Cloud Computing project Report

An online business can use load balancing and auto-scaling to support unexpected usage peaks and save money when usage is lower.

Amazon web services (AWS) provide elastic load balancing to automatically distribute incoming web traffic across multiple Amazon EC2 instances.

If one instance is getting too much traffic, some of it can be rerouted to another instances for processing. With load balancing we can add or remove Amazon EC2 instances as when needed without disrupting the overall flow of information.

The main features of load balancing is distribution

- Load balancing can automatically re-route to other running instances or equally distributes to all the their available instances.
- Node failures If one instance fails, it reroutes the traffic to other instances.

Managing applications using load balancing has a lots of benefits. It minimizes the risk of overloading one single instance and provides health monitoring for Amazon EC2 instances. Health check monitoring helps monitoring the traffic, handles incoming requests and checks continually to make sure that the load-balancing running between the instances function properly.

We can create and manage security groups with the load balancing to provide additional security and networking options. We can also set up load balancer in single availability zones or multiple availability zones to load balance incoming application traffic across Amazon EC2 instances. An availability zone is like a datacenter, which contains collection of computers in the physical locations, which will protect our application with the single location failure. We can access elastic load balancer through AWS management console, which is simple and intuitive web interface.

Auto scaling is one of the important services to build highly available applications. It eliminates the single point of failure by distributing traffic across availability zones. It can be self-healing by launching new instances, if any of the instances failed. We need to enable load balancer before configuring auto scaling.

We need to launch instances even for auto scaling. When usage is lower, we can switch off the server or instances to save money. Auto scaling can automatically increase the number of instances during high availability and in demand to performance maintenance. Auto Scaling is useful for both to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

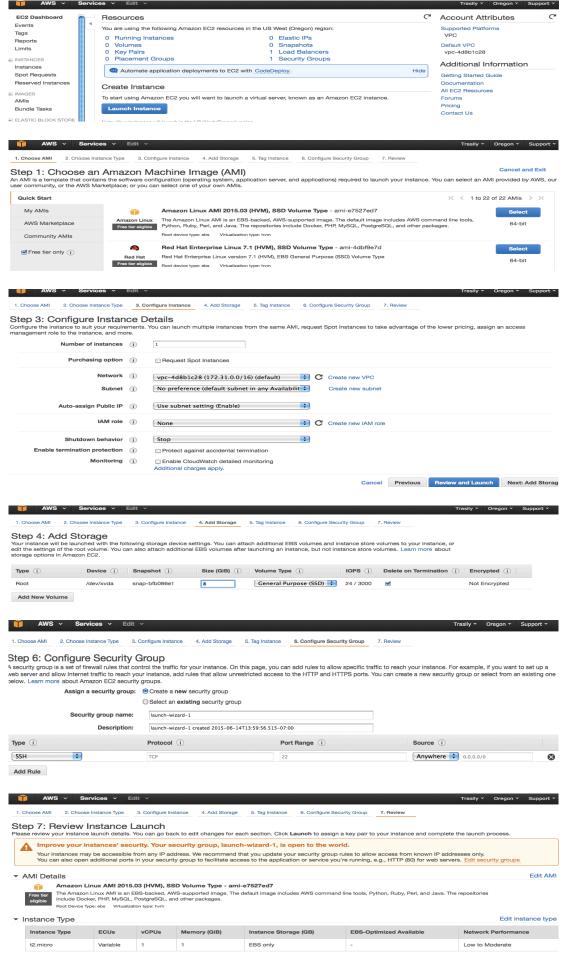
Load Balancing – For example: In a large organization, there are 5 servers and the network is equally distributing between all the servers. If one server fails, it automatically reroutes to the other four servers using dynamic load balancing.

Auto scaling – For example: In a flower shop, the web orders will be high during some occasion. But other normal days, orders will be less. During peak hours, we never know when we will get large number of orders. So we have to be ready with availability of servers using auto scaling. Auto scaling will automatically turn off the servers during less usage to avoid high cost. Also in online shopping, there will be high load on the servers during sale, daytime, weekends, etc. Auto scaling will automatically minimize the number of servers during low traffic (e.g. early morning).

