Scalability & High Availability - Load balancing

08:12 PM

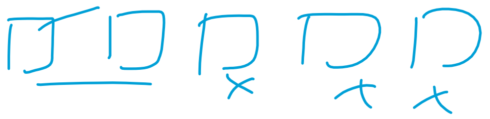
up high

Scalability & High Availability

1) up
2) high process

- Scalability means that an application / system can handle greater loads by adapting.
- There are two kinds of scalability:
  - Vertical Scalability
  - Horizontal Scalability (= elasticity)
- **Scalability is linked but different to High Availability**

base

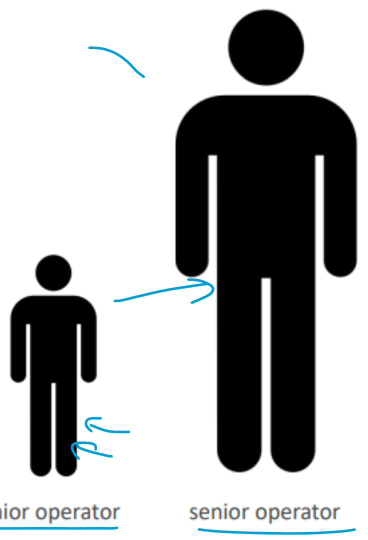


- Let's deep dive into the distinction, using a call center as an example

Vertical Scalability

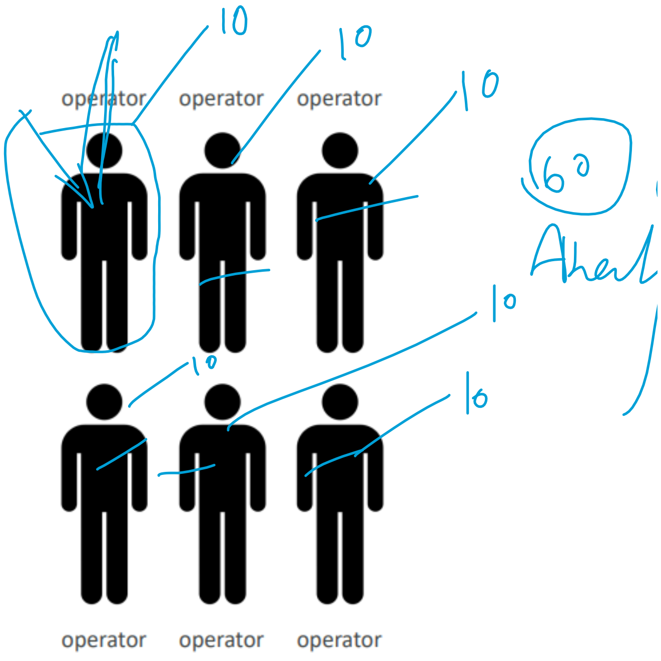
- Vertically scalability means increasing the size of the instance
- For example, your application runs on a t2.micro
- Scaling that application vertically means running it on a t2.large
- Vertical scalability is very common for non distributed systems, such as a database.
- RDS, ElastiCache are services that can scale vertically.
- There's usually a limit to how much you can vertically scale (hardware limit)

Aws



Horizontal Scalability

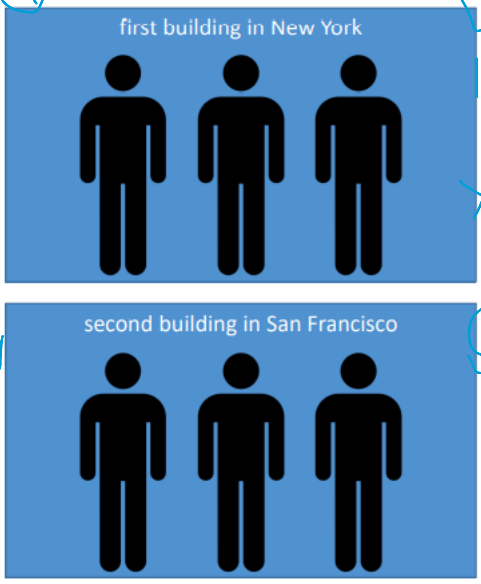
- Horizontal Scalability means increasing the number of instances / systems for your application
- Horizontal scaling implies distributed systems.
- This is very common for web applications / modern applications
- It's easy to horizontally scale thanks the cloud offerings such as Amazon EC2



