

VISOKO UČILIŠTE ALGEBRA

PROJEKTNI ZADATAK

Implementacija računarstva u oblaku - Azure

Antonio Janach

Zagreb, lipanj 2021.

Sadržaj

1. Uvod.....	1
2. Zahtjevi infrastrukture	2
3. Opis infrastrukture.....	3
4. Topologija infrastrukture	4
5. Razrada projekta - projektno rješenje	5
5.1. Kreiranje resursne grupe	5
5.2. Mrežna konfiguracija u Azure platformi	6
5.3. Kreiranje virtualnih mašina u Azure platformi	8
5.4. NGINX - konfiguracija <i>Load Balancer-a sa SSL/TLS enkrijcijom</i>	9
5.5. WordPress instalacija.....	11
5.5.1. WordPress instalacija na WP01	11
5.5.2. WordPress instalacija na WP02	13
Popis slika.....	15
Popis tablica.....	15
Zaključak	16
Literatura	17

1. Uvod

Ideja upotrebe samog rješenja u oblaku bazira se na sistemu plaćanja resursa po korištenju usluge. Korištenje rješenja u oblaku, korisnici i klijenti ne kupuju opremu niti ulažu u izgradnju informacijskih sustava. Već se ta oprema iznajmljuje od pružatelja usluge. Prednosti koje nosi takav način poslovanja je taj što se poslodavci mogu lakše usredotočiti na razvoj systemske infrastrukture, razvoj softverskih rješenja i ostalih rješenja bez potrebe za kupnjom fizičkog hardvera.

Rješenja u oblaku koncept je koji se u zadnjih par godina često spominje, no takva rješenja su već dostupna godinama. Neko od poznatih davatelja usluga takvih rješenja su:

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- IBM Cloud
- Digital Ocean

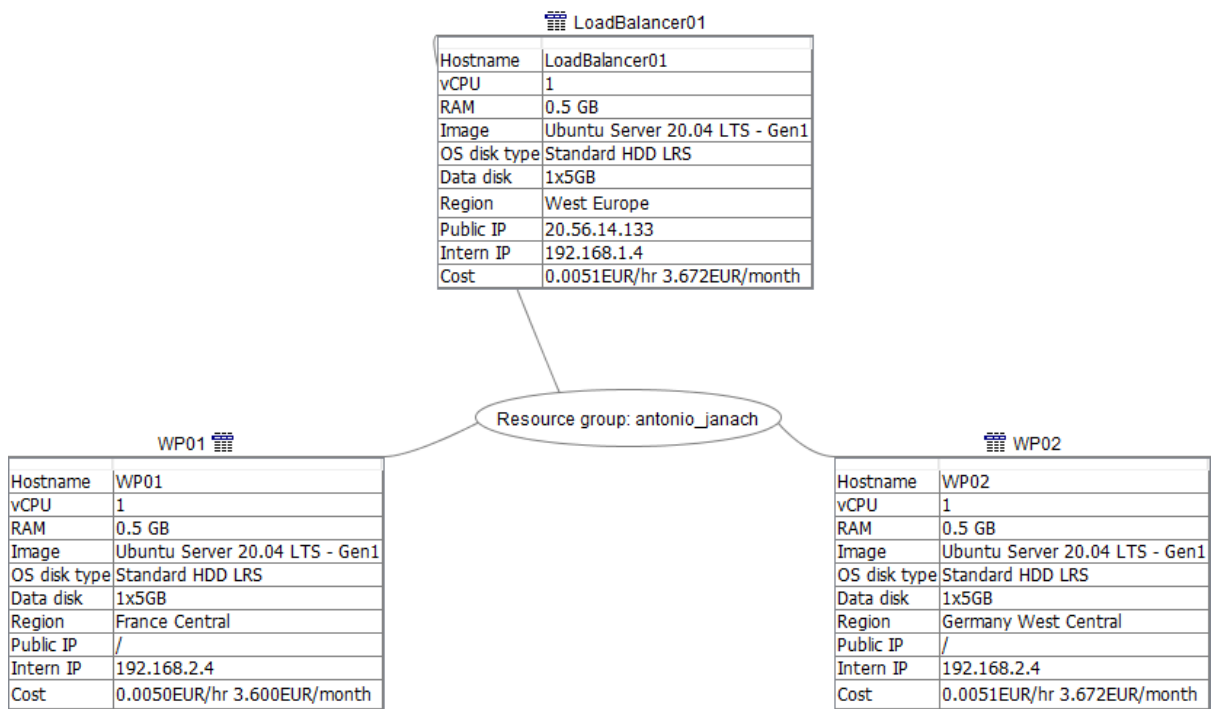
Iako su navedeni davatelji usluga za rješenja u oblaku već duži niz godina prisutni, u posljednje vrijeme sve se više tvrtki, manjih i većih, banaka a i drugih institucija okreće takvim rješenjima i za vlastite proizvode. Tome je pridonijela tvrdnja da su se brzine interneta posljednjih godina znatno povećale. Samim time trenutno je i više nego ikad dostupno obavljanje raznih zadataka s udaljenih lokacija koristeći internet.

Cilj ovog projekta je kreirati tri objekta u jednu resursnu grupu na Azure platformi. Prvi objekt mora biti javno dostupan, a to je load balancer. Taj prvi objekt mora preusmjeriti zahtjev koji mu je dan od strane korisnika prema pozadinskim WordPress serverima kojima je IP privat. Što bi značilo da je WordPress sadržaji moguće pristupiti samo preko Load Balancer-a. Drugi objekt i treći objekt su WordPress serveri. Prvi objekt mora preusmjeriti svoj dan zahtjev prema WordPress sadržaju koristeći HTTPS promet.