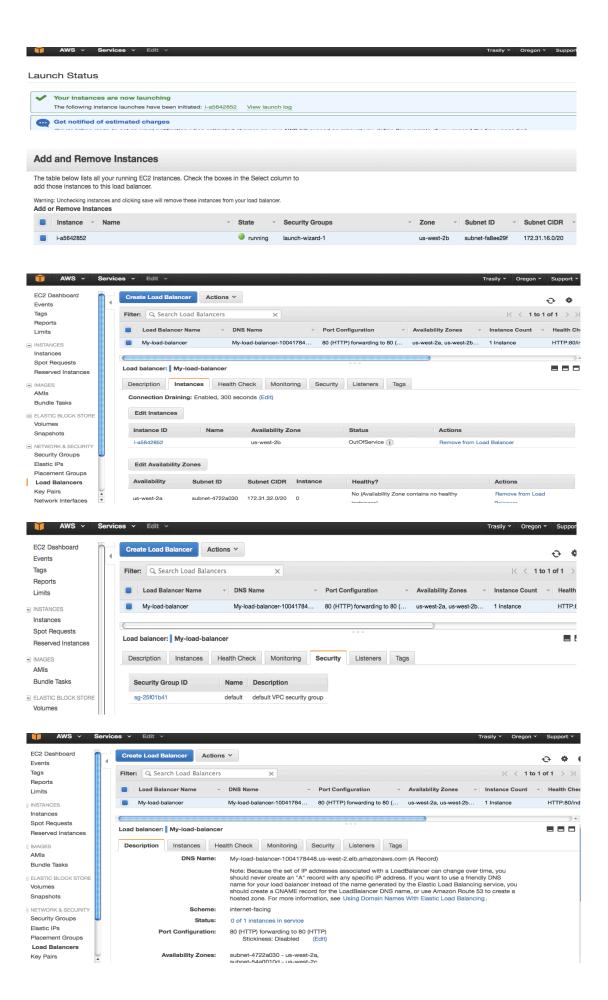
First we need to create an AWS account. We can see how to launch Instance, create load balancer, configure health checks, assign security groups, register, review settings, create load balancer and create auto scaling.

Creating Load balancer: Trasily * N. California * Support * EC2 Dashboard Resources Account Attributes You are using the following Amazon EC2 resources in the US West (N. California) region: Supported Platfo 0 Running Instances 0 Elastic IPs Default VPC Key Pairs
 Placement Groups Load Balancers
 Security Groups vpc-34b76b51 Instances Spot Requests Additional Information Getting Started Guide Reserved Instances Create Instance All EC2 Resources AMIs To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance Bundle Tasks AWS V Services V Edit V 1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EQ2 Instances 6. Add Tags 7. Revie Step 1: Define Load Balancer **Basic Configuration** This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You walso need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80. Load Balancer name: My-load-balance Create LB Inside: My Default VPC (172.31.0.0/16) Create an internal load balancer: (what's this?) Enable advanced VPC configuration: Listener Configuration: Load Balancer Port • HTTP HTTP -80 80 Add 1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review Step 2: Assign Security Groups You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time. Assign a security group: Oreate a new security group ● Select an existing security group Filter VPC security groups Security Group ID Name Description Actions sq-25f01b41 default VPC security group default Copy to new AWS - Services - Edit -1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EG2 Instances 6. Add Tags 7. Review Step 4: Configure Health Check Over load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatic removed from the load balancer. Customize the health check to meet your specific needs. Ping Protocol HTTP Ping Port 80 Ping Path /index.html **Advanced Details** Response Timeout (i) Health Check Interval i 30 Unhealthy Threshold (i) 2 ÷
Healthy Threshold (i) 10 ÷ 1. Define Load Balancer 2. Assign Security Groups 3. Configure Security Settings 4. Configure Health Check 5. Add EC2 Instances 6. Add Tags 7. Review Step 7: Review ▼ Define Load Balancer Edit load balancer definition Load Balancer name: My-load-balancer
Scheme: internet-facing
Port Configuration: 80 (HTTP) forwarding to 80 (HTTP) ▼ Configure Health Check Edit health check Ping Target: HTTP:80/index.html
Timeout: 5 seconds
Interval: 30 seconds
Unhealthy Threshold: 2
Healthy Threshold: 10 ▼ Add EC2 Instances Cross-Zone Load Balancing: Enabled
Connection Draining: Enabled, 300 seconds
Instances: Load Balancer Creation Status Successfully created load balance Load balancer My-load-balancer was successfully created Note: It may take a few minutes for your instances to become active in the new load balancer

Launch Instances and add to load balancer:



Security Groups Edit security around



Create Autoscaling Group and add instances:

