

# 01-video-notes-tuesday

---

- [Week 8 - AWS: ALB, High Availability, Security Groups \(#tier-1-group-1\)](#)

## Recap odradjenog

- Sta se odradilo prije vezano za nas homework, cloudwatch alert.

## Nastavak - kreiranje nove instance

- Koristili smo AWS Linux AMI i da smo trebali puno toga podesiti da bi ona radila kako treba.
- Mozemo li iskoristiti ove stvari za neke buduće EC2 instance koje ćemo da podizemo?
- Kreiranje nove instance
  - idemo na Search `EC2` - selektujemo `Instances` i `Launch an instance`.
  - Name: `ec2-web-server-week-8`
  - odradimo selektovanje `AWS Linux` Amazon Machine Image-a (AMI).
  - Odabiremo `t2.micro`
  - Kreiramo key pair - key pair name: `week-8`
- Network settings - defaultni VPC ostavljamo takav kakav jeste jer jos nismo obradjivali VPC.

## Dodavanje novih security grupa

- Security grupe
  - Odabiremo `Select existing security group` opciju, međjutim mi ovdje nemamo security grupu sa prethodnog predavanja pa ćemo otvoriti novi EC2 Dashboard i otici u dio za security groups.
    - U ovom dijelu Dzenan je uklonio neke svoje dvije security grupe koje je ranije kreirao. Zatim je kreirao novu security grupu:
  - `EC2 - Security Groups - Create Security group`
    - Name: `web-server-week-8`
  - Po defaultu nova security grupa ima otvorene portove za `outbound` saobraćaj a treba da dodamo `inbound` saobraćaj.
    - Type: `SSH` - Protocol: `TCP` - Port range: `22` - Source: `Anywhere` - Ima security praksa ali dok nemamo svoj VPC i svoju VPN konekciju morati ćemo tako da radimo. Idemo na novi rule za HTTP protokol koji će dozvoliti pristup našoj web aplikaciji kroz browser.
    - Type: `HTTP` = Protocol: `TCP` - Port Range: `80` - Source: `Anywhere`
- Description: Security group created for web server in week-8.

- Kliknemo na `Create security group`.
- Razlika izmedju GiB i GB.

## nastavak sa novokreiranom instancom

- Nakon kreiranja security grupe vracamo se na nasu instancu. Reloadamo i selectujemo novokreiranu security groupu pod opcijom `Security groups`.
- `Configure storage` - ostavljamo na defaultnih 8 GB - gp3.
- Razlika izmedju GiB i GB - razlika je u tome da je GiB preciznija mjera.
- Kada je EC2 instanca lansirana - ona je u statusu pending, cekamo da je Instance state - Running i Status check - OK. Na dijelu:
- `Actions` - `Monitor and troubleshooting` mozemo pronaci logove za instancu.
- Tu isto mozemo kliknuti na opciju `Get Instance screenshot` - gdje cemo dobiti prikaz ekrana nase EC2 instance, da vidimo sta se desava i ako ima neki problem tu ce biti prikazan.

## SSH-iranje u novokreiranu instancu

- Promijenimo permisiju nad kljucem:
  - `chmod 600 *.pem`
  - `ssh -i "instance-key.pem" ec2-user@javna-ip-adresa`
- Prebacujemo se na root:
  - `sudo su -`
- Otvaramo javnu IP adresu u nasem browseru. Nista nece biti prikazano.
- Instaliramo nginx
  - `yum install nginx -y`
  - `systemctl start nginx` - da startujemo, i kada osvježimo browser dobiti cemo welcome message.
- Ne zelimo welcome message, umjesto toga zelimo da prikaze nasu javnu IP adresu. Idemo u dio gdje se nalazi nasa stranica:
  - `vi /usr/share/nginx/html/index.html` - brisemo welcome code i paste-ujemo kod za prikaz nase javne IP adrese.
- `systemctl enable nginx` - da se enable nginx pri bootu instance.
- Stopirao instancu i dao summary sta se odradilo do sada:
  - Uzeli smo Amazon Linux AMI i od njega napravili EC2 instancu.
  - 1 Instalirali nginx
  - 2 Uradili update aplikacije - custom html
  - 3 `systemctl enable nginx`

- To su osnovna 3 koraka koja smo odradili na našem početnom AMI image-u.
- Kad pokrenemo instancu ponovo, dobili smo novu IP adresu, pasteovali je u browser i sad bi trebala se pojaviti nova javna IP adresa.

## Sta ako trebamo podici novu instancu koja ce imati sve karakteristike prethodne? - Custom AMI image

- Morali bi ponovo sve instalirati ispočetka i ponoviti tri prethodna koraka gore.
- Umjesto toga mozemo kreirati `custom AMI image`. Sta to znaci?
- To znaci da:
  - Oznacavamo nasu ec2 instancu (u kvadraticu pored `Name` polja u `Instances` dijelu)
  - zatim kliknuti na `Actions` - `Image and templates` - `Create image`.
  - Image name: ami-week-8
  - Image description: Custom AMI image created from EC2 instance in week 8
  - `No reboot` - ako omogucimo (oznacimo Enable) instanca se nece restartovati tokom procesa kreiranja AMI-a.

