Google Kubernetes Engine

Contents

- 1 Regisztráció a Google Cloud Platform-ban
 2 Google Cloud SDK beállítása
 2.1 SDK install
 2.2 SDK init
 2.2 SDK init ◆ 2.3 Bejelentkezés és kubectl
 ♦ 2.3.1 kubectl kontextus • 3 Klaszter létrehozás ♦ 3.1 Node pool 4 Hálózat 4.1 SSH
 4.2 Service-ek publikus elérése
 - 4.3 Ingress

Regisztráció a Google Cloud Platform-ban

https://console.cloud.google.com Kubernetes Engine API aktiválása a projektben:

688	bining			Project ID
API	APIs & Services	Ŧ	>	adamproject1 Dashboard ject number
Ť	Support		>	Library 5600007398
Θ	IAM & admin		>	Credentials to project settings
Ť				

Majd a keres?be írjuk be, hogy Kubernetes. Ekkor bejön



Kattintsunk az Enable-re.



...

A baloldali f? menüben megjelent a Kubernetes Engine menüpont:



Google Cloud SDK beállítása

SDK install

Vegyük fel az új DNF repository-t:

```
# sudo tee -a /etc/yum.repos.d/google-cloud-sdk.repo << EOM
[google-cloud-sdk]
name=Google Cloud SDK
baseurl=https://packages.cloud.google.com/yum/repos/cloud-sdk-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOM</pre>
```

Installáljuk fel

dnf install google-cloud-sdk

SDK init

Inicializáljuk az sdk-t. Ez meg fog nyitni a root nevében futó böngész?t, ahol be kell jelentkezni a google account-ba.

```
# gcloud init
...
You must log in to continue. Would you like to log in (Y/n)?
...
```

Harmadik lépésként létre kell hozni egy projektet:

gcloud projects create adamproject1
Create in progress for [https://cloudresourcemanager.googleapis.com/v1/projects/adamproject1].

Listázzuk ki az összes projektet:

. . . .

# gcioua projects list		
PROJECT_ID	NAME	PROJECT_NUMBER
adamproject1	adamproject1	206600067398
leafy-firmament-183906	My Project 23184	1046937073531
optimum-agent-244420	My First Project	174610655273

Válasszuk ki az új projektet

gcloud config set project adamproject1

az új projektet hozzá kell rendeli egy billing account-hoz (amihez a bankszámlát beállítottuk
be kell rajta kapcsolni a Kubernetes Engine API-t.

https://console.cloud.google.com/apis/library/container.googleapis.com?project=adamproject1

Projekt törlése:

. .

```
# gcloud projects delete leafy-firmament-183906
Your project will be deleted.
```

Do you want to continue (Y/n)? Y

```
Deleted [https://cloudresourcemanager.googleapis.com/v1/projects/leafy-firmament-183906].
```

You can undo this operation for a limited period by running ' commands below. See https://cloud.google.com/resource-manager/docs/creating-managing-projects for information on shutting down projec \$ gcloud projects undelete leafy-firmament-183906

Bejelentkezés és kubectl

A gcloud bejelentkezési adatait a config list paranccsal lehet lekérni:

```
# gcloud config list
[core]
account = myemail@gmail.com
disable_usage_reporting = True
project = adamproject1
```

Your active configuration is: [default]

A disable_usage_reporting anonim adatokat küld a Google-nek, nem a projekthez kapcsolódik.

A config-helper paranccsal pedig a tokent is listázni lehet:

```
# gcloud config config-helper
configuration:
    active_configuration: default
    properties:
        core:
            account: myemail@gmail.com
            disable_usage_reporting: 'True'
            project: adamproject1
```

credential: access_token: XXXXX id_token: YYYY token_expiry: '2019-06-22T08:43:33Z' sentinels: config_sentinel: /root/.config/gcloud/config_sentinel

kubectl kontextus

A kubectl konfigurációs fájlja itt van: ~/.kube/config. Minikube esetében ez így néz ki:

```
apiVersion: v1
kind: Config
clusters:
- cluster:
    certificate-authority: /root/.minikube/ca.crt
    server: https://192.168.42.224:8443
name: minikube
contexts:
    preferences: {}
users:
- name: minikube
user:
    client-certificate: /root/.minikube/client.crt
    client-key: /root/.minikube/client.crt
    cluster: minikube
user: minikube
name: minikube
```

Miden egyes **kubecti** parancs a kiválasztott kontextusban fut le. Minden kontextus tartalmaz egy klasztert és egy user-t, amivel végrehajtjuk a parancsokat a klaszteren. A **current-context** mutatja meg, hogy melyik kontextus van kiválasztva.

