

Informatikai technológiák laboratórium 2. (BMEVIMIA429)

Felhőalapú szolgáltatások vizsgálata (Mérési útmutató)

Szatmári Zoltán, Tóth Áron
Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék

2014. november 19.

1 Mérési környezet

A mérés során egy közös AWS fiók keretében fogunk dolgozni, így minden mérőpár által létrehozott objektumot láthatunk. Fokozottan ügyelni kell arra, hogy mások munkáját ne akadályozzuk, ne rontsuk el, ezért minden esetben a létrehozott objektumokat egyértelműen, a neptun kódok segítségével nevezzük el. Figyeljünk, hogy végig a us-east régióban dolgozzunk, itt van a tanszéknek olyan kvótája, hogy a méréshez szükséges mennyiségű erőforrást tudjunk használni.

2 Mérési feladat, motiváció

A mérés kerettörténetét adja most az, hogy online karácsonyi marketing kampányt szeretnénk indítani és úgy reklámozni a forradalmian új szolgáltatásunkat, melynek keretében lehetőségünk van online rendelni a karácsonyfát házhozszállítással.

Ennek a marketingakciónak keretében futni fog egy komolyabb erőforrásokat igénylő banner kampány nagy látogatottságú oldalakon. Rögtön nagyban gondolkozunk, és a kampányt ki szeretnénk terjeszteni az egész világra, hiszen kiterjedt kapcsolataink vannak európai, ázsiai és amerikai karácsonyfa-kereskedő hálózatokkal, akik vállalták a szolgáltatásunk helyi lebonyolítását. A banner kampány ennek megfelelően a legnagyobb hazai és külföldi online hirdetési médiumon fog megjelenni lehetőleg a nyitólapon. Ezen felül egy

nagy méretű Facebook promóciós kampányt is indítunk, ezért akár óránkénti több milliós letöltési szám várható a bannerként használt képek esetében.

A szolgáltatáshoz tartozik ezen felül egy webes alkalmazás, ahol az ügyfelek regisztrálni tudnak és le tudják adni a megrendelésüket. Kezdetben az alkalmazás legyen egy egyszerű kis PHP alapú oldal, amely az adatokat SQL adatbázisba menti, így a rendeléseket a háttérben dolgozó adminisztrátorok feldolgozhatják és továbbíthatják a szállító partnerek felé.

A feladat megoldása során több különböző problémát azonosítottunk:

- Nem ismert, hogy hogyan reagál a piac a szolgáltatásunkra, így lehet, hogy egyáltalán nem lesz népszerű, de elképzelhető, hogy hatalmas hype-ot váltunk ki és több millió megrendelés fut be.
- A banner kampány az aktivitás óráiban óriási sávszélességet és letöltési számot produkál a világ minden területén, miközben a kampány időszakán kívül senki sem fogja a bannereket letölteni.

A fenti okok miatt úgy döntöttünk, hogy nem építünk ki saját infrastruktúrát, hanem az Amazon felhőalapú szolgáltató szolgáltatásait vesszük igénybe. Ezzel minimalizálhatjuk a terheléssel járó költséget, hiszen forgalom alapján fizetünk, valamint biztosíthatjuk a skálázhatóságot az esetleges hirtelen népszerűség kezelésére.

1. Gyűjtsük össze, és indokoljuk röviden meg, melyek azok a szolgáltatások, amit ezen leírás alapján igénybe fogunk venni a tervezett feladatok ellátásához!

3 EC2 alapok

Webes menedzsment felületen igényeljük egy egyszerű virtuális gépet és vizsgáljuk meg annak tulajdonságait.