## Sta radimo:

- Na početku smo imali jedan AMI image koji smo koristili kao template za nasu EC2 instancu, i sad hocemo da od te EC2 instance napravimo nas custom AMI image koji ce sadrzavati sve izmjene koje smo manuelno dodali pa necemo morati ponovo sve da radimo kada budemo dizali novu EC2 instancu.
- Ta nova instanca ce automatski sadrzavati sve izmjene koje smo mi prethodno kreirali na prvoj EC2 instanci kroz nas AMI image.

## Nastavak kreiranja AMI image-a

- u `Instance volumes` vidimo da osnovni dio tog AMI image-a je i ovaj storage (EBS ili Elastic Block Storage).
- Kliknemo na `Create Image`
- Kreiran je novi AMI image koji ima svoj ID. Kada kliknemo na taj ID vidimo da se kreira nas AMI image koji ima `Root device name` `/dev/xvda`.

## Root device - brisanje prethodne instance

- EBS koji je bio zakacen na prethodnu EC2 instancu.
- Sad mozemo izbrisati prethodnu EC2 instancu - `Instance state` - `Terminate`.
- Medjutim AMI image koji smo mi prethodno kreirali i dalje postoji bez obzira sto smo terminirali EC2 instancu.

- Dalje ako smo dobili zadatak da podignemo novu instancu koja ce imati prethodno instalirano sve sto smo radili na prosloj uklonjenoj EC2 instanci, mi mozemo kreirati novu instancu iz ovog AMI image-a.

## Kreiranje nove instance iz prethodno kreiranog AMI image-a

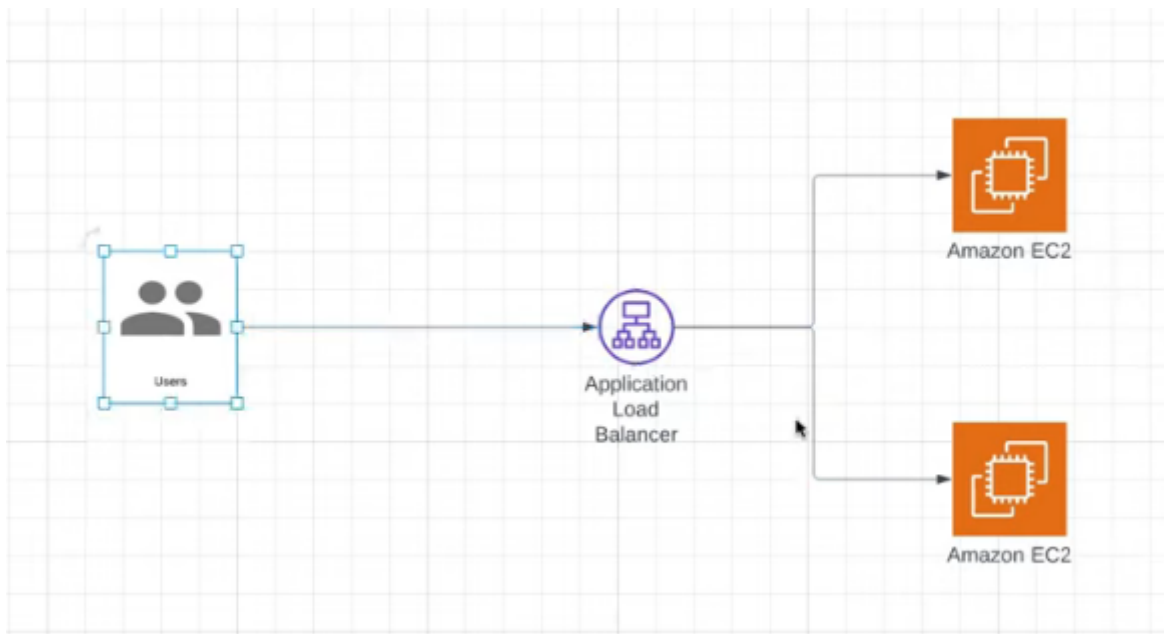
- Vracamo se na `Instances` - `Launch Instance`
- U dijelu gdje biramo AMI image kliknemo na `My AMIs` vidimo da nam je dostupan AMI image kojeg smo mi prethodno kreirali (ami-week-8).
- Takodjer ako se vratimo na `Instances` - ispod imamo `Images` section gdje imamo `AMIs` i gdje mozemo vidjeti nas prethodno kreiran AMI image.
- Kliknemo na `Launch Instance from AMI` i kreiramo novu EC2 instancu.
  - Name: ec2-web-server
  - Key pair (login): odabiremo key pair
  - Firewall (security groups): odabiremo security group koju smo ranije kreirali pod nazivom `web-server-week-8`
  - Configure storage: po defaultu ostavljamo 8 GB - gp3.
  - Kliknemo na Launch instance da se kreira instanca.

