

# For Further Questions?



[WWW.ETHANS.CO.IN](http://WWW.ETHANS.CO.IN)

# What is Cloud Computing

Wikipedia defines cloud computing as:

**"Cloud computing is internet-based computing in which large groups of remote servers are networked to allow the centralized data storage, and online access to computer services or resources."**

The **National Institute of Standards and Technology (NIST)** gives the following definition of cloud computing:

**"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."**

# What is Cloud Computing

- Users should be able to provision and release resources on-demand
- The resources can be scaled up or down automatically, depending on the load
- The provisioned resources should be accessible over a network
- Cloud service providers should enable a pay-as-you-go model, where customers are charged based on the type and quantum of resources they consume

- **Infrastructure as a Service (IaaS)** provides users the capability to provision processing, storage, and network resources on demand.
- In **Platform as a Service (PaaS)**, the service provider makes certain core components, such as databases, queues, workflow engines, e-mails, and so on, which are available as services to the customer. The customer then leverages these components for building their own applications.
- In the **Software as a Service (SaaS)** model, typically, third-party providers using a subscription model provide end-user applications to their customers.
- Example IaaS, PaaS & SaaS

# Cloud Service Models

On premise	IaaS	PaaS	FaaS	SaaS
Functions	Functions	Functions	Functions	Functions
Application	Application	Application	Application	Application
Runtime	Runtime	Runtime	Runtime	Runtime
Operating system	Operating system	Operating system	Operating system	Operating system
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Networking	Networking	Networking	Networking	Networking
Storage	Storage	Storage	Storage	Storage
Hardware	Hardware	Hardware	Hardware	Hardware





- Region:
  - Is a geographical area
- Availability Zone(AZ):
  - Is a data centre
  - 2 or more within a region



- Edge Location
  - Is a CDN endpoint for Cloudfront
  - Used to cache files closer to a user's physical location







## Networking and Content Delivery


-  VPC - Virtual data centre in the cloud
-  CloudFront - CDN, used for caching
-  DirectConnect - Dedicated network connection to AWS
-  Route53 - DNS






## Compute

-  EC2 - Virtual machine in the cloud
-  ECS - Docker containers
-  Lambda - Serverless
-  Elastic Beanstalk - Web apps without infrastructure code





## Storage

-  S3 - Object storage
-  Glacier - Archive
-  Storage Gateway - Hybrid storage
-  EFS - File system




## Migration

-  Snowball - Petabyte scale data solution
-  DMS - Database migration <-> AWS, AWS <-> AWS
-  Server Migration (SMS) - Migrate on premise servers to AWS




## Database

-  RDS - relational
-  DynamoDB - NoSQL
-  ElastiCache - in-memory cache
-  Redshift - data warehouse





## Management Tools

-  Cloudwatch - Monitoring & alerting
-  Cloudformation - Infrastructure as code
-  CloudTrail - Audit all API calls
-  OpsWorks - Chef configuration management



## Management Tools continued

-  Config - Configuration resources rules
-  Trusted Advisor - Reduce costs and improve security
-  Service Catalog - Organization catalogue

## Security, Identity, and Compliance





-  IAM - access control
-  Inspector - agents, security
-  Certificate Manager - SSL/TLS certificates
-  Directory Service - directory store (Active Directory)

## Developer Tools

-  CodeCommit - Managed Git source control
-  Code Build - compiles source code, runs tests
-  CodeDeploy - automated deployments on premise & AWS
-  Code Pipeline - CI/CD



## Artificial Intelligence

-  Lex - speech to text, natural language understanding
-  Polly - text to speech
-  Rekognition - image recognition
-  Machine Learning - apply complex algorithms, predictions

# Setting up AWS Account

Point your browser to <http://aws.amazon.com/> and click on **Create a Free Account**.



The screenshot shows the AWS login page. At the top is the Amazon Web Services logo. Below it is the heading "Sign In or Create an AWS Account". The text says: "You may sign in using your existing Amazon.com account or you can create a new account by selecting 'I am a new user.'". There is a label "My e-mail address is:" followed by a text input field containing the placeholder "yourRegisteredEmail". Below this are two radio button options: "I am a new user." and "I am a returning user and my password is:". The second option is selected and highlighted with a red border. Below the selected option is a password input field with a masked password ".....". At the bottom is a yellow button with a play icon that says "Sign in using our secure server". Below the button are two links: "Forgot your password?" and "Has your e-mail address changed?".



## Sign In or Create an AWS Account

You may sign in using your existing Amazon.com account or you can create a new account by selecting "I am a new user."

My e-mail address is:

☐ I am a new user.