Mikor létrehozunk egy cluster-t a gcloud containter cluster create paranccsal, akkor a gcloud automatikusan hozzáad egy új kontextust a .kube/config fájlhoz:

gcloud container clusters create adam1

kubeconfig entry generated for adam1.

Nézzük meg a kontextust:

kubectl config current-context
gke_adamproject1_europe-west6_adam1

Láthatjuk, hogy a gcloud átállította a kontextust a GKE klaszerünkre.

És most nézzük bele a config fájlba:

```
apiVersion: v1
kind: Config
clusters:
- cluster
     certificate-authority-data: DATA+OMITTED
server: https://34.65.220.36
  name: gke_adamproject1_europe-west6_adam1
  cluster:
certificate-authority: /root/.minikube/ca.crt
server: https://192.168.42.224:8443
  name: minikube
contexts:
  context:
     cluster: gke_adamproject1_europe-west6_adam1
  user: gke_adamproject1_europe-west6_adam1
name: gke_adamproject1_europe-west6_adam1
  context:
     cluster: minikube
     user: minikube
user: minikube
name: minikube
current-context: gke_adamproject1_europe-west6_adam1
preferences: {}
users:
  name: gke_adamproject1_europe-west6_adam1
  user:
     auth-provider:
        config:
cmd-args: config config-helper --format=json
          cmd-path: /usr/lib64/google-cloud-sdk/bin/gcloud
expiry-key: '{.credential.token_expiry}'
token-key: '{.credential.access_token}'
        name: gcp
  name: minikube
  user:
     client-certificate: /root/.minikube/client.crt
     client-key: /root/.minikube/client.key
```

Láthatjuk, hogy a GKE klaszterhez user-hez a beállításokat a gcloud config config-helper paranccsal gy?jti be.

Ahhoz hogy visszaváltsunk a minikube kontextusra, csak ki kell adni config use-context parancsot.

```
# kubectl config use-context minikube
Switched to context "minikube".
```



Note Ha kitöröljük a GKE klasztert a gcloud container clusters delete paranccsal, akkor a kubernetes config fájlból is ki fogja törölni a bejegyzést.

Klaszter létrehozás

A gcloud container cluster create parancsnak els? sorban azt kell megadni, hogy milyen típusú virtuális gépet szeretnénk készíteni, ez jelent?sen befolyásolni fogja az árat. Az elérhet? virtuális gépe listáját a machine-types list paranccsal listázhatjuk:

\$ gcloud compute machine-	types list		
NAME ZONE	C	PUS MEMOR	RY_GB DEPRECATED
f1-micro us-centr	al1-f 1	0.60	
g1-small us-centr	al1-f 1	1.70	
n1-highcpu-16 us-centr	al1-f 1	6 14.40)
n1-highcpu-2 us-centr	al1-f 2	1.80	
n1-highcpu-32 us-centr	al1-f 3	2 28.80)

A cluster létrehozásához meg kell adni azt is, hogy milyen régióban és azon belül milyen lehetséges zónákban jöjjenek létre a Kubernetes node-ok. A lehetséges zónákat a régiókkal együtt a **compute zones list** paranccsal listázhatjuk, ebb?l kell válogatni:

#	gcloud	compute	zones	list	grep	europe-west6	
e	urope-we	est6-a		eu	cope-v	vest6	UP
e	urope-we	est6-b		eu	cope-v	vest6	UP
e	urope-we	est6-c		eu	cope-v	vest6	UP

Láthatjuk, hogy a europe-west6 régió három zónából áll, itt fogjuk létrehozni a klasztert. A zónákat vessz?vel elválasztva kell megadni, space nélkül.

```
gcloud container clusters \
    create adam1 \
    --region europe-west6 \
    --node-locations europe-west6-a,europe-west6-b,europe-west6-c, \
    --machine-type n1-standard-1 \
    --enable-autoscaling \
    --num-nodes 1 \
    --max-nodes 3 \
    --min-nodes 1
```



Warning

A num-nodes értéké zónákra fog vonatkozni. Ha a node-locations-ben három zónát sorolunk föl, és a num-nodes értéke 2, akkor összesen 6 node fog létrejönni