## Dalje

- Ako uzmemo javnu IP adresu novokreirane instance otvorimo u browseru vidjet cemo output nase jave IP adrese bas kao na prethodnoj instanci.
- Vidimo koliko nam je olaksao posao kreiranje AMI image-a kao template-a za nove instance.

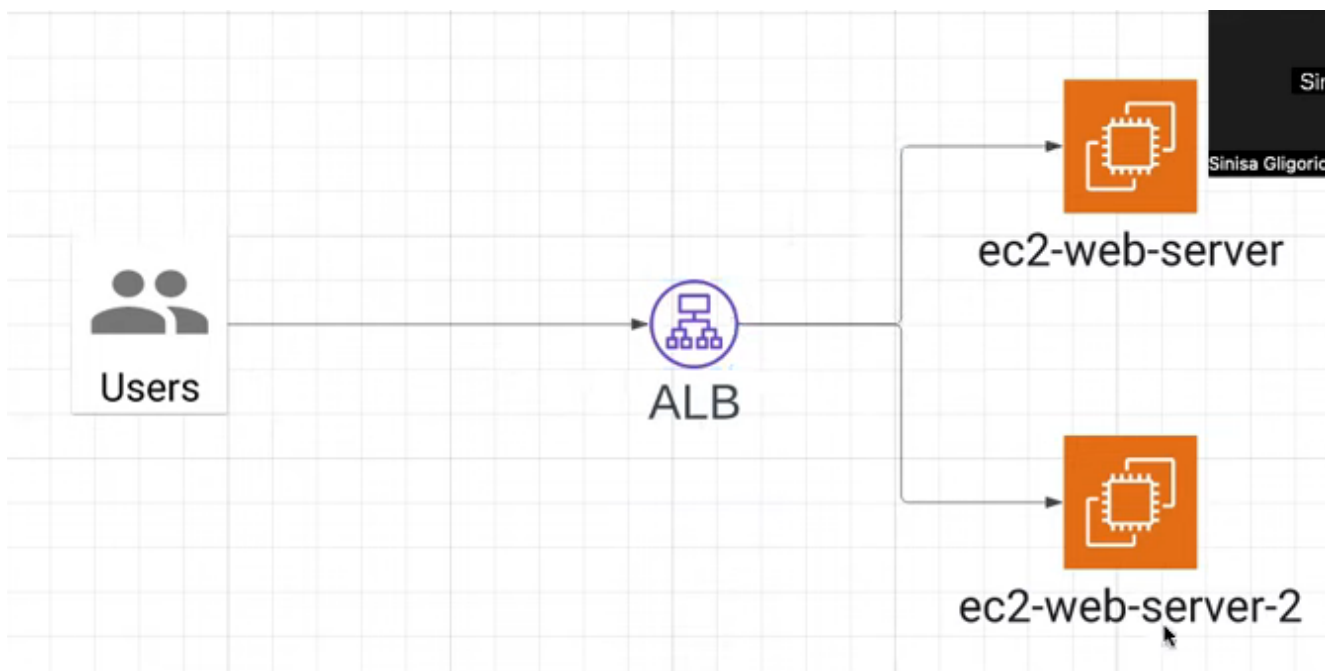
## Visoka dostupnost aplikacija - load balancer

- Trenutno imamo svoju EC2 instancu koja ima svoju public IP i imamo vise korisnika koji pristupaju toj EC2 instanci direktno preko njene IP adrese.
- Kakav god problem da se dogodi sa ovom EC2 instancu mi mozemo podici novu iz nased AMI image-a (AMI Week 8) i nasim korisnicima poslati novu IP adresu koju ce oni iskoristiti da posjete nasu web stranicu.
- Ali to nije efektivan nacin kako bi mi oporavili svoju aplikaciju u slucaju da dodje do problema. Jest neki disaster recovery plan ali nije bas najefektivniji.
- To mozemo da rijesimo da ispred nasih instanci postavimo `load balancer`.
- Load balancer ce da raspoređuje saobraćaj po nasim EC2 instancama u pozadini i u slucaju da jedna od te dvije EC2 instance ne bude dostupna nas load balancer ce zahvaljujuci svom `health check`-u prepoznati koja od ove dvije instance nije dostupna i nastaviti da preusmjerava saobraćaj prema instanci koja je dostupna.