2. Zahtjevi infrastrukture

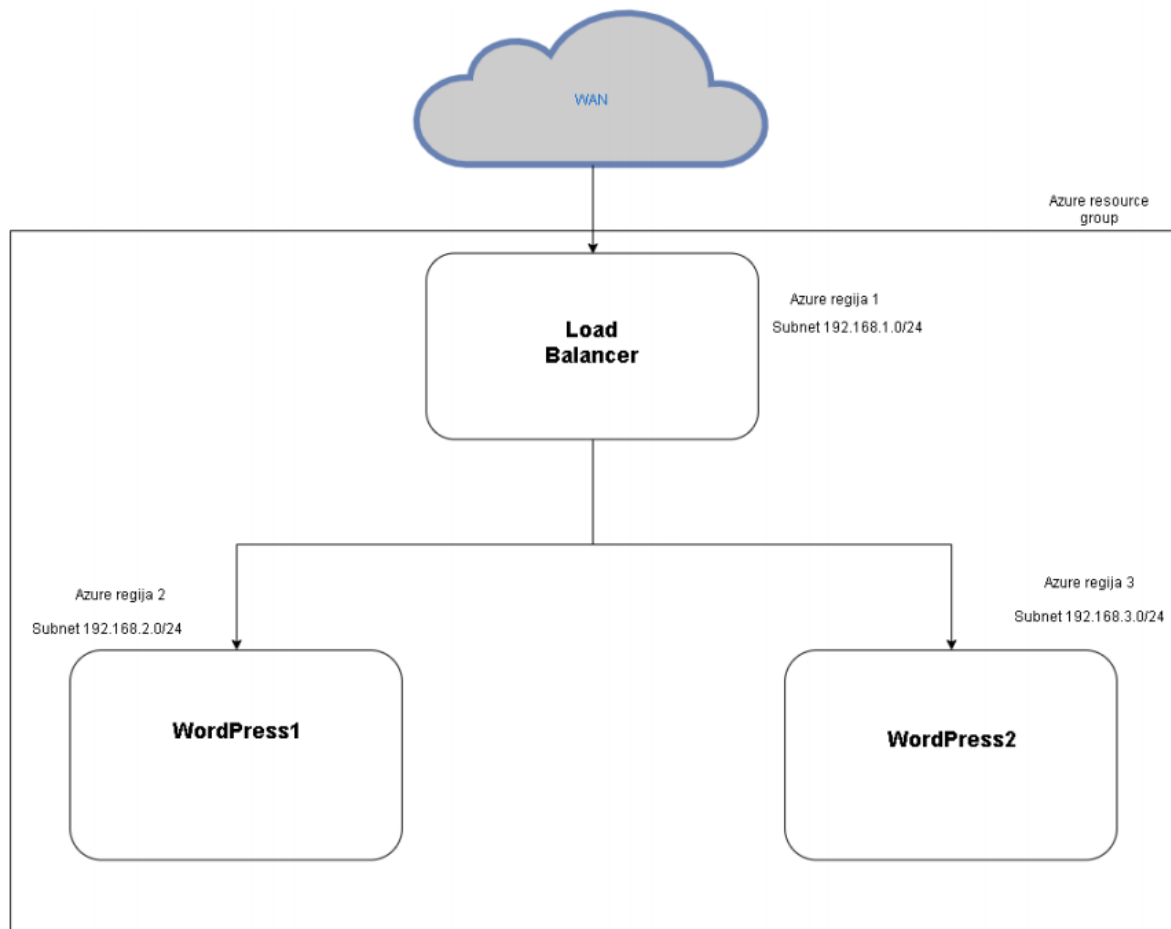
- a) U Azureu napravite jednu resource grupu imena ime_prezime
- b) Prva premisa projekta je napraviti kompletnu konfiguraciju tako da cjelokupna infrastruktura troši što manje novca unutar rješenja koje ste dizajnirali. Primjera radi, ako odaberete rješenje kroz virtualke, odaberite regije gdje su virtualke najjeftinije, sa najjeftinijim storage-om i sa najjeftinijom konfiguracijom. Virtualke moraju biti Ubuntu 18.04, minimalne po svim pitanjima. To dokumentirajte u dokumentaciji projekta – koje ste točno izbore napravili pri konfiguraciji.
- c) U resource groupi instalirajte tri objekta (virtualne mašine, app servise, kako odaberete). Svaki objekt mora biti u posebnoj Azure regiji. Shodno tome, konfigurirajte po jednu Azure virtualnu mrežu za svaku regiju. Neka to budu adrese 192.168.1.0/24, 192.168.2.0/24 i 192.168.3.0/24.
- d) Prvi objekt (VM, app servis, neki drugi Azure servis po odabiru) mora biti javno dostupan L7 load balancer/reverse proxy. Primjera radi, možete ga instalirati kao Ubuntu virtualnu mašinu sa Apacheom, nginx-om ili haproxy-em (nekoliko potencijalnih sugestija). Što god odabrali, mora biti u pitanju objekt koji će moći dobiti permanent public IP. Kada mu dodijelite permanentni javni IP, nakon toga, dobiti ćete host zapis u domeni vuacloud koji će glasiti na vaše imeprezime (dakle, imeprezime.vua.cloud).
- e) Load balancing prema pozadinskim WordPress serverima mora ići po Azure virtualnim mrežama, i pozadinski WordPress serveri ne smiju imati javne IP adrese.
- f) Load balancing algoritam može biti koji god – least connections, round robin, nebitno.
- g) Drugi objekt mora biti WP instalacija sa bar jednim aktivnim člankom, sadržaja “ovo je prva web stranica studenta Ime Prezime”.
- h) Treći objekt mora biti WP instalacija sa bar jednim aktivnim člankom, sadržaja “ovo je druga web stranica studenta Ime Prezime”.
- i) Ako se sa strane Interneta LB-u pristupa preko HTTP, mora biti napravljena automatska redirekcija na HTTPS. Iskoristite certifikat dostupan na IE. Možete (ali i ne morate) složiti SSL offload (da LB/RP prihvaća konekcije na HTTPS i u pozadini komunicira sa WP serverima isključivo preko HTTP protokola).
- j) Datoteke od WordPressa mogu biti u virtualki ili izvan virtualke (blob, CDN, što god odaberete, ali mora biti u Azure infrastrukturi). Ne mora postojati sinhronizacija sadržaja između te dvije virtualke, ali ako želite napraviti implementaciju nekog oblika replikacije pa da sadržaj bude sync'd, slobodno, i u dokumentu navedite kako ste to ostvarili. Isto se odnosi i na MySQL bazu/baze.
- k) Kada ne radite na projektu, ugasite virtualne mašine ako ste njih odabrali.
- l) U dokumentaciji navesti tekstualno koliko vam je Azure naveo da će svaki objekt koštati.