AP

High Availability

- High Availability usually goes hand in hand with horizontal scaling
- High availability means running your application / system in at least 2 data centers (== Availability Zones)
- The goal of high availability is to survive a data center loss
- The high availability can be passive (for RDS Multi AZ for example)
- The high availability can be active (for horizontal scaling)



High Availability & Scalability For EC2

- Vertical Scaling: Increase instance size (= scale up / down)
  - From: t2.nano - 0.5G of RAM, 1 vCPU
  - To: u-12tb1.metal – 12.3 TB of RAM, 448 vCPUs

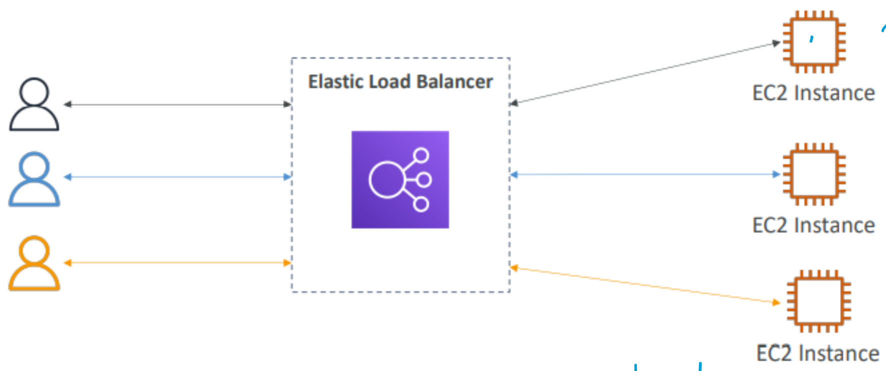
max

how?

- Horizontal Scaling: Increase number of instances (= scale out / in)
  - Auto Scaling Group
  - Load Balancer
- High Availability: Run instances for the same application across multi AZ
  - Auto Scaling Group multi AZ
  - Load Balancer multi AZ

What is load balancing?

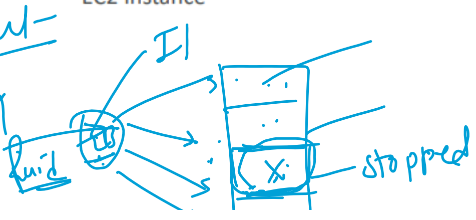
- Load Balances are servers that forward traffic to multiple servers (e.g., EC2 instances) downstream



when

Why use a load balancer?

why?



- Spread load across multiple downstream instances
- Expose a single point of access (DNS) to your application
- Seamlessly handle failures of downstream instances
- Do regular health checks to your instances
- Provide SSL termination (HTTPS) for your websites
- Enforce stickiness with cookies
- High availability across zones
- Separate public traffic from private traffic

http

a AZ

b AZ

Why use an Elastic Load Balancer?

- An Elastic Load Balancer is a managed load balancer
  - AWS guarantees that it will working
  - AWS takes care of upgrades, maintenance, high availability
  - AWS provides only a few configuration knobs
- It costs less to setup your own load balancer but it will be a lot more effort on your end
- It is integrated with many AWS offerings / services
  - EC2, EC2 Auto Scaling Groups, Amazon ECS
  - AWS Certificate Manager (ACM), CloudWatch
  - Route 53, AWS WAF, AWS Global Accelerator

(PaaS)

IAAS

SaaS

Health Checks

- Health Checks are crucial for Load Balancers
- They enable the load balancer to know if instances it forwards traffic to are available to reply to requests
- The health check is done on a port and a route (/health is common)
- If the response is not 200 (OK), then the instance is unhealthy