## Kreiranje IALB-a koji cemo staviti ispred nase EC2 instance

- Trenutno imamo pokrenutu jednu EC2 instancu - znamo da nam trebaju minimalno dvije EC2 instance iza naseg load balancera zato sto ako jedna prestane sa radom, load balancer proslijedjuje incoming traffic na drugu.
- Za ovu svrhu kreiramo dodatnu EC2 instancu:
  - Idemo u `Images` - `AMIs`
  - Oznacimo nas AMI i kliknemo na `Launch Instance from AMI`
  - Imenujemo nasu instancu: `ec2-web-server-2`
  - Odabiremo isti AMI image, week-8 key pair, istu security grupu kao prije, isti storage i kliknemo na `Launch instance`.
  - Idemo nazad na `Instances` da provjerimo novokreiranu instancu.
- Sada cemo imati 2 web servera ispred kojih cemo staviti nas load balancer.



- 
- Application Load Balancer skraceno mozemo nazvati `ALB`.
- sada cemo nas ALB konfigurirati da proslijedjuje saobraćaj prema ovim EC2 instancama.

## Konfigurisanje ALB-a

- Da provjerimo da li nasi serveri run-iraju, mozemo filtrirati na `Instances` tako sto u Find kucamo `Instance state = running` i output ce biti 2 instance koje run-iraju tj. nasa 2 web servera.
- Sad cemo postaviti ta 2 web servera iza load balancera.
- Na lijevoj strani idemo u tab ispod Instances i Images koji se zove `Load Balancing` i kliknemo na `Load Balancers`.
  - Kliknemo na `Create load balancer`

## Digresija na tipove ALB-a:

- Biramo izmedju 3 load balancera: application, network i gateway.
- Gateway koristimo ako supportamo third party aplikacije ili resurse.
- Network koristimo kada imamo jedan dio resursa unutar AWS clouda a drugi unutar naseg on-premises datacentra jer moze da koristi IP adresu naseg servera unutar on-premises datacentra. Radi na mreznom nivou OSI layera.
- Aplikativni ALB radi na 7. nivou OSI layera (aplikativni nivo) - imamo opciju da radimo manipulaciju nad headerom naseg requesta. Sa network load balancerom nemamo tu mogucnost.
- `Vazno` je znati tipove load balancera kako za day-to-day rad tako i za certifikaciju.

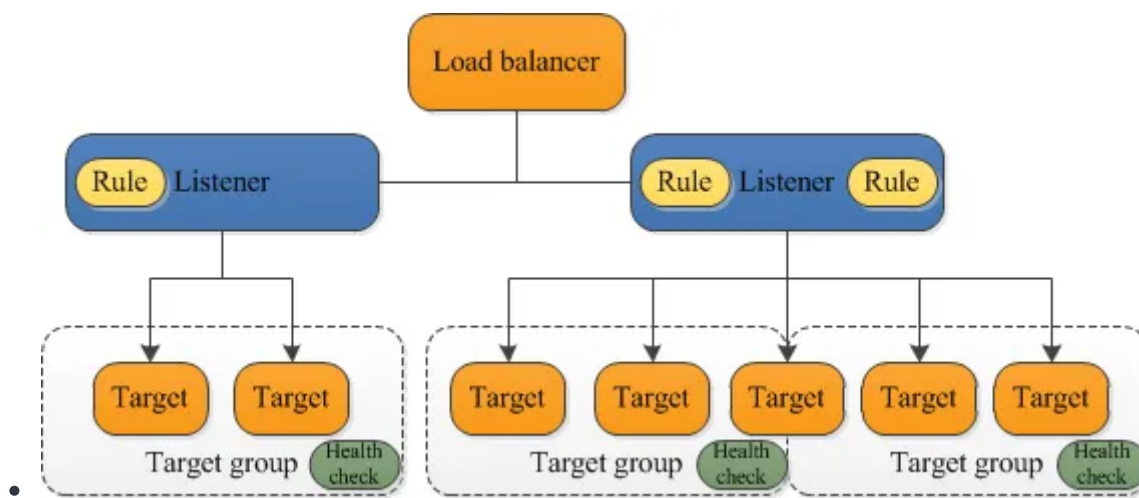
## Nastavak konfigurisanja load balancera

- Nazad na `Load Balancer types` kliknemo na `Create` ispod `Application Load Balancer` jer nam taj tip treba.
  - Load Balancer name: alb-web-server-week-8