To in:	spect the conte	nts of your clus	ster, go to: 1	https://console.c	loud.google.com	/kubernetes	/workload_/	gcloud/europe-wes	t6/adam1?project=adamp
kubec	onfig entry gen	erated for adam1							
NAME	LOCATION	MASTER_VERSION	MASTER_IP	MACHINE_TYPE	NODE_VERSION	NUM_NODES	STATUS		
adam1	europe-west6	1.12.8-gke.10	34.65.149.1	3 n1-standard-1	1.12.8-gke.10	3	RUNNING		

Listázzuk a cluster-eket:

# gclo	ud container	clusters list					
NAME	LOCATION	MASTER_VERSION	MASTER_IP	MACHINE_TYPE	NODE_VERSION	NUM_NODES	STATUS
adam1	europe-west@	5 1.12.8-gke.10	34.65.250.80	nl-standard-1	1.12.8-gke.10	3	RUNNING

És nézzük meg a web-es konzolon is. Menjünk a Kubernetes Engine felületre, és itt válasszuk a Cluster menüpontot: https://console.cloud.google.com/kubernetes

Name ^	Location	Cluster size	Total cores	Total memory	Notifications	Labels			
🥝 adam1	europe-west6	3	3 vCPUs	11.25 GB			Connect	1	î

# kubectl get nodes				
NAME	STATUS	ROLES	AGE	VERSION
gke-adam1-default-pool-b9ed6fb1-d955	Ready	<none></none>	2m11s	v1.12.8-gke.10
gke-adam1-default-pool-c1ab468d-slqn	Ready	<none></none>	2m8s	v1.12.8-gke.10
gke-adam1-default-pool-e3b15f3f-98dl	Ready	<none></none>	2m7s	v1.12.8-gke.10

gcloud container clusters delete adam1 --region europe-west6
The following clusters will be deleted.
 - [adam1] in [europe-west6]

Do you want to continue (Y/n)? Y

Node pool

Minden cluster-hez rendelve van egy node pool amit így listázhatunk:

gcloud container node-pools list --region europe-west6 --cluster adam1
NAME MACHINE_TYPE DISK_SIZE_GB NODE_VERSION
default-pool n1-standard-2 100 1.13.10-gke.0

Ezt a konzolon, a klaszter részletek oldal alján láthatjuk:

e pools					0	Columns 👻
Status	Version	Number of nodes	Machine type	Image type	Autos	caling
🥝 ок	1.13.10-gke.0	6 (2 per zone)	n1-standard-2	Container-Optimized OS (cos)	Off	t
4	e pools Status Status	e pools Status Version OK 1.13.10-gke.0	e pools Status Version Number of nodes OK 1.13.10-gke.0 6 (2 per zone)	E pools Status Version Number of nodes Machine type OK 1.13.10-gke.0 6 (2 per zone) n1-standard-2	Version Number of nodes Machine type Image type OK 1.13.10-gke.0 6 (2 per zone) n1-standard-2 Container-Optimized OS (cos)	E pools Status Version Number of nodes Machine type Image type Autos OK 1.13.10-gke.0 6 (2 per zone) n1-standard-2 Container-Optimized OS (cos) Off

A pool-t a cluster resize paranccsal méretezhetjük át:

gcloud container clusters resize [CLUSTER_NAME] --node-pool [POOL_NAME] --num-nodes [NUM_NODES]

Ez az adam1 cluster-re az alábbi:

Do you want to continue (Y/n)? y

Resizing adam1...done. Updated [https://container.googleapis.com/v1/projects/adamproject1/zones/europe-west6/clusters/adam1].



Warning

A node számot zónánkét kell érteni. Ha 2-re állítjuk és három zónában van a klaszterünk, akkor összesen 6 node lesz a pool-ban

Hálózat

A GKE klaszterben a node-nak van egy external és egy internal IP-je. Az external IP egy publikus IP cím. Ha készítünk egy nodePort-os Kubernetes service-t, akkor a node publikus IP -n és a service portján elérhet?ek lesznek a service-hez tartozó pod-ok. Listázzuk a node-okat a -o wide kapcsolóval. Az **EXTERNAL-IP** oszlopban valós, publikus (dinamikusan kiosztott) IP címek szerepelnek.

