**UNIVERSITATEA DE STAT DIN MOLDOVA**

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**DEPARTAMENTUL DE INFORMATICĂ**

**GRĂDINARU MĂDĂLINA**

**Cloud Computing**

**Lucrare de Laborator nr.5:**

„ Implementarea HPA în Google Cloud ”

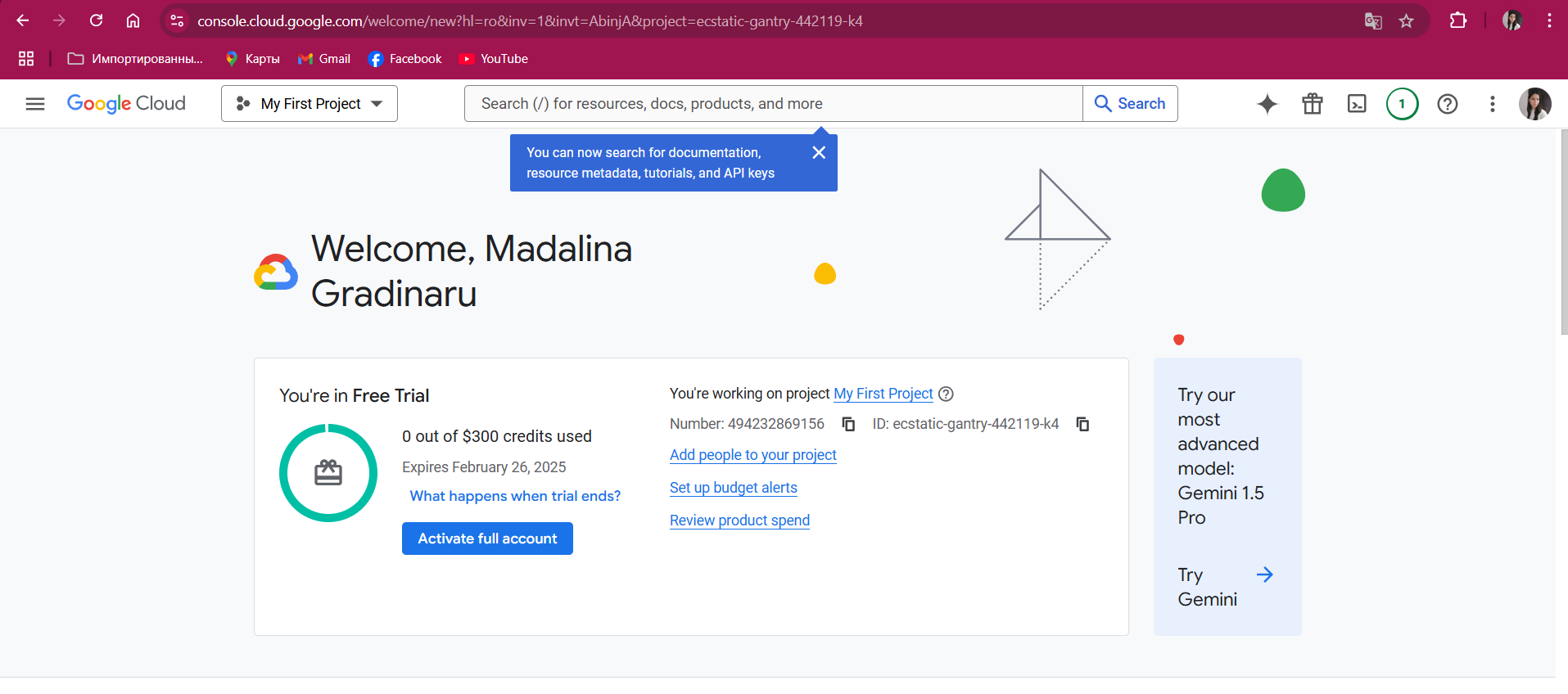
Profesor: Aurelia Prepeliță, dr., conf.univ.

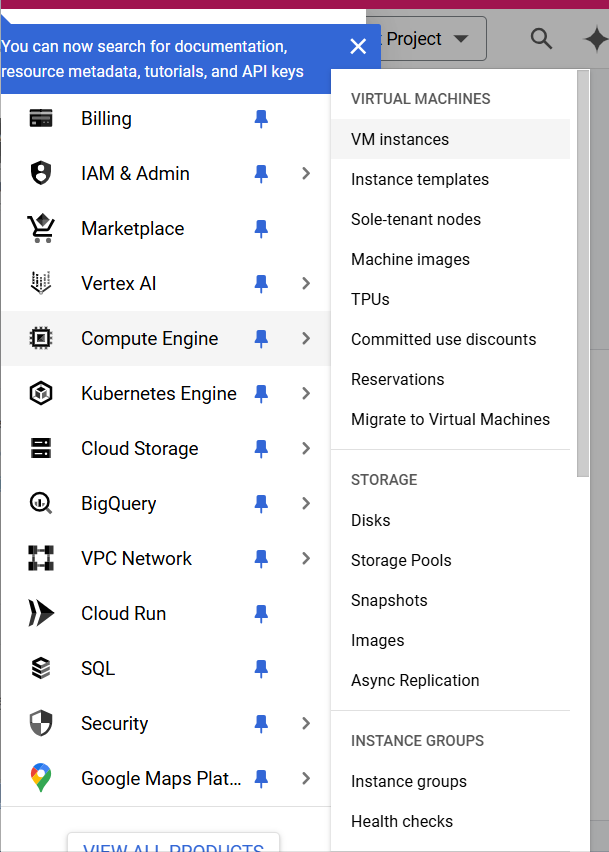
**CHIȘINĂU – 2024**

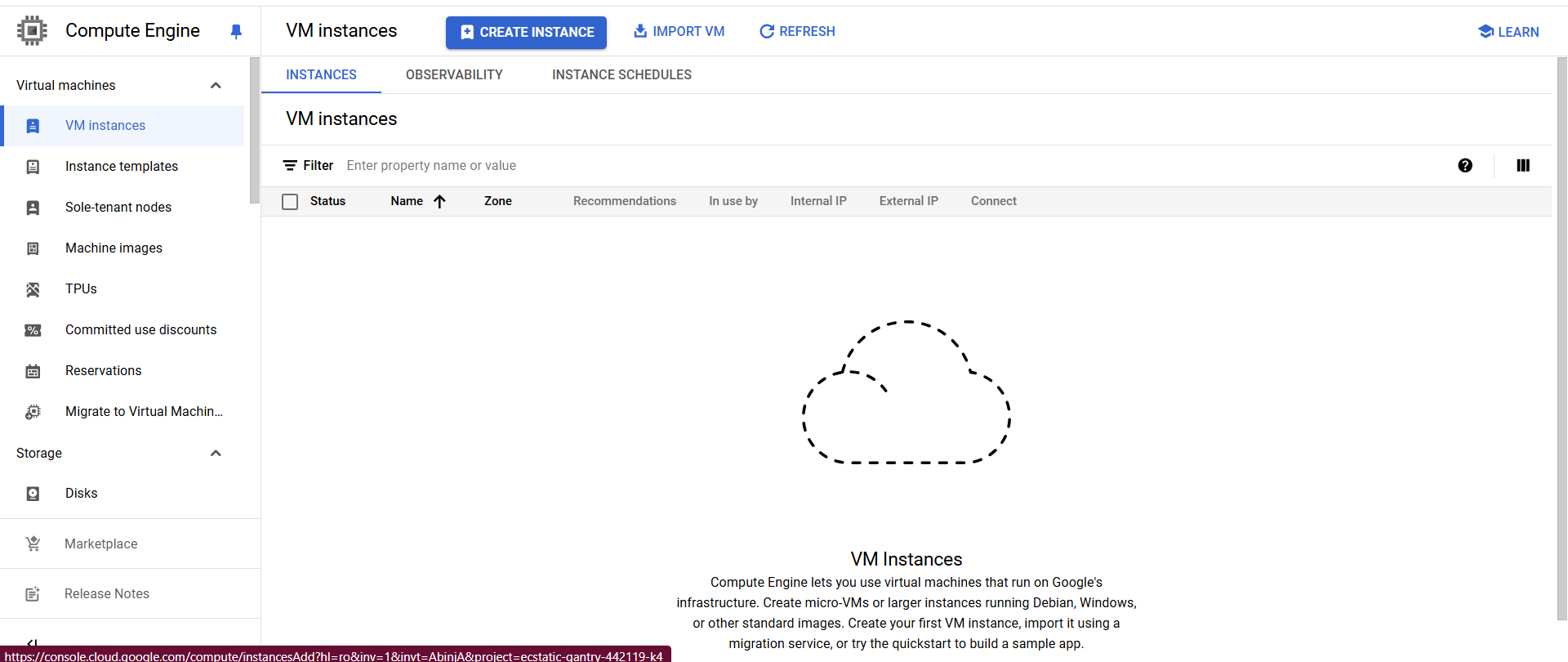
# ****1. Introducere:****

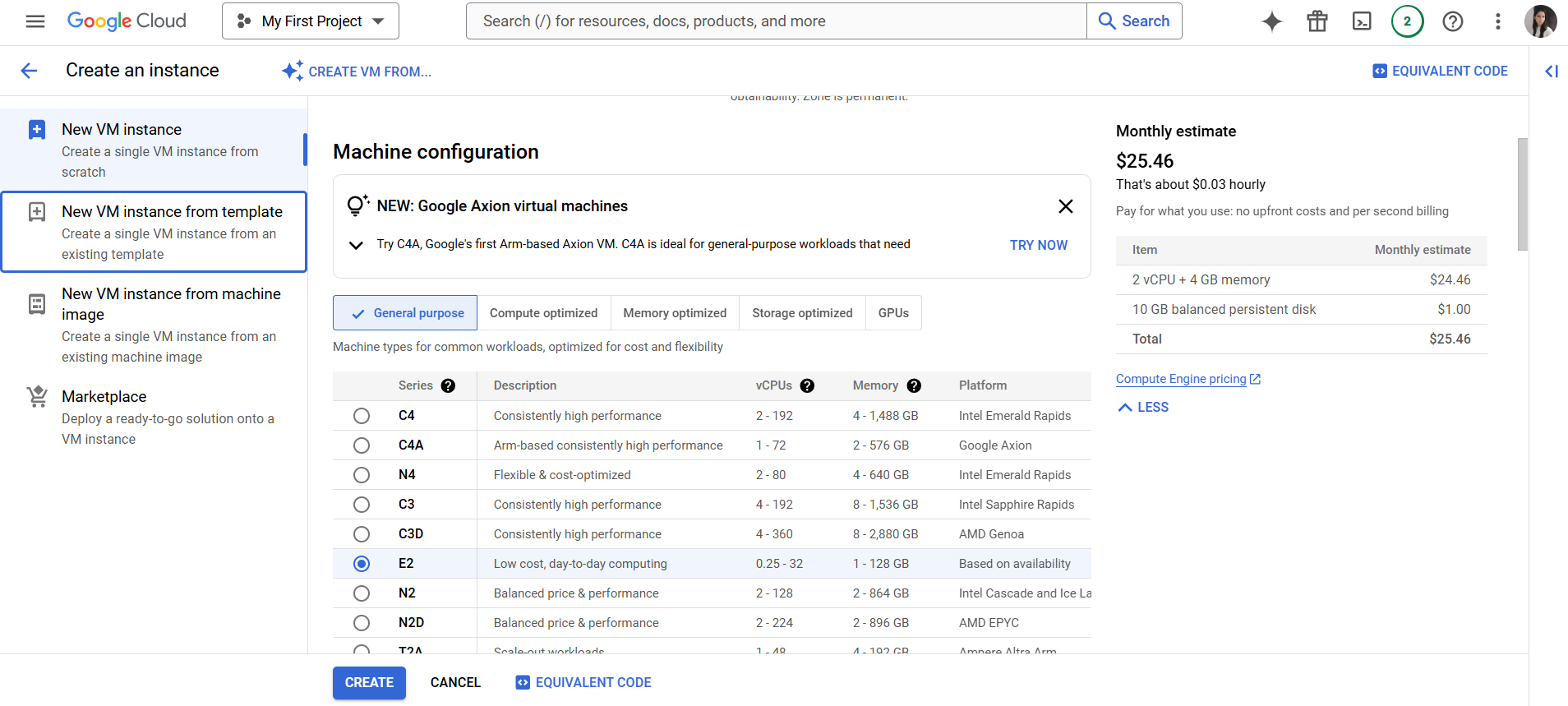
# 2. Instalarea Google Cloud SDK C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{E725A847-D3D5-43A5-8A1C-BA967344DCF3}.png