- Scheme: kliknemo na `Internet-facing` (za pristup sa interneta) jer ako izaberemo `internal` onda nas load balancer nece biti dostupan van AWS sto znaci da mu mozemo pristupiti samo ako se nalazimo unutar naseg VPC-a.
- IP address type po defaultu - IPv4.
- VPC - samo defaultni trenutno.
- Mappings - availability zone za nas load balancer - minimalno dvije izabrati, npr. us-east-1a i us-east-1b.
- Security groups
  - `Vazno` - kreiramo posebnu sec grupu za nas ALB i trebamo imati na umu da smo prethodno kreirali security grupu za nasa 2 web servera gdje imamo dozvoljene portove 22 i 80.
- Koje inbound rules treba imati sec grupa na ALB-u? Mi znamo da nasoj aplikaciji ce pristupati korisnici preko ALB-a na nacin da ce svoj request upucivati ALB-u a ALB ce kasnije taj request proslijedjivati web serverima.
  - Na osnovu toga mozemo reci da sec grupa na ALB-u ima inbound port 80.
  - Kliknemo `Create new security group`
  - Sec group name: alb-sg-web-server
  - Description: Sec Group used for ALB for web servers.
  - Inbound rules: `Add rule` samo HTTP, source Anywhere.
  - Kliknemo na Create Security group.
- Nakon kreiranja sec grupe vracamo se nazad na konfigurisanje ALB-a:
  - refresujemo Security groups dio i odabiremo sec group koju smo kreirali za load balancer.

## Kreiranje target grupe pri konfiguraciji ALB-a

- Sta su target grupe? Predstavljaju logicko grupiranje EC2 instanci ispred kojih se nalazi ALB.
- Unutar zadane target grupe cemo postaviti nase dvije EC2 instance, tj. nasa 2 servera i onda nas ALB ce znati prema kojim EC2 instancama da proslijedjuje saobracaj - zato sto se ova target grupa veze za load balancer.

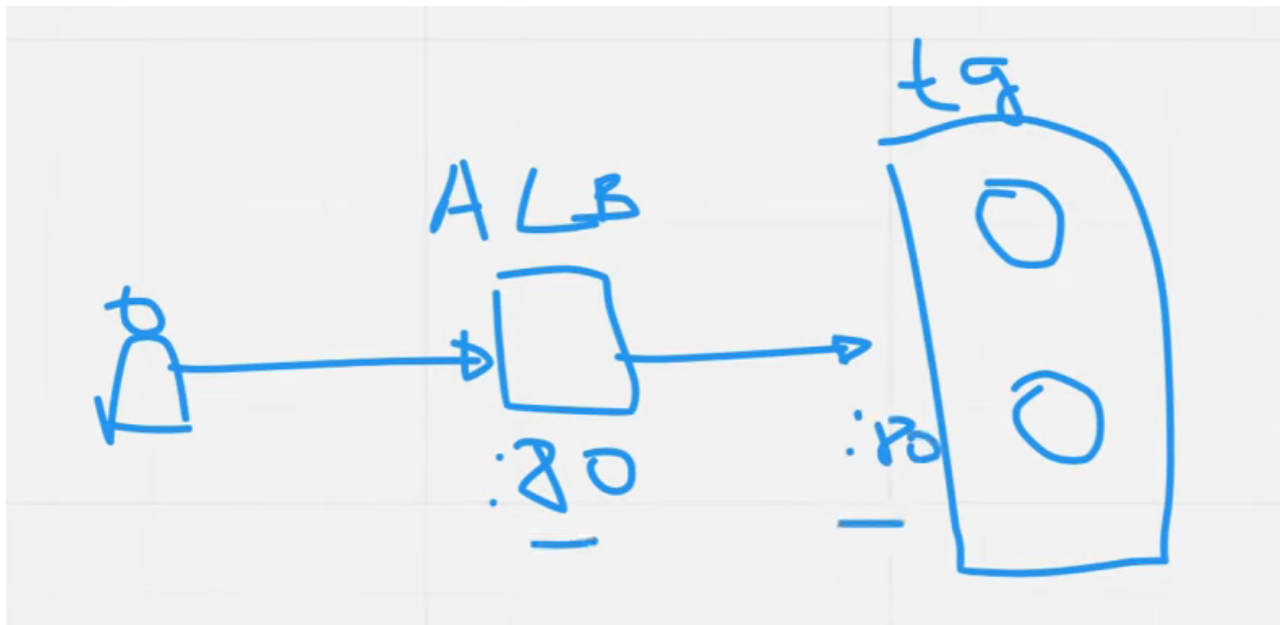


- Kako ALB proslijeđuje requestove? Tako što se definiše `listener rule` koji će definisati na koji način mi proslijeđujemo requestove prema našim EC2 instancama koje se nalaze unutar te target grupe.
- Da definišemo target grupu, nakon kreiranja nove security grupe, kliknemo na `Create target group` i poslije toga:
  - Choose a target type: Instances
  - Name: tg-alb-web-server
  - VPC po defaultu.
  - Health check - komunikacija između ALBa i naše EC2 instance, ping-a je kako bi provjerio je li u healthy stanju. Možemo provjeravati neku specifičnu putanju preko Health check path a možemo raditi health check i preko port-a (Advanced health check settings). U ovom slučaju kliknemo na `Override` i port 80 - u ovom slučaju health check se radi preko porta 80.
  - Kliknemo na `Next` i idemo na Register targets gdje kliknemo na naše dvije EC2 instance (označimo ih samo).
  - Kliknemo na `Include as pending below`.
  - Kad vidimo da su dodane ispod, kliknemo na `Create target group`.
- Sad imamo kreiranu target grupu u koju se dodaju 2 EC2 instance, treba pričekati pola minute - minut da budu u statusu Healthy.

