**UNIVERSITATEA DE STAT DIN MOLDOVA**

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**DEPARTAMENTUL DE INFORMATICĂ**

**GRĂDINARU MĂDĂLINA**

**Cloud Computing**

**Lucrare de Laborator nr.7:**

„Inițiere Terraform”

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

**CHIȘINĂU – 2024**

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

Terraform este un instrument open-source creat de HashiCorp, utilizat pentru automatizarea și gestionarea infrastructurii ca cod (IaC). Este extrem de popular în domeniul DevOps și este folosit pentru a defini, previzualiza și gestiona infrastructura într-un mod declarativ.

**Beneficii ale Terraform:**

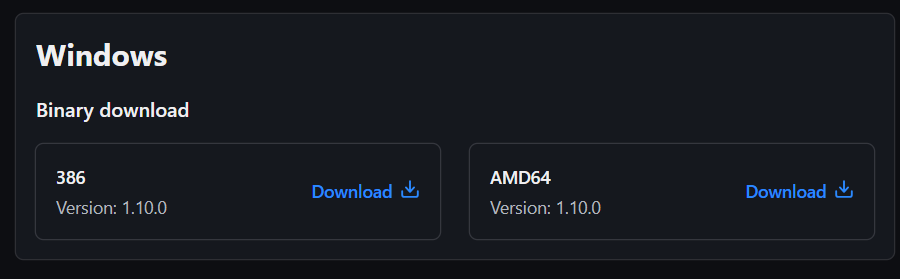
1. **Automatizare completă**: Permite gestionarea infrastructurii fără intervenții manuale.
2. **Portabilitate**: Același set de cod poate fi utilizat pentru mai multe medii și provideri.
3. **Reutilizabilitate**: Modulele Terraform facilitează utilizarea codului reutilizabil pentru implementări frecvente.
4. **Colaborare**: Statele Terraform pot fi partajate între echipe, iar planurile oferă transparență în modificări.

**Exemple de utilizare:**

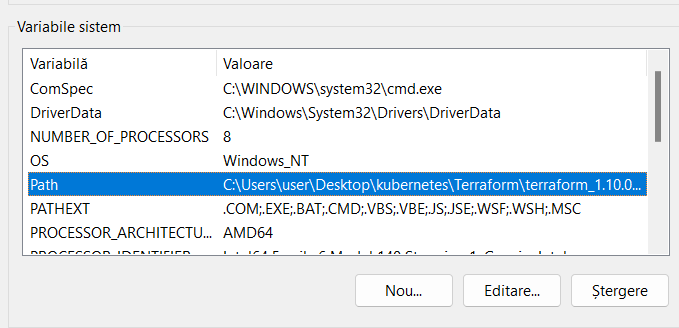
1. Crearea unei arhitecturi complete pe AWS, incluzând instanțe EC2, baze de date RDS și load balancere.
2. Gestionarea clusterelor Kubernetes prin integrarea cu providerul său specific.
3. Implementarea rețelelor complexe, inclusiv subneturi, VPC-uri și reguli de firewall.

# 2. Instalarea Terraform

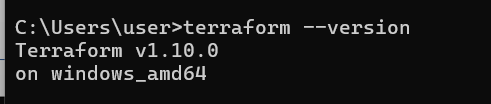
<https://developer.hashicorp.com/terraform/install?product_intent=terraform>



Am salvat terraform.exe în directorul Terraform și am adăugat calea către acest director în variabila Path:

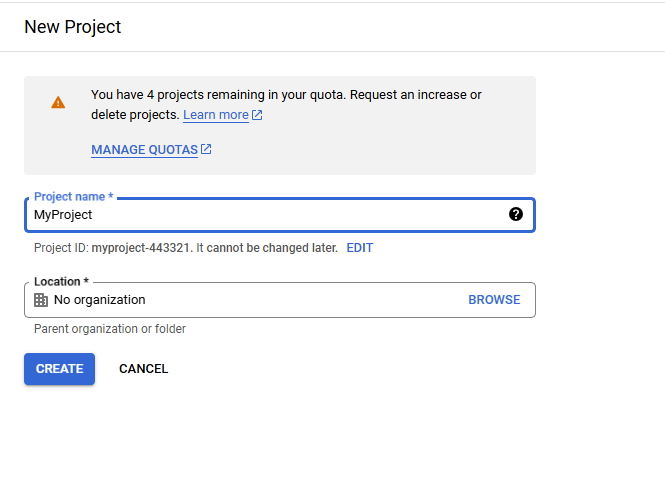


Verificăm instalarea Terraform.