core

responses

4xx

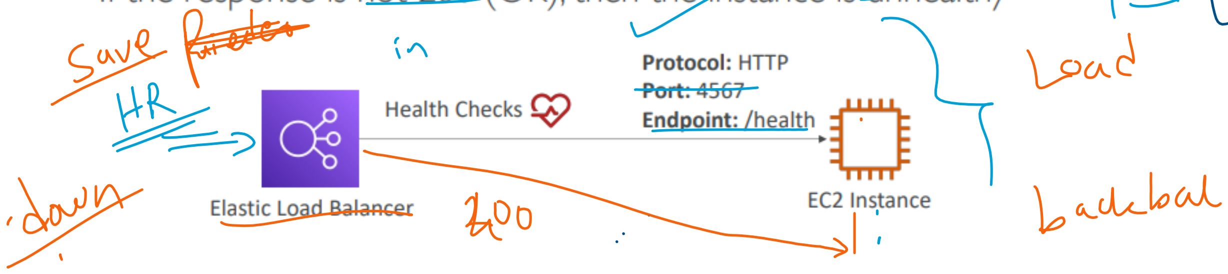
2xx

3xx

4xx

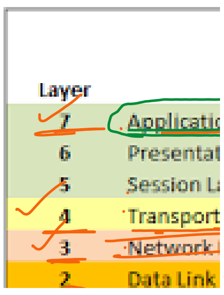
5xx

80, 443



Types of load balancer on AWS

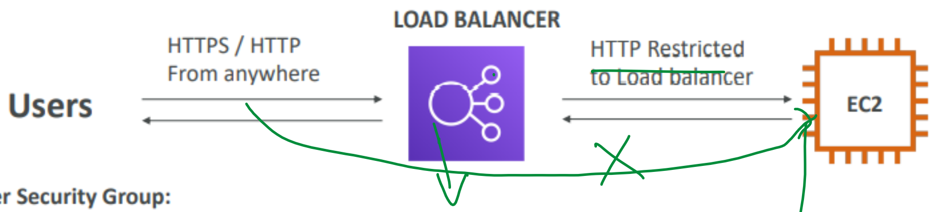
- AWS has 4 kinds of managed Load Balancers
- Classic Load Balancer (v1 - old generation) – 2009 – CLB
  - HTTP, HTTPS, TCP, SSL (secure TCP)
- Application Load Balancer (v2 - new generation) – 2016 – ALB
  - HTTP, HTTPS, WebSocket



Layer	
7	Applicati
6	Presentat
5	Session L
4	Transport
3	Network
2	Data Link

- Network Load Balancer (v2 - new generation) – ~~2017 – NLB~~
  - TCP, TLS (secure TCP), UDP
- Gateway Load Balancer – ~~2020 – GWLB~~
  - Operates at layer 3 (Network layer) – IP Protocol
- Overall, it is recommended to use the newer generation load balancers as they ~~provide more features~~
- Some load balancers can be setup as internal (private) or external (public) ELBs

Load Balancer Security Groups



Load Balancer Security Group:

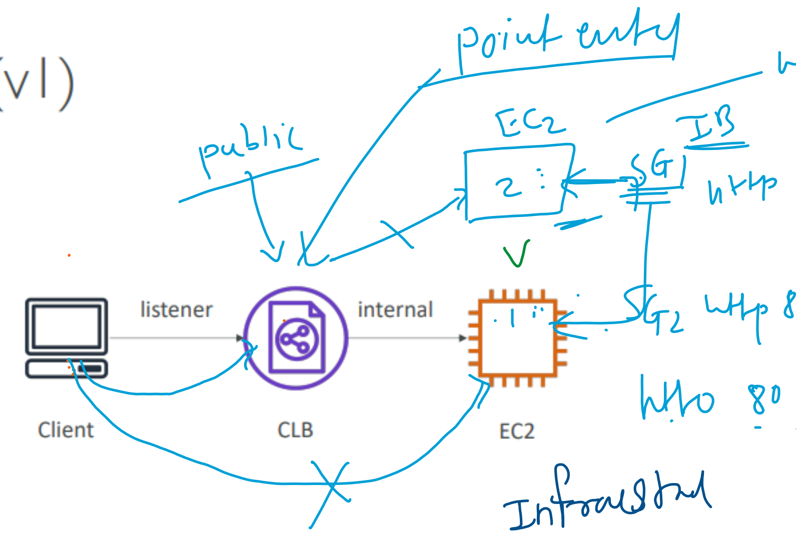
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	~~0.0.0.0/0~~	Allow HTTP from an...
HTTPS	TCP	443	~~0.0.0.0/0~~	Allow HTTPS from a...

Application Security Group: Allow traffic only from Load Balancer

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	sg-054b5ff5ea02f2b6e (load-b	Allow Traffic only...