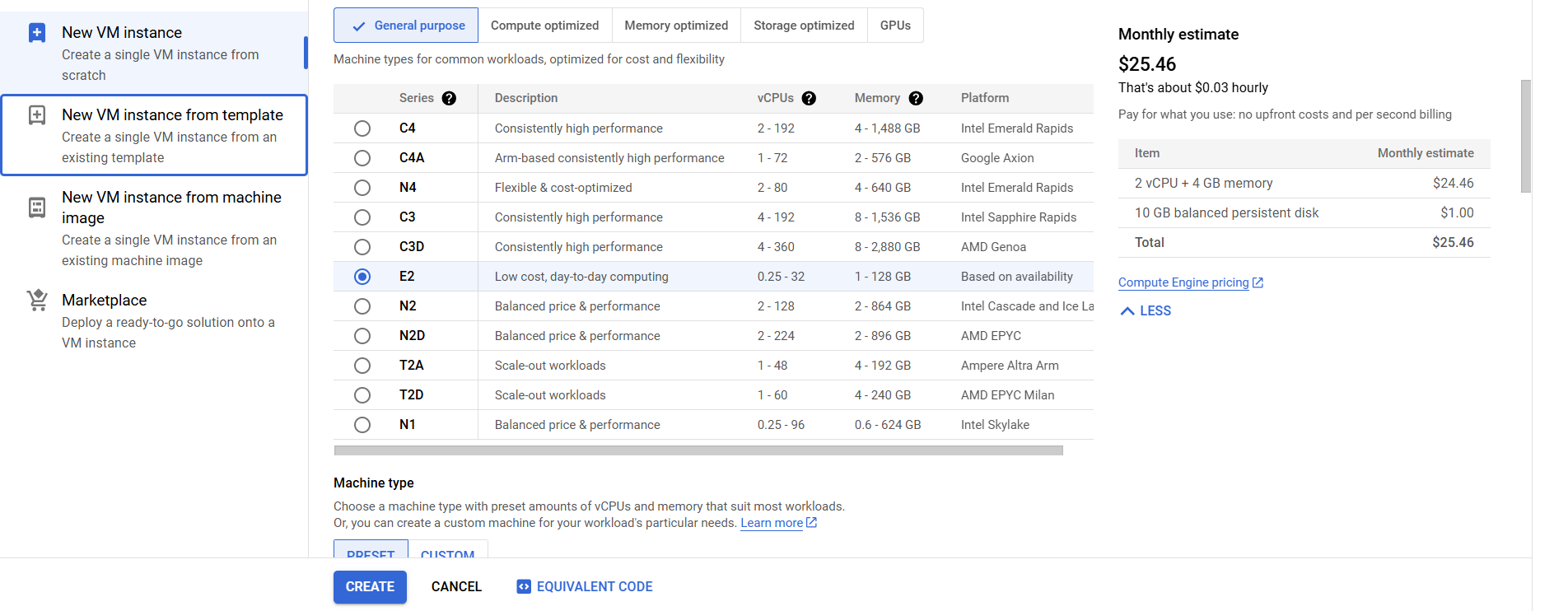
# 3. Instalare Kubernetes ca parte a Google Cloud SDK

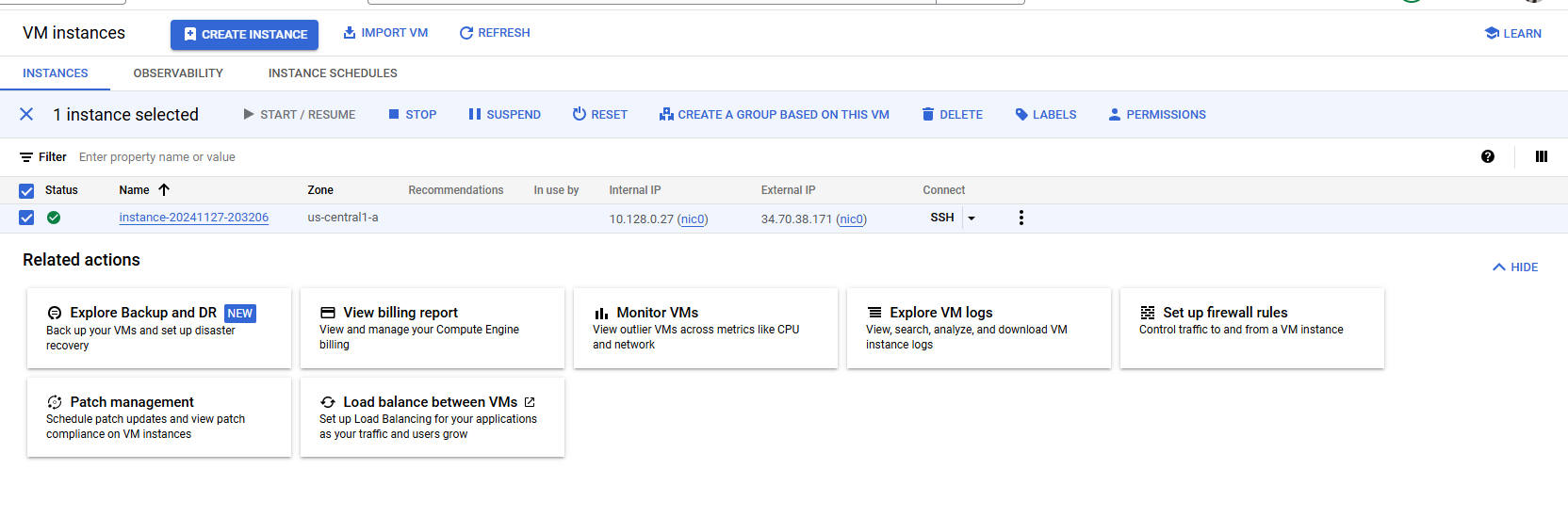


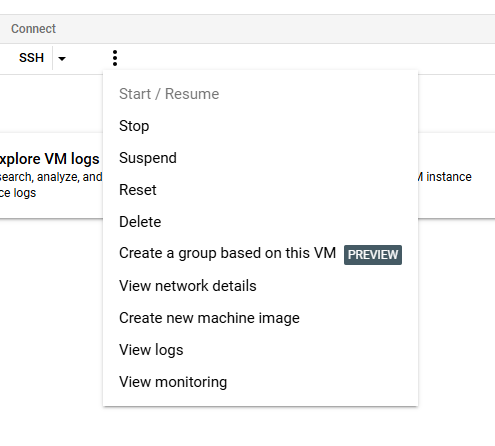
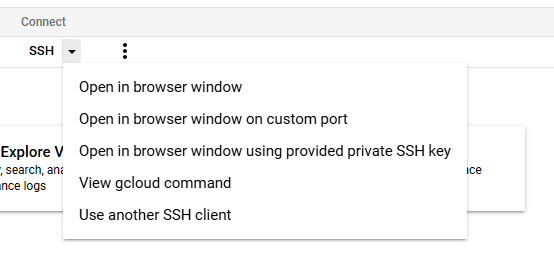