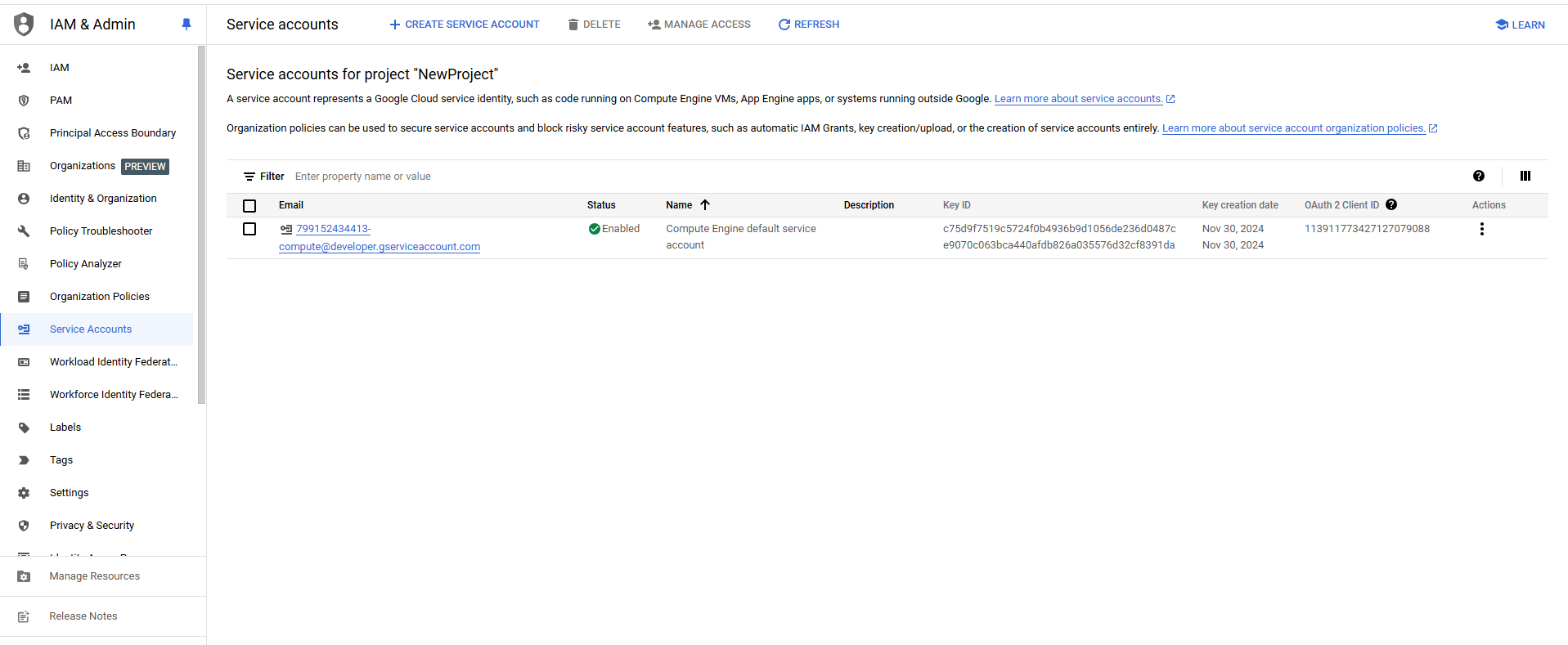
# 3. Crearea unui proiect pe GCP

**Google Cloud Platform (GCP)** este platforma cloud oferită de Google, care permite crearea și gestionarea infrastructurii pe serverele Google. Crearea unui proiect este primul pas pentru a interacționa cu resursele din GCP.



# 4. Lansăm un serviciu Kubernetes Engine API pentru acest proiect GCP

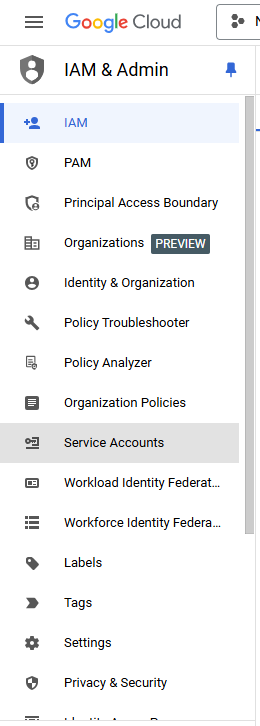
**Kubernetes Engine API** din Google Cloud permite gestionarea și orchestrarea containerelor folosind Kubernetes. Este esențial pentru a crea și gestiona un cluster Kubernetes pe GCP.

