Servers Introduction

*Potential Servers include EC2, Virtualization, User data, Metadata and Networking for EC2.*

**Amazon S3** can store Static Objects such as *Archives, Backup, Logs, Metadata, Cold Storage, Static Websites, Images & Videos.*

Hosting a Static Website on Amazon S3

**Amazon S3** Stores HTML/CSS/JavaScript Pages of the Static Website, Automatically-Assigning an Endpoint URL (*Used to Access Website*).

|  |  |  |
| --- | --- | --- |
| **Advantages to Hosting on S3** | | |
| *Don’t Need to Manage Infrastructure.* | *Automatically Scales to Handle Increased Traffic.* | *Provides Low-Cost Option for Static Website Hosting.* |

|  |  |
| --- | --- |
| **Hosting on S3 Use Cases** | |
| *Websites that don’t contain Server-Side Script.* | *Websites that Change Infrequently.* |
| *Websites that need Scale for Occasional Traffic Increases* | *Customers who don’t want to manage Infrastructure.* |

***There are Regional Differences to Bucket URLs defined in the following:***

* US Regions have (-) in Bucket URLs.
* EU Regions have (.) in Bucket URLs.

*The Following Steps dictate how you’d Start Hosting a Static Website on Amazon S3:*

1. Create an S3 Bucket to Store Website Content by.
2. Configure the S3 Bucket Enabling Website Hosting & Grant Public Read Permissions for Bucket Contents and include an Index Document (*index.html*).
3. Upload Website Content to Bucket using the AWS Management Console/AWS CLI.
4. You can now access the website through the Endpoint URL assigned to it my amazon.

***Custom Domain Names are possible through the following steps:***

1. Client enters the Custom URL.
2. Route 53 Maps Client Custom URL to Website Endpoint URL.
3. Client Connects to Website Endpoint URL.

Computing on AWS

**Steps to Launching an EC2 Instance**

1. Choose an Amazon Machine Image (AMI) template that EC2 uses to launch instances.
2. Choose an Instance Type, Specifying RAM, Cores, CPU, Storage and Networking Capacity.
3. Specify a Key Pair if connecting through SSH/RDP (Public + Private Key).
4. Specify a Network Type of either EC2 or VPC, Specify a DNS if you’d like.
5. Assign a Security Group which is Firewall Rules that Control Traffic to/from the instance and which ports.
6. Specify the Storage Options (Ephemeral/EBS Volume).
7. Assign Roles/Permissions with your Instance Profile through AWS IAM if needed.
8. Decide on Userdata(*Instance ID/Type/AMI ID/Public Host-name*) when an Instance Launches.

Managing AWS Instances

A diagram of a diagram

Description automatically generatedA screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| **Instance State Characteristics** | |
| **Billing** | Billed for Instance Use only in the following cases: Instance is in Running State or Stopping State due to Stop-Hibernate Action. |
| **Data Persistence** | Instance Store Volume Data is preserved while instance is running/rebooted and erases all other states. |
| Attached EBS Volume Data is preserved in instance states with exceptions:   * Root Device Volume deleted when instance terminated by default. * If Volume’s DeleteOnTermination attribute is active, it happens. |
| **IP Address** | IPv4 Changes every time an Instanced is Launched/Started. |
| Elastic IP Address remains associated with Instance until it’s Terminated. |

AWS Elastic Beanstalk

PaaS Facilitating Quick Deployment, Scaling & Application Management through Infrastructure Provisioning/Configuration, Load Balancing, Application Deployment, Automatic Scaling & Health Monitoring.

|  |  |  |
| --- | --- | --- |
| **Elastic Beanstalk Features** | | |
| Supports Web Applications written for Common Platforms. | Gives Control Over Key Runtime Config Options (i.e. – Instance Type, DB, EC2 Scaling). | No Charge, Only pau for Resources used by Underlying Service. |

|  |  |  |
| --- | --- | --- |
| **Elastic Beanstalk Advantages** | | |
| *Increases Developer Productivity.* | *Built-In Scalability* | *Reduced Management Complexity.* |

Elastic Load Balancing

**Scaling** is the Ability to Increase/Decrease Compute Capacity to meet Fluctuating Demand and can be done Manually/Automatically.

|  |  |
| --- | --- |
| **Auto-Scaling Benefits** | |
| *Fault Tolerance.* | *High Availability.* |
| *Performance.* | *Cost Optimization.* |

**Auto-Scaling** requires several components: *Route 53 connects to an Elastic Load Balancer, which connects to an EC2 Auto-Scaling Group which holds Multiple Instances.*

**Elastic Load Balancing** Automatically Distributes Incoming Traffic across Multiple Targets (IP Addresses, EC2 Instances, Containers, Monitoring Target Health routing only to Healthy Targets and Automatically Scales Load Balancer Capacity for Incoming Traffic.

|  |  |
| --- | --- |
| **Elastic Load Balancing Use Cases** | |
| Secure Access through Single Point. | Decouple Application Environment. |
| Provide High Availability & Fault Tolerance. | Increase Elasticity & Scalability. |

|  |  |  |
| --- | --- | --- |
| **Elastic Load Balancing Features** | | |
| *High Availability* | *Health Checks* | *Security Features* |
| *TLS Termination* | *Layer4/7 Load Balancing* | *Operational Monitoring* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Elastic Load Balancer Types** | | | |
| **Application** | **Gateway** | **Network** | **Classic** |
| Provide Advanced Load Balancing for Traffic (HTTP/S) | Load Balancing for Virtual Appliances & All Types of IP Traffic. | Load Balancing for TCP Traffic. | Load Balancing for HTTP/S, SSL/TLS and TCP Traffic on EC2-Classic. |
| Flexible Application Management | Virtual Appliance Management | Extreme Performance & Static App IP | Existing Apps on EC2-Classic. |
| Layer 7 | Layer3/4 | Layer 4 | Layer 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Application Load Balancer Features** | | | |
| Path-Based & Route-Based Hosting. | Native IPv6 Support. | Dynamic Ports. | Additional Supported Request Protocols. |
| Deletion Protection & Request Tracking. | Enhanced Metrics & Access Logs. | Targeted Health Checks. |  |
| **Gateway Load Balancer Features** | | | |
| Provides Level 3 Gateway & 4 Load Balancing | Passes all Layer 3 Traffic through 3RD Party Virtual App. | Supports IP Protocols. |  |
| Provides Enhanced Metrics & Access Logs. | Provides Targeted Health Checks. | Provides Deletion Protection & Request Tracking. |  |