☒ I am a returning user and my password is:

[Forgot your password?](#)

[Has your e-mail address changed?](#)

# Setting up AWS Account

**Contact Information**

*\* Required Fields*

**Full Name\***

**Company Name**

**Country\***

**Address\***


**City\***

**State / Province or Region\***

**Postal Code\***

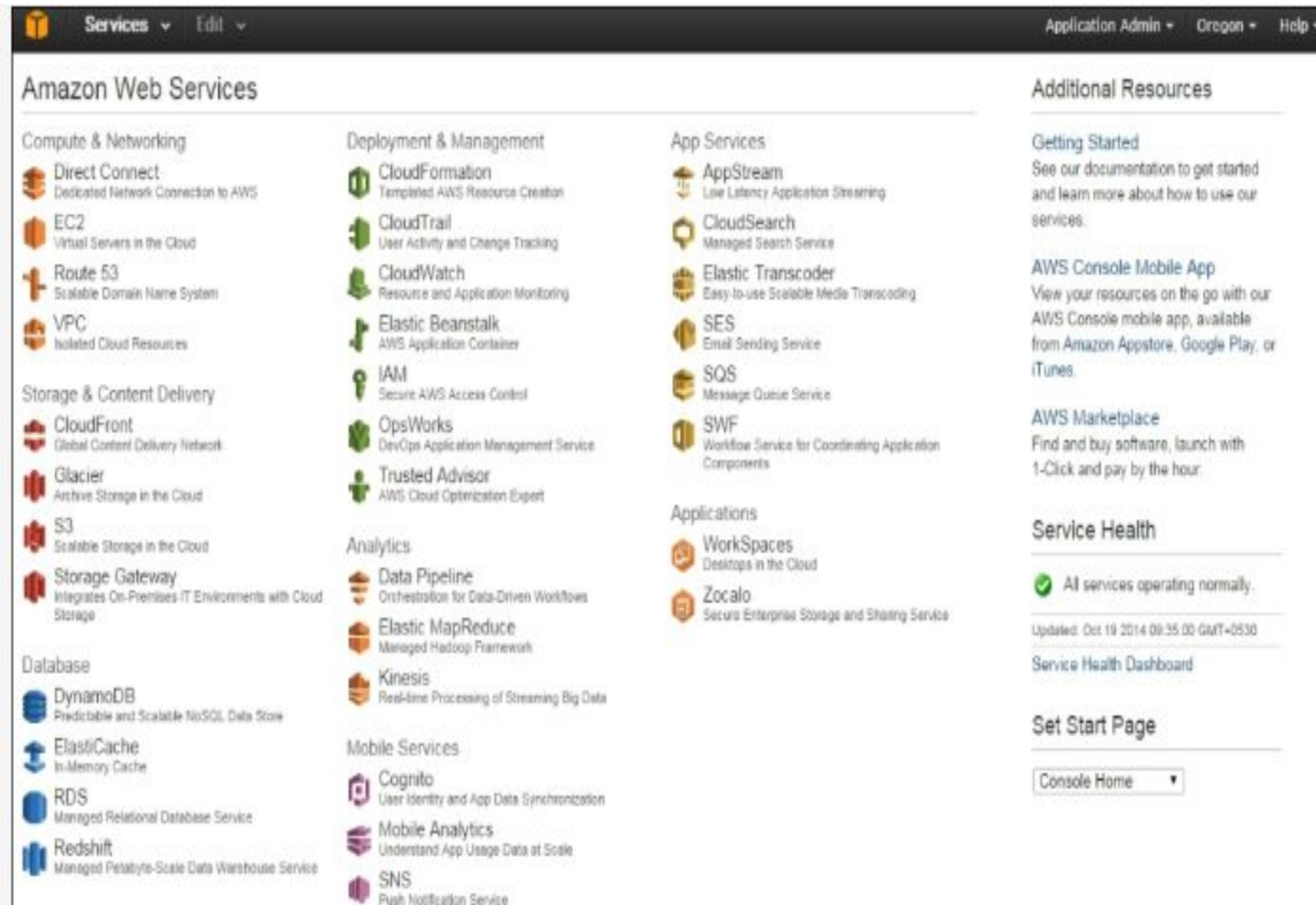
**Phone Number\***

 English Amazon Web Services Sign Up

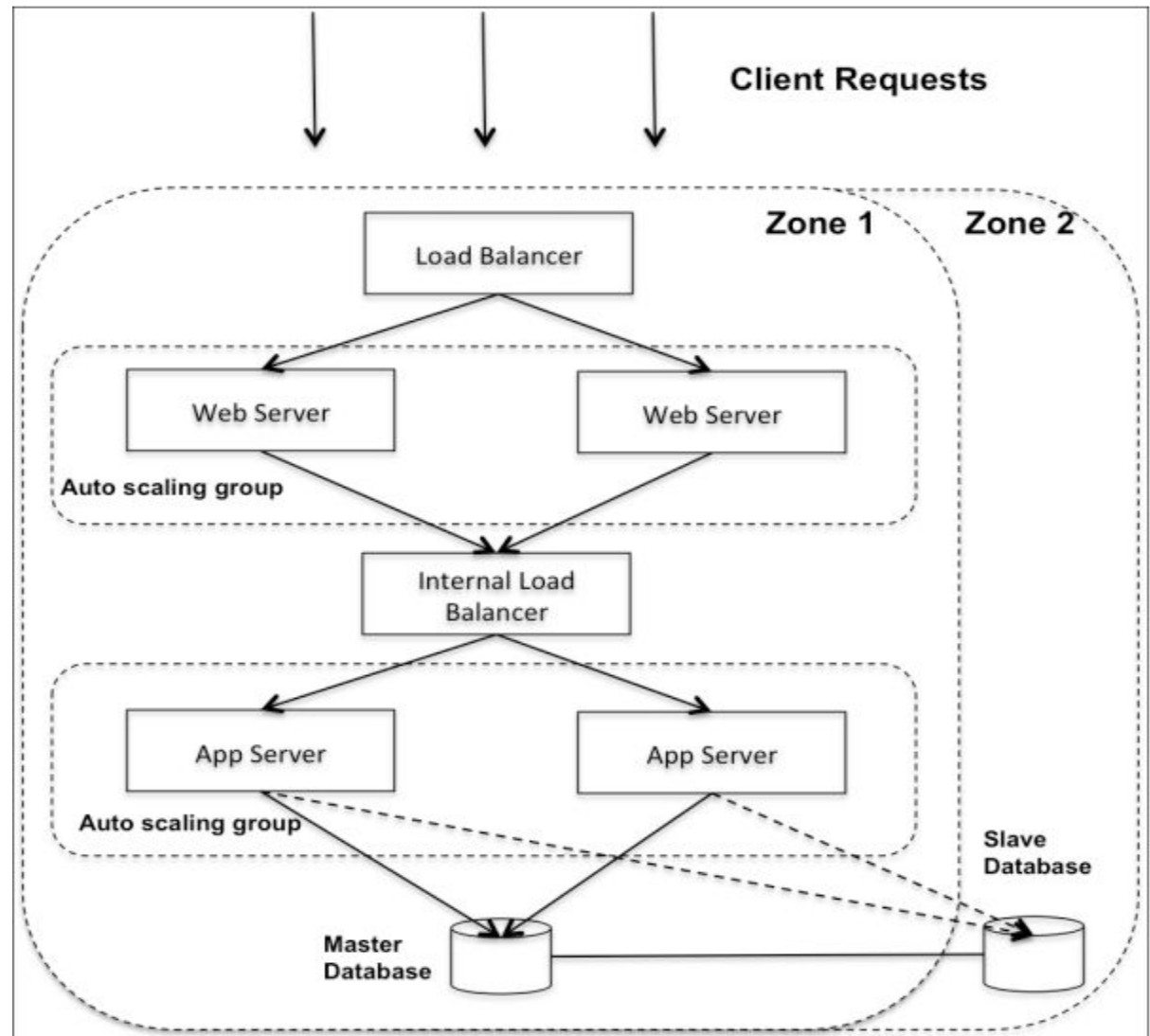
Progress bar showing steps: Contact Information (✓), Payment Information (✓), Identity Verification (✓), Support Plan (●), Confirmation (●)

# AWS Management Console



# Designing Cloud Applications

## Multi-tier Architecture



**Amazon Elastic Compute Cloud (EC2)** :Amazon EC2 is a web service that provides compute capacity in the AWS cloud.

**Amazon Elastic Block Storage (EBS)**:Amazon EBS is highly available and durable persistent block level storage volumes for use with Amazon EC2 instances.

**Elastic IP addresses** allow you to allocate a static IP address, and programmatically assign it to an instance.

**Amazon CloudWatch**: We can enable monitoring on EC2 instance using it.

We can also distribute incoming traffic by using the **Elastic Load Balancer (ELB)** service.

We can create auto scaling groups using the auto scaling feature to automatically scale your capacity based on CloudWatch.

**Amazon S3:** Amazon S3 is a highly durable and distributed data store.

**Amazon Glacier:** Amazon Glacier is low-cost storage service that is typically used for archiving and backups.

**Amazon RDS:** It provides an easy way to setup, operate, and scale a relational database in the cloud. Database options available from AWS include MySQL, Oracle, SQL Server, PostgreSQL, and Amazon Aurora (in preview at this time)

**Amazon DynamoDB :** It is a NoSQL database service offered by AWS. It supports both document and key-value pairs, data models, and has a flexible schema.

**Amazon ElastiCache:** If your application is read-intensive, then you can use the AWS ElastiCache service to significantly boost the performance of your applications. ElastiCache supports Memcached and Redis in-memory caching solutions.

**Amazon Simple Queue Service (Amazon SQS)** is a reliable, highly-scalable, hosted, and distributed queue for storing messages as they travel between computers and application components.

**Amazon Virtual Private Cloud (Amazon VPC)** allows you to extend your corporate network into a private cloud contained within AWS. Amazon VPC uses the IPSec tunnel mode that enables you to create a secure connection between a gateway in your data center and a gateway in AWS.