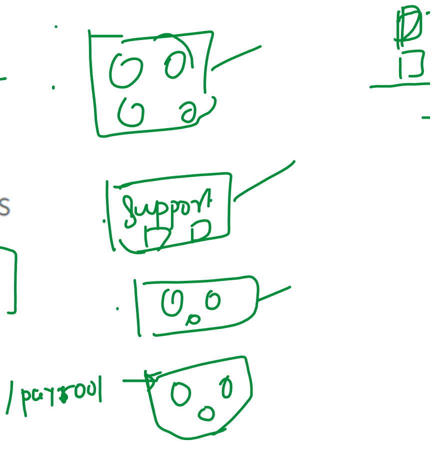
Classic Load Balancers (v1)

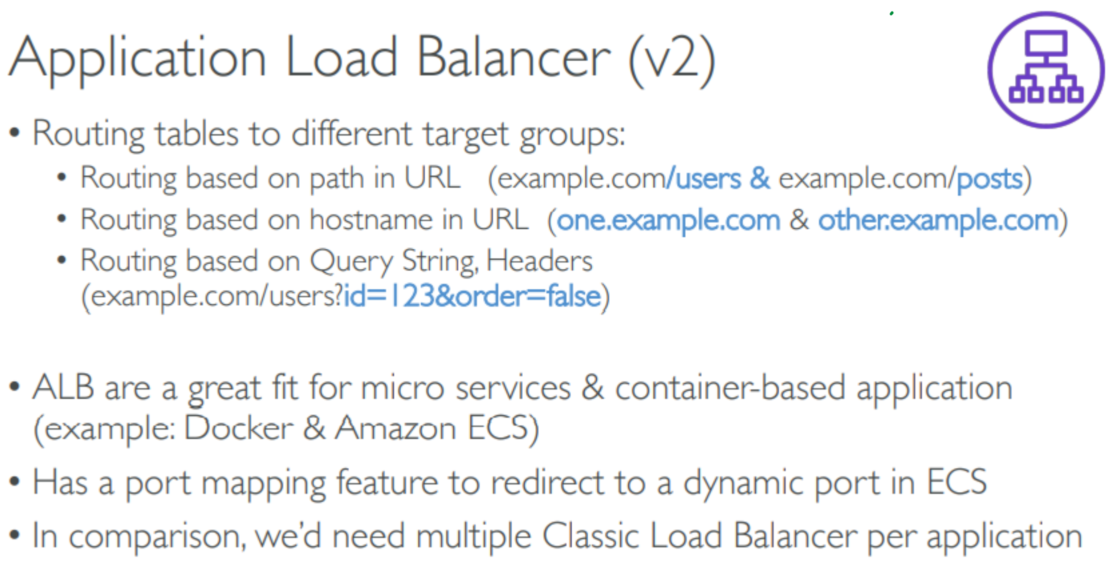
- Supports TCP (Layer 4), HTTP & HTTPS (Layer 7)
- Health checks are TCP or HTTP based
- Fixed hostname XXX.region.elb.amazonaws.com



Application Load Balancer (v2)