## Nazad na ALB konfiguraciji

- Odabiremo target grupu (tg-alb-web-server) koju smo prethodno kreirali i sad mi ovdje bukvalno povezujemo target grupu sa load balancerom.
- Sta se odradilo do sada:
  - Imamo naš ALB koji smo povezali sa njegovom target grupom, unutar target grupe imamo naše 2 EC2 instance. ALB će primiti saobraćaj po portu 80 i proslijediti će requestove koje primi od strane korisnika prema target grupi. Isto, target grupa prima requestove na portu 80 jer smo tako definisali.





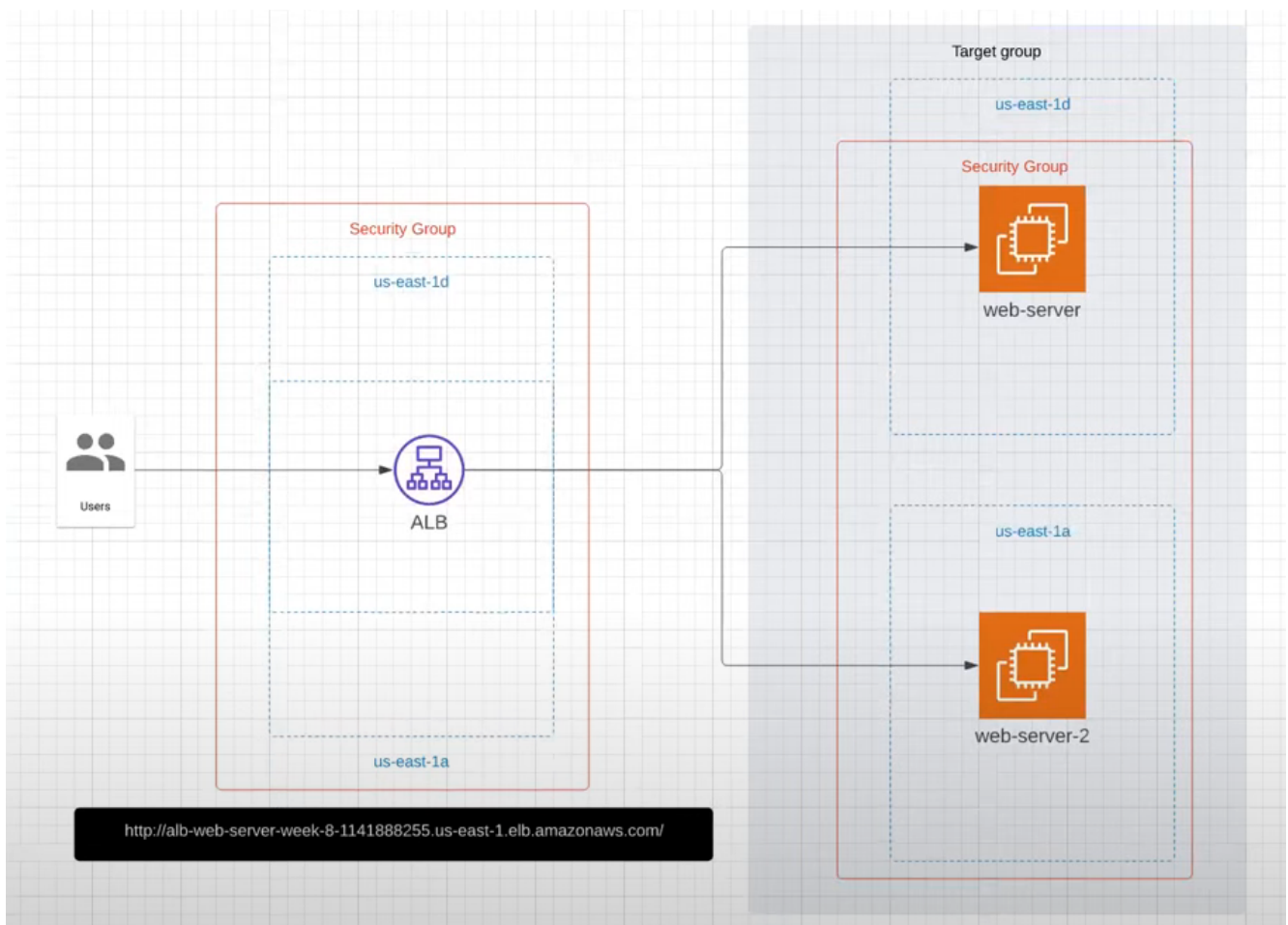
- o Medjutim, pored toga, nas ALB ima mogucnost da radi health check te on konstantno ping-a EC2 instance i provjerava da li su one dostupne na portu 80. Onog trenutka kad ne budu dostupne, on ce prestati da im proslijedjuje saobracaj.
- o Ukoliko instanca ne bude dostupna, ALB proslijedjuje request prema sljedecoj koja zadovolji health check.
- o Sljedeci korak je kliknuti `Creat load balancer`.