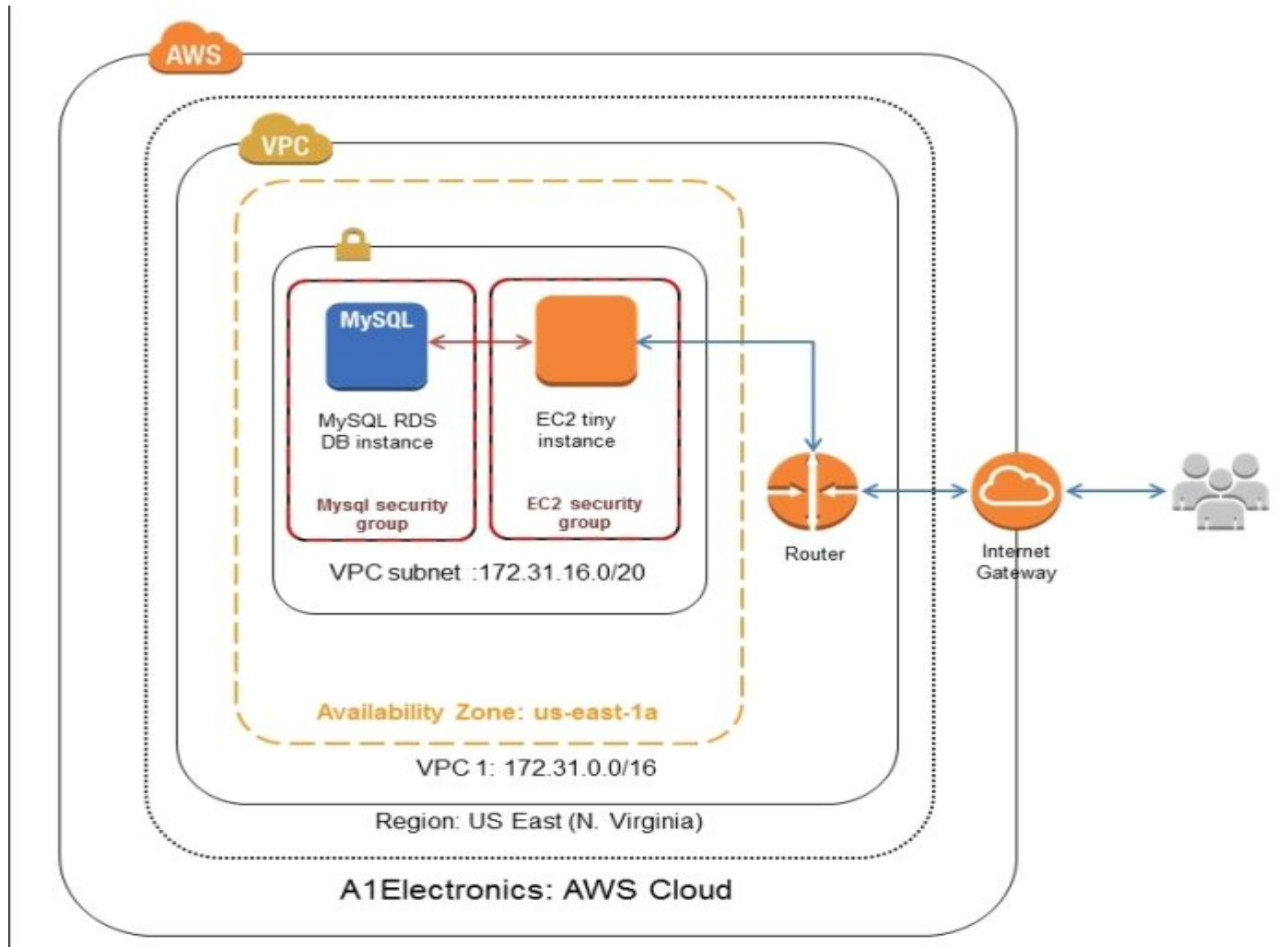
**Amazon Route 53** is a highly-scalable DNS service that allows you to manage your DNS records by creating a hosted zone for every domain you would like to manage.

**AWS Identity and Access Management (IAM)** enables you to you to control access to AWS services and resources.

**Amazon CloudWatch** is a monitoring service for your AWS resources. It enables you to retrieve monitoring data, set alarms, troubleshoot problems, and take actions based on the issues arising in your cloud environment.



# Setting up AWS Infrastructure



**Region:** AWS services are hosted in multiple locations around the world and these are known as regions. The regions are connected through the public internet.

**Availability Zone:** Availability zones (AZ) can be treated as traditional data centers within a region. AZs in the same region are designed to provide infrastructure redundancy in the event of a catastrophic outage, such as earthquakes, snowstorms, Godzilla attacks, and so on. The number of AZs in a region is region specific. In our example, we select the **us-east-1a** AZ.

**EC2 Instance:** This is a virtual server on which you run your applications. These come in various flavors to meet your computing demand. A high compute EC2 instance also has high network I/O memory associated with it. You cannot have a low compute EC2 instance with high memory and network I/O. EC2 instances have fixed CPU to memory ratios. It is best to select a micro instance for development, since it is free. More on EC2 instance types is available at <http://aws.amazon.com/ec2/instance-types/>.

**Security Groups:** A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. The security group can be configured by a set of rules for inbound and outbound traffic. The rules define the network protocol, port, and source and destination IP address ranges to accept or send your data to.

**Virtual Private Cloud (VPC):** VPC lets you provision a private, isolated section of the AWS cloud where you can launch AWS resources in a virtual network, using custom-defined IP address ranges. It is like your own private data centre.

**Subnets:** Subnets are logical segments of a VPC's address range that allow you to designate to a group of your resources based on security and operational needs

**Router:** Each VPC comes with a default router in order to communicate with resources outside the VPC. For example, connecting to a database server in other VPCs.