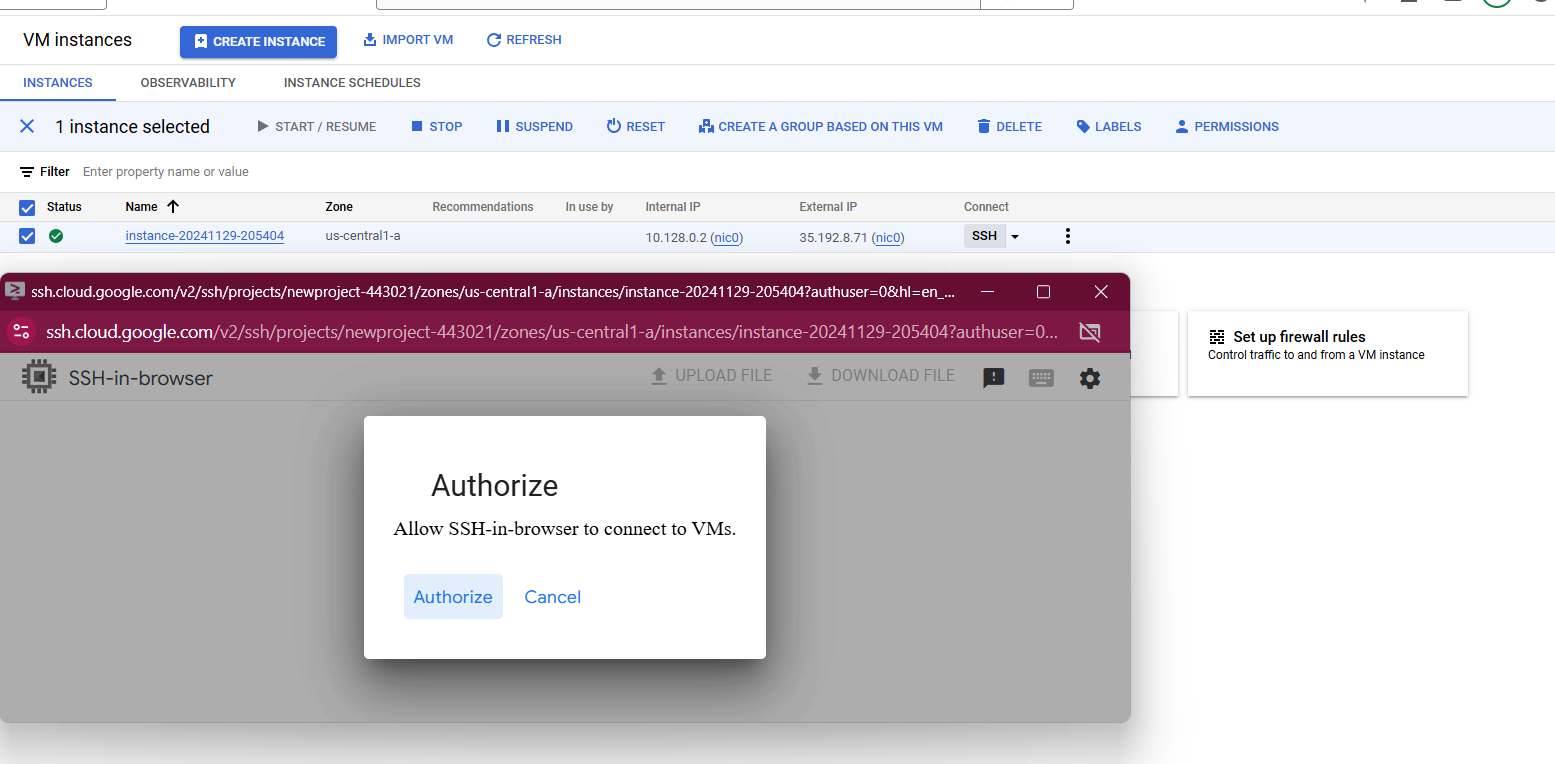


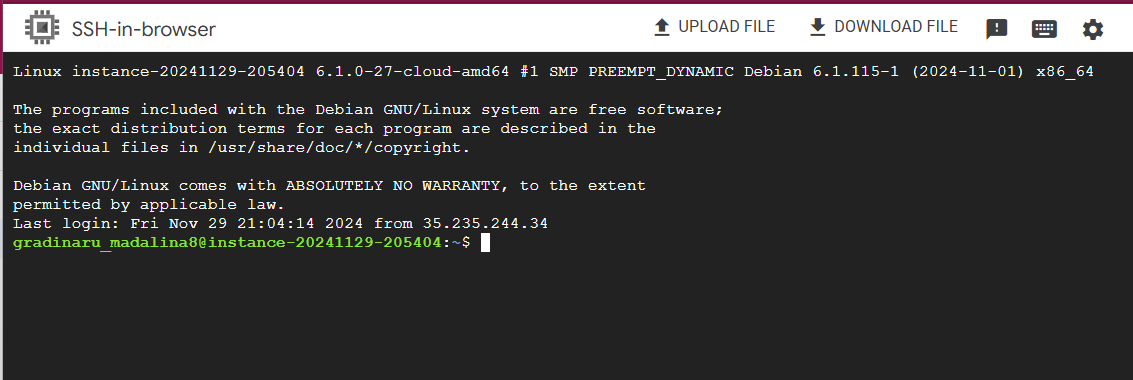


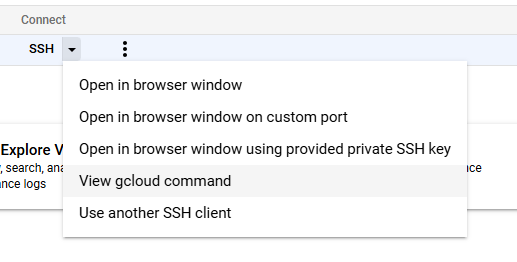


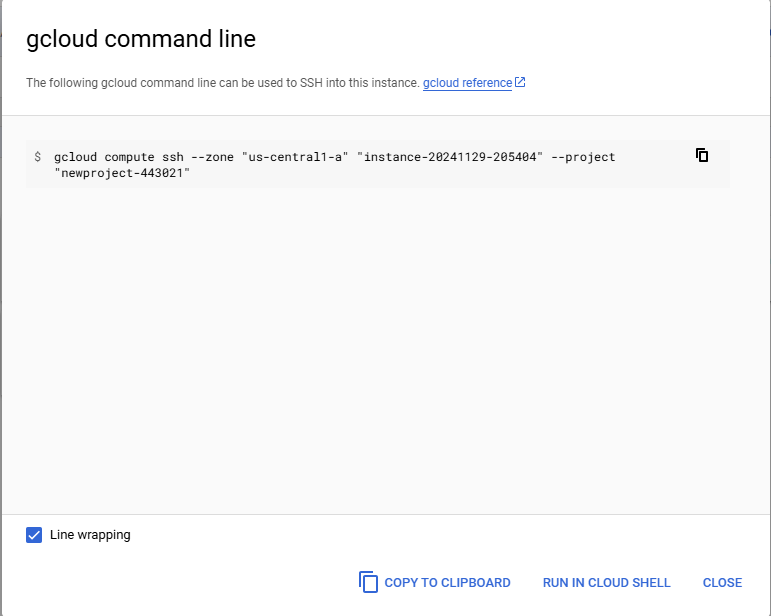


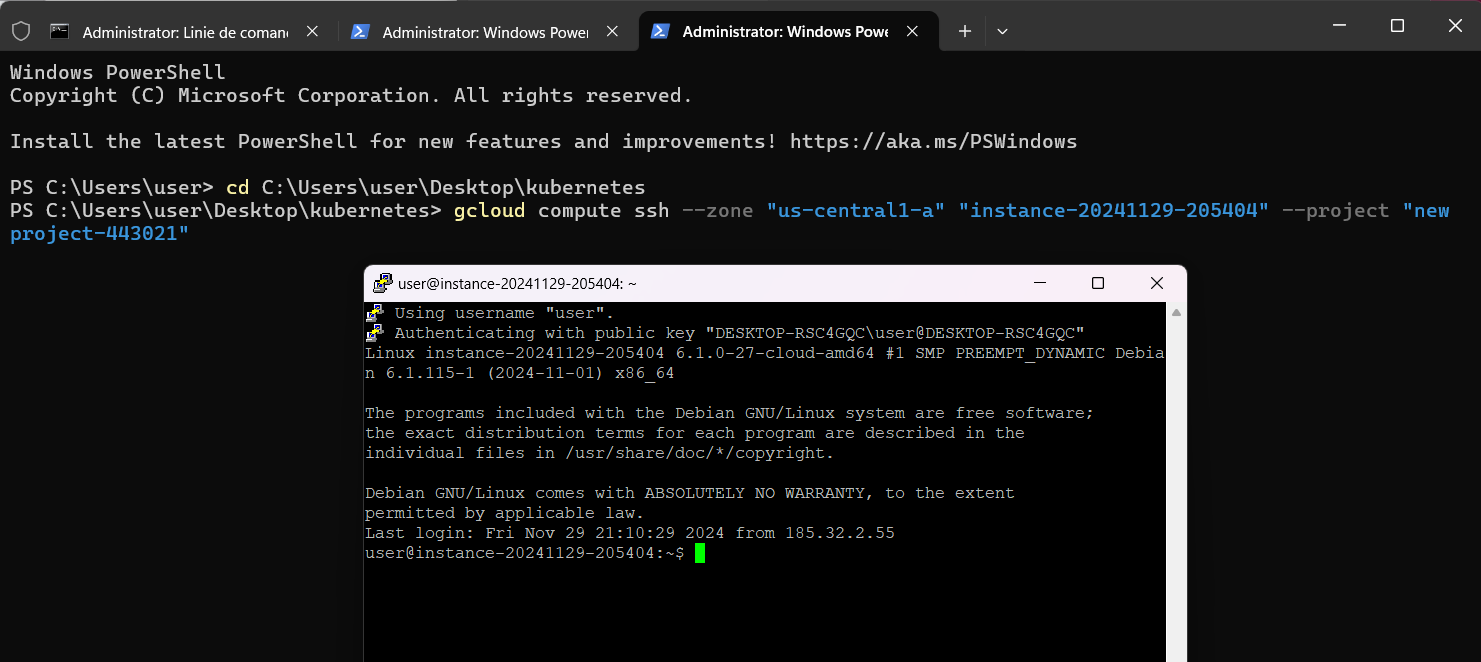
Verificam ce ne ofera GC pentru lucru cu VM:  


Apasand pe acel buton, ne conectam la masina noastra virtuala:  


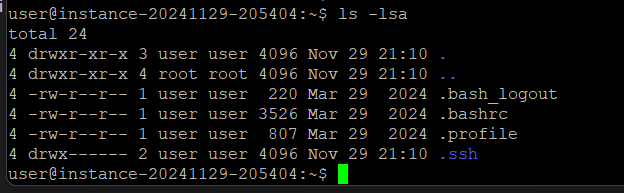
si se deschide ssh in browser, deci ne-am conectat la masina noastra virtuala:  


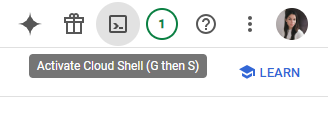
De asemenea ne putem conecta la VM, local:  


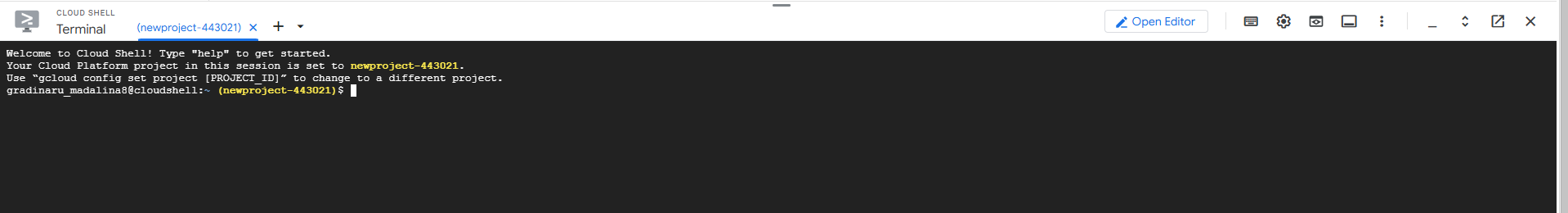


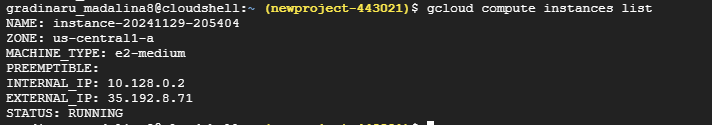
Ruland comanda in Powershell, se activeaza VM in directorul de lucru prin ssh.  


Deci am reusit sa ma conectez la instanta virtuala. A aparut promptul Linux.  
Putem rula comenzi si sa vedem toate fisierele si directoarele din directorul curent.

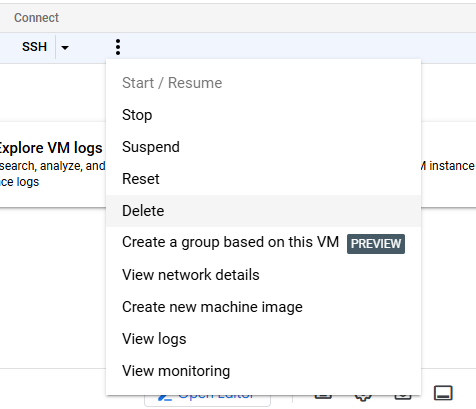
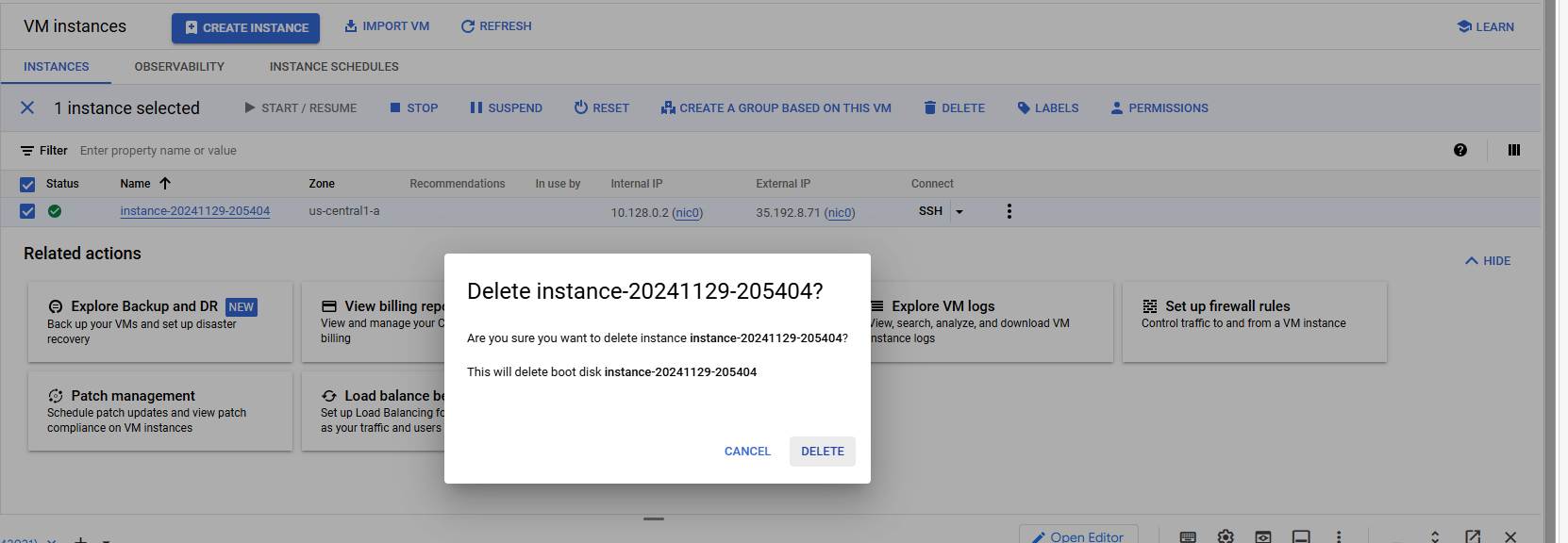


Un alt tip de conectare este prin terminal din Google Cloud:  