# kubectl get nodes -o wide						
NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP
gke-adam1-default-pool-67fe18ad-3v96	Ready	<none></none>	14m	v1.12.8-gke.10	10.172.0.12	34.65.193.16
gke-adam1-default-pool-75a1a9b3-slpl	Ready	<none></none>	14m	v1.12.8-gke.10	10.172.0.13	34.65.84.196
gke-adam1-default-pool-eaf55b7a-9kb3	Ready	<none></none>	14m	v1.12.8-gke.10	10.172.0.11	34.65.22.212

A node-ok egy VPC (Virtaul Private Cloud) hálózatra vannak kötve. Azt, hogy milyen portok vannak nyitva a VPC-n, t?zfal szabályok szabályozzák (firewall-rules).

# gcloud compute firewall-r	ules list						
NAME	NETWORK	DIRECTION	PRIORITY	ALLOW	DENY	DISABLED	
default-allow-icmp	default	INGRESS	65534	icmp		False	False
default-allow-ssh	default	INGRESS	65534	tcp:22		False	

Láthatjuk, hogy a klaszerünkben a 22-es (SSH) port alapértelmezetten nyitva van.

A VPC-vel kapcsolatos beállításokat a Web-es konzolon a VPC network menüpotban végezhetjük el:

NETWORKING			
VPC ne	twork 🖡	>	VPC networks
Networ	k services	>	External IP addresses

Itt megtekinthetjük a fent listázott t?zfal szabályokat:

83	Firewall rules	Filter resources				
×	Routes	Name	Туре	Targets	Filters	Protocols /
Ŷ	VPC network peering	gke-adam1-c44e3334-all	Ingress	gke-adam1-c44e3334-node	IP ranges: 10.56.0.0/14	tcp udp
×	Shared VPC					sctp; icmp;
	a l 1/00	gke-adam1-c44e3334-ssh	Ingress	gke-adam1-c44e3334-node	IP ranges: 34.65.119.177/32, 2 more 👻	tcp:22

Az **External IP addresses** menüpontban megtekinthetjük a Google Cloud Platform-on használt publikus IP címeinket. Jelenleg a három node-os Kubernetes klaszterünk mindegyik node-jához tartozik egy publikus IP cím:

Name	External Address	Region	Туре 🗸	Version	In use by
-	34.65.22.212	europe-west6	Ephemeral 👻	IPv4	VM instance gke-adam1-default-pool-eaf55b7a-9k
-	34.65.84.196	europe-west6	Ephemeral 👻	IPv4	VM instance gke-adam1-default-pool-75a1a9b3-sl
-	34.65.193.16	europe-west6	Ephemeral 👻	IPv4	VM instance gke-adam1-default-pool-67fe18ad-3v

SSH

A Cluster létrehozásakor minden olyan SSH kulcsot megkaptunk, amivel a három node-ra be tudnunk lépni SSH-val. És ahogy ezt láthattunk is a default hálózaton a 22-es port nyitva van mind három node-on:

#	gcloud	compute	firewall-rules	list		
Nž	AME		NETWORK	DIRECTION	PRIORITY	ALLOW
de	efault-a	allow-ssł	n default	INGRESS	65534	tcp:22

A describe paranccsal bele is nézhetünk:

```
# gcloud compute firewall-rules describe default-allow-ssh
allowed:
- IPProtocol: tcp
ports:
    - '22'
description: Allow SSH from anywhere
direction: INGRESS
kind: compute#firewall
name: default-allow-ssh
...
```

Egy adott node-ra a gcloud compute ssh <felhasználó név>@<node név> paranccsal lehet belépni.



Ha nem adunk meg felhasználó nevet, akkor a root-al fog a google sdk megpróbálni belépni, ami nem lehetséges. **Permission denied (publickey)**.. Ennek széles irodalmva van a neten, nincs igazából konklúzió. Pl: https://stackoverflow.com/questions/26193535/error-gcloud-compute-ssh-usr-bin-ssh-exited-with-return-code-255

Az SSH-hoz használt user-t nem kell el?re létrehozni. Ha olyan nevet adunk meg, ami még nem létezik, akkor els? alkalommal létre fogja hozni. Fontos tehát, hogy mindig megadjunk egy a root-tól különböz? felhasználó nevet a @ el?tt, különben nem fog menni a belépés. Próbáljunk meg belépni az els? node-ra, aminek a nevét a **kubecti get nodes** parancsból nézhetjük ki.