1. From the EC2 dashboard, click on **Security Groups** from the navigation pane and then on the **Create Security Group** button.
2. Create a security group for EC2 instances to allow the following
  1. Web traffic from any IP address on port 8080 (default Tomcat server port)
  2. SSH traffic for remote login from any IP address
  3. ICMP traffic to ping the EC2 instance from a public Internet

# Setting up AWS Infrastructure

### Create Security Group

Security group name ⓘ

sq-EC2WebSecurityGroup

Description ⓘ

Security rules to access the ec2 instances

VPC ⓘ

vpc-3f30a65a (172.31.0.0/16) \*

\* denotes default VPC

Security group rules:

Inbound

Outbound

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
SSH ▾	TCP	22	Anywhere ▾ 0.0.0.0/0 ⓘ
Custom TCP Rule ▾	TCP	8080	Anywhere ▾ 0.0.0.0/0 ⓘ
All ICMP ▾	ICMP	0 - 65535	Anywhere ▾ 0.0.0.0/0 ⓘ

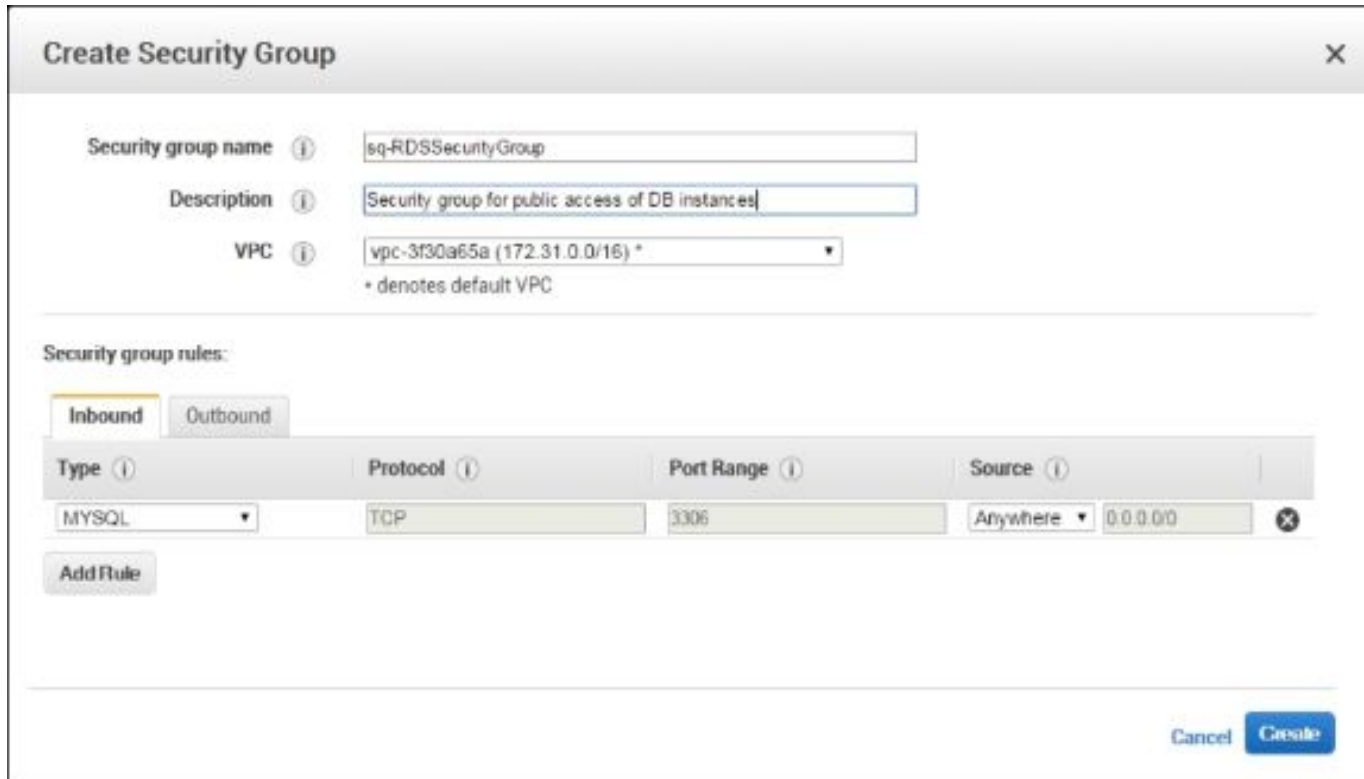
Add Rule

Cancel

Create

# Setting up AWS Infrastructure

Create a security group for MySQL RDS instances to allow access from the Internet.



The screenshot shows the 'Create Security Group' dialog in the AWS Management Console. The 'Security group name' is 'sq-RDSecurityGroup', the 'Description' is 'Security group for public access of DB instances', and the 'VPC' is 'vpc-3f30a65a (172.31.0.0/16) \*'. Under 'Security group rules', the 'Inbound' tab is selected, showing a rule for 'MYSQL' using 'TCP' on port '3306' from 'Anywhere' (0.0.0.0/0). There are 'Add Rule', 'Cancel', and 'Create' buttons.

**Create Security Group** [X]

Security group name ⓘ sq-RDSecurityGroup

Description ⓘ Security group for public access of DB instances

VPC ⓘ vpc-3f30a65a (172.31.0.0/16) \*  
\* denotes default VPC

Security group rules:

**Inbound** Outbound

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
MYSQL ▼	TCP	3306	Anywhere ▼ 0.0.0.0/0 [X]

Add Rule

Cancel Create

## Creating EC2 instance key pairs

1. From the EC2 dashboard, click on **Key Pairs** from the navigation pane and then on the **Create Key Pair** button.
2. Enter e2accesskey when prompted with a dialog box asking to enter the key pair name. This key pair name will be used while configuring the EC2 instances.