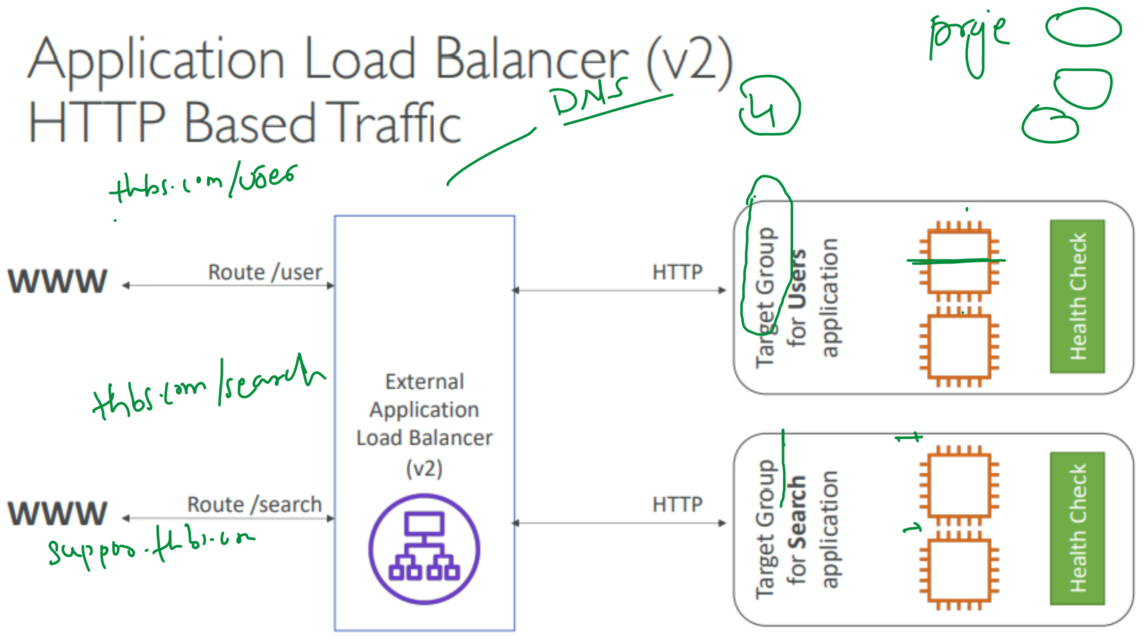
- Application load balancers is ~~Layer~~ 7 (~~HTTP~~)
- Load balancing to multiple ~~HTTP applications~~ across machines (target groups)
- Load balancing to multiple applications on the ~~same machine~~ (ex: containers)
- Support for HTTP/2 and ~~WebSocket~~
- Support redirects (from HTTP to ~~HTTPS~~ for example)





Application Load Balancer (v2)

- Routing tables to different target groups:
  - Routing based on path in URL (example.com/users & example.com/posts)
  - Routing based on hostname in URL (one.example.com & other.example.com)
  - Routing based on Query String, Headers (example.com/users?id=123&order=false)
- ALB are a great fit for micro services & container-based application (example: Docker & Amazon ECS)
- Has a port mapping feature to redirect to a dynamic port in ECS
- In comparison, we'd need multiple Classic Load Balancer per application



Application Load Balancer (v2) HTTP Based Traffic

http://13.127.248.158/static-html-css-web/#[object%20Object]

1. **Create 3 - ec2 machine --- SG - allow http:80 --->0.0.0.0**
   three --> clone -- ip/folder/index.html --->

1. ABL - Target groups--
   1. group1 - first, second ec2
   2. Group2 - three
2. Start creating ALB (internet facing)

   Lister --> **http:80 ----> we decide which target group we need to transfer traffic to**

   --- edit
   Add more-- Rules / conditions based on which we want to distribute traffic

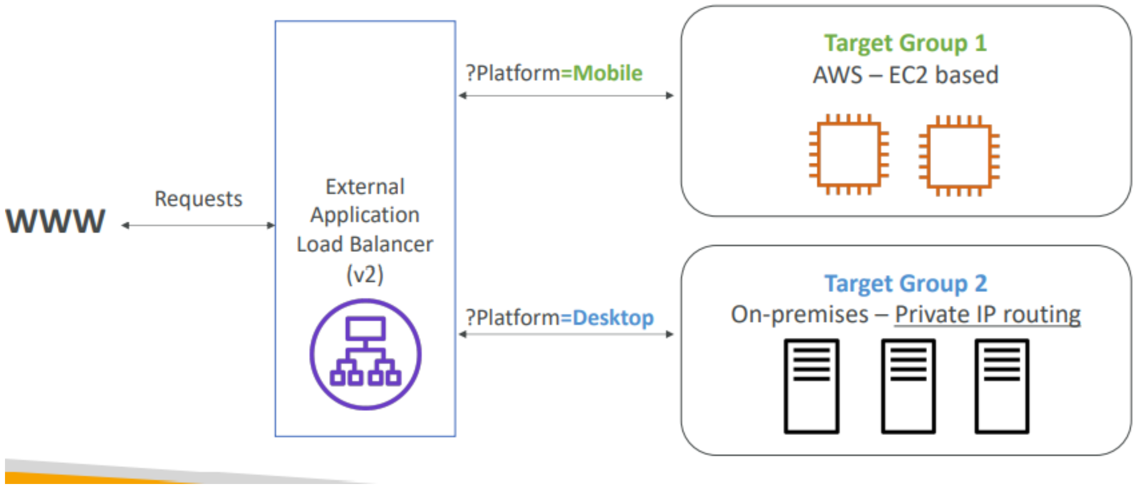
4. DNS of ALB --> rule we will test

Application Load Balancer (v2) Target Groups

- EC2 instances (can be managed by an Auto Scaling Group) – HTTP
- ECS tasks (managed by ECS itself) – HTTP
- Lambda functions – HTTP request is translated into a JSON event
- IP Addresses – must be private IPs