gcloud compute ssh adam2@gke-adam1-default-pool-67fe18ad-3v96

No zone specified. Using zone [europe-west6-b] for instance: [gke-adaml-default-pool-67fe18ad-3v96]. Updating project ssh metadata...?Updated [https://www.googleapis.com/compute/v1/projects/adamproject1]. Updating project ssh metadata...done. Waiting for SSH key to propagate.

adam2@gke-adam1-default-pool-67fe18ad-3v96 ~ \$

Service-ek publikus elérése

A Kubernetes nodok external IP címe publikus internet cím. De ahogy azt láthattuk alapértelmezetten csak néhány port van nyitva a **VPC** default hálózaton. Kubernetes-ben a POD-ok elérhet?ek nodePort típusú service-ekkel bármelyik node external IP címén. Mivel GKE-ben ezek publikus IP címei a VM-eknek, ezért a megfelel? hálózati szabály hozzáadásával a nodePort típusú service-ek is elérhet?ek lesznek a publikus internetr?l. Telepítsünk föl egy teszt pod-ot, amiben egy hello world webserver fog futni a 8080-ás porton.

```
metadata:
  name: web
  namespace: default
spec:
  selector
    matchLabels:
       run: web
  template:
    metadata:
labels:
         run: web
     spec:
       containers:
       - image: gcr.io/google-samples/hello-app:1.0
imagePullPolicy: IfNotPresent
         name: web ports:
           containerPort: 8080
            protocol: TCP
```

kubectl apply -f web-deployment.yaml

Majd hozzunk hozzá létre egy nodePort típusú szolgáltatást.

apiVersion: v1 kind: Service metadata: name: web-service spec: type: NodePort ports: port: 8080 selector: run: web0

kubectl apply -f web-service.vaml

Most keressük meg a service node portját. Mivel nem specifikáltuk, a Kubernetes random választott egyet:

# Kubecti get	svc web-sei	rvice			
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
web-service	NodePort	10.60.0.58	<none></none>	8080:32315/TCP	27s

Nincs más dolgunk, mint hogy a Google Cloud Platform-n kinyissuk a t?zfalat a 32315 portra. (A test-node-port az általunk kitalált neve a t?zfal szabálynak)

gcloud compute firewall-rules create test-node-port --allow tcp:32315

Most keressük meg valamelyik node publikus IP címét:

# kubectl get node -o wide						
NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP
gke-adam1-default-pool-5a644461-csxd	Ready	<none></none>	7m37s	v1.12.8-gke.10	10.172.0.15	34.65.84.196

Írjuk be a böngész?be ezt a címet: http://34.65.84.196:32315/



Juhuuu!

Ingress

Az ingress alapból benne van a GKE cluster-ben, és egy saját google implementáció. Tehát nekünk semmi dolgunk nincs a telepítéssel.

https://cloud.google.com/kubernetes-engine/docs/tutorials/http-balancer

Az Ingress kontrollerek a Kubernetes service bels? pontjára csatlakoznak rá, tehát nem a nodePort-ra. Ismétlés képpen: háromféle portot különböztetünk meg a service-ekben:

- port: ezen a porton érhet? el a service a cluster-en belül. Ennek nincs köze ahhoz hogy a cluster-n kivül (a node IP címén keresztül) hol
- optil. ezen a porton enter? er a service a cluster en beloi. Entrek mics koze annoz nogy a cluster en kvor (a node in cliner kresztul) nor érher? el a szolgáltatás. Ha ezt nem definiáljuk, akkor ugyan az lesz mint a target-port
 target-port: ez azt mondja meg, hogy a pod-on melyik port-ra továbbítsa az üzeneteket a service. Ha nem adjuk meg külön, akkor a deployment-ben megadott -containerPort értékét fogja kapni.
 nodePort: Azt mondja meg, hogy a cluster-en kívül melyik porton lesz elérhet? a szolgáltatás a node-ok publikus IP címén keresztül. Ha nem adjuk meg akkor random választ a Kubernetes, ahogy ezt az el??? fejezetben láthattuk is.

Ha statikus IP-t akarunk, akkor azt külön kell csiálni, de fontos, hogy aztán kitöröljük, mert külön számláz érte.

name: basic-ingress
spec:
backend:
serviceName: web-service
servicePort: 8080