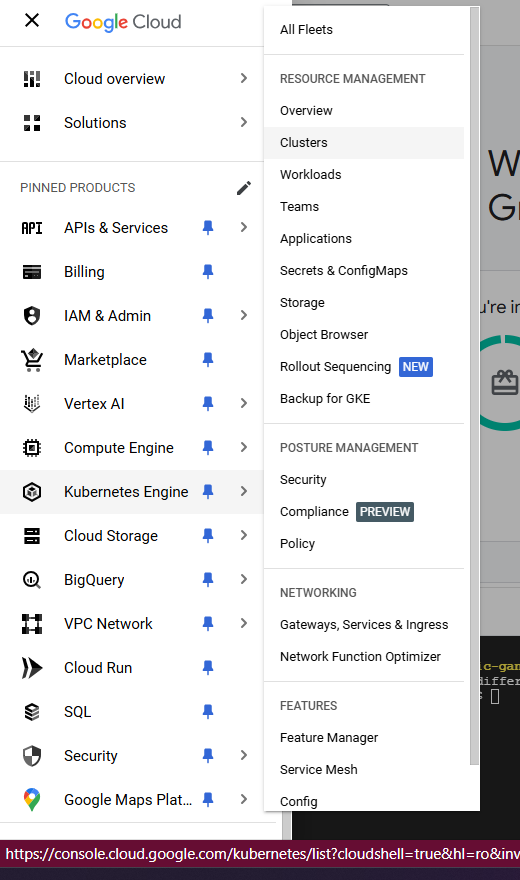


Aici deja putem rula comenzi, ex:  


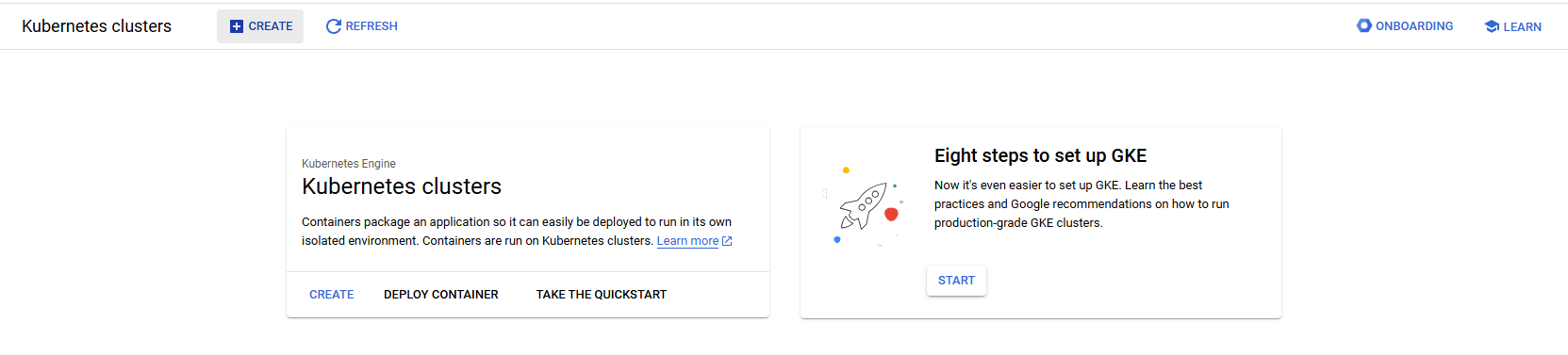
Apoi stergem masina virtuala creata, de altfel se iau bani din cont.

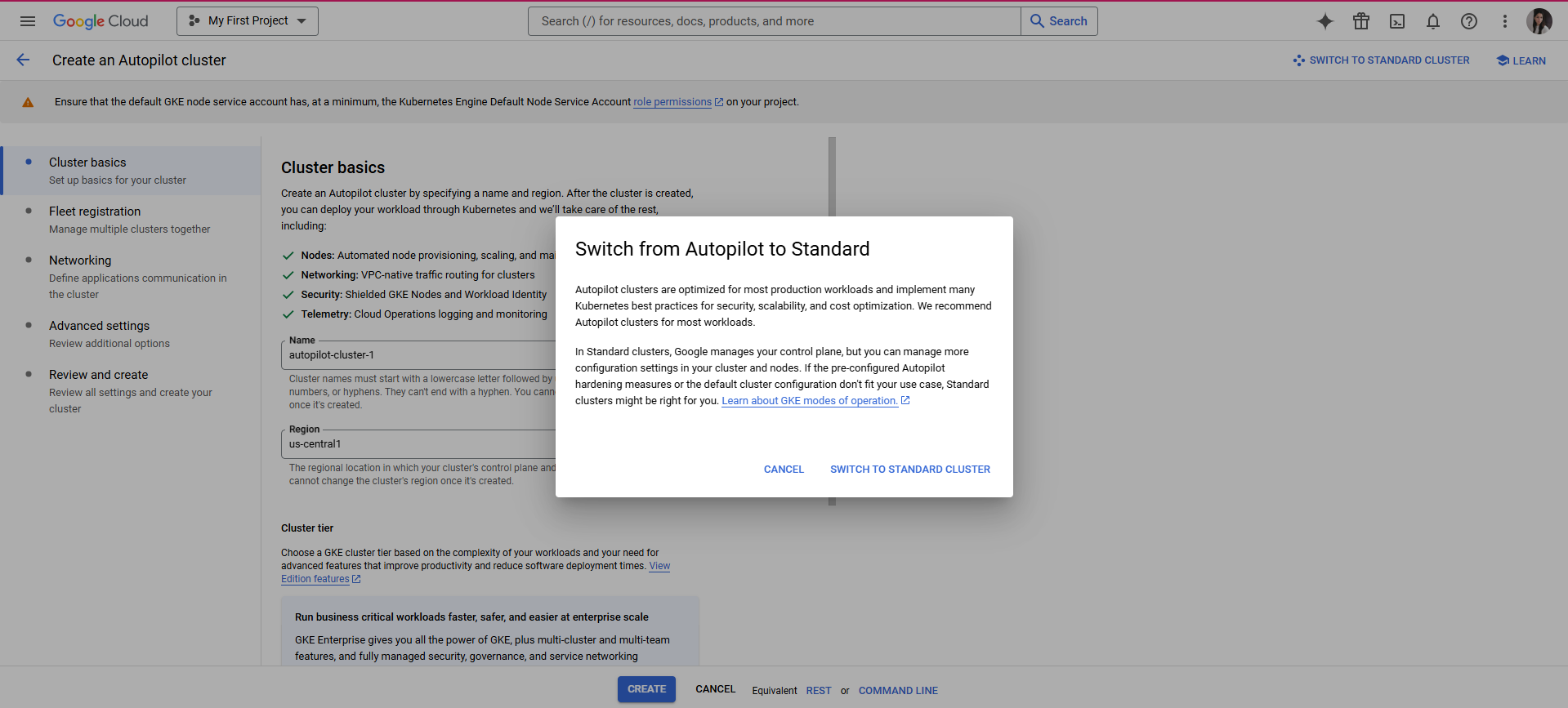
  


Exemplu de creare a unui Cluster:

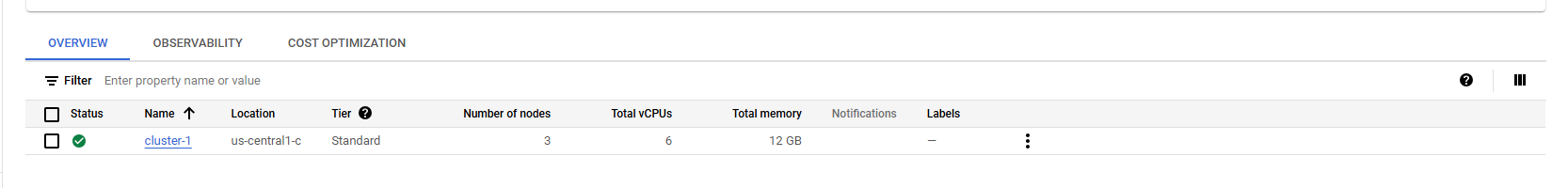


Facem Enable  
Selectez create

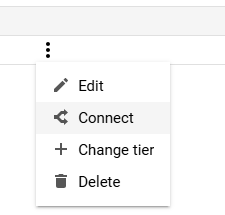


Alegem varianta standart de creare a Clusterelor  


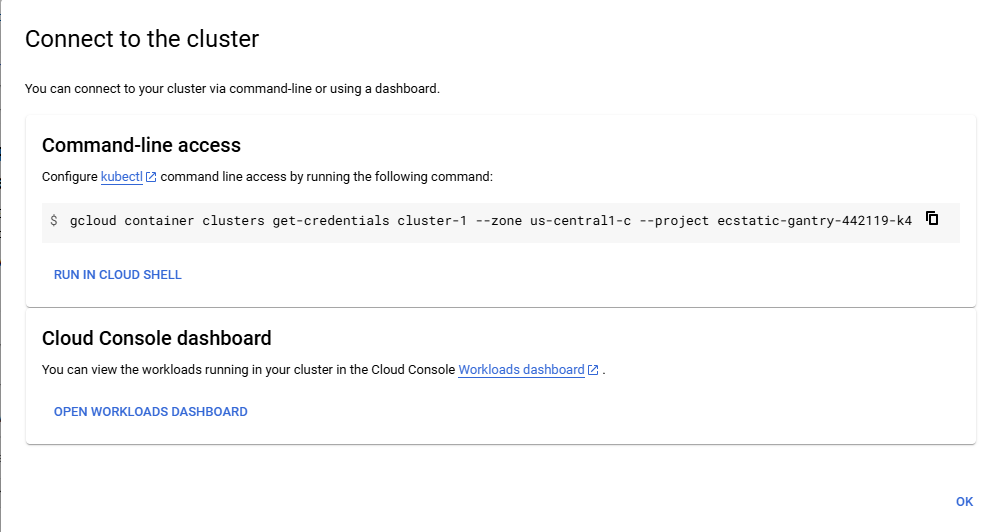
Alegem să creem un cluster de tip Zonal, deoarece este mai ieftin. Regional, care lucreaza pe mai multe zone.  
Observăm că clusterul meu s-a creat cu succesȘ

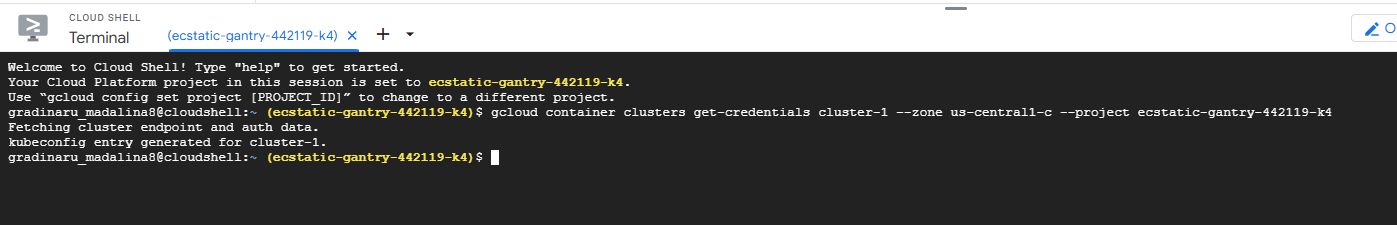


Alegem aceasta optiune

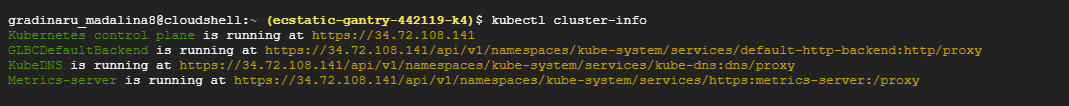


Rulam comanda in CloudShell pentru a asigura conectarea



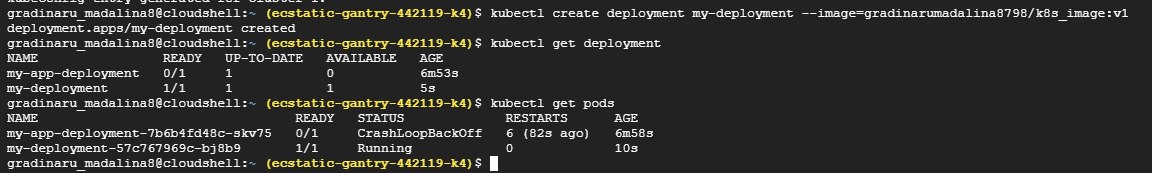


Comanda kubectl cluster info asigura că clusteru a fost conectat corect si toate IP serviciilor sunt disponibile.