ELB Load Balancers & Listeners

**Listeners** are Processes Defining Port/Protocol Load Balancer Listens on. Each LB needs at least 1 Listener to accept traffic, Up to 50 Listeners can Exist on a LB and Routing Rules are Defined on Listeners.

**Target Groups** contain Registered Targets Providing Support to Resources i.e. – *EC2/Container Instances*. One Target can have Several Target Group Registrations.

Amazon EC2 Auto-Scaling

*A Service that aids ensuring Application Availability through Automatically Launching/Terminated Instances based on User-Defined Scaling Options. There are different forms including:*

|  |  |
| --- | --- |
| **Auto-Scaling Forms** | |
| **Dynamic Scaling** | Scales Auto-Scaling Group Capacity as Traffic Changes occur. |
| **Target Tracking Scaling (*Dynamic*)** | Increase/Decrease Current Group Capacity based on Target Value for Specific Metric. |
| **Step Scaling (*Dynamic*)** | Increase/Decrease Current Group Capacity based on Set of Scaling Adjustments from Alarm Breach Size. |
| **Simple Scaling (*Dynamic*)** | Increase/Decrease Current Group Capacity based on Single Scaling Adjustment. |
| **Predictive Scaling** | Increases Auto-Scaling Group Capacity in advance of Daily/Weekly Traffic Flow Patterns, based on Forecast Load & Scheduled Minimum Capacity for Periodic Spikes in Conjunction with Dynamic Track. |

|  |  |
| --- | --- |
| **Auto-Scaling Jargon** | |
| **Capacity** | Min/Max Group Size you want for your Auto-Scaling Group, representing the Initial Capacity at the time of creation. |
| **Scaling In/Out** | Increase in CPU Utilization outside Desired Range causes the Group to Scale Out & Add Instances, the opposite for scaling in. |
| **Instance Health** | Health Status (*Healthy/Unhealthy*) Notification originates from sources (i.e. – *EC2, ELB, Custom Health Checks*). |
| **Termination Policy** | Determines which Instances are Terminated First during Scale-In Events. |
| **Launch Template** | Specifies Instance Config Info (i.e. – *AMI ID, Instance Type, Key Pair, Security Groups, etc*) |
| **Lifecycle Hooks** | Provide Opportunity for User Action before Scale Event Completion. |
| **Steady-State Group** | Recreates Instance Automatically if Unhealthy/ AZ Failes, stills have Potential Downtime while Instance Recycles. |

|  |  |  |
| --- | --- | --- |
| **Auto-Scaling Policy** | | |
| Amazon CloudWatch Alarms: *% of Load for Time Duration*. | Target-Tracking Policy: *Targeted/Expected CPU Utilisation Considered.* | Scheduled Actions: *Set Time/Date (Month/Week/Year).* |

***You can follow these steps to Auto-Scale based on a Schedule:***

1. Create Scheduled Action telling EC2 Auto-Scaling to perform at specified times.
2. Specify Start Time for when it takes effect & new max/min defined sizes.
3. Auto-Scaling Updates Group to match Capacity Values at specified times.

|  |  |  |
| --- | --- | --- |
| **Auto-Scaling Best Practices** | | |
| *CloudWatch: 1 minute Frequency Metric Data Collection.* | *EC2 Auto-Scaling: Turn on Auto-Scaling Group Metric.* | *EC2 Instances: Avoid Burstable Performance Instance Types.* |
| *Alarm Sustain Period: Configure Alarms for Maintained State Changes.* | *Cooldown Period: Suspend Scaling Activities Scaling In/Out for Simple Policy.* | *Instance Warm-Up Period: Specify Seconds for Newly-Launched Instance to warm.* |

Amazon Route 53

**Amazon Route 53** is a Scalable Domain Name System (DNS) Web Service which allows for the following features:

|  |  |  |
| --- | --- | --- |
| **Route 53 Features** | | |
| *Register/Transfer Domain Name.* | *Resolve Domain Names to IP Addresses.* | *Support High Availability & Lower Latency.* |
| *Connect to Infrastructure.* | *Distribute Traffic across Regions.* |  |

|  |  |
| --- | --- |
| **Routing Policies** | |
| **Simple Routing Policy** | Single Resource performs Given Function for Domain |
| **Weighted Routing Policy** | Routes Traffic to Multiple Resources in User-Specified Proportions. |
| **Latency Routing Policy** | Resources In Multiple AWS Regions to Region with Lowest Latency. |
| **Failover Routing Policy** | Configure Active-Passive Failover. |
| **Geolocation Routing Policy** | Route Traffic based on User Location. |
| **Geoproximity Routing Policy** | Route Traffic based on Resource Location. |
| **Multivalue Answer Routing Policy** | Route 53 Responds to DNS Queries with up to 8 Health Randomly-Selected Records. |
| **IP-Based Routing Policy** | Route Traffic based on User Location & IP Address of Traffic’s Origin. |