3. Opis infrastrukture



Slika 1: prikaz opisa infrastrukture kroz umnu mapu

4. Topologija infrastrukture



Slika 2: prikaz topologije infrastrukture

5. Razrada projekta - projektno rješenje

5.1. Kreiranje resursne grupe

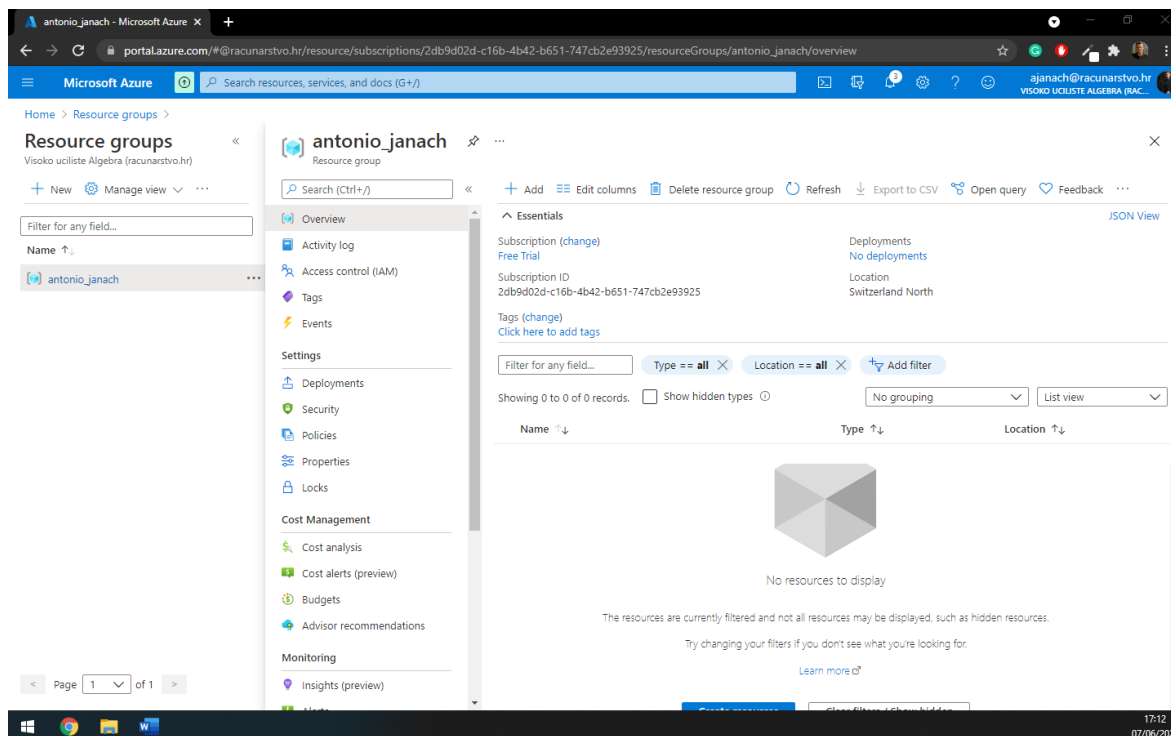
Trenutno postoje dva modela implementacija resursa u Azureu, a to su Resource Manager model i klasični model implementacije. Azure je izvorno nudio samo klasični model implementacije, u tom modelu je svaki resurs nezavisan i nije bilo načina za grupiranje povezanih resursa zajedno. U 2014. godini Azure predstavlja Resource Manager koji donosi koncept grupacije resursa.

Resursna grupa je spremnik za resurse koji dijele zajednički životni ciklus. Grupe resursa je spremnik izrađen u Azure regiji koju odredi korisnik, u kojoj su smješteni resursi Azure oblaka. Resursna grupa može biti skup virtualnih računala, aplikacijskih servisa, ili nekih drugih servisa koje Azure kao takav nudi.

Resource Manager model implementacije nudi neke od prednosti:

- Implementacija, upravljanje i nadzor svih servisa neko rješenja kao jedne grupe
- Opetovano implementiranje rješenja tijekom životnog vijeka uz sigurnost da su resursi implementirani na konzistentan način
- Primjenjivanje politike kontrole pristupa nad svim resursima u grupi resursa gdje se te politike automatski primjenjuju na novo dodane resurse
- Primjenjivanje oznake resursa za logično organiziranje svih resursa u Azure pretplati
- Definiranje ovisnosti među resursima

Kako bi se kreirala resursna grupa potrebno je otvoriti Azure portal gdje je iz izbornika potrebno odabrati „Resource groups“. Potom pritisnuti na „Add“ te upisati podatke koji su traženi.



Slika 3: resursna grupa

Od sada nadalje u resursnu grupu mogu se instalirati objekti, u ovog slučaju projekta unutar ove resursne grupe nalazit će se tri objekta.