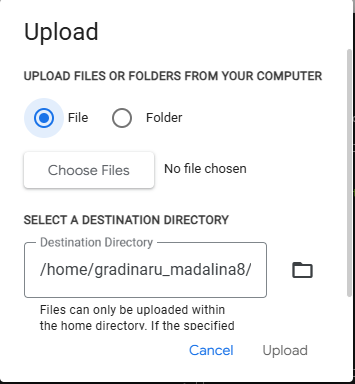
kubectl create deployment

kubectl get deployments



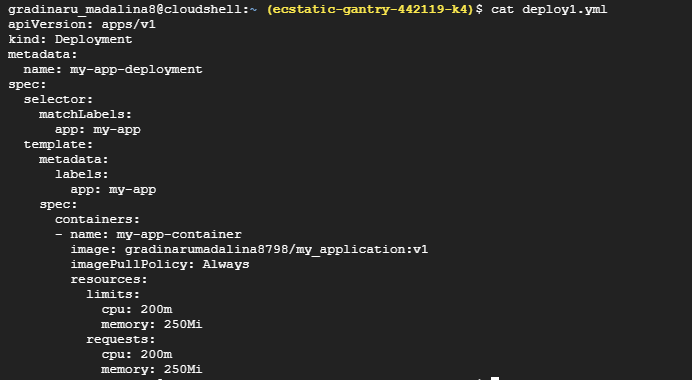


Facem upload la fisiere din directorul propiectului

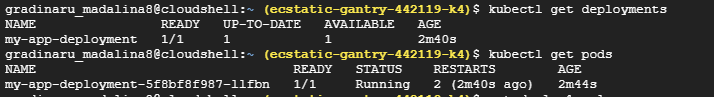


C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{199C250D-A5A7-44AC-9BC0-8D51DA0EF192}.png

Acum creem deployment cu deploy1.yml



C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{82484BE5-0C6C-4864-B7D6-695BB9A38BCF}.png



Creem si un serviciu de tip Load Balancer pentru a lansa aplicatia noastra:  
C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{3B11C821-C1E6-46D8-B3D5-AA85E4A59795}.png

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{451D6929-EA90-48A0-B281-E9FA99FAC881}.png

**Configurăm autoscalarea pe baza utilizării CPU pentru deployment-ul my-app-deployment folosind Kubernetes Horizontal Pod Autoscaler (HPA).**

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{113CDCA1-B067-4B31-A130-BDC2C9D17BDC}.png

gradinaru\_madalina8@cloudshell:~ (ecstatic-gantry-442119-k4)$ kubectl autoscale deployment/my-app-deployment --cpu-percent=15 --min=1 --max=10

**horizontalpodautoscaler.autoscaling/my-app-deployment autoscaled**

**--cpu-percent=15**: Autoscaler-ul va încerca să mențină utilizarea CPU-ului la aproximativ 15%. Dacă utilizarea CPU depășește această valoare pe un pod, Kubernetes va lansa mai multe poduri pentru a distribui încărcătura.

**--min=1**: Numărul minim de poduri va fi 1. Dacă nu este necesară scalarea, rămâne un singur pod.

**--max=10**: Numărul maxim de poduri poate ajunge până la 10, dacă este necesar, pentru a susține cerințele de trafic.

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{E2A56BA4-E161-4038-828D-86CDB6D567D3}.png

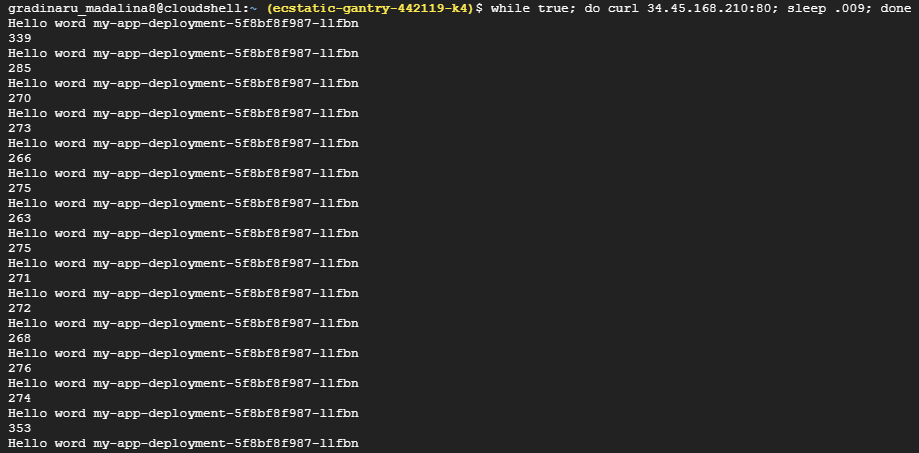
**TARGETS** arată utilizarea **CPU** țintită de **15%**. În cazul tău, valoarea este **<unknown>** pentru că încă nu există date despre utilizarea CPU-ului.

**REPLICAS** arată că, momentan, există **1 pod activ**.

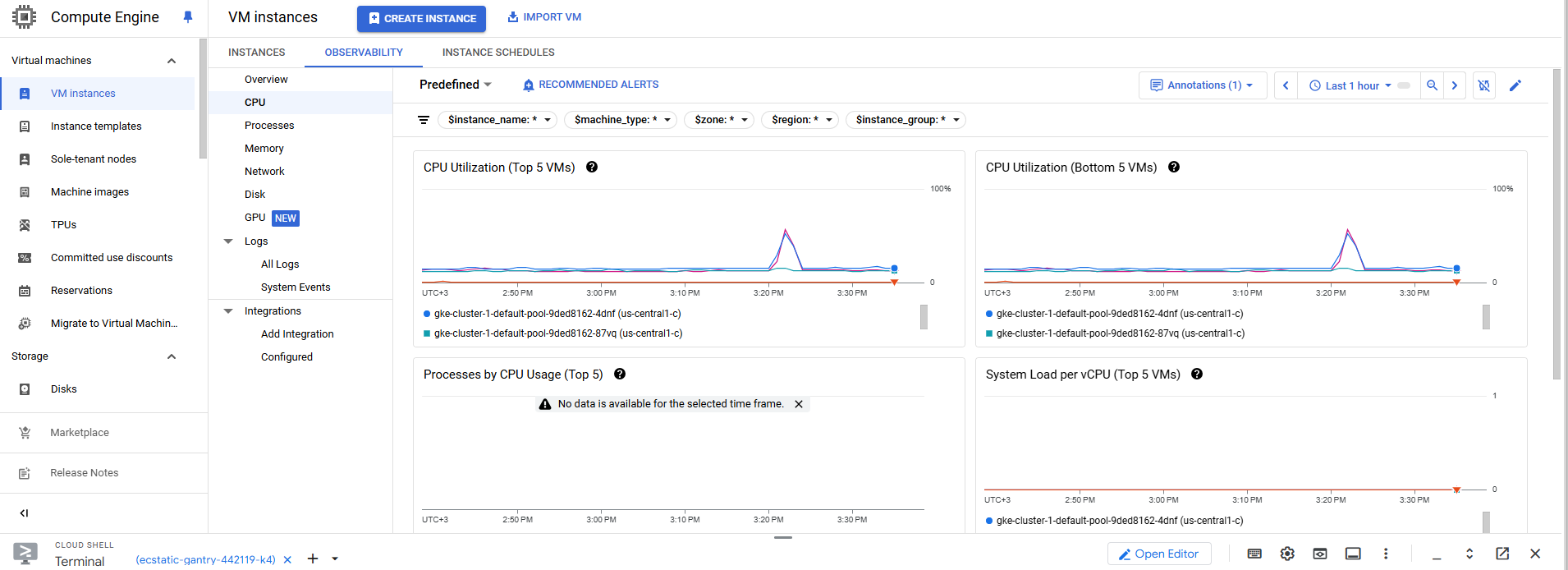
**MINPODS** și **MAXPODS** indică limitele de scalare: **1 pod minim** și **10 poduri maxim**.

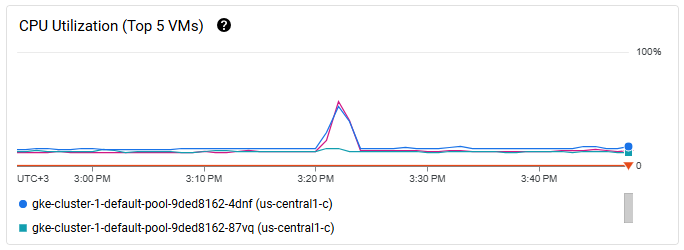
**Testare**

Acest test face un număr mare de cereri HTTP la aplicație pe portul 80 pentru a simula încărcătura (traficul HTTP). Cererile sunt făcute într-un interval foarte scurt (sleep .009), ceea ce provoacă o utilizare mare a CPU-ului pe podul curent.

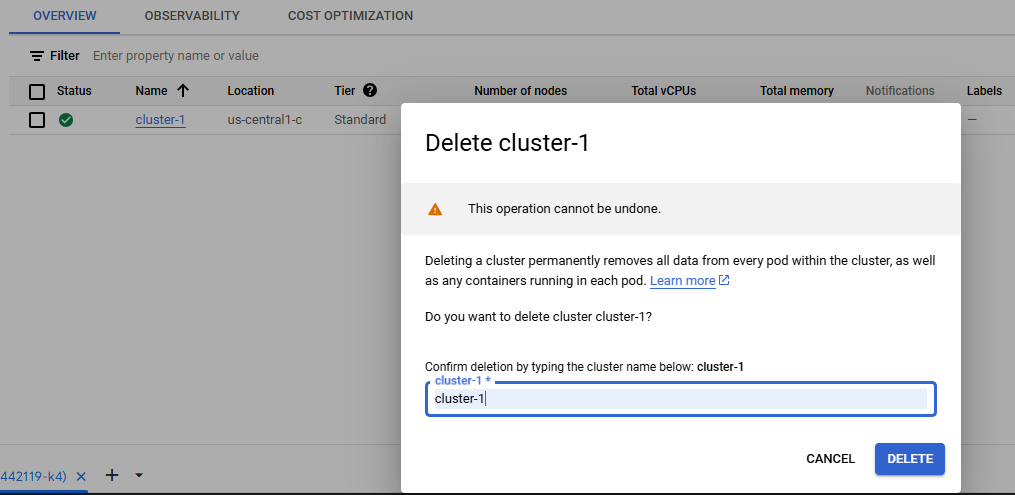


C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{FEFA288E-A0A3-4902-9737-82626B4CD526}.png