**Accesăm secțiunea IAM**  


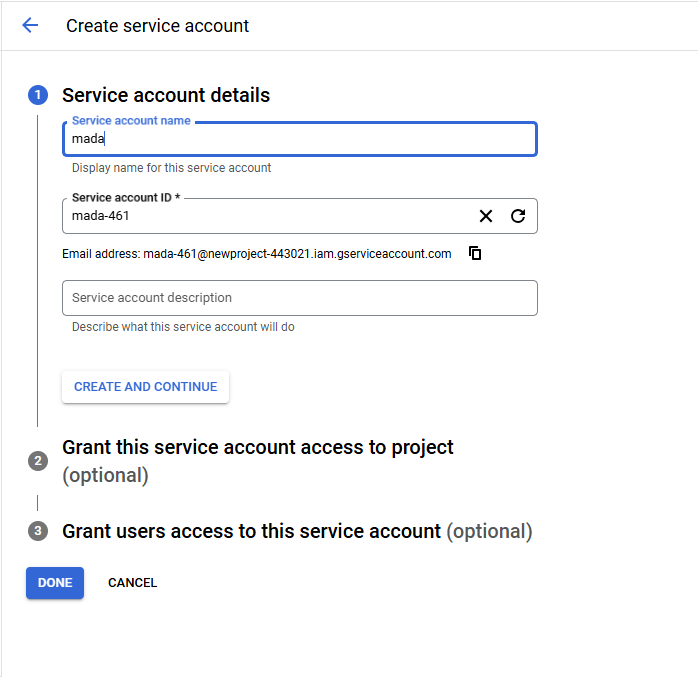
# Creăm un cont de serviciu

**IAM (Identity and Access Management)** este sistemul GCP folosit pentru a controla accesul la resursele Google Cloud. Contul de serviciu este folosit pentru a oferi permisiuni Terraform pentru a interacționa cu GCP.

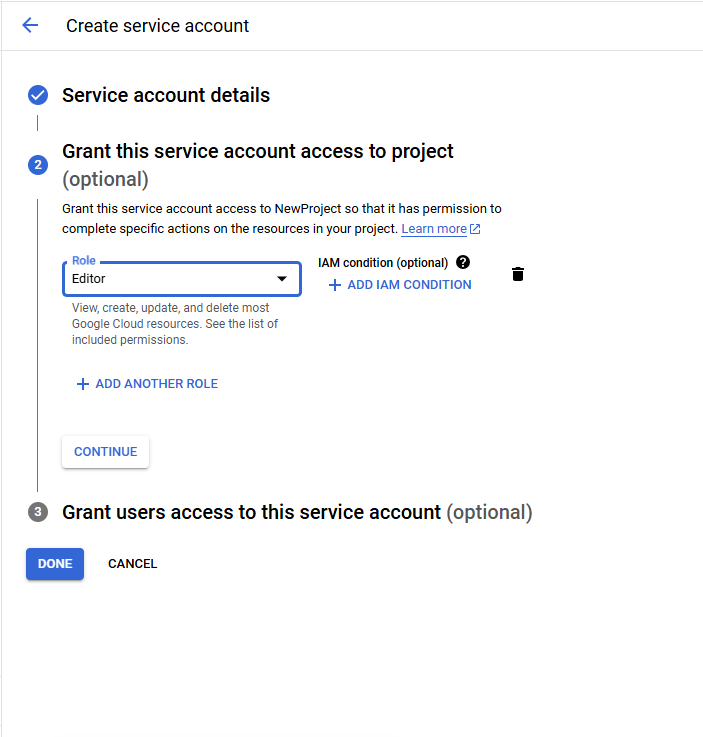
Din meniul din stânga paginii, alegem Service Accounts și apăsăm butonul Create Service Account



