1. Inspector
2. WAF & Shield
3. Resource Access Manager
4. Cognito
5. Secrets Manager
6. GuardDuty
7. Key Management Service
8. CloudHSM

**1, Inspector**:

We use it for Ec2 servers only, when we implement and going to handover project to client, we run Inspector to check unknown errors in Ec2 like Network problems, security threats

**Lab:**

1, install 3 Ec2 Linux servers Web Server1, Web Server2, Web Server

|  |
| --- |
| #! /bin/bash  yum update -y  yum install nginx -y  service nginx start  echo "<h1>Web server-1</h1>" >> /usr/share/nginx/html/index.html |

Install inspector and enable scanning on it (Google: aws inspector agent install)

* Login to Web Server1 in Putty **&** Login to Web Server2 in Putty
* Sudo su – (root user)
* curl -O <https://inspector-agent.amazonaws.com/linux/latest/install> (install inspector agent)
* bash install

Now add Tags to Ec2 Linux servers to enable scanning

Web Server1 & Web Server2------------------------Tags: Env & Production

Web Server3---------------------------------------------Tags: Env & Development

2, Inspector

Security, Identity, & Compliance -> Inspector -> **Assessment targets** (that means whom to target or scan) -> Name\*: "AWSB26" & Use Tags\*: Env & Production (Because I want to scan only Web Server1 & Web Server2)

Security, Identity, & Compliance -> Inspector -> **Assessment templates** (what to scan i.e., AWSB26) -> Name\*: "AWSB26 template" & Target name\*: "AWSB26" (which we created in Assessment targets) &

Rules packages\*:

Network Reachability-1.1

Security Best Practices-1.0

Common Vulnerabilities and Exposures-1.1

CIS Operating System Security Configuration Benchmarks-1.0 ("Center for Internet Security" is an organization that provides AMI benchmarks)

&

Duration\*: 1 Hour

[we can check after 1 Hour for problems]

**Workflow**: **[3 Ec2 servers – 3 Ec2 servers (Tags) – inspector (Ec2 tags) – inspector (scans)]**

**2, WAF & Shield:**

**AWS WAF**: Web Application Firewall

We have “Security Groups” to block only traffic to Web Server or DB Server, but in “WAF” we block traffic to whole Region.

Here we can use those 3 Ec2 servers

**Lab:**

We create **Application load balancer (**HTTP**)**

1, EC2 -> Load Balancing -> **Target Groups** -> Target group name: AWSB26-WAF & Protocol: HTTP & VPC

EC2 -> Target groups -> select our created "AWSB26-WAF" -> goto menu and select "Targets" ->

Register targets -> select Web Server1, Web Server2, Web Server3

2, EC2 -> Load Balancing -> **Load Balancers** -> Create Load Balancer -> Application Load Balancer "Create" -> Name: AWSB26-WAF-ALB & Select VPC and Availability zones and Subnets in AZ -> -> Select an existing security group -> Target group: "Existing target group" & Name: "AWSB26-WAF"

**HTTPS:443**

EC2 -> Load Balancing -> **Load Balancers** -> AWSB26-WAF-ALB -> select "Listeners" in menu ->You can see HTTP:80 -> click on "Add listener" -> Protocol : HTTPS **&** Default action(s): Forward to.. : "AWSB26-WAF" target group **&** Security policy: “ELBSecurityPOlicy-FS-1-12019-08” **&** Default SSL certificate: (create “Certificate Manager” under IAM [**Refer 20th Document**])

**HTTP:80 Redirect to HTTPS:443**

EC2 -> Load Balancing -> **Load Balancers** -> AWSB26-WAF-ALB -> select "Listeners" in menu ->You can see HTTP:80 click on "AWSB26-WAF View/edit rules" -> click top on “pencil” to edit **&** click left on “pencil” to edit **& 1. Redirect to...** HTTPS:443

EC2 -> Load Balancing -> **Load Balancers** -> AWSB26-WAF-ALB -> select "Descriotion" in menu -> **copy** DNS name: AWSB26-WAF-ALB-1169407024.us-east-1.elb.amazonaws.com (and add in Route53)

3, Networking & Content Delivery -> **Route 53** -> Hosted zones -> select "dheerajpalvai.xyz" -> Create record -> Simple routing -> Define simple record -> Record name: www & Value/Route traffic to: "Alias to Application and Classic Load Balancer" : [us-east-1] & A record

--------------

4, WAF: here we need to block some internet router provider IPS (https://whatismyipaddress.com/ [73.119.241.21]) to block Region

Security, Identity, & Compliance -> WAF & Shield -> **AWS WAF** -> **IP sets** -> Create IP set -> IP set name: "AWSB26-Blocked-IPs" **&** In IP addresses box we blocked Ips (73.119.241.21/32) should give subnet /32

Security, Identity, & Compliance -> WAF & Shield -> **AWS WAF** -> **Web ACLs** -> Create web ACL -> Name: "AWSB26-ACLs" & under Associated AWS resources - optional click on Add AWS resources -> select “**Application Load Balancer**” and select our ALB -> -> here AWS give some managed rules, so goto “Add rules” and select “Add managed rule groups” : here if we want to set our own rules, select “Add my own rules and rule groups” -> select “Add my own rules and rule groups” -> IP set -> Name: “Blockmyipset” **&** choose IP set: "AWSB26-Blocked-IPs"

