Security Groups, etc.,.
   b. Responsible for ensuring access to the web site from around the world and load balancing them at every required levels say at Web server and App server.
2. Web Server Tier:
   a. Manages Web Servers via AWS EC2, AMI images, etc.,
   b. Responsible for configuring and maintaining web server instances to handle the web requests.
3. App Server Tier:
   a. Manages App Servers via AWS EC2, AMI images, etc.,
   b. Responsible for configuring and maintaining App server instances to handle the application requests.
4. Database Tier:
   a. Manages Database Servers and Data via AWS RDS, IOPS Volumes, etc.,
   b. Responsible for configuring and maintaining the backend database access, security, performance and availability. Also manage replication, backup,

An AWS Cloud Architecture for a Web Hosting

Route 53:
Provides External DNS services
http://www.mywebapp.com

Elastic Load Balancer ELB:
Spreads traffic to
Web Server Autoscaling groups

Web Server Security Group:
Acts as External Firewall

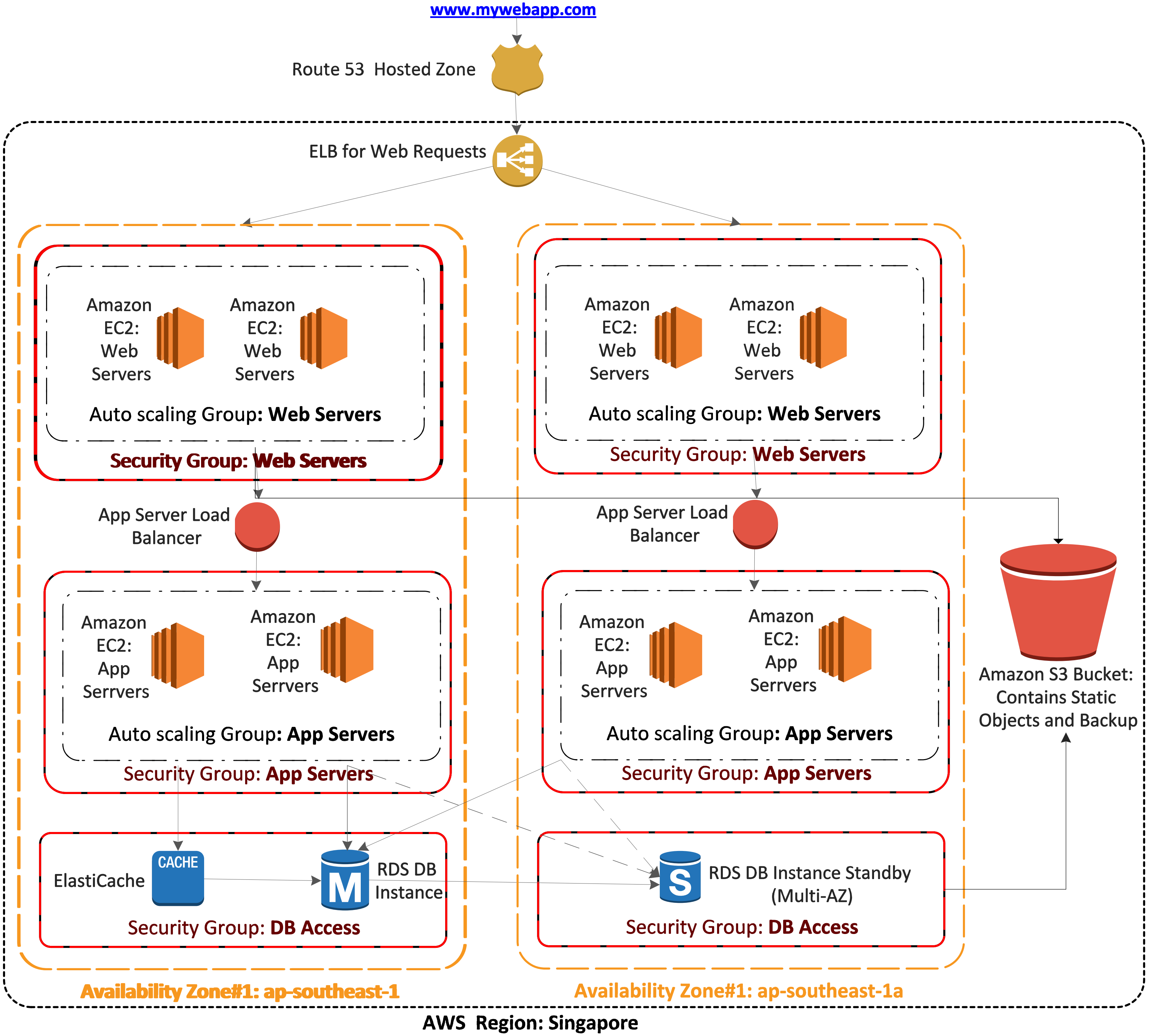
Auto Scaling Web Tier:
Group of EC2 instances
Handling Web requests

App Server load Balancer:
Can be a Software load
balancer to distribute
App server requests

App Server Security Group:
Acts as Backend Firewall
controlling access to
App servers

Auto Scaling App Tier:
Group of EC2 instances
hosting the applications

DB Tier:
RDS DB instance supporting
HA with multi-AZ deployment.
Standby remains covering DR
scenarios as well.