1. Jelentkezzünk be az AWS webes menedzsment felületen és válasszuk ki az EC2 menedzsment felületet! (<https://console.aws.amazon.com>)
2. Hozzunk létre egy SSH kulcsot, majd nevezzük el a neptun kód segítségével! A PEM fájlt mentjük el (pl. az asztalra), hiszen később szükségünk lesz rá. (Állítsuk át az elmentett fájl jogosultságát úgy, hogy csak a tulajdonos férjen hozzá (600), különben később az SSH kliens nem fogadja el.)
3. Hozzunk létre egy Security Group-ot, majd adjuk meg, hogy az SSH (22-es port) és a Web (80-as port) mindenhol elérhető legyen!
4. Indítsunk egy t1.micro méretű virtuális gép példányt a „Classic Wizard” segítségével az „Ubuntu Server 14.04 LTS (PV)” 64 bites AMI sablonból! Figyeljünk, hogy paravirtualizált (PV) image-et válasszunk, majd a Wizard-on végighaladva ne felejtsük el megadni a korábban elkészített Security Group-ot, SSH kulcsot és a „Name” tag-et (itt a neptun kód szerint nevezzük el a példányt)!

5. Várjuk meg, míg elindul a virtuális gép, majd jelentkezünk be rá ssh segítségével! Figyeljünk, hogy a felhasználónév „ubuntu” legyen és az imént letöltött kulcsot használjuk a hitelesítéshez! (Az, hogy ubuntu a felhasználói név a sablon leírásából derül ki.)
6. Vizsgáljuk meg a kapott szervert és mutassunk be néhány erőforráskorlátot (memória, lemez, hálózati paraméterek), amit a kiválasztott erőforrássablon meghatároz!

4 EC2 parancssori alapismeretek

Parancssori kliens segítségével hajtsuk végre az előzőkhez hasonló lépéseket és hozzunk létre egy virtuális gépet a már meglévő ssh kulcs és security group jellemzőkkel!

1. Indítsuk el az ITLab_AWS virtuális gépet a labor PC-n, mert ott előre telepítve megtalálható az EC2 használatához szükséges parancssori eszközkészlet és a hitelesítéshez szükséges adatok is beállításra kerültek. Bejelentkezéshez használjuk a `meres/LaborImage` felhasználói név és jelszó párost!
2. Teszteljük a parancssori eszközkészletet. Használjuk a parancssori automatikus kiegészítés lehetőségét, hogy felfedezzük a parancsokat: írjuk be egy parancssorba, hogy `ec2-descr` és a TAB billentyűvel nézzük meg milyen lehetőségeink vannak. Kérjük le az elérhető régiók, kulcspárok, és futó virtuális gép példányok listáját!
3. Indítsunk el egy új virtuális gép példányt az előző feladatban megadott paraméterek segítségével (`ec2-run-instances`). Keressük ki az AMI ID-t az előző virtuális gép adatai közül, adjuk meg ezen felül a régió, security group és a kulcspár nevét. Jegyezzük fel a visszatérésben kapott instance ID-t.
4. Nevezzük el a virtuális gépet azaz adjuk meg hozzá a „Name” tag-et. (`ec2-create-tags`)
5. Vizsgáljuk meg a webes menedzsment konzolon, mit kaptunk!
6. Jelentkezünk be a gépre és vizsgáljuk meg alapvető paramétereit!

5 S3 alapok

Egy kiválasztott képet feltöltünk egy tárolóba és megvizsgáljuk annak elérhetőségét.

1. A webes menedzsment felületen lépünk át az S3 menedzsment oldalára.
2. Hozzunk létre egy tárolót (bucket), nevezzük el a neptun kódunk alapján! Figyeljünk, hogy ebben az esetben csak kisbetűket és számokat használjunk az elnevezésben, mert később a CloudFront használatánál csak azt tudjuk felhasználni!

3. Keressünk valamilyen egyszerű képet az interneten amit a „banner kampányban” használni fogunk és mentjük el az asztalra. Ez lesz a teszt banner, amit tárolni fogunk, majd HTTP fölé ki fogunk szolgálni.
4. Töltsük fel a kiválasztott képet a saját tárolónkba!
5. Állítsuk be, hogy mindenki számára elérhető legyen a feltöltött fájl. (Properties/Permissions)
6. Nézzük meg, hogy hogyan érhető el webről a képünk:
`http://bucketname.s3.amazonaws.com/kepem.jpg`
7. Próbáljuk parancssorból elérni a tárolónk tartalmát! Töltsük le a képet a méréshez használt virtuális gépre az `s3cmd` parancssori hívással!