Security, Identity, & Compliance -> WAF & Shield -> **AWS WAF** -> **Rule groups** (here we can create our own security rules and attach to Web ACLs)

**Workflow**: **[3 Ec2 servers – ALB – WAF blocking IPs or counties – WAF (ALB, WAF IPs)]**

**[Here we Blocked some IPs for website “dheerajpalvai.xyz” which is of “Application Load Balancer” using WAF]**

**[This WAF sits in front of ALB, website doesn’t go to ALB – That WAF contains list of blocked Ips]**

**-> [WAF is used for only ALB & CDN which are internet facing application]**

**[In real time we don’t expose our server to internet, so we don’t use this much]**

**AWS Shield**:

1, DDos – Distributed Denel of servicer

DDos attacks – DDos attacks means attacking server by sending more traffic.

If we send DDos attacks to any server, that server cannot handle that traffic

e.g., in amazon we have “big billion-dollar sale” and flipkart (retail online sale) is competitor to amazon,

if flipkart want to disturb amazon sale then flipkart sends more traffic amazon servers using DDos, so that amazon server gets down

2, How to handle DDos attacks?

In real time the sufferer from DDos attacks (amazon) use **AWS Shield** to slow down traffic, they (amazon) will open separate ports or separate server for DDos attacks and all the traffic from DDos comes to that ports we called this process as **honeypot**,

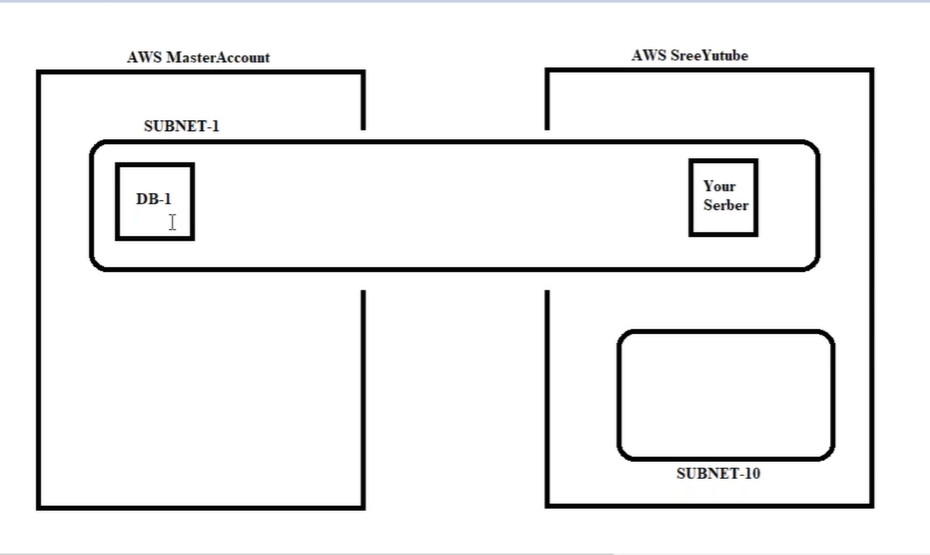
But an experienced hacker will able to identify this technique

Finally, there is no chance to escape from DDos attacks

**AWS Firewall Manager**:

This is also AWS WAF firewall only to block regions or countries, but it is used for all **AWS accounts** and manage centrally

**3, Resource Access Manager**: sharing subnet to other account



Here in “Resource Access Manager” we are sharing our subnet from main account to Sub-account within the same region in two accounts, without using VPC peering

**Lab**:

1,

Main account ([dheerajpalvai@gmail.com](mailto:dheerajpalvai@gmail.com))

Security, Identity, & Compliance -> **Resource Access Manager** -> Create a resource share -> Name: "AWSB26-RAM" & Select resource type: "Subnets" & Add AWS account number: 989878044275(this is sub-account account number)

2,

Sub-account ([dheeraj.nc24@gmail.com](mailto:dheeraj.nc24@gmail.com))

Ec2 -> Launch instance -> here in place of VPC and Subnet selection we get Main account VPC and Subnet

**4, Cognito**: sign-in types

Cognito is basically used for developers for Web applications

There are 2 types of logins in web applications

Manage User Pools

Manage Identity Pools

Manage User Pools: is like registering and sign-in

Manage Identity Pools: is like sign-in with **Gmail** account or **Facebook** account

**5, Secrets Manager**: storage of passwords

We can **store all DB passwords** or **certificates** or **API tokens** here we can select passwords from here without entering manually

**6, GuardDuty**: security guard

intrusion prevention service: is reactive (like Security Guard guard to your home)

intrusion detection service: is proactive (like CCTV camera to your home)

note: GuardDuty is “intrusion detection service”

it scans all types of duties and stores in Log files

**7, Key Management Service**: encrypted keys

Here we have some of the keys which are Encrypted with our data, so that no one can steal that data

**8, CloudHSM**: Hardware Security Module is a physical device, we can keep all keys here