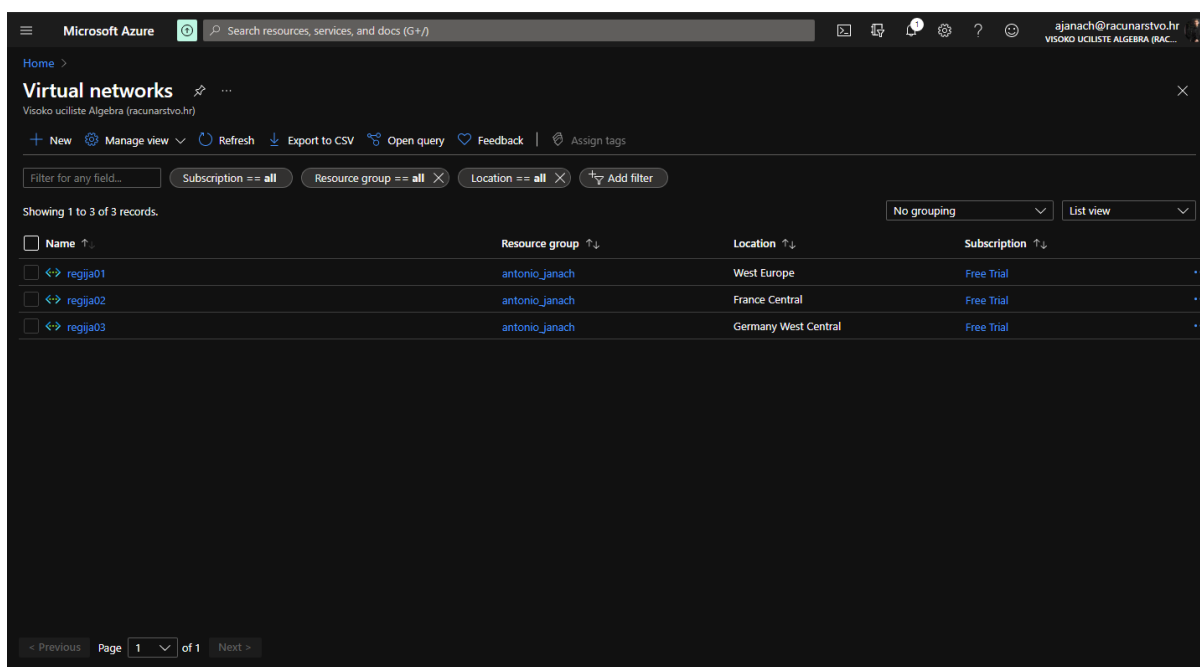
5.2. Mrežna konfiguracija u Azure platformi

Postoji mnogo mrežnih komponenti uključenih u umrežavanju računarstva u oblaku, a neke od njih su *intra-cloud* mreža koja povezuje lokalne infrastrukture u oblaku kao što je LAN podatkovni centar korišten za umrežavanje servera. *Intra-cloud* mreža obavlja tri tipa prometa:

- Mreža između virtualnih mašina na istom serveru
- Mreža između servera i sustava pohrane
- Mreža između virtualnih mašina na različitim serverima

Jezgrena transportna mreža (WAN/MAN) koriste korisnici kako bi pristupili uslugama unutar podatkovnog centra, a inter-cloud mreža služi za povezivanje cloud infrastrukture zajedno koje mogu imati iste ili različite vlasnike.

U ovome projektu cilj je kreirati tri virtualna mrežna adaptera. Svaki mrežni adapter gleda se kao objekt u resursnoj grupi. Što bi značilo da svaki objekt mora biti u posebnoj regiji i svaka regija mora imati drugi subnet. Prema zahtjevu projekta to su: 192.168.1.0/24, 192.168.2.0/24 i 192.168.3.0/24.



Slika 4: prikaz kreirana tri virtualna mrežna adaptera

Detaljne specifikacije triju kreiranih virtualnih mrežnih adaptera prikazano je na sljedećoj tablici:

Tablica 1: specifikacije pojedinih virtualnih mrežnih adaptera

Name	Resource group	Region	Address space	Subnet name	Subnet
Regija01	Antonio_janach	West Europe	192.168.1.0/24	Regija_subnet01	192.168.1.0/24
Regija02	Antonio_janach	France Central	192.168.2.0/24	Regija_subnet02	192.168.2.0/24
Regija03	Antonio_janach	German West Central	192.168.3.0/24	Regija_subnet03	192.168.3.0/24

Sljedeće što je potrebno konfigurirati je *peering* na svakome od mrežnih virtualnih adaptera. Pomoću *peerin* usluge postiže se povezivanje s partnerom u određenoj regiji. Odnosno, *peering* omogućuje jednostavno povezivanje dviju ili više virtualnih mreža u Azure infrastrukturi.

Na svakome od virtualnih mrežnih adaptera postavljen je peering tako da virtualna mašina na kojoj je konfiguriran tako da virtualna mašina na kojoj je konfiguriran *Load Balancer* može komunicirati s WP01 i WP02 virtualnom mašinom, dok WP01 i WP02 mogu komunicirati samo s LoadBalancer virtualnom mašinom. Ovo je nužan korak u implementaciji jer se svaka kreirana virtualna mašina nalazi u drugoj regiji, te se na ovoj način omogućuje komunikacija između njih. U ovome projektu namjerno sam koristio regije koje su jedna blizu druge zbog bolje i kvalitetnije povezanosti samih virtualnih mašina, da sam kreirao virtualne mrežne adaptere u drugim regijama morao bih na toj regiji kreirati virtualne mašine te bi možda smanjilo cijenu same virtualne mašine zbog odabira druge regije, no to nisam htio jer sam gledao na samu kvalitetu povezanosti.

Tablica 2: opis konfiguracije *peering-a*

Ime regije	Međusobno povezane regije	Međusobno povezane VM
Regija01	Regija01 - regija02 Regija01 - regija03	LoadBalancer01 - WP01 LoadBalancer01 - WP02
Regija02	Regija02 - regija01	WP01 - LoadBalancer01
Regija03	Regija03 - regija01	WP02 - LoadBalancer01

5.3. Kreiranje virtualnih mašina u Azure platformi

Prema zahtjevima u projektu potrebno je kreirati tri objekta (virtualna mašina, app servis ili neki drugi servis) virtualnih mašina tako da svaki objekt koji je kreiran troši što manje novaca, a da se opet mogu zadovoljiti svi zahtjevi u projektu.