## DNS name - Listeners - Inicijalizacija instanci

- Poslije toga vidimo nas load balancer u State: Provisioning.
- Taj load balancer dobija svoj `DNS name` koji mozemo koristiti u browseru da pristupimo nasim EC2 instancama tj. koristimo endpoint od ALB-a, jer krajnjeg korisnika ne zanimaju IP adrese instanci nego njega zanima koji je to endpoint load balancera.
- Na opciji `Listeners` mozemo vidjeti protocol i Default routing rule gdje vidimo da sav saobracaj preusmjerava (Forward to) nasoj target grupi. Kad kliknemo na tu target grupu vidimo da se radi inicijalizacija nasih ec2 instanci.
- Sta znaci inicijalizacija? Znaci da te ec2 instance postaju vidljive load balanceru, znaci dodaju se u target grupu. Treba imati na umu, dok Health Status nije Healthy, ALB im nece preusmjeravati saobracaj.
- To mozemo vidjeti ako paste-amo nas DNS point u web browser.

## Ista availability zona za ALB i EC2 instance

- `EC2` instance (obje) moraju biti u istoj availability zoni u kojoj se nalazi nas load balancer inace im nece proslijediti requests.
- Prilikom kreiranja ALB-a moramo dodati availability zone nasih instanci.



- 
- Na našem ALB-u imamo security grupu koja dozvoljava pristup po portu 80.

## Ogranicavanje pristupa po portu 80

- Web serveri (instance) imaju svoju security grupu koja svima dozvoljava pristup po portu 80 - što nama sad više ne odgovara jer ne želimo da korisnik može pristupiti EC2 po portu 80 nego samo preko ALB-a.
- Cilj je ograniciti pristup - da su serveri dostupni po portu 80 samo za saobraćaj koji dolazi od strane ALB-a.
- Kako to možemo odraditi?
  - Vraćamo se na našu Management Console i pod `Network & Security` biramo `Security groups`.
  - Tu vidimo `web-server-week-8` i `alb-sg-web-server` security grupe.
- Najbolja praksa je sljedeća: - 01:09:50 na videu - u slučaju da treba review.
  - Vratiti se na `Security groups` i našu `web-server-week-8` security group i kliknuti na `Edit inbound rules`.
    - Uklonimo postojeće rule, kliknemo na `Add rule`, kućamo HTTP, Source - izabrati Custom, ukucati `alb-sg-web-server`.
    - U Description upisemo: Allow inbound traffic via port 80 only from ALB.
    - `Save rules`