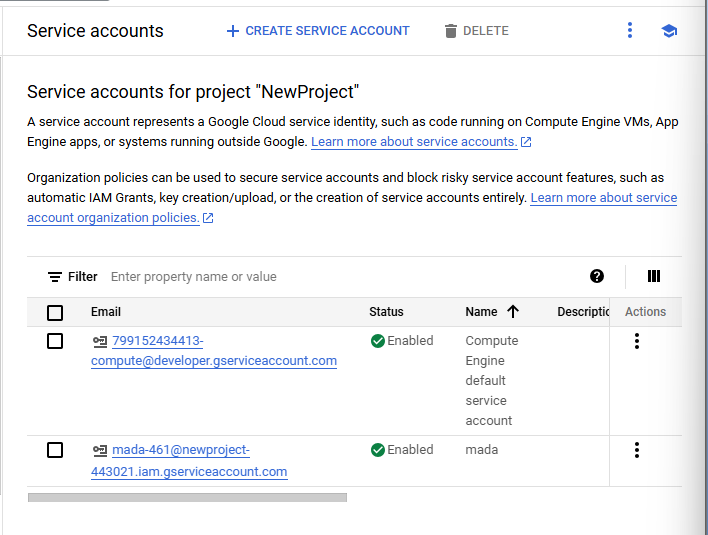
**Atribuim un nume serviciului**



**Selectăm un rol**

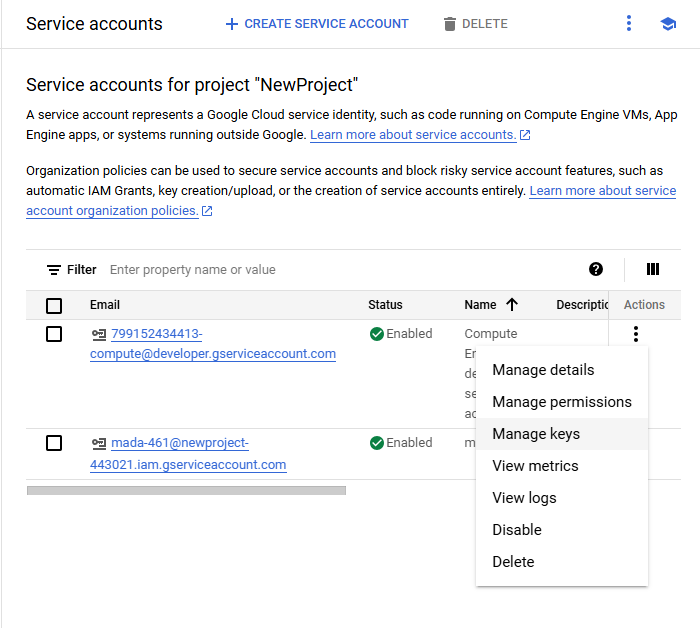


**Opțional putem să setăm anumite permisiuni pentru un utilizator**



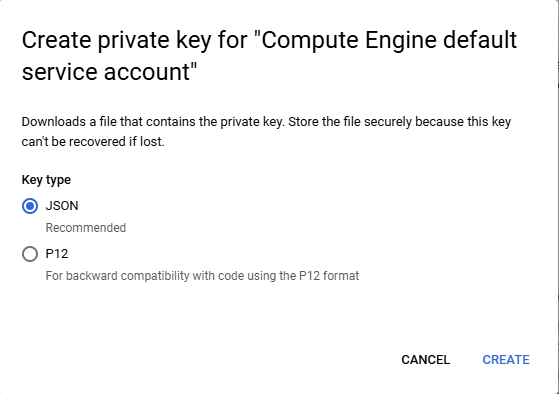
# Creem o cheie pentru acest cont de serviciu:

**1. Apăsăm pe cele trei puncte din dreapta contului**

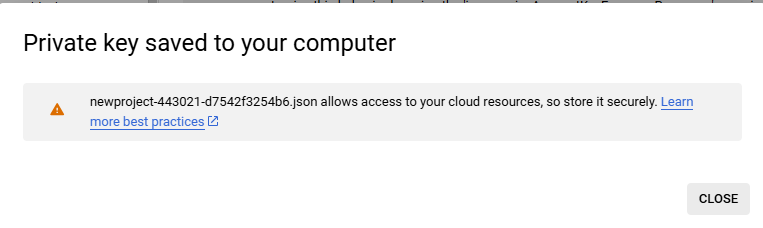


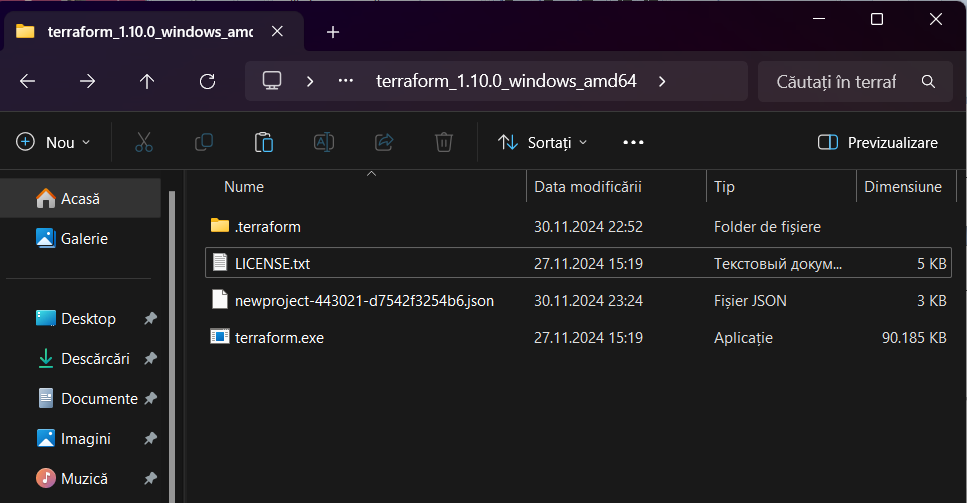
**2. Din fereastra care s-a deschis lăsăm setată ca default ca această cheie să fie salvată în format .json.**

Aceasta este necesară pentru a permite Terraform să acceseze și să gestioneze resursele din proiectul Google Cloud în mod programatic.