Tablica 3: detaljne specifikacije virtualnih mašina

Naziv VM-a	Resursna grupa	Regija	Image	Size	OS disk type	Data disks	Cijena
LoadBalancer01	Antonio_janach	West Europe	Ubuntu Server 20.04 LTS - Gen1	Standard B1ls (1 vcpu, 0.5 GiB memory)	Standard HDD LRS	1x5GB - HDD	0.0051 EUR/hr
WP01	Antonio_janach	France Central	Ubuntu Server 20.04 LTS - Gen1	Standard B1ls (1 vcpu, 0.5 GiB memory)	Standard HDD LRS	1x5GB - HDD	0.0050 EUR/hr
WP02	Antonio_janach	Germany West Central	Ubuntu Server 20.04 LTS - Gen1	Standard B1ls (1 vcpu, 0.5 GiB memory)	Standard HDD LRS	1x5GB - HDD	0.0051 EUR/hr

Kad su virtualne mašine kreirane i upaljene spojio sam se na njih i na svakoj mašini napravljen je update i upgrade zajedno s net-toolsima.

```
Sudo apt-get update && sudo apt-get upgrade -y  
Sudo apt-get install net-tools -y
```

Kao objekt koristio sam virtualne mašine jer su virtualne mašine najjeftiniji objekt koji se može zakiputi od Azure platforme. Kad kažem najjeftiniji mislim na to da je app servis skuplji jer se na njemu mogu konfigurirati već neke gotove stvari, dok je virtualna mašina jeftinija jer ju je potrebno ručno konfigurirati, odnosno virtualna mašina zahtijeva više vremena da bude na njoj konfigurirano željeno.

5.4. NGINX - konfiguracija Load Balancer-a sa SSL/TLS enkrijcijom

NGINX je HTTP i obrnuti *proxy* poslužitelj te općeniti TCP/UDP proxy poslužitelj koji je otvorenog koda i besplatan. Osim što se koristi za služenje HTTP zahtjeva može se koristiti i za slanje i primanje elektroničke pošte preko SMTP/POP3 protokola. Najvažnija karakteristika NGINX-a odlikuje stabilnost, skalabilnost, visoke performanse, jednostavno postavljanje uz malu potrošnju resursa. Primjer popularnih stranica koje koriste NGINX su: Netflix, WordPress, GitHub, SoundCloud.

Kako bi NGINX radio ispravno potrebno ga je instalirati i pravilno konfigurirati, tj. postaviti parametre. Prema zahtjevu koji je zadan u projektu potrebno je konfigurirati NGINX tako da balansira promet prema Web serverima na tako da do Web servera nije moguće doći putem javne IP adrese, već se do sadržaja dolazi tako što se u tražilicu upisuje domena na koju glasi load balancer te onda on poslužuje korisnika s podacima koji su na Web serveru.

```
#instalacija nginx servisa:
Sudo apt-get install nginx -y
#pokretanje ninx servisa:
Sudo systemctl start nginx
#omogućavanje servisa da se pokrene kad i sustav:
Sudo systemctl enable nginx
#konfiguracija nginx.conf datoteke:
Vim /etc/nginx/nginx.conf
    user www-data;
    worker_processes auto;
    pid /run/nginx.pid;
    include /etc/nginx/modules-enabled/*.conf;
    events {
        worker_connections 1024;
    }
    http {
        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        #dodavanje web servera nad kojima će se izvršavati LB:
        upstream backends {
            ip_hash; #algoritam koji LB koristi
            server 192.168.2.4; #privatna adresa WP01 poslužitelja
            server 192.168.3.4; #privatna adresa WP02 poslužitelja
        }
        server {
            listen 80 default_server;
            server_name _;
            return 301 https://$host$request_uri; #ključna linija koda za redirect
        }

        include /etc/nginx/mime.types;
        default_type application/octet-stream;
        server {
            listen 443 ssl;
            server_name _;
            location / {
                sub_filter http://ajanach.vua.cloud https://ajanach.vua.cloud;
                sub_filter_once off;
            }
        }
    }
}
```

```

proxy_redirect http://ajanach.vua.cloud https://ajanach@vua.cloud;
proxy_set_header Accept-Encoding "";
proxy_buffering off;
add_header X-Frame-Options SAMEORIGIN;
add_header Strict-Transport-Security: max-age=3456000;
add_header X-XSS-Protection "1; mode=block";
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
allow all;
proxy_pass http://backends;
}

#Dodavanje putanje do SSL certifikata:
ssl_certificate "/home/janach01/certifikati/fullchain3.pem";
ssl_certificate_key "/home/janach01/certifikati/privkey3.pem";
ssl_session_cache shared:SSL:1m;
ssl_session_timeout 5m;
ssl_ciphers HIGH:!aNULL:!MD5;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
ssl_prefer_server_ciphers on;
}

#dodan log zapisi:
access_log /var/log/nginx/access.log;
error_log /var/log/nginx/error.log;

gzip on;

include /etc/nginx/conf.d/*.conf;
#include /etc/nginx/sites-enabled/*;
}

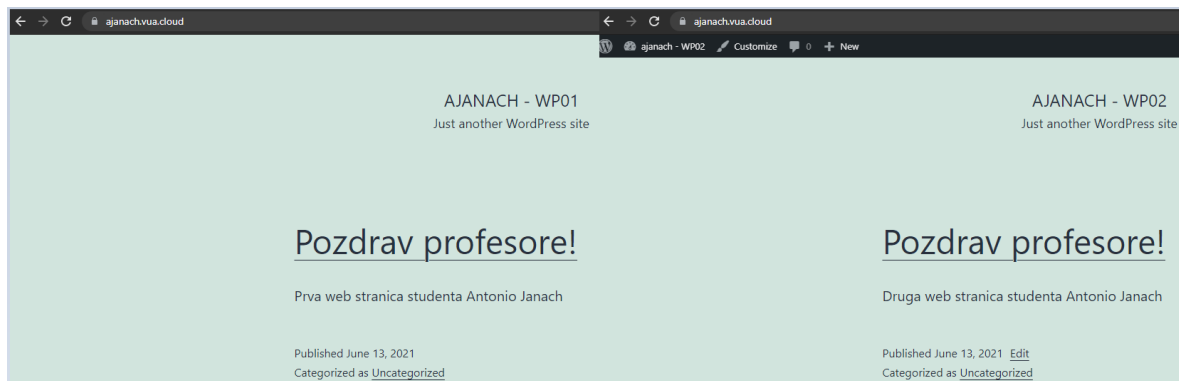
```

Zatim je potrebno zakomentirati učitavanje konfiguracije na putanji `/etc/nginx/sites-enabled/` kako se ne bi učitavala dodatna konfiguracija:

```
#include /etc/nginx/sites-enabled/*;
```

Za kraj potrebno je propustiti HTTP i HTTPS portove na svim virtualnim mašinama kako bi klijenti dobili odgovor od web poslužitelja na zahtjev.