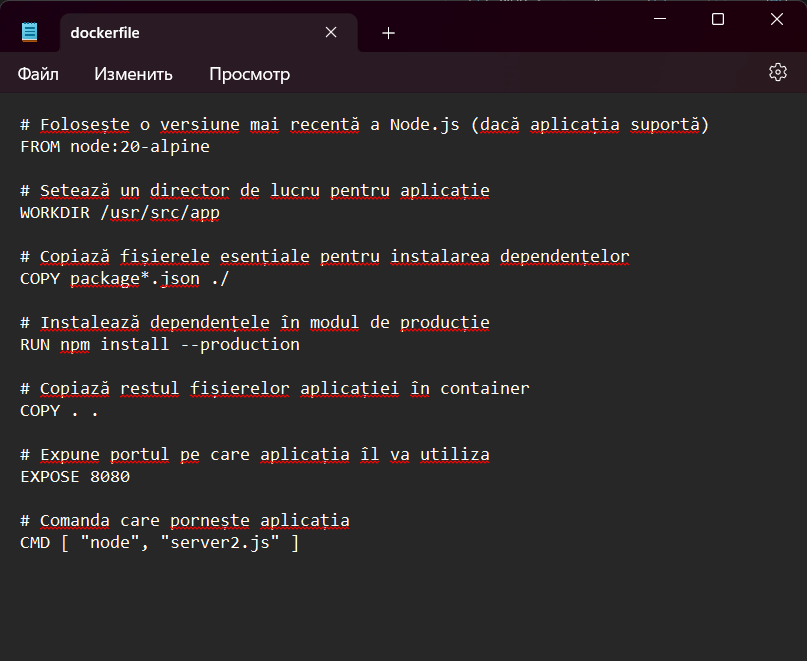


Ștergem clusterul

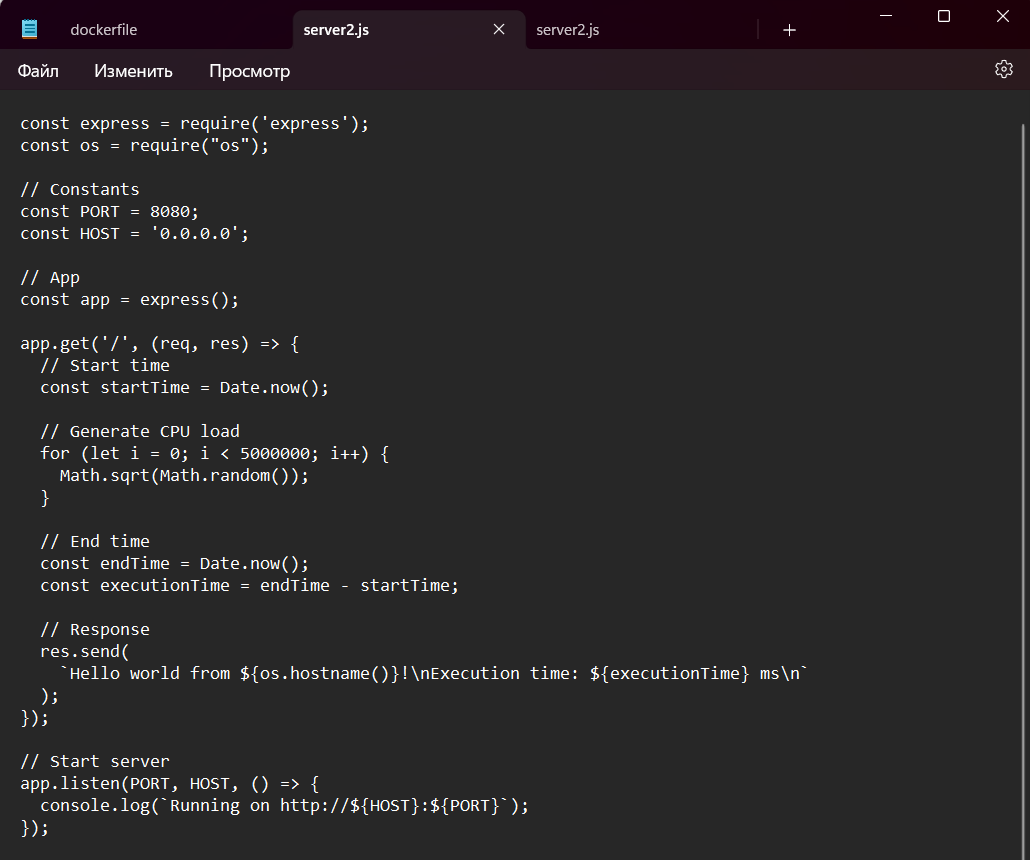


**Exemplul 2:**

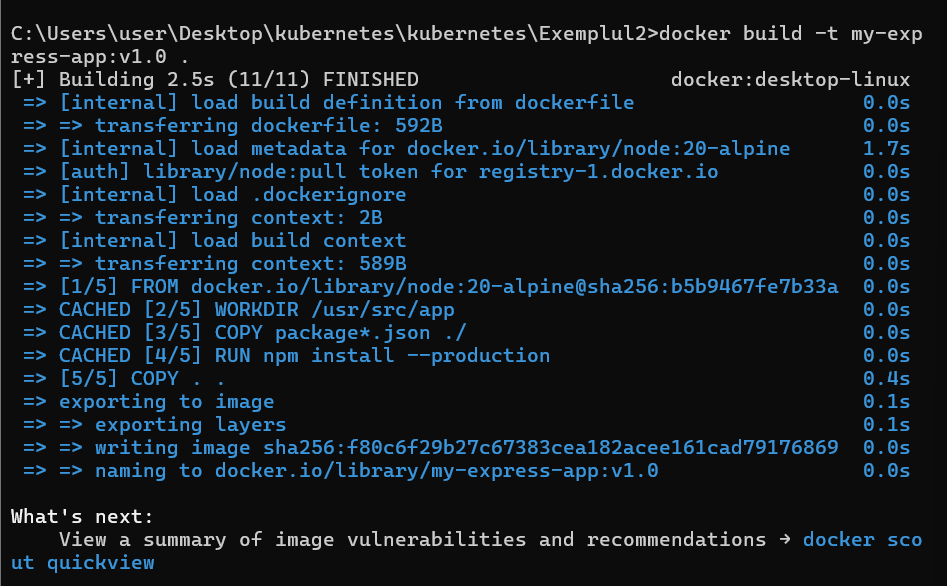
Utilizăm fișierele din lab2  
Dockerfile



server2.js

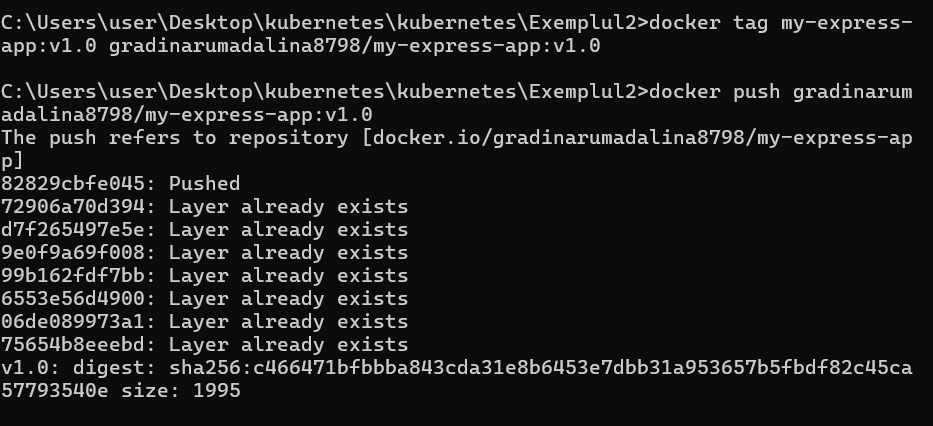


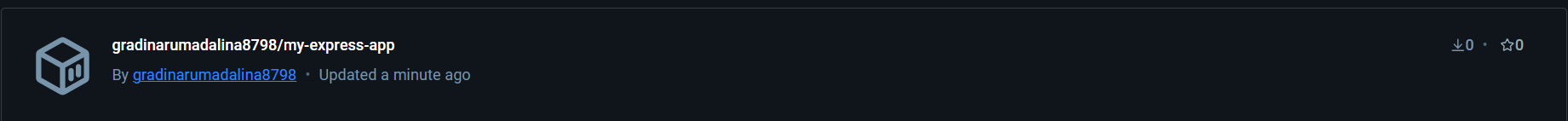
construim imaginea cu docker build .





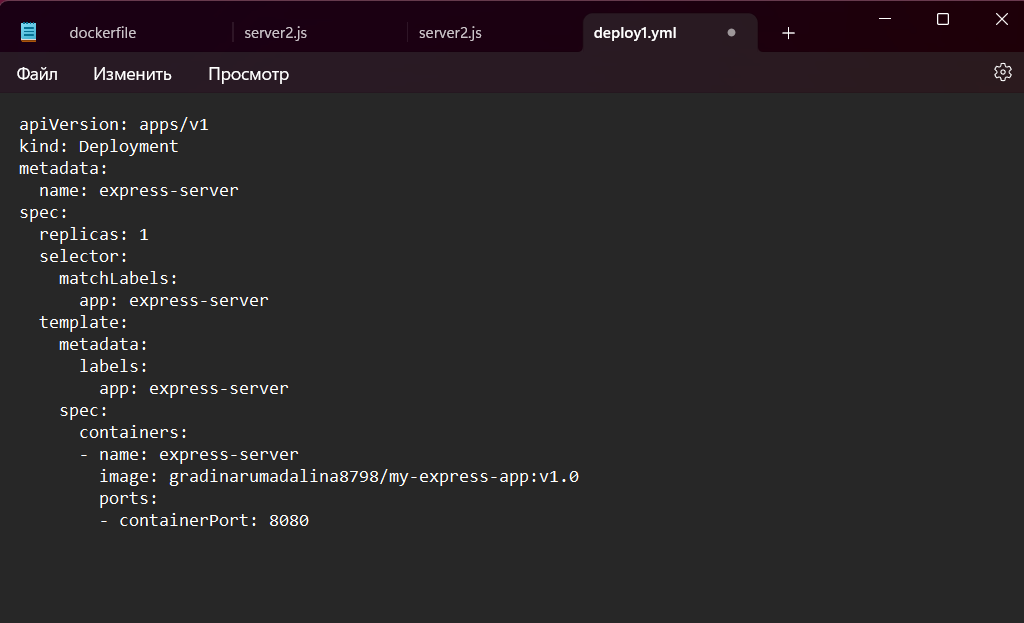
Îi atribui un tag și îi fac push pe Docker HUB

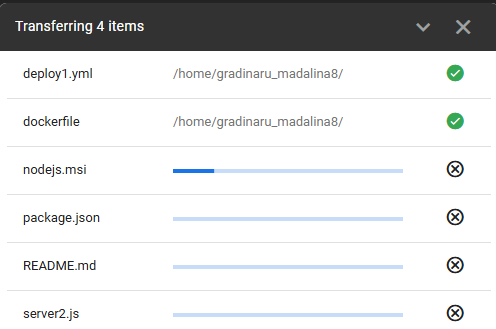


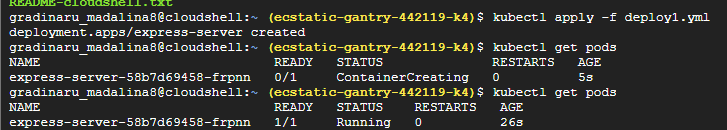


Deci, imaginea a fost creată cu succes.

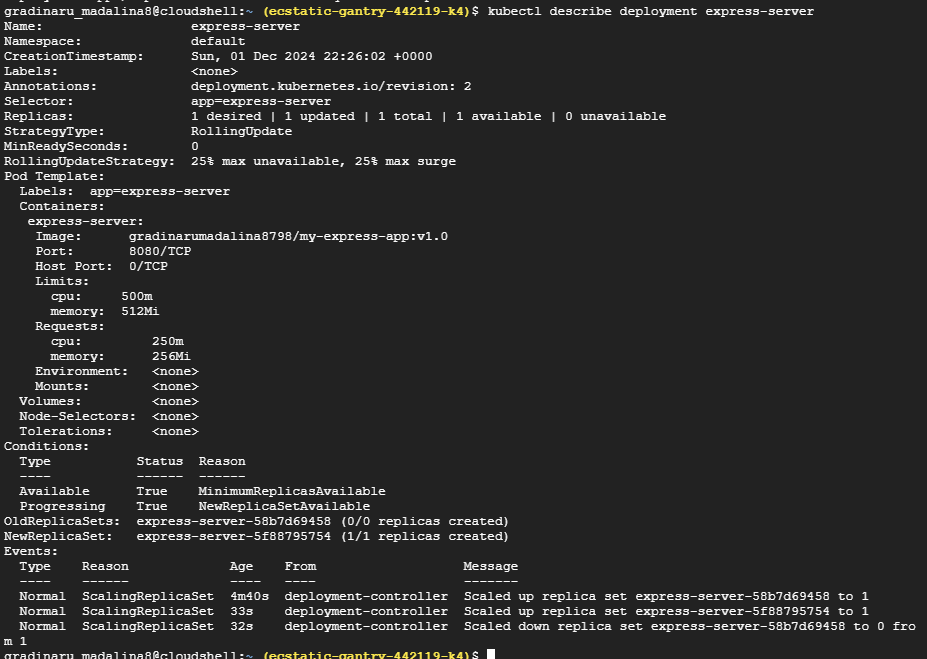
Descriem fișierul deploy1.yml pentru a putea in baza lui și a imaginii să creem un deployment.