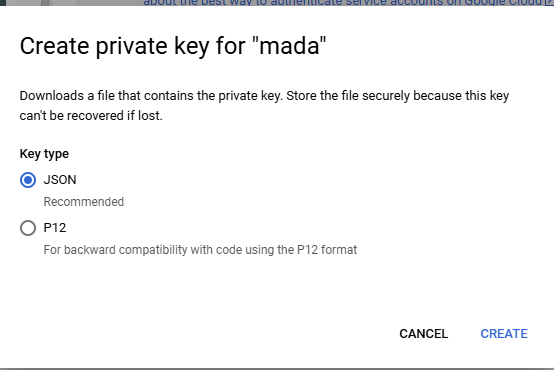


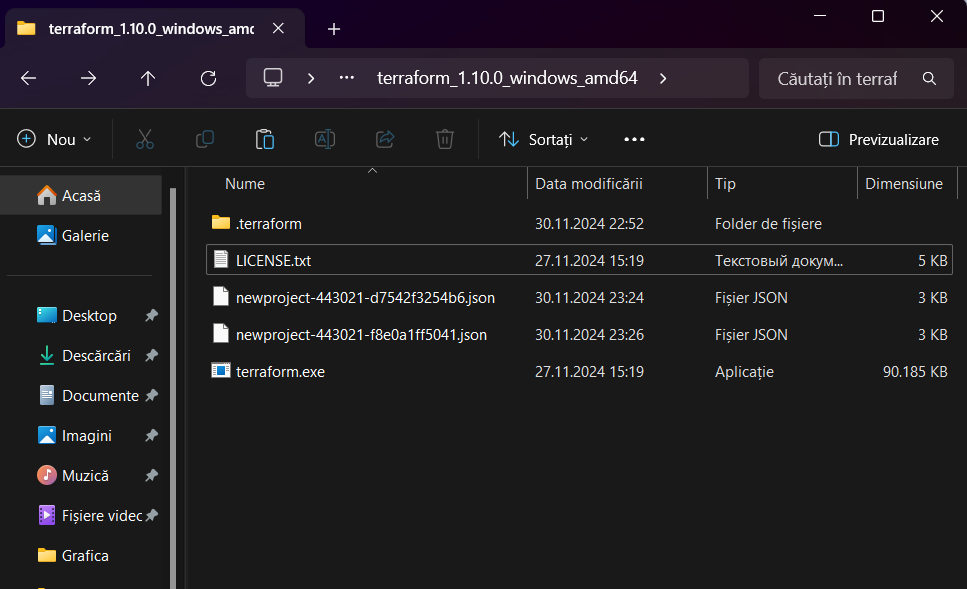
**3. Salvăm cheia în directorul în care se află fisierul terraform.exe**





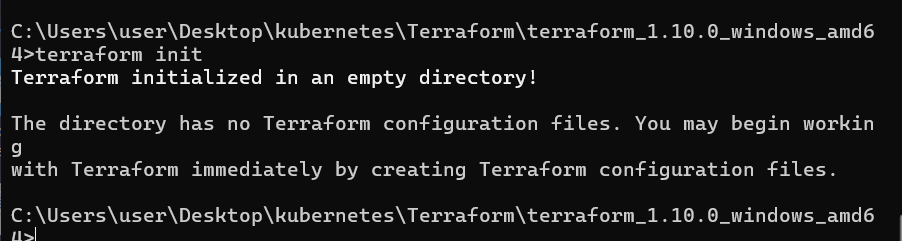
**De asemenea, am realizat o cheie și pentru contul de serviciu personalizat.**

