18.18.Apr.AWS IAM, Key Pair, SG,EBS EC2

18 April 2023 07:50 PM

AWS

- Service Model
  - IaaS = AWS, Azure, GCP
    - EC2(Virtual Cloud Server- Elastic Cloud Computing)
  - PaaS = AWS, Azure, GCP
    - S3, IAM- Identity Access Management- Users, Groups, Policy=Permission , Roles
  - SaaS = Salesforce

IAM

1. Groups - Set of Users + Policy
2. User - who uses aws services
   a. Web Console
   b. CLI
   c. SDK
   d. REST
   e. IaaC
3. Policy- JSON format permission

IAM --> Group(developers) --> Policy(S3FullAccess) --> user (tom)

Accessing AWS all services

1. Web Console(Website) = root [Email, Password], iam normal user [account id, username, password]
2. CLI(Command)
3. SDK (Python/Java/Go)
4. REST API Call
5. IaaC:
   a. CloudFormation(Blueprint)
   b. 3rd Party(Terraform)

AWS User

- Root User = Owner(Full Permissions) = email, password
- IAM User = Employee(Specific Permissions) = Account ID, Username, Password

URL:	https://m111.signin.aws.amazon.com/console
UserName:	tom
Password:	6@f5F{9'

IAM Service

- Identity = tom --> Developers
- Access = S3, EC2 using Policy
- Manage

1. Create iam group
   a. Attach Policy(S3FullAccess)
2. Create iam user

    a. Console Access
    b. Autogenerated Password
    c. Attach this user to developers group
    d. Uncheck Next Password
3. Policy
    a. Attach to Group(developers)

EC2(Virtual Cloud Server)

Compute Engine:
- Virtual= Shared(Instance)
- Cloud= Hardware will be taken care by AWS
- Server= Logical Server

SAI (16GB) --> 4GB(Rent--> Virtualization = VirtualBox), 8GB
Hyper V
VMware
KVM
Microsoft Virtual Player
Oracle VirtualBox

AMI = Amazon Machine Image = Pre Installed OS(Free Tier)+ Extra Packages (Windows Free/Paid)
Instance Type: t2.micro(CPU, RAM)

EBS= Elastic Block Storage = Virtual Hard Disk =8 GB

Security Group = Firewall
- Rules
    - In Bound = Who can come
    - Out Bound = who can go from your server to anywhere in internet

Rule(Inbound/Outbound):
- Language(Protocol) = HTTP
- Door(Port) = 80
- Come From (Source/Destination) = 11.22.33.44/32 = IP Range = CIDR Block
    - IPv4 : 0.0.0.0/0
    - IPv6 : ::/0

CIDR Block:
Ex1: 192.168.0.0/28

Start: 192.168.0.**0**
End : 192.168.0.**15**

32-28 = 2^4 = 16 16

Ex2: 10.0.0.0/26

Start: 10.0.0.**0**
End : 10.0.0.63

32-26 = 2^ 6 = 64

1. Create firewall for Web server on port **80 --> HTTP** for any one.

2. Create Firewall for System Admin to Access through **SSH=22** protocol for any one.

Prepare Web Server

1. AMI : Ubuntu
2. EBS : 8GB
3. Security Group: sgalltraffic = all traffic(all protocols, all ports) , Anywhere
4. Key Pair: mujahed.pem key
5. User Data: After System Ready ---> Execute some Commands(CI/CD Jenkins Server)

```
#!/bin/bash

# Update packages
sudo apt-get update

# Install Java
sudo apt install -y default-jdk

# Install Jenkins dependencies
sudo apt install -y git

# Install Jenkins
sudo wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
sudo echo "deb https://pkg.jenkins.io/debian-stable binary/" >> /etc/apt/sources.list.d/jenkins.list
sudo apt update
sudo apt install -y jenkins

# Start Jenkins and enable it to start on boot
systemctl start jenkins
systemctl enable jenkins
```

$ systemctl status jenkins

```
#!/bin/bash
sudo apt update
sudo apt install python3-pip -y
sudo mkdir /etc/ansible
sudo bash -c 'cat<<EOF > /etc/ansible/hosts
[ws]
127.0.0.1
EOF'
sudo apt install ansible -y
```

```
ssh-keygen -b 3072 -t rsa -f ~/.ssh/sshkey -q -N ""
cat ~/.ssh/sshkey.pub | cat >> ~/.ssh/authorized_keys
sudo ansible --version > /tmp/ansibleVersion.txt
```

```
$ cat /tmp/ansibleVersion.txt
```

```
$ ssh   -i mujahed.pem UserName@OnlyPublicIP
```