6 CloudFront alapok

Az imént feltöltött képet CDN technológia segítségével is elérhetővé tesszük az interneten.

1. Hozzunk létre egy új terjesztést (Distribution).
2. Adjuk meg az alapul szolgáló S3 tároló adatait, a többi adatot hagyjuk alapértelmezetten.
3. Kis idő elteltével, miután elkészült, olvassuk le a hozzáféréshez használható domain nevét, majd nézzük meg a webes elérhetőségét a képünknek:
`http://domain.name/kepem.jpg`

7 RDS alapok

Létrehozunk egy MySQL adatbázis példányt és megvizsgáljuk elérhetőségeit.

1. Hozzunk létre egy DB Security Group-ot és adjunk hozzáférési jogosultságot a korábban létrehozott EC2 security groupnak.
2. Hozzunk létre egy MySQL alapú adatbázis példányt a most elkészített security group-ba. Legyen a példány mérete `t1.micro` és az adatainál minden azonosítót (db és felhasználói név) válasszunk a `neptun` kódunknak. Biztonsági mentést ne igényeljünk.
3. Telepítsük az egyik korábban létrehozott virtuális gépünkre a MySQL klienst
 - `sudo apt-get update`
 - `sudo apt-get install mysql-client`
4. Teszteljük a kapcsolatot! Ehhez keressük ki az adatbázis példányunk eléréséhez használható URL-t és a parancssori klienssel lépünk be a virtuális gépről!

8 Webalkalmazás telepítése

Építsünk egy egyszerű webalkalmazást az eddigi építőelemeinkből: virtuális gépek, mint webserverek és az adatbázis.

1. Telepítsünk egy minimális webkiszolgálót mindkét virtuális gépünkre!
 - `sudo apt-get update`
 - `sudo apt-get install apache2 libapache2-mod-php5 php5-mysql`
 - `sudo /etc/init.d/apache2 restart`
2. Vizsgáljuk meg az alapértelmezett weboldal elérhetőségét az interneten, az egyik és másik webserververünk publikus URL-jének böngészőbe írásával!
3. Telepítsük a mérésvezető által javasolt webes alkalmazást a rendszerre!

9 Load balancer

1. Készítsünk egy egyszerű PHPInfo webes alkalmazást, hogy később legyen mivel tesztelni a terhelés elosztót. (A `/var/www/html` mappába hozzunk létre egy `info.php` fájlt és tartalma legyen: „`<?php phpinfo(); ?>`”)
2. Hozzunk létre egy terheléelosztót a saját kettő virtuális gépünket alapul véve!
3. Teszteljük a webes tartalom elérését a terhelés elosztó URL-jét használva! Figyeljünk, hogy a DNS terjedés miatt ez kb. 3-5 perc múlva fog csak működni. Mit tapasztalunk?

10 Elastic Beanstalk

Vizsgáljuk meg, hogy hogyan lehet az előző építőkövekből összeállított infrastruktúra nagy részét egyetlen PaaS szolgáltatással helyettesíteni!

1. Hozzunk létre egy PHP 5.3 típusú alkalmazást az Elastic beanstalk segítségével! Mindenféle szükséges azonosítónak használjuk megint a neptun kódot!
2. Vizsgáljuk meg és értelmezzük, hogy mi történik a háttérben! Nézzük meg az Events fület a részletező oldalon és az EC2 menedzsment konzolt.
3. Kis idő után elkészül a környezet. Nézzük meg a részletező oldalon a különböző paramétereit és látogassuk meg böngészővel az URL-jével jelzett helyet!
4. Töltsük fel a mérésvezető által ajánlott alkalmazást új verzióként és vizsgáljuk meg működését!

11 A mérés befejezése

Állítsuk le az általunk létrehozott infrastruktúrát és töröljük az általunk létrehozott objektumokat! Legyünk körültekintőek és minden tartozékot szüntessünk meg!