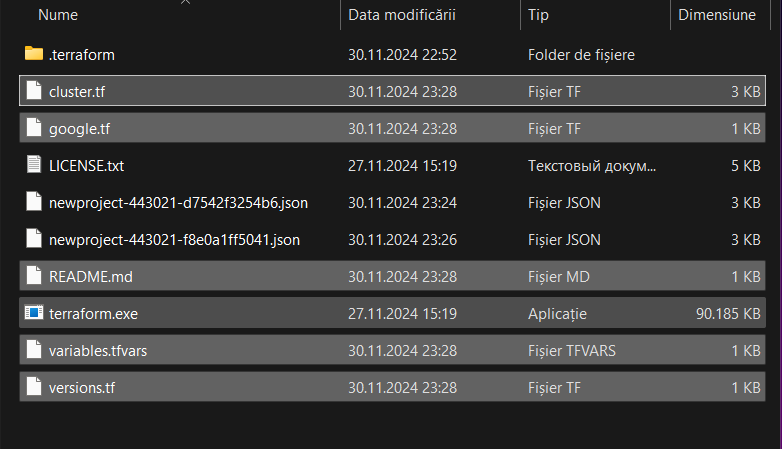




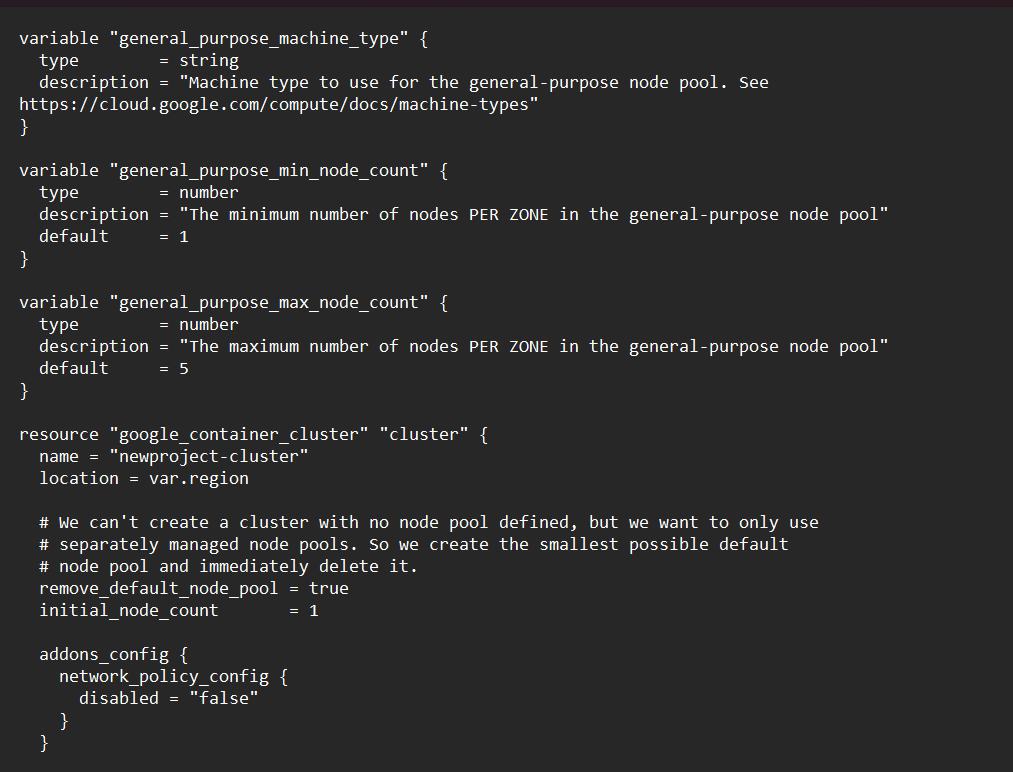
# 6. Crearea resurselor. Run terraform commands:

## terraform init

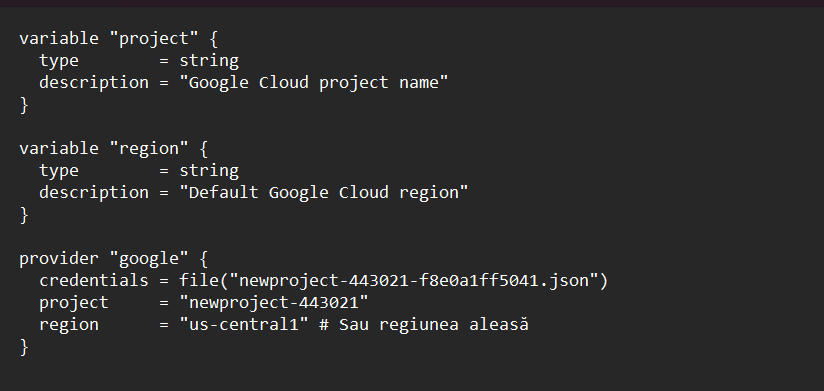


**Am inclus in folderol proiectului fisierele necesare  
**

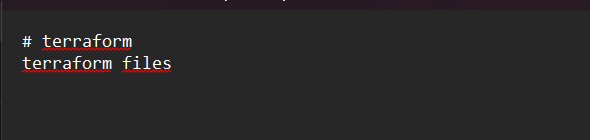
**cluster.tf**: Definirea și configurarea clusterului Kubernetes și a node pool-urilor.



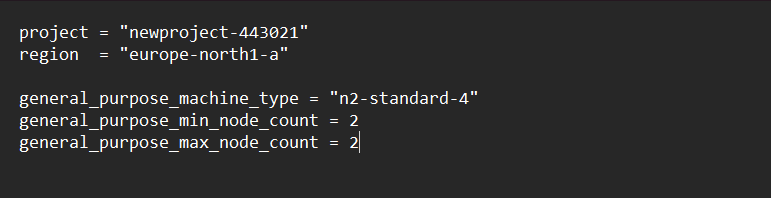
**google.tf**: Configurarea providerului Google pentru Terraform.