Konfiguracija NGINX-a je uspješno konfigurirana i radi na način kad se u Web tražilicu upiše <http://ajanach.vua.cloud> otvara <https://ajanach.vua.cloud>. Što znači da je certifikat postavljen na razini *load balancer-a* i da zahtjeve poslužuje prema Web serverima koristeći *ip hash* algoritam.



Slika 5: prikaz uspješne konfiguracije virtualne mašine na kojoj se pokreće Load Balancer

5.5. WordPress instalacija

WordPress je besplatan CMS, otvorenog koji je nastao 2003. godine kao blog sustav s malim dijelom koda i s malim brojem korisnika. Do današnjeg dana izrastao je u najveći CMS sustav s velikim brojem dodataka, widgeta i tema kojega koriste milijuni web sjedišta i kojeg posjećuju deseci milijuna posjetitelja svakoga dana. U pozadini je podržan PHP-om i MySQL-om. Pomoću WordPressa se mogu kreirati poslovna web-sjedišta, e-commerce (web trgovine), forumi, portfolia, blogovi, magazini itd.

5.5.1. WordPress instalacija na WP01

Prema zahtjevima u projektu podignut je LAMP server na WP01 virtualnoj mašini. Podignut je Apache servis, kreirana mysql baza podataka i instaliran PHP verzije 7.4. Datoteke WordPress-a spremaju se na disk od 5GB. Na disku je bilo potrebno kreirati particiju, formatirati disk u xfs datotečni sustav, zatim montirati particiju na jednu od folder-a i trajno zalijepiti konfiguraciju u fstab tako da se pokreće svaki puta kada se pokrene Ubuntu virtualna mašina.

Prikaz konfiguracije:

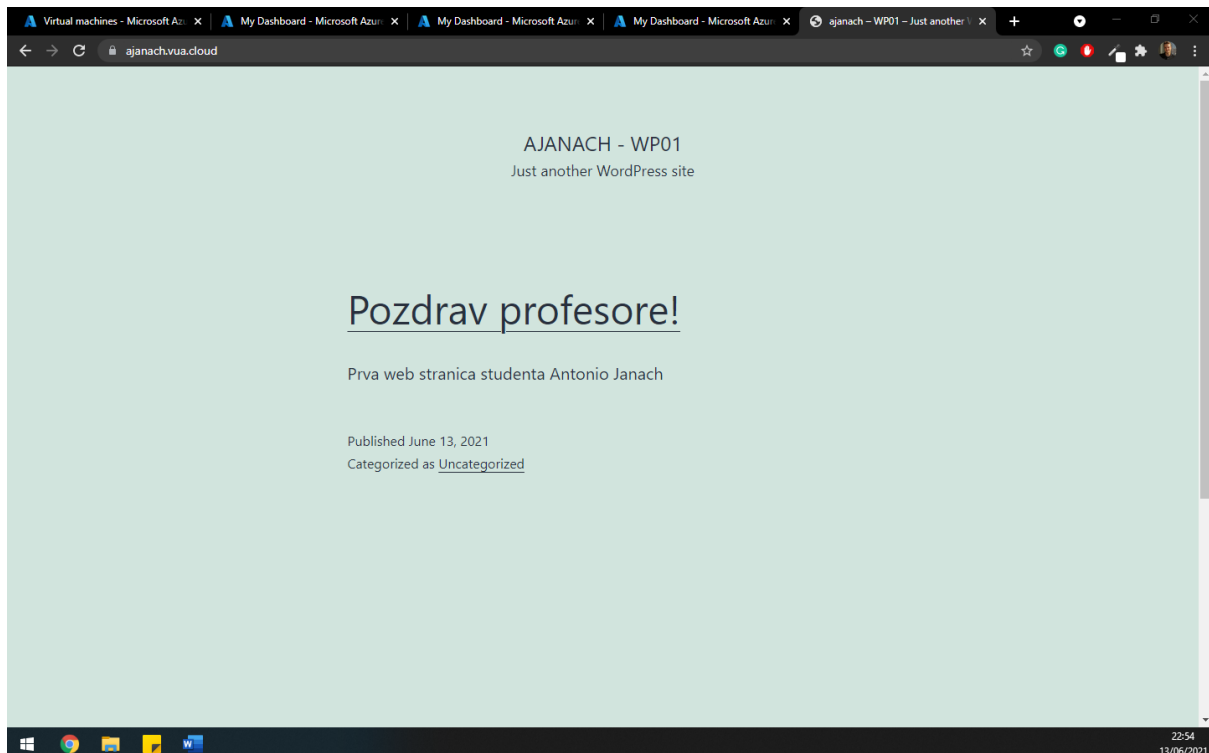
```
#Prerequisite instalacija:
sudo apt-get install mariadb-server apache2 php php-cli php-json php-gd php-mbstring
php7.4-common php-xml php7.4-mysql wget -y
#pokretanje potrebnih servisa
sudo systemctl start apache2
sudo systemctl enable apache2
sudo systemctl start mariadb
sudo systemctl enable mariadb
#kreirati particiju:
sudo fdisk /dev/sdb <<EOF n p 1 2048 41943039 w EOF
#kreiranje file sistema, direktorija i mount na logički volument koji će biti predviđen za
WordPress datoteke:
sudo mkfs.xfs /dev/sda1
sudo mkdir /mnt/wordpress
sudo mount /dev/sda1 /mnt/wordpress
#trajni mount konfiguracije da bude dostupna i nakon pokretanja:
tail -n 1 /etc/mtab >> /etc/fstab
#provjera da li je sve ispravno unešeno u mtab-a u fstab, točnije napravimo remount iz
fstab-a:
sudo mount --all
#konfiguracija mysql baze podataka:
sudo mysql_secure_installation
sudo mysql -u root -p
MariaDB [(none)]> CREATE DATABASE WP01;
MariaDB [(none)]> CREATE USER 'WP01'@'localhost' IDENTIFIED BY 'Pa$$w0rd1234';
MariaDB [(none)]> GRANT ALL ON WP01.* TO 'WP01'@'localhost';
exit
#pozicionirati se na putanju gdje će biti pohranjene datoteke WordPress-a, zatim preuzeti
najnoviju verziju WordPressa i otpakirati ga:
sudo wget https://wordpress.org/latest.tar.gz
sudo tar xvzf latest.tar.gz
```

```

#dodjela prava nad direktorijem:
sudo chmod -R 775 /mnt/wordpress/
sudo chown -R www-data:www-data /mnt/wordpress/
sudo chmod -Rf 775 /mnt/wordpress/
sudo chown www-data:www-data /mnt/wordpress/
#urediti konfiguracijski file na putanji /etc/apache2/sites-available/000-default.conf
Vim /etc/apache2/sites-available/000-default.conf
    <VirtualHost *:80>
        ServerAdmin root@localhost
        DocumentRoot /mnt/wordpress/wordpress

        <Directory "/mnt/wordpress/wordpress">
            Options Indexes FollowSymLinks
            AllowOverride all
            require all granted
        </Directory>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
    </VirtualHost>
#ponovno pokrenuti apache2 i mysql bazu:
Sudo systemctl restart apache
Sudo systemctl restart mariadb