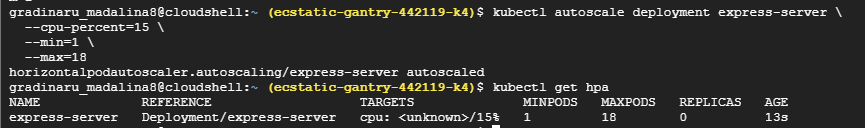
Facem upload pentru fisierele din directorul de lucru pe Google Cloud din Shell.  


Creem deploymentul si verificăm succesul creării  


Pentru acest deployment setam resoursele din linia de comanda  
C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{0D7EE04C-6495-40F2-9B51-EE5C33F43799}.png

Verificăm  


Urmează să facem autoscalarea



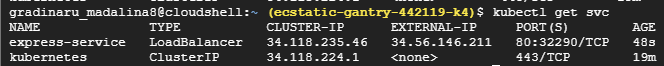
**--cpu-percent=15**: Setează pragul de utilizare a CPU la 15%. Dacă utilizarea CPU depășește acest procent, HPA va crea mai multe replici pentru a reduce încărcătura pe fiecare pod.

**--min=1**: Setează numărul minim de replici la 1, ceea ce înseamnă că dacă utilizarea CPU este sub pragul setat, va rămâne cel puțin un pod.

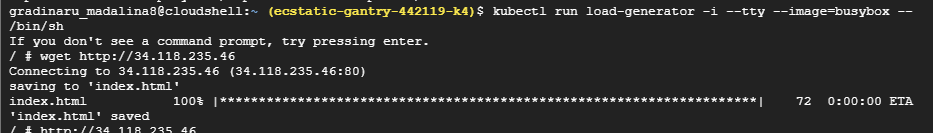
**--max=18**: Setează numărul maxim de replici la 18, ceea ce înseamnă că HPA nu va crea mai multe de 18 replici, indiferent de cerințele de scalare.

Lansăm un serviciu in baza deploymentului

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{E2E08E5C-E6BB-47B2-A1EE-59D81357241C}.png  
Obținem IP-ul extern

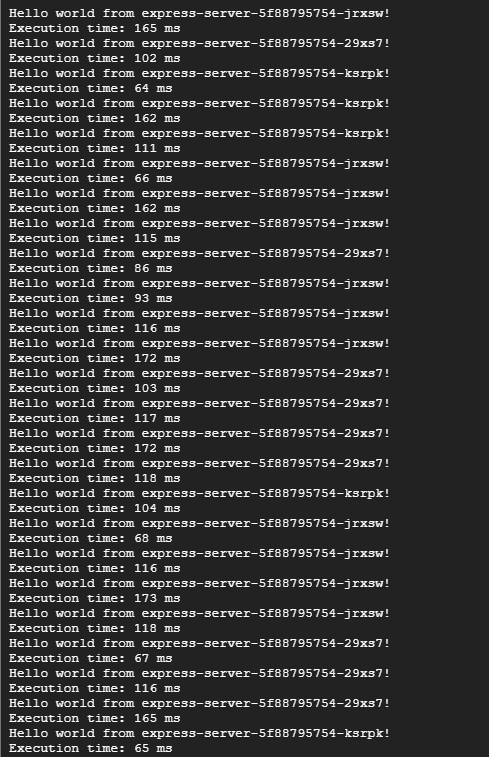


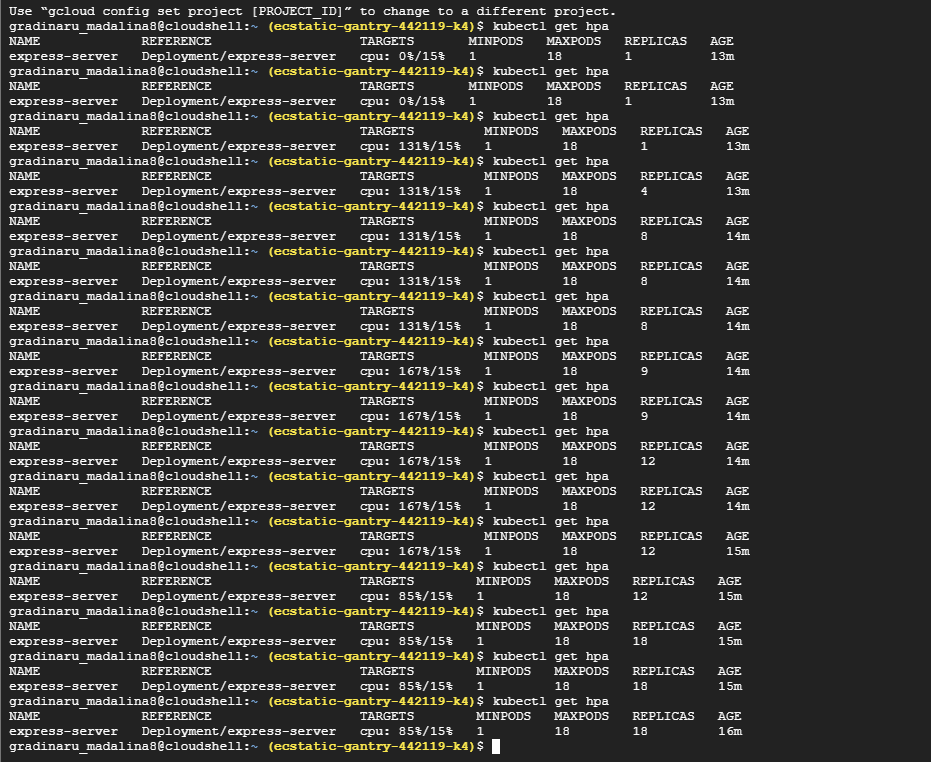
Lansăm aplicația și îi atribuim mai multe requesturi



C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{21C6E26A-202D-48E0-A2AE-5F6AF840D9D2}.png

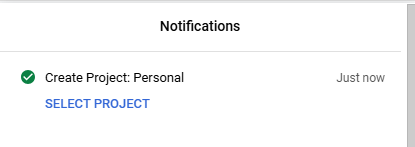
C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{31DE34A4-1DA3-430C-8F38-403F1D64A81D}.png



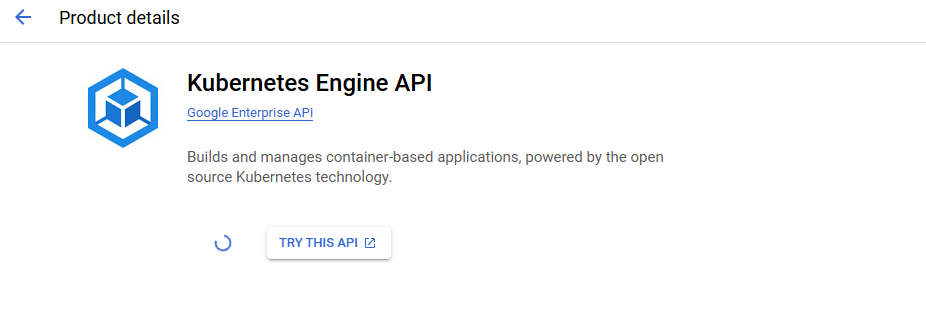


**Exemplul 3**

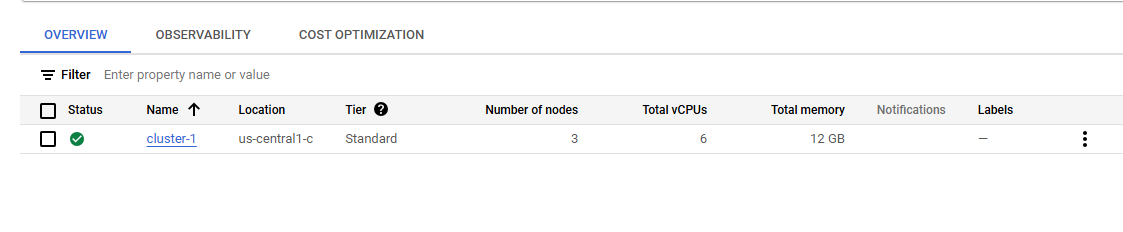
**1. Accesăm Google Cloud și creem un nou proiect**

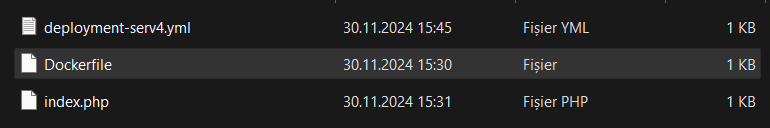


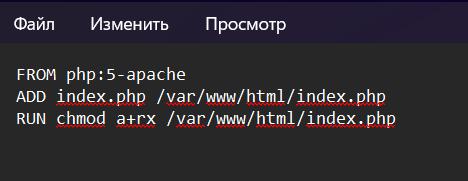
**2. Setăm Enable pe Kubernetes Engine API**



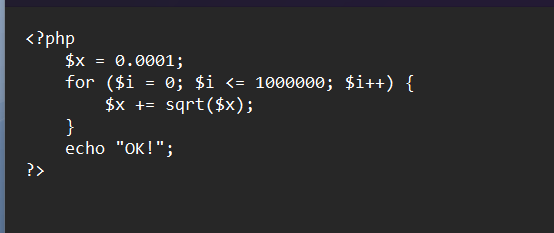
3. Creăm un cluster local, standart :

