05 Networking and Elasticity

July 2, 2020

1 Networking

Networks reliably carry loads of data around the globe allowing for the delivery of content and applications with high availability

Cloud Networking Includes: - network architecture - network connectivity - application delivery - global performance - delivery

1.1 Network Connectivity

Include services that are reliable and cost-effective and that route users to internet applications

Every computer has an IP adress. Now most people do not remember the IP adress of google (74.125.21.147) so they just use `google.com`

1.1.1 DNS

When someone enters `google.com`, there is a DNS or Domain Name System that asks a domain authority which IP maps to `google.com`. The domain authority is the service name `google.com` was registered with. You are then routed to that IP adress

2 Route 53

- Route 53 is AWS cloud DNS
- It is a reliable and scalable service
- allows you to register a domain name
- routes internet traffic to the resources
- checks the health of your resources
- allows you to route users based on user's geographic location

3 Cloud Elasticity

One of the main benefits of the cloud is that it allows you to stop guessing about capacity when you need to run your applications. Elasticity scales your services automatically

4 EC2 Auto Scaling

EC2 Auto Scaling is a service that monitors your EC2 instance and automatically adjucts by adding or removing EC2 instances based on conditions you define in order to maintain application

availability and provide peak performance to your users

4.1 Features

- Automatically scale in and out based on needs
- Include automatically with Amazon EC2
- Automate how your Amazon EC2 instances are managed

4.2 Integration

EC2 autoscaling is integrated with amazons messaging services called Simple Notification Service (SNS) to alert you when an EC2 instance is launched or teminated

5 Additional Services

There is also AWS Auto Scaling which allows you to scale other services such as DynamoDB

6 Elastic Load Balancing

Elastic Load Balancing automatically distributes incoming application traffic across multiple services

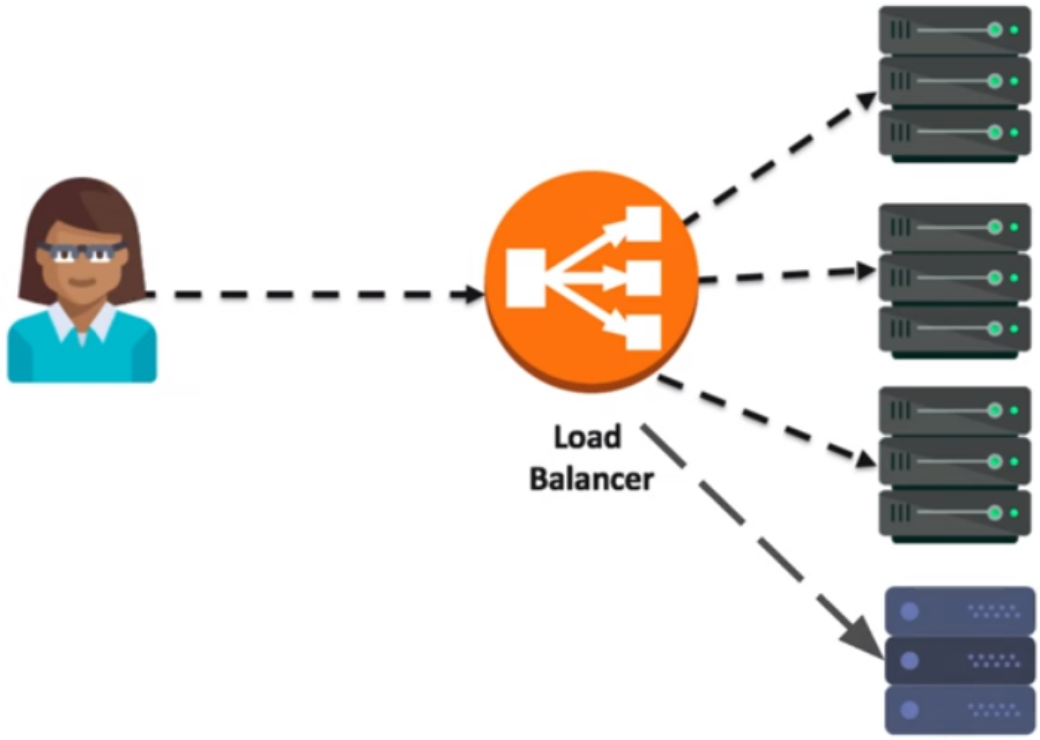
Elastic Load Balancer is a service that: - Balances load between two or more servers - Stands in front of a web server - Provides redundancy and performance - Elastic load balancing works with EC2 instances, containers, IP adresses and Lambda functions - You can configure Amazon Ec2 instances to only accept traffic from a load balancer