Note - Make sure you select the correct AWS region from the EC2 dashboard to create the keys because key pairs can't be shared across regions

As soon as you create the key pair, your private key will be immediately downloaded to your computer. Secure this private key. This private key file can be only downloaded once during the creation of the keys. You cannot change access keys in your EC2 instances once they have been assigned.

## Creating Roles

1. From the IAM dashboard, click on **Roles** in the navigation pane and then on the **Create New Role** button
2. Create a role named ec2Instance for our EC2 instances that have access to all the AWS provided services, as shown in the following screenshot:

The screenshot shows the 'Create Role' wizard in the AWS IAM console. On the left, a sidebar lists the steps: 'Step 1: Set Role Name' (selected), 'Step 2: Select Role Type', 'Step 3: Establish Trust', 'Step 4: Set Permissions', and 'Step 5: Review'. The main area is titled 'Set Role Name' and contains the instruction 'Enter a role name. You cannot edit the role name after the role is created.' Below this, there is a 'Role Name' label and a text input field containing 'ec2instances'. A small note below the input field states: 'Maximum 64 characters. Use alphanumeric and '+', '@', '-' characters'. At the bottom right of the main area, there are two buttons: 'Cancel' and 'Next Step'.



# Setting up AWS Infrastructure

Create Role

Step 1: Set Role Name

**Step 2: Select Role Type**

Step 3: Establish Trust

Step 4: Set Permissions

Step 5: Review

## Select Role Type

• AWS Service Roles

Amazon EC2

Allows EC2 instances to call AWS services on your behalf.

Select

• AWS Directory Service

Allows AWS Directory Service to manage access for existing directory users and groups to AWS services.

Select

• AWS Lambda

Allows Lambda Function to call AWS services on your behalf.

Select

• AWS S3 Invocation for Lambda Functions

Allows S3 to invoke Lambda functions in your account.

Select

• AutoScaling Notification Access

Allows AWS AutoScaling to publish to SNS and SQS notification targets in your account.

Select

Cancel

Previous

Next Step

# Setting up AWS Infrastructure

Create Role

Step 1: Set Role Name

Step 2: Select Role Type

Step 3: Establish Trust

**Step 4: Set Permissions**

Step 5: Review

## Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

Select Policy Template

Administrator Access

Provides full access to AWS services and resources.

Select

Power User Access

Provides full access to AWS services and resources, but does not allow management of Users and groups.

Select

Read Only Access

Provides read-only access to AWS services and resources.

Select

AWS CloudFormation Read Only Access

Provides access to AWS CloudFormation via the AWS Management Console.

Select

Cancel

Previous

Next Step

# Setting up AWS Infrastructure

## Create Role

Step 1: Set Role Name

Step 2: Select Role Type

Step 3: Establish Trust

Step 4: Set Permissions

**Step 5: Review**

## Review

Review the following role information. To edit the role, click an edit link, or click **Create Role** to finish.

**Role Name** ec2Instances [Edit Role Name](#)

**Role ARN** arn:aws:iam:295846325849:role/ec2instances

**Trusted Entities** The identity provider ec2.amazonaws.com

**Permissions** Power User Access [Edit Permissions](#)

[Cancel](#)

[Previous](#)

[Create Role](#)

## Creating EC2 Instance

1. From the EC2 dashboard, click on **Instances** in the navigation pane and on the **Launch** instance. This will start a process of provisioning an EC2 instance.
2. The next step is to choose an operating system for the EC2 instance; this is done by choosing the correct **Amazon Machine Image (AMI)** as per our requirements. Select the **Ubuntu Server 14.04 LTS (HVM) SSD Volume Type** AMI, as shown in the following screenshot:

# Setting up AWS Infrastructure

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review





Cancel and Exit

## Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start ( < 1 to 22 of 22 AMIs > )

- My AMIs
- AWS Marketplace
- Community AMIs
- ☒ Free tier only (1)

 Amazon Linux Free tier eligible	<b>Amazon Linux AMI 2014.09.1 (HVM) - ami-b65ed3de</b> The Amazon Linux AMI is an EBS backed image. It includes the 3.14 kernel, Ruby 2.1, PHP 5.5, PostgreSQL 9.3, Docker 1.2, the AWS command line tools, and repository access to many other packages. Root device type: ebs    Virtualization type: hvm	Select 64-bit
 Red Hat Free tier eligible	<b>Red Hat Enterprise Linux 7.0 (HVM), SSD Volume Type - ami-a8d369c0</b> Red Hat Enterprise Linux version 7.0 (HVM), EBS General Purpose (SSD) Volume Type Root device type: ebs    Virtualization type: hvm	Select 64-bit
 SUSE Linux Free tier eligible	<b>SUSE Linux Enterprise Server 12 (HVM), SSD Volume Type - ami-aeb632c6</b> SUSE Linux Enterprise Server 12 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled. Root device type: ebs    Virtualization type: hvm	Select 64-bit
 Ubuntu Free tier eligible	<b>Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-9eaa1cf5</b> Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical ( <a href="http://www.ubuntu.com/cloud/services">http://www.ubuntu.com/cloud/services</a> ). Root device type: ebs    Virtualization type: hvm	Select 64-bit