```



Slika 6: prikaz uspješno instaliranog WordPress-a na WP01 virtualnu mašinu

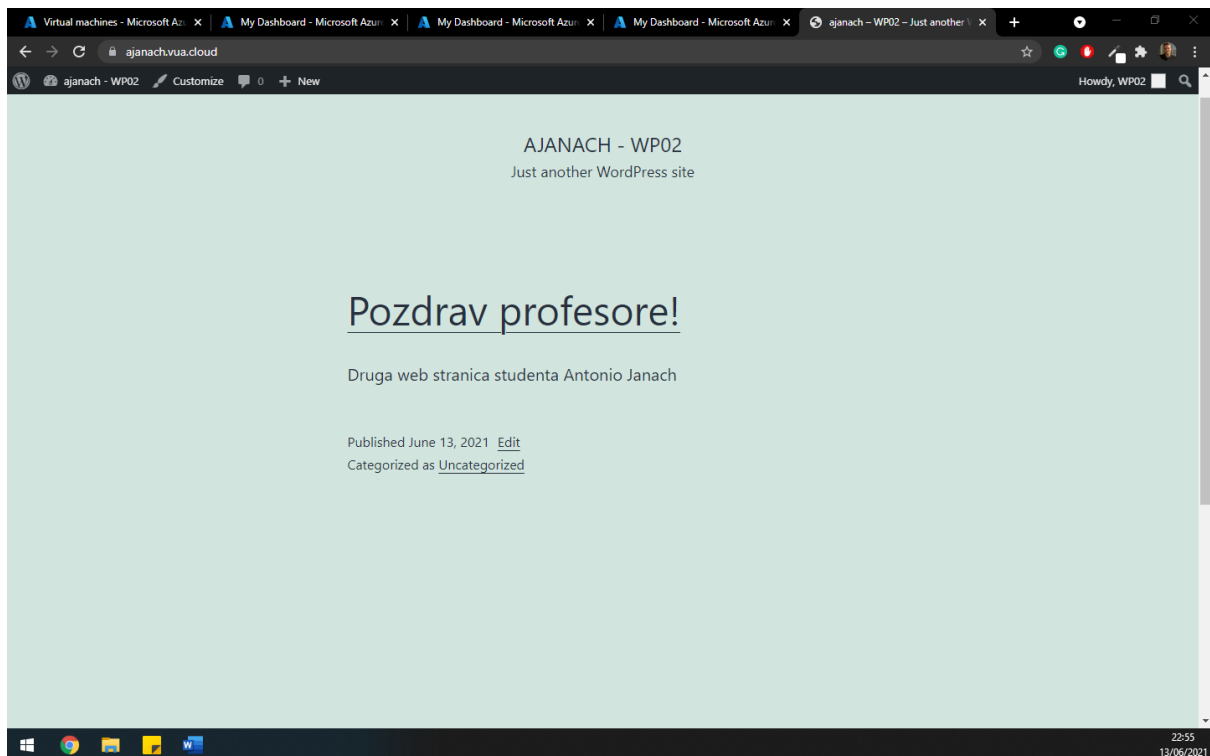
5.5.2. WordPress instalacija na WP02

Prema zahtjevima u projektu podignut je LAMP server na WP02 virtualnoj mašini. Podignut je Apache servis, kreirana mysql baza podataka i instaliran PHP verzije 7.4. Datoteke WordPress-a spremaju se na disk od 5GB. Na disku je bilo potrebno kreirati particiju, formatirati disk u xfs datotečni sustav, zatim montirati particiju na jednu od folder-a i trajno zalijepiti konfiguraciju u fstab tako da se pokreće svaki puta kada se pokrene Ubuntu virtualna mašina.