6.1 Redundancy

Means if you lose a server, the load balancer will request from the other working servers

6.2 Performance

Good performance means that if the balancer is facing bottlenecks, it will add other servers



7 EC2 Auto Scaling

1. **Create a Launch Configuration**

   - On the AWS Management Console page, type `EC2` in the `Find Services` box and then select `EC2`.
   - Scroll down to the `Auto Scaling` section on the left-hand menu and click `Auto Scaling Groups`.
   - Click the `Create Auto Scaling group` button.
   - Review the steps and click on `Get started`.
   - Create a launch configuration by first selecting an Amazon Machine Image (AMI). In the row for `Amazon Linux 2 AMI (HVM), SSD Volume Type`, click the `Select` button.

   Note: An AMI is a template for an instance that indicates the operating system, an application server, and applications.

   - Confirm that `t2.micro` is selected.
   - Click `Next: Configure details`.
   - Enter a name of your choosing in the `Name` field.
   - Expand the `Advanced Details` section.
   - Next to `IP Address Type`, click on `Assign a public IP address to every instance.`
   - Click `Next: Add Storage`. Review the screen.
   - Click `Next: Configure Security Group`.
   - Ensure `Create a new security group` is selected.
   - Click `Review`.
   - Click on `Create launch configuration`.
   - On the `Select an existing key pair or create a new key pair`, select `Create a new key pair`, enter a key pair name in the `Key pair name` field, and click `Download Key Pair`.
   - Click on `Create launch configuration`.

2. **Create an Auto Scaling Group**

   - On the `Create Auto Scaling Group` page, enter a group name of your choosing in the `Group name` field, ensure the `Group size` is set to `1`, for `Network` leave the default value. If no default value is shown, click on `Create new VPC`, and select the first `Subnet` by clicking in the `Subnet` field.
   - Click `Next: Configure scaling policies`.
   - Ensure that `Keep this group at its initial size` is selected.
   - Click `Review`.
   - Review the selected options and click `Create Auto Scaling group`.
   - Click `Close`.

3. **Verify your Auto Scaling Group**

   - Verify that the group has launched your EC2 instance by first ensuring the auto scaling group you just created is selected and examining the `Details` tab shown on the bottom of the screen.
   - Click the `Activity History` tab. The status of your instance should be `Successful`, which means the instance is launched.
   - Click on the `Instances` tab. Notice the `Lifecycle` column states `InService`.

4. **Test Auto Scaling**

   - Click on the `Instances` tab.
   - Under the `Instance ID` column, click on the blue Instance ID link.
   - You will be taken to the Amazon EC2 console Instances page.
   - Your instance should be selected.
   - Click the `Actions` button, scroll down to `Instance State`, and select `Terminate`. Then select `Yes, Terminate`.
   - In the left-hand navigation pane, click `Auto Scaling Groups`.
   - Click the `Instances` tab. You will eventually see a new instance appear. If the new instance doesn't appear, click refresh occasionally to update the list.
   - Click on the `Activity History` tab to review the history for the Instance.

5. Delete Auto Scaling Resources

   - At the top of the screen, click the `Actions` button next to the `Create Auto Scaling group`.
   - Click the `Delete` option.