# Setting up AWS Infrastructure

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

## Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
<input checked="" type="checkbox"/>	General purpose	t2.micro <small>Free tier eligible</small>	1	1	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate
<input type="checkbox"/>	General purpose	m3.medium	1	3.75	1 x 4 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.large	2	7.5	1 x 32 (SSD)	-	Moderate
<input type="checkbox"/>	General purpose	m3.xlarge	4	15	2 x 40 (SSD)	Yes	High
<input type="checkbox"/>	General purpose	m3.2xlarge	8	30	2 x 80 (SSD)	Yes	High

Cancel Previous Review and Launch Next: Configure Instance Details

# Setting up AWS Infrastructure


1. Choose AMI   2. Choose Instance Type   **3. Configure Instance**   4. Add Storage   5. Tag Instance   6. Configure Security Group   7. Review


### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

**Number of instances** ⓘ

**Purchasing option** ⓘ ☐ Request Spot Instances

**Network** ⓘ   Create new VPC

**Subnet** ⓘ   Create new subnet  
4090 IP Addresses available

**Auto-assign Public IP** ⓘ

**IAM role** ⓘ

**Shutdown behavior** ⓘ

**Enable termination protection** ⓘ ☒ Protect against accidental termination

**Monitoring** ⓘ ☐ Enable CloudWatch detailed monitoring  
Additional charges apply.

**Tenancy** ⓘ   
Additional charges will apply for dedicated tenancy.

▸ Network interfaces

▸ Advanced Details

Cancel   Previous   Review and Launch   **Next: Add Storage**

# Setting up AWS Infrastructure

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

## Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more about storage options in Amazon EC2.](#)

Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Delete on Termination	Encrypted
Root	/dev/sda1	snap-1f006ddb	30	General Purpose (SSD)	90 / 3000	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

General Purpose (SSD) volumes provide the ability to burst to 3,000 IOPS per volume, independent of volume size, to meet the performance needs of most applications and also deliver a consistent baseline of 3 IOPS/GiB. [Set my root volume to General Purpose \(SSD\).](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more about free usage tier eligibility and usage restrictions.](#)

Cancel Previous **Review next launch** Next: Tag instance



# Setting up AWS Infrastructure

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

### Step 5: Tag Instance

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Learn more about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)
Name	A1ElectronicsEcommerce

Create Tag (Up to 10 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

# Setting up AWS Infrastructure

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Tag Instance6. Configure Security Group7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: ☐ Create a new security group  
☒ Select an existing security group

Security Group ID	Name	Description	Actions
<input type="radio"/> sg-aa9b68ce	sg-RDSSecurityGroup	Security rules to access RDS instances	<a href="#">Copy to new</a>
<input type="radio"/> sg-1302e577	default	default VPC security group	<a href="#">Copy to new</a>
<input checked="" type="radio"/> sg-979767f3	sg-EC2WebSecurityGroup	Security rules to access the ec2 instances	<a href="#">Copy to new</a>

Inbound rules for sg-979767f3 (Selected security groups: sg-979767f3)

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ
Custom TCP Rule	TCP	8080	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
All ICMP	All	N/A	0.0.0.0/0


[Cancel](#)[Previous](#)[Review and Launch](#)

# Setting up AWS Infrastructure

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Tag instance6. Configure Security Group7. Review

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**Improve your instance's security. Your security group, sg-EC2WebSecurityGroup, is open to the world.**

Your instance may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details

Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-9a41c66

Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>)

root device type: xfsvirtualization type: hvm

Edit AMI

Instance Type

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Edit instance type

Security Groups

Security group name: sg-EC2WebSecurityGroup

Description: Supports HTTP port 8080, ICMP, SSH

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0
Custom TCP Rule	TCP	8080	0.0.0.0

Edit security groups

Cancel

Previous

Launch

Slide 43

[www.ethans.co.in](http://www.ethans.co.in)

# Setting up AWS Infrastructure

## Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Choose an existing key pair