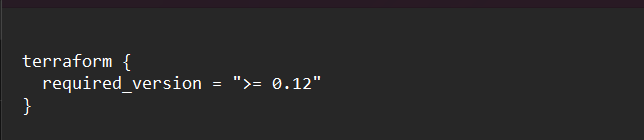
**README.md**: Documentația proiectului, cu instrucțiuni de configurare și utilizare.



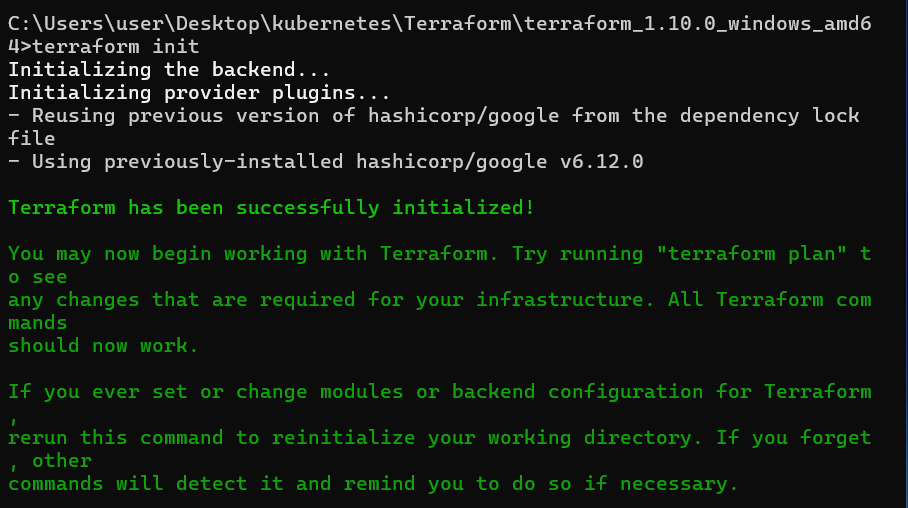
**variables.tfvars**: Setarea valorilor pentru variabilele utilizate în fișierele de configurare.



**versions.tf**: Asigură compatibilitatea între versiunile Terraform și ale providerilor.



# ****6. Inițializarea și Crearea Resurselor****

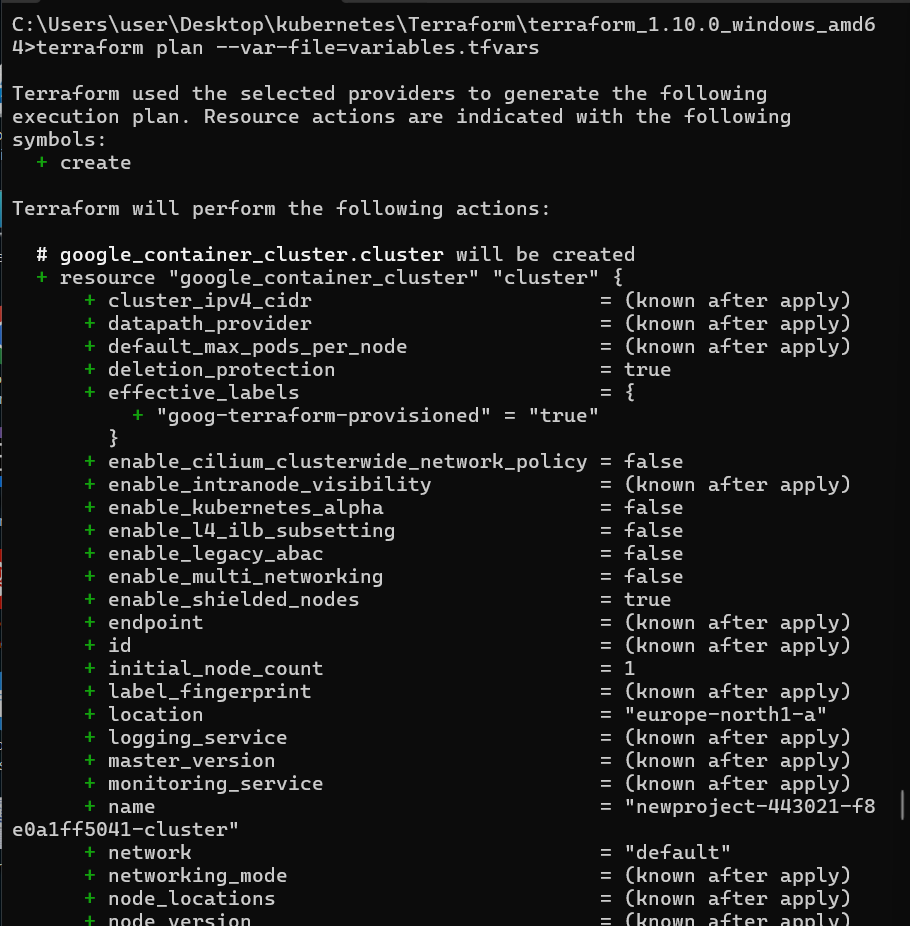
**Odată ce filierele sunt în director, reinițializăm proectul.  
**