www.mywebapp.com
Route 53 Hosted Zone
ELB for Web Requests
Amazon EC2: Web Servers
Amazon EC2: Web Servers
Auto scaling Group: Web Servers
Security Group: Web Servers
Amazon EC2: Web Servers
Amazon EC2: Web Servers
Auto scaling Group: Web Servers
Security Group: Web Servers
App Server Load Balancer
App Server Load Balancer
Amazon EC2: App Servers
Amazon EC2: App Servers
Auto scaling Group: App Servers
Security Group: App Servers
Amazon EC2: App Servers
Amazon EC2: App Servers
Auto scaling Group: App Servers
Security Group: App Servers
Amazon S3 Bucket: Contains Static Objects and Backup
ElastiCache
CACHE
M
RDS DB Instance
Security Group: DB Access
S
RDS DB Instance Standby (Multi-AZ)
Security Group: DB Access
Availability Zone#1: ap-southeast-1
Availability Zone#1: ap-southeast-1a
AWS Region: Singapore

Key components of the architecture, challenges they address and added benefits they provide:

End User Facing:

No Downtime of the solution/web site: One of the key factor to gain more users and retain them is to ensure your solution is always available. To achieve such an high availability you need to make provisions to handle failures at every stage of your solution. Amazon brings up capabilities to handle failure at almost every stage of your web based solution. Like use of AWS Multi-Availability Zones (ap-southeast-1 and ap-southeast-1a in the diagram) so that users will be taken to a working instance within a chosen region (Singapore in the diagram). In the above architecture, users in Singapore region, will be accessing website www.mywebapp.com from both AZs if any of them fails users will be seamlessly redirected to the other zone. Once the failed components in AZ are fixed, users will be sent to both the AZs. This all happens without any user notifications and network team interventions to redirect the users.

Faster web page responses: As web users may origin for anywhere around the world, you need to ensure the faster loading of web pages/applications to the end users. The simplest logic to achieve this is to have your solution geographically deployed close to your user locations. Amazon has various regions that covers almost all major areas around the world. By hosting your web site on Amazon Route 53 hosted zones, you can seamlessly redirect your users to their nearest location to load the web site. With use of Amazon CloudFront and S3 you can make static file access much faster as well as add unparalleled IO performance to your RDS, web pages and application data.

Security of User Data: Since business users data is critical and needs to be protected from unauthorized access, security turns to be a key factor to choose any web service provider. AWS “Security Groups” allows you to configure the access rules at each level of the system. In the above architecture, at “Security Group: Web Servers” you can define rule that only https/http access to your web servers. By using “Security Group: App Servers” you can define rule such that only your web servers can access your app server on specific ports and protocols. You can setup similar abstraction for your Database as well by using “Security Group: DB Access”. In this way access to your resources is abstracted in general from outside and one has to meet all the security constraints defined at each level of the solution to access the underlying data. By use of HTTPS, you secure user data that is in transit over network.

Infrastructure Side:

Faster Server provisioning: Server resource availability is an often on-going requirement in every IT company which slows down the overall turnaround of the services to the clients. With AWS EC2 instances, you can spin up number of VM instances in moments. With AMI setup, you can even have all the needed customizations setup in one image and have instance created readily out from them. This enables you to deliver VM instances to meet the peak loads. Further to that using Auto Scaling creates capacity groups of servers that can grow or shrink on demand.

Automated distribution of load: AWS ELB does auto load balancing of the user web requests across the web servers group so that server are loaded equally and end user experience is improved. This all is taken care automatically for you.

Simplified IT NOC Operations: With the ability of auto scaling the resources, you may be worried about the reliability and operations of the overall system. To simply this job Amazon ships CloudWatch feature which reports health status monitoring data for Amazon EC2 instances. This monitoring data is indeed used by Auto Scaling to determine expanding or shrinking your EC2 instances. This assures the faster scaling of your resources at no loss of reliability.

Simplified Storage Administration : With AWS Elastic Block Store feature to host your storage volumes, you create a consistent volumes for your instances so that they start up with correct drives/storages across the system restarts. This feature enables applications or systems to reliably access the same raw data.

Simplified Database Maintenance: AWS supports almost all the major DBs including Oracle, MySQl, MS SQL. With RDS, your DB is automatically backed up with support for point in time recovery. The key feature of RDS deployment in Multi-AZ configuration is you can protect your DB against unexpected failures. In the above architecture, I've chosen "M" RDS DB instance which acts as master/primary. When you access you site from either of the multi-AZs, all your requests at DB level would be sent to the master DB instance in ap-southeast-1 zone. You can create a copy of your master DB and configure it in other AZ ap-southeast-1a as a Stand by DB. This will auto failover your solution to the stand by DB in your ap-southeast-1a zone should there be any downtime with your master DB instance. Once master DB comes online you can switch over from Stand by to Master DB. Further to that by use of EBS for placing DB files, you gain high performance as well as persistence DB files even when your DB host fails.

Conclusion: With all of the above AWS features and capabilities the proposed architecture takes the start-up company to the cloud with a cost-effective, highly-scalable and fault-tolerant infrastructure. You may want to just review the pricing of the AWS components on pay-as-you-use mode to make a call on migration to AWS.