- ALB can route to multiple target groups
- Health checks are at the target group level

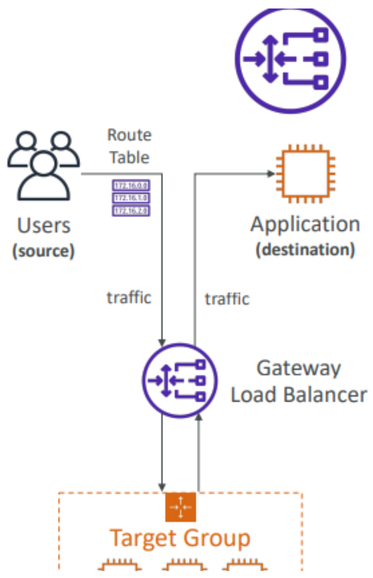
Application Load Balancer (v2) Query Strings/Parameters Routing



Gateway Load Balancer

- Deploy, scale, and manage a fleet of 3rd party network virtual appliances in AWS
- Example: Firewalls, Intrusion Detection and Prevention Systems, Deep Packet Inspection Systems, payload manipulation, …

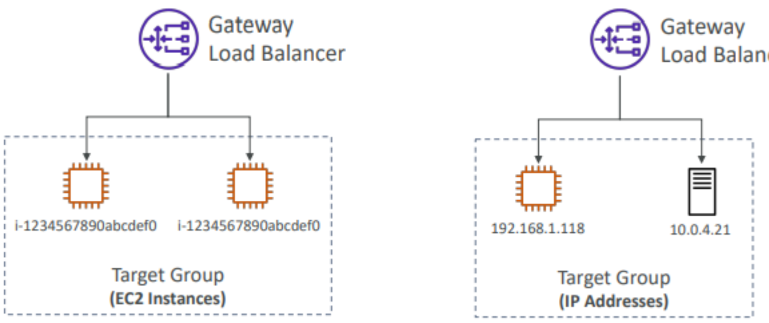
- Operates at Layer 3 (Network Layer) – IP Packets
- Combines the following functions:
  - **Transparent Network Gateway** – single entry/exit for all traffic
  - **Load Balancer** – distributes traffic to your virtual appliances



- Uses the **GENEVE** protocol on port **6081**

Gateway Load Balancer – Target Groups

- **EC2 instances**
- **IP Addresses** – must be private IPs



SSL/TLS - Basics

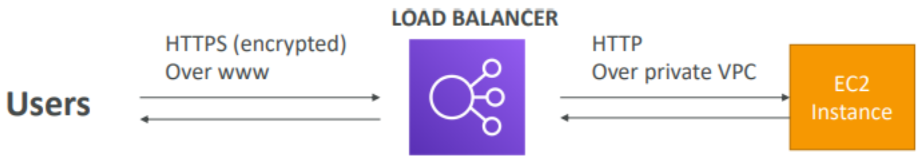
- An SSL Certificate allows traffic between your clients and your load balan to be encrypted in transit (in-flight encryption)

- **SSL** refers to Secure Sockets Layer, used to encrypt connections
- **TLS** refers to Transport Layer Security, which is a newer version
- Nowadays, **TLS certificates are mainly used**, but people still refer as SSL

- Public SSL certificates are issued by Certificate Authorities (CA)
- Comodo, Symantec, GoDaddy, GlobalSign, Digicert, Letsencrypt, etc…

- SSL certificates have an expiration date (you set) and must be renewed

Load Balancer - SSL Certificates



- The load balancer uses an X.509 certificate (SSL/TLS server certificate)
- You can manage certificates using ACM (AWS Certificate Manager)
- You can create upload your own certificates alternatively
- HTTPS listener:
  - You must specify a default certificate
  - You can add an optional list of certs to support multiple domains
  - **Clients can use SNI (Server Name Indication) to specify the hostname they reach**
  - Ability to specify a security policy to support older versions of SSL / TLS (legacy clients)