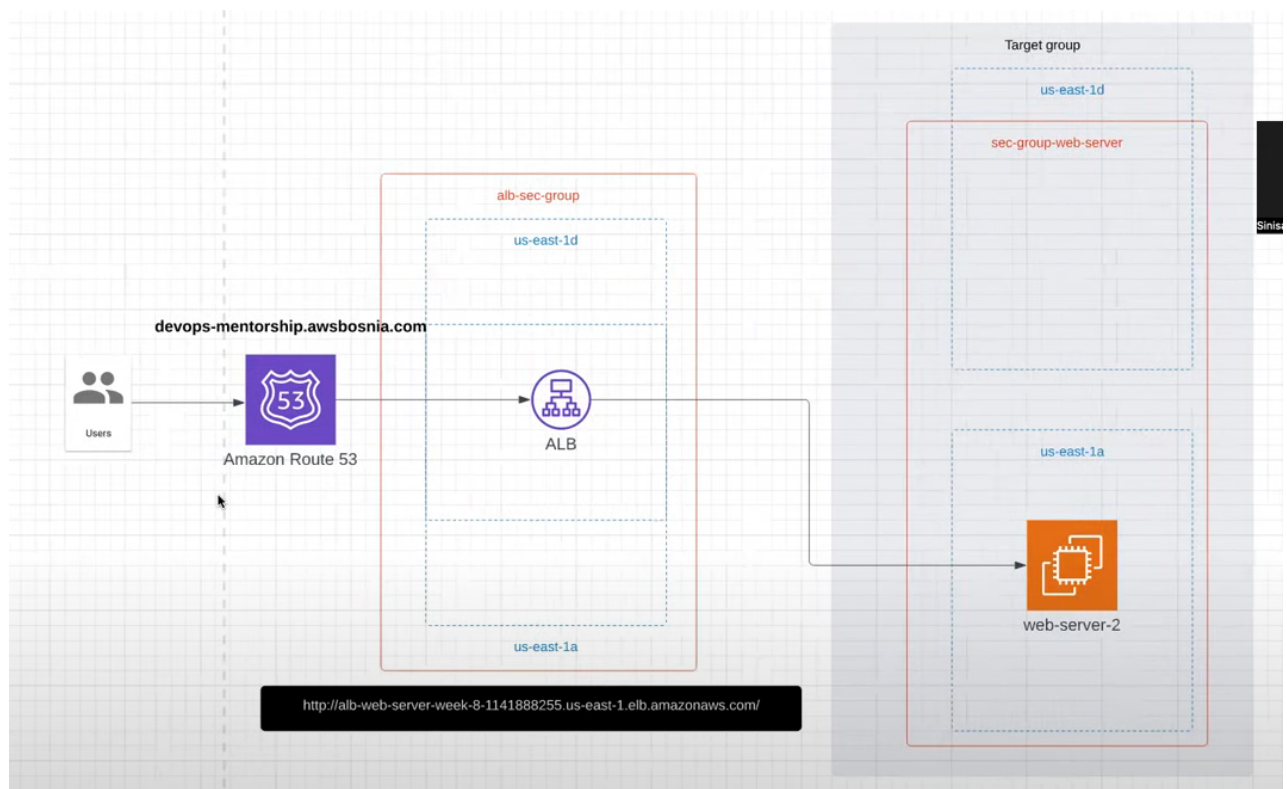
- Na ovaj nacin smo vidjeli kako sec grupe se mogu iskoristiti da ogranice saobraćaj ne samo sa IP adresa nego gdje je source neka druga sec grupa.
- `sudo cat /var/log/nginx/access.log` - da se pogleda Health check koji ALB salje instanci. Requestovi dolaze sa vise razlicitih IP adresa - ovom slucaju 3 jer smo nas ALB stavili u 3 availability zone.

## Dodavanje domena - Amazon Route 53

- Krajnjem korisniku ne saljemo DNS name naseg load balancera nego bi trebali da imamo nego domensko ime, koje mozemo imati unutar servisa `Amazon Route 53`.
- Ovaj servis nam omogucava da imamo nasu domenu i da imamo DNS server gdje cemo mi pristupati nasoj aplikaciji preko domenskog imena a Route 53 ce vrsiti razvrstavanje tog domenskog imena i vrsiti preusmjerenje.
- Dzenan demonstrira na svom drugom racunu kako sve radi koristeći svoju domenu `awsbosnia.com` - demo na min. 01:15:50

## Sta ce se dogoditi ako jedan od nasih web servera postane nedostupan?

- Primjer kad terminiramo instancu, sta se desava na nivou ALB-a?
- Na nivou ALB-a imamo target grupu koja registruje da jedna od dvije instance nije healthy i onda cemo imati total targets 1, necemo vidjeti drugu instancu.
- U browseru cemo dobiti Gateway timeout nakratko, nasa aplikacija ce nastaviti da radi medjutim jedan request je prosao prema unhealthy instanci.
- Mozemo odraditi fine tuning da load balancer ranije prepozna da instanca nije healthy i da ne prosljedjuje requests prema istoj.
- Na ALB Dashboardu u `Monitoring` mozemo vidjeti broj healthy host-ova koji je pao sa 2 na 1, tako da imamo situaciju da ALB preusmjerava traffic prema samo jednom web serveru.



- Bukvalno sta treba da radimo je da podizemo novu EC2 instancu nakon alarma po prethodnoj proceduri preko AML image-a - sto smo odradili prije.
  - Idemo na svoju target group i kliknemo na `Register targets` gdje otvoriti ce novi window da dodamo novu instancu tako sto kliknemo da selektujemo novu instancu, te zatim kliknemo na button - `Include as pending below`.

## Auto Scaling grupa - samo manji uvod

- Ovo je bio manuelni proces povratka tj. dodavanja nove instance, imamo visoku dostupnost.
- Sve je radilo super dok imamo jednu healthy instancu, ali sta ako je bio problem sa obje instance? Morali bi rucno da podignemo 2 EC2 instance i dodamo unutar load balancera, sto znaci da nam treba neki mehanizam koji ce to raditi automatski. Taj mehanizam se naziva Auto Scaling grupa.

## plan za iduce predavanje

- O Auto Scaling grupama u cetvrtak.
- Iduci utorak
  - <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-attaching-volume.html>

## ▼ Storage

### ▼ Amazon EBS

#### ▼ EBS volumes

##### ► EBS volume types

Size and configuration constraints

Create a volume

**Attach a volume to an instance**

Attach a volume to multiple instances

Make a volume available for use

View volume details

Replace a volume

Replace a root volume

Monitor the status of your volumes

Detach a volume from an instance

Delete a volume

◦ Fault testing

- ALB fine tuning