Prikaz konfiguracije:

```
#Prerequisite instalacija:
sudo apt-get install mariadb-server apache2 php php-cli php-json php-gd php-mbstring
php7.4-common php-xml php7.4-mysql wget -y
#pokretanje potrebnih servisa
sudo systemctl start apache2
sudo systemctl enable apache2
sudo systemctl start mariadb
sudo systemctl enable mariadb
#kreirati particiju:
sudo fdisk /dev/sdb <<EOF n p 1 2048 41943039 w EOF
#kreiranje file sistema, direktorija i mount na logički volument koji će biti predviđen za
WordPress datoteke:
sudo mkfs.xfs /dev/sda1
sudo mkdir /mnt/wordpress
sudo mount /dev/sda1 /mnt/wordpress
#trajni mount konfiguracije da bude dostupna i nakon pokretanja:
tail -n 1 /etc/mtab >> /etc/fstab
#provjera da li je sve ispravno unešeno u mtab-a u fstab, tocnije napravimo remount iz
fstab-a:
sudo mount --all
#konfiguracija mysql baze podataka:
sudo mysql_secure_installation
sudo mysql -u root -p
MariaDB [(none)]> CREATE DATABASE WP02;
MariaDB [(none)]> CREATE USER 'WP02'@'localhost' IDENTIFIED BY 'Pa$$w0rd1234';
MariaDB [(none)]> GRANT ALL ON WP02.* TO 'WP02'@'localhost';
exit
#pozicionirati se na putanju gdje će biti pohranjene datoteke WordPress-a, zatim preuzeti
najnoviju verziju WordPressa i otpakirati ga:
sudo wget https://wordpress.org/latest.tar.gz
sudo tar xvzf latest.tar.gz
#dodjela prava nad direktorijem:
sudo chmod -R 775 /mnt/wordpress/
sudo chown -R www-data:www-data /mnt/wordpress/
sudo chmod -Rf 775 /mnt/wordpress/
sudo chown www-data:www-data /mnt/wordpress/
#urediti konfiguracijski file na putanji /etc/apache2/sites-available/000-default.conf
Vim /etc/apache2/sites-available/000-default.conf
<VirtualHost *:80>
    ServerAdmin root@localhost
    DocumentRoot /mnt/wordpress/wordpress
    <Directory "/mnt/wordpress/wordpress">
```

```
Options Indexes FollowSymLinks
AllowOverride all
require all granted
</Directory>
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
#ponovno pokrenuti apache2 i mysql bazu:
Sudo systemctl restart apache
Sudo systemctl restart mariadb
```



Slika 7: prikaz uspješno instaliranog WordPress-a na WP02 virtualnu mašinu

Popis slika

Slika 1: prikaz opisa infrastrukture kroz umnu mapu	3
Slika 2: prikaz topologije infrastrukture	4
Slika 3: resursna grupa	5
Slika 4: prikaz kreiranja tri virtualna mrežna adaptera	6
Slika 5: prikaz uspješne konfiguracije virtualne mašine na kojoj se pokreće Load Balancer	10
Slika 6: prikaz uspješno instaliranog WordPress-a na WP01 virtualnu mašinu	12
Slika 7: prikaz uspješno instaliranog WordPress-a na WP02 virtualnu mašinu	14

Popis tablica

Tablica 1: specifikacije pojedinih virtualnih mrežnih adaptera	6
Tablica 2: opis konfiguracije peering-a	7
Tablica 3: detaljne specifikacije virtualnih mašina	8

Zaključak

Računarstvo u oblaku sve više se koristi u poslovne i privatne svrhe što znači da je dobro prihvaćen u svijetu. Znajući kako danas internet ima važnu ulogu u svakome poslovanju, jedna od najvećih prednosti korištenja ove vrste usluga jest da korisnik plaća pružatelju usluga koliko sredstva koristi, za razliku od kupnje vlastitih resursa gdje se resursi najčešće ne koriste u potpunosti. Računarstvo u oblaku je tehnologija koja još nije postigla svoj vrhunac, te je još uvijek u razvoju. Računarstvom u oblaku pomaže kako bi izbjegli kupovinu svog potrebnog softvera i hardvera tako što uz prihvatljivu cijenu može se iznajmiti sav potreban hardver i softver za obavljanje svojih poslova. Računarstvo u oblaku idealan je za tvrtke koje te počinjju svoje poslovanje jer ne moraju graditi vlastitu infrastrukturu, jer imaju mogućnost korištenja usluga u oblaku. Računarstvo u oblaku pruža veću sigurnost i privatnost tvrtkama važnih podataka. Ono što je vrlo vjerojatno moguće, a to je da ćemo se u budućnosti susresti s operacijskim sustavima koji nisu instalirani lokalno, već će pristup biti moguć preko interneta.

Za rješavanje zahtijeva koji su u projektu navedeni podignute su tri virtualne mašine u Azure okolinu. Svaka virtualna mašina nalazi se u drugoj regiji. Na prvoj virtualnoj mašini konfiguriran je servis Load Balancer koji promet raspoređuje prema pozadinskim WordPress serverima.

U ovome projektu na praktičan način je prikazano kako poslužitelji istovremeno dobivaju velike količine zahtijeva i upita na koje moraju odgovoriti od strane korisnika, pojavljuje se potreba da mrežni poslužitelji na što bolji i brži način odgovaraju na spomenute upite kako bi svojim korisnicima pružili što brži pristup uslugama. Servis koji omogućuje dostupnost i brži pristup uslugama je *Load Balancer*. Upotrebom navedenog dokazano je da je prometno opterećenje moguće optimizirati s nekoliko različitih algoritama, ovisno o korisnikovim potrebama.

Literatura

- [1] <https://wordpress.org/support/article/how-to-install-wordpress/>
- [2] <https://www.digitalocean.com/community/tutorials/how-to-install-wordpress-with-lamp-on-ubuntu-18-04>
- [3] <https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/>
- [4] <https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts>
- [5] <https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts>
- [6] <https://docs.microsoft.com/en-us/azure/?product=featured>
- [7] <https://docs.microsoft.com/en-us/azure/azure-portal/>
- [8] <https://docs.microsoft.com/en-us/azure/app-service/>
- [9] <https://docs.microsoft.com/en-us/azure/virtual-machines/>
- [10] <https://docs.microsoft.com/en-us/azure/storage/>
- [11] <https://docs.microsoft.com/en-us/azure/?product=networking>
- [12] <https://docs.microsoft.com/en-gb/azure/virtual-machines/network-overview>
- [13] <https://devblogs.microsoft.com/premier-developer/understanding-cidr-notation-when-designing-azure-virtual-networks-and-subnets/>
- [14] <https://vnt-software.com/azure-vms-pricing/>
- [15] <http://www.securityandit.com/network/wordpress-with-nginx-ssl-offloading/>
- [16] <https://www.nginx.com/resources/wiki/start/topics/examples/loadbalanceexample/>
- [17] <https://aws.amazon.com/blogs/compute/deploying-an-nginx-based-http-https-load-balancer-with-amazon-lightsail/>
- [18] https://www.server-world.info/en/note?os=Ubuntu_20.04&p=nginx&f=9