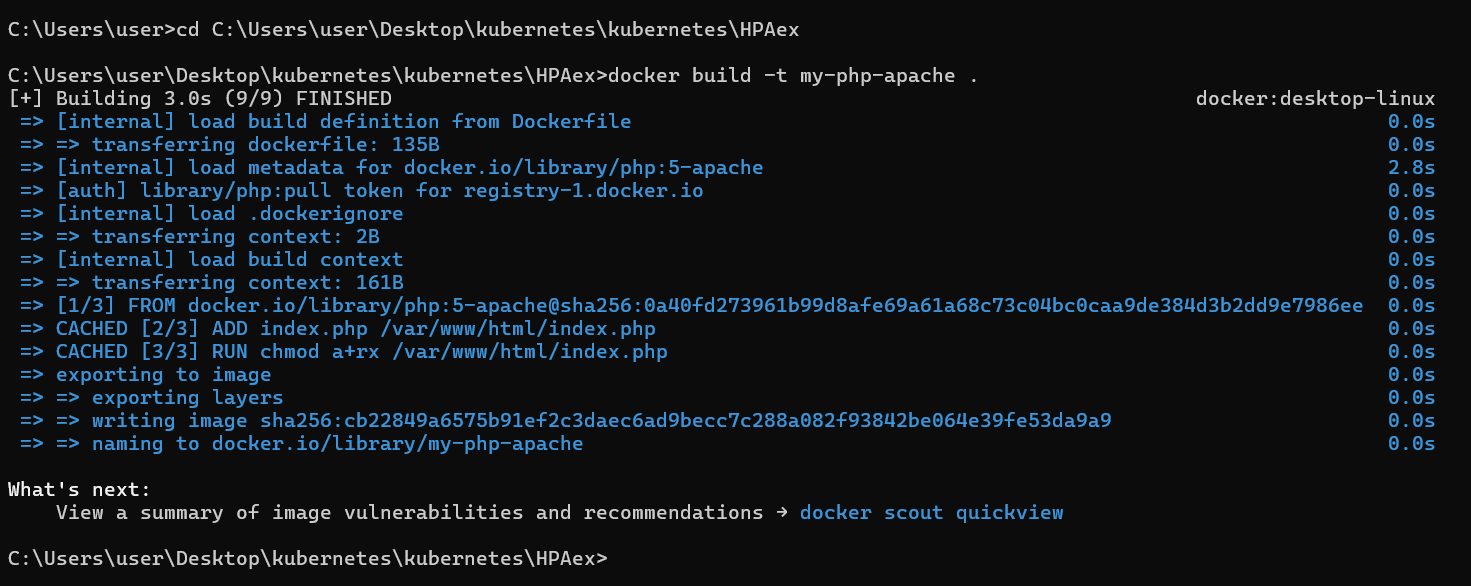


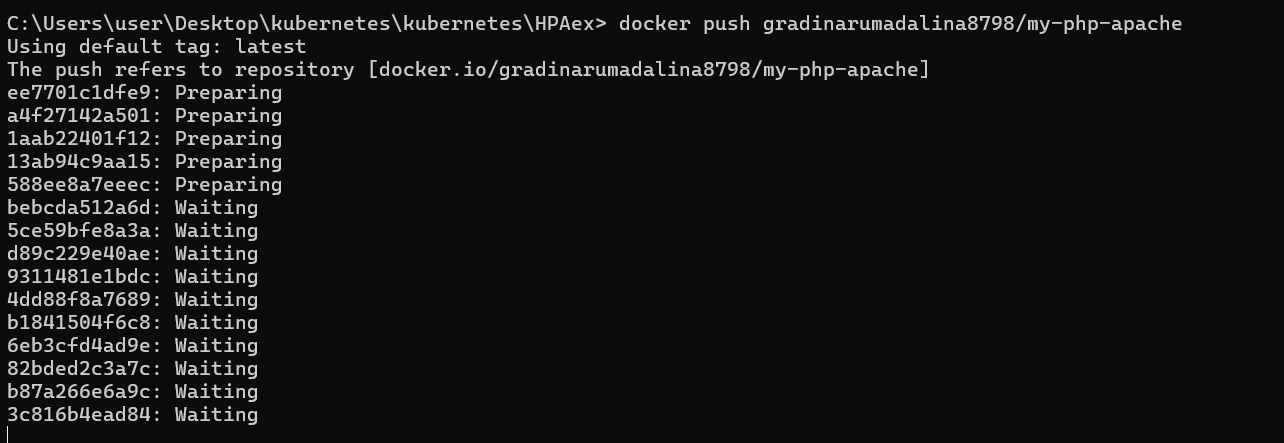
4. Pregătim fișierele necesare:  


Dockerfile:  


index.php:

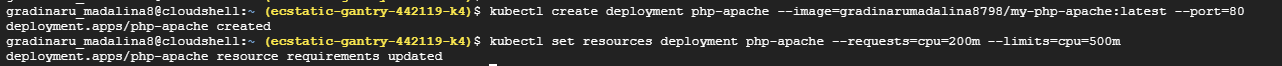


5. Ne mutăm în directorul principal și creem imaginea docker:  


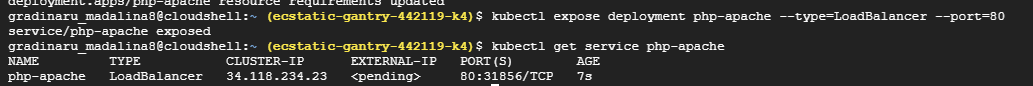
6. După ce s-a creat cu succes, o trimitem către remote cu un tag:  


**3. Deploy și expunerea aplicației**

1. **Creați o implementare Kubernetes pentru imagine**:



1. **Expuneți implementarea ca un serviciu**:
2. **Obțineți adresa IP externă**



C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{D51635D5-0298-4E33-8490-D395A5006849}.png

**4. Crearea Horizontal Pod Autoscaler (HPA)**

1. **Creați HPA folosind kubectl autoscale**:

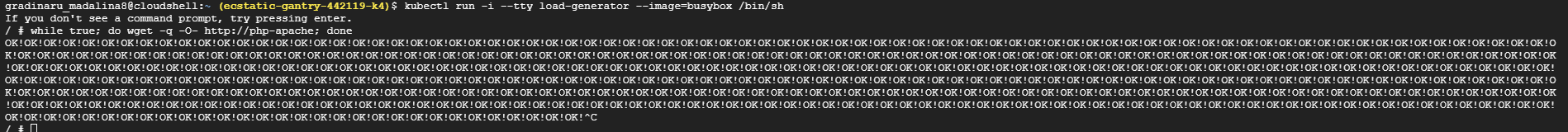
C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{A957D263-E6DB-4F0F-82C9-84457E9078D6}.png

1. **Verificați starea HPA**

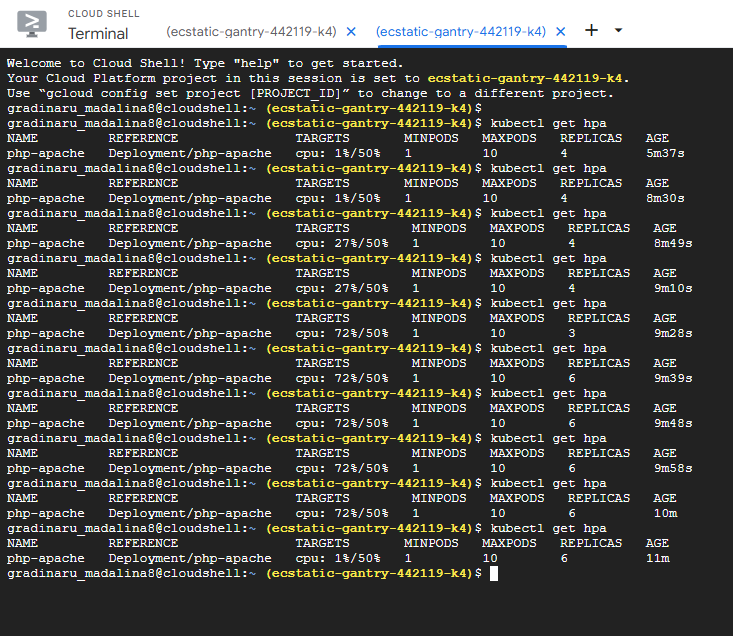
C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{515ECA43-A80B-4377-94A1-D419C16D6E37}.png

**Generarea unei încărcări artificiale**

1. **Creați un container busybox pentru generarea de trafic**:
2. **Trimiteți cereri repetitive către server**:

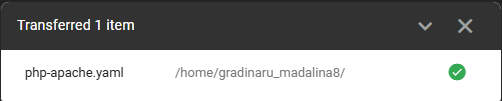


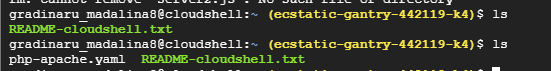
1. **Monitorizați starea HPA**:

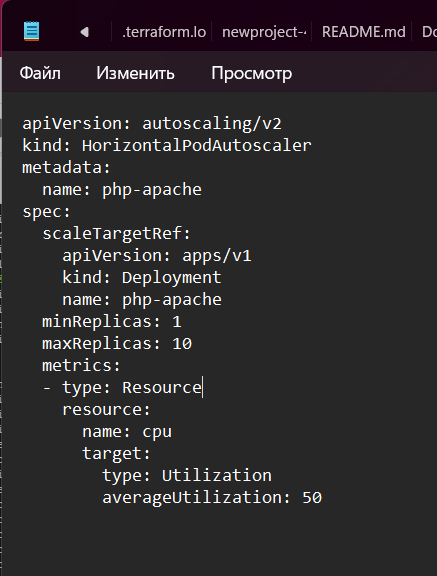
observăm că numărul de replici a tot crescut odată ce avansa timpul de cereri repetitive.  
După oprire (Ctrl+C), numărul lor a ajuns la pragul minim.  


**Crearea declarativă a HPA**

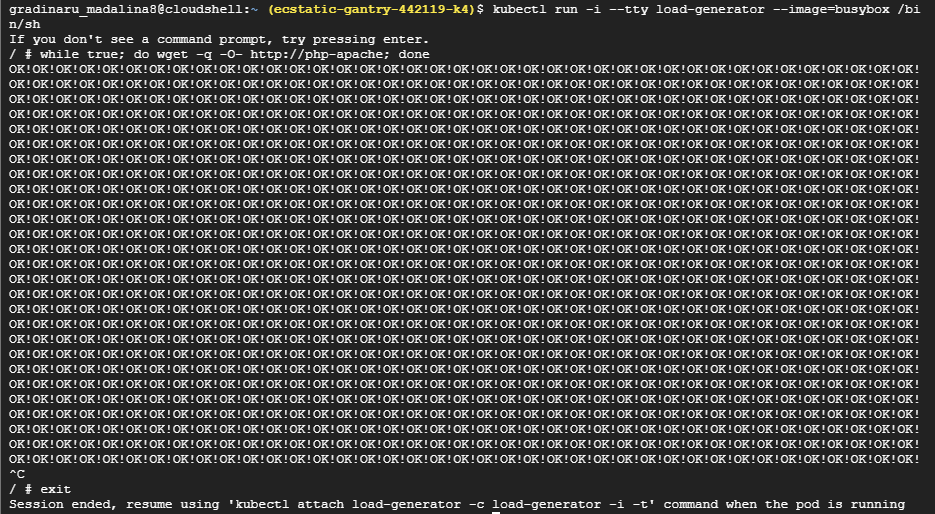
1. **Descărcați fișierul de configurare HPA**:



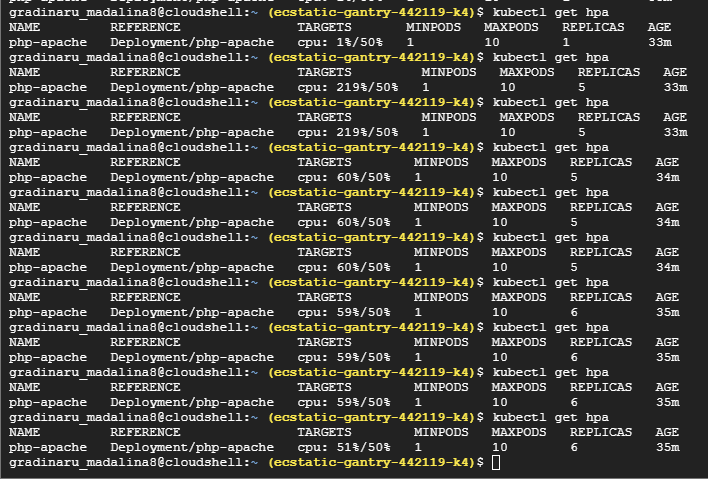


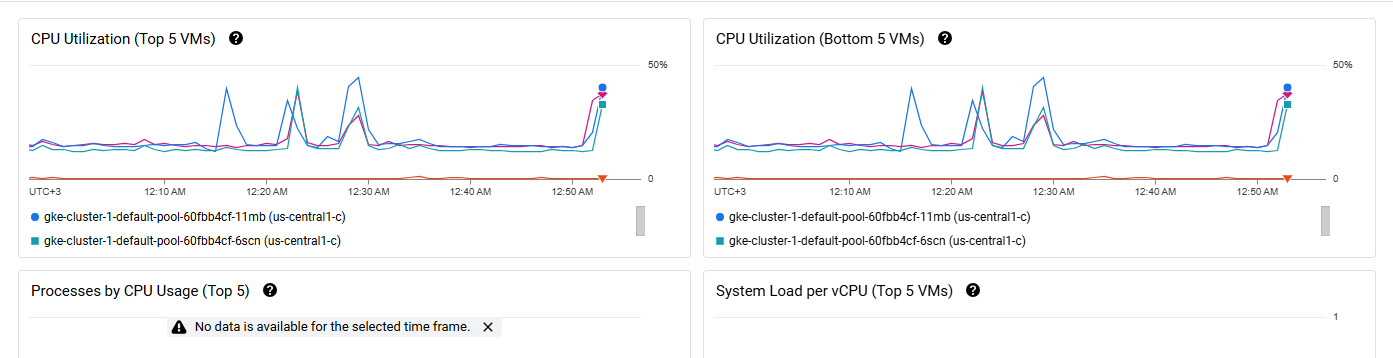


1. **Aplicați fișierul HPA**:



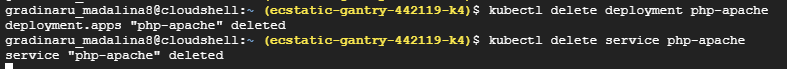
1. **Verificați starea HPA**:





**8. Curățarea resurselor**

1. **Ștergeți implementarea și serviciul**:



1. **Ștergeți HPA**:

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{41721299-89CF-4F8A-81E3-48F9646F4DB5}.png