Select a key pair

ec2AccessKey

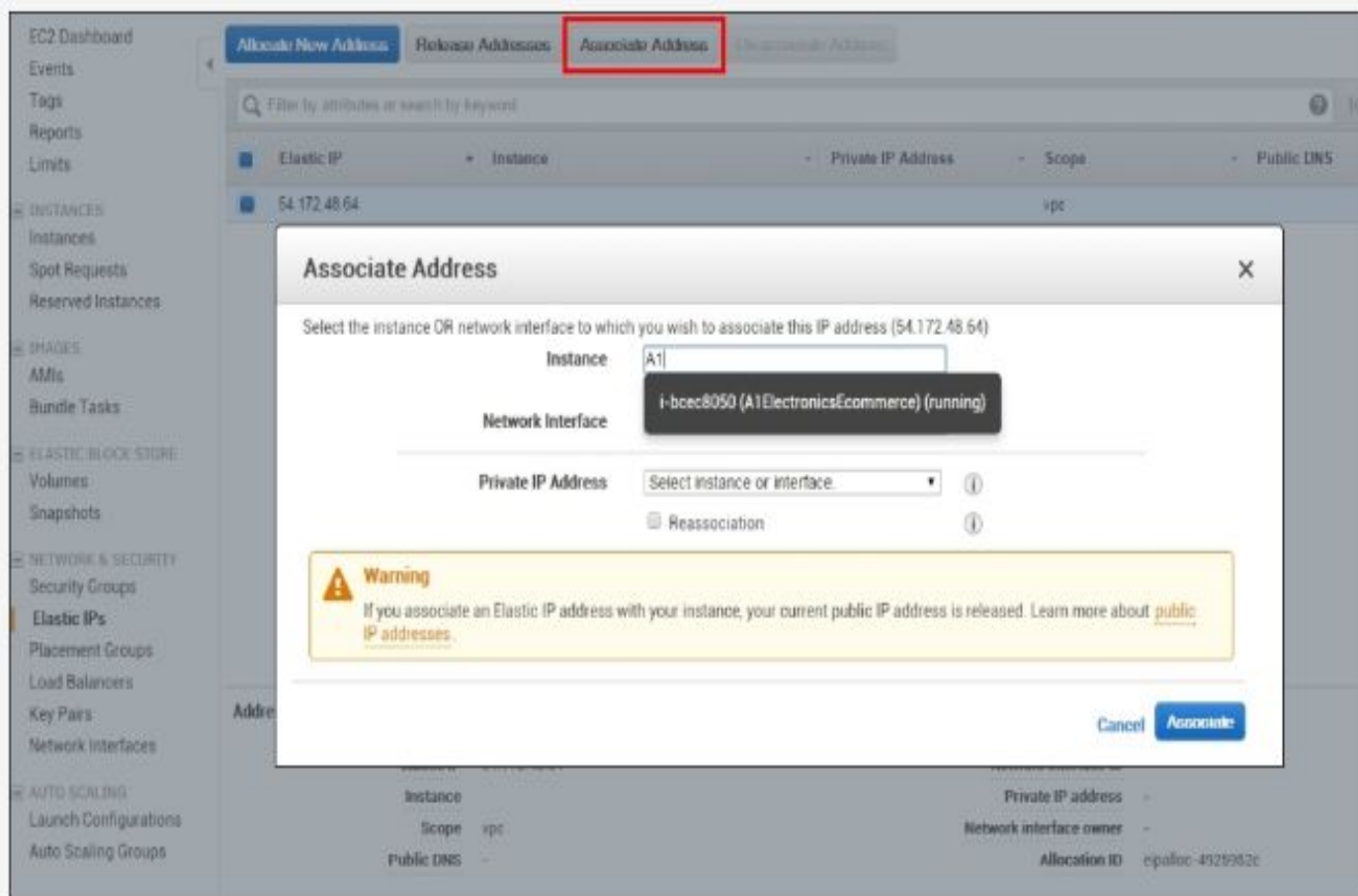
☒ I acknowledge that I have access to the selected private key file (ec2AccessKey.pem), and that without this file, I won't be able to log into my instance.

Cancel

Launch Instances

# Setting up AWS Infrastructure

## Assigning Elastic IP



# Setting up AWS Infrastructure

The screenshot shows the AWS Management Console interface for an EC2 instance. The left sidebar contains navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, IMAGES, ELASTIC BLOCK STORE, and NETWORK & SECURITY. The main content area displays the details for the instance 'A1ElectronicsEcommerce' (Instance ID: i-b0ec8050). The instance is in a 'running' state and is located in the 'us-east-1a' availability zone. A red box highlights the following information:

Property	Value
Public DNS	ec2-54-172-48-64.compute-1.amazonaws.com
Public IP	54.172.48.64
Elastic IP	54.172.48.64

Other instance details include:

- Instance ID: i-b0ec8050
- Instance state: running
- Instance type: t2.micro
- Private DNS: ip-172-31-16-179.ec2.internal
- Private IPs: 172.31.16.179
- Secondary private IPs: None
- VPC ID: vpc-3f30a65a
- Subnet ID: subnet-6e15c737
- Network interfaces: eth0
- Source/dest. check: True
- EBS-optimized: False
- Root device type: ebs
- Availability zone: us-east-1a
- Security groups: sg-EC2WebSecurityGroup, view rules
- Scheduled events: No scheduled events
- AMI ID: ubuntu-trusty-14.04-amd64-server-20140927 (ami-9ea1cfc0)
- Platform: -
- IAM role: ec2Instances
- Key pair name: ec2AccessKey
- Owner: 295846325849
- Launch time: December 17, 2014 11:44:30 PM UTC+5:30 (22 hours)
- Termination protection: True

A	: ( <b>address</b> ) Actual IP address of the Domain
AAAA	: Ipv6 address record which maps the hostname to a 128 bit Ipv6 address
CNAME	: ( <b>canonical name</b> ) Makes one domain alias for another domain name
MX	: ( <b>mail exchange</b> ) List of mail exchange servers
PTR	: ( <b>pointer record</b> ) Maps an Ipv4 address to the CNAME on the hos
NS	: ( <b>name server</b> ) It highlights which Name Server is authoritative for the domain
SOA	: ( <b>State of Authority</b> ) It stores important information like timestamp when any change to Domain was made
SRV	: ( <b>Service</b> ) To define TCP service
TXT	: ( <b>Text</b> ) This lets the admin to insert text in the DNS record