**Ne asigurăm că fișierul google.tf conține numele fișierul json correct în secțiunea de credențiale.**



## terraform plan --var-file=variables.tfvars

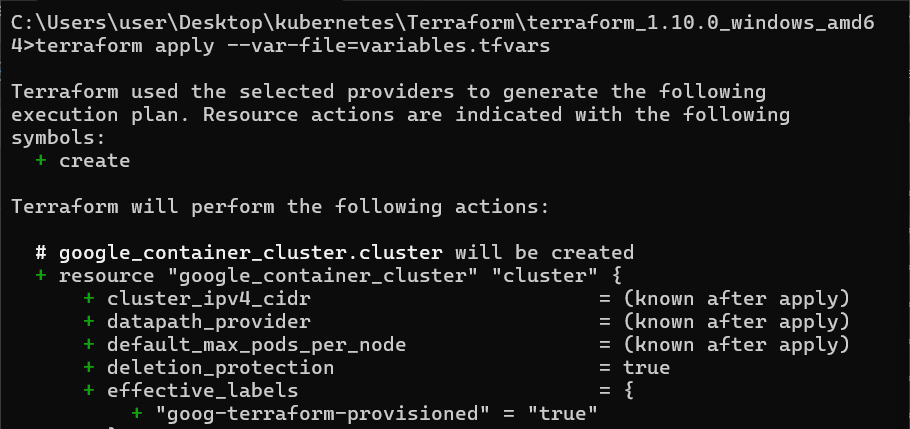
generează un plan detaliat care arată ce resurse vor fi create, modificate sau șterse.



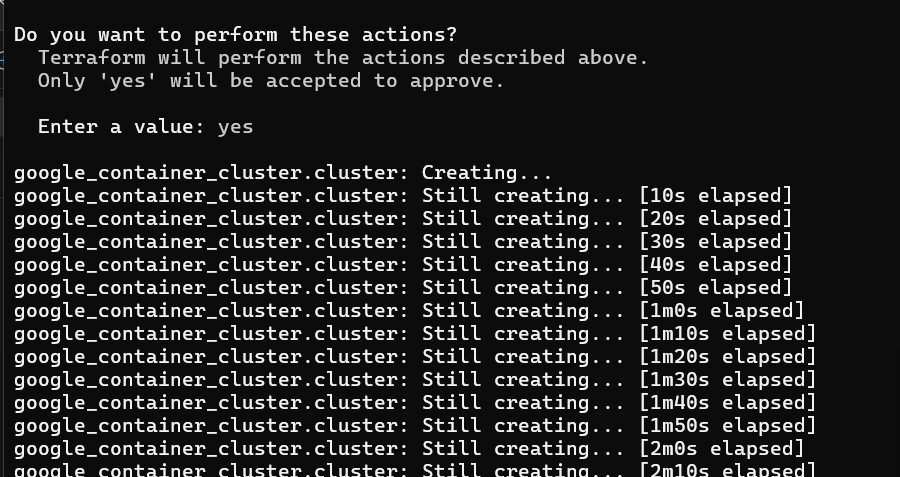


## terraform apply --var-file=variables.tfvars

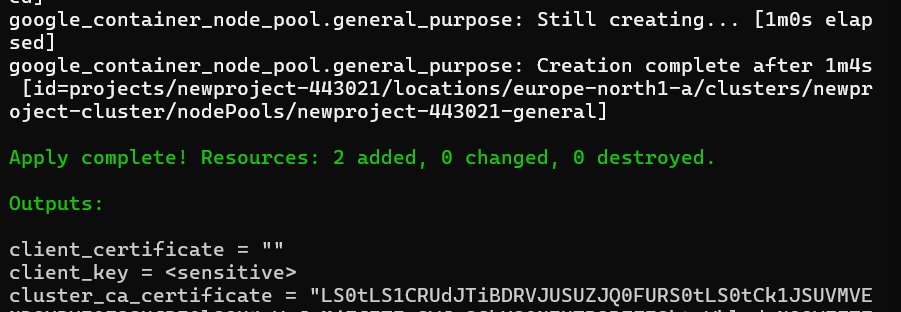
aplică efectiv modificările, creând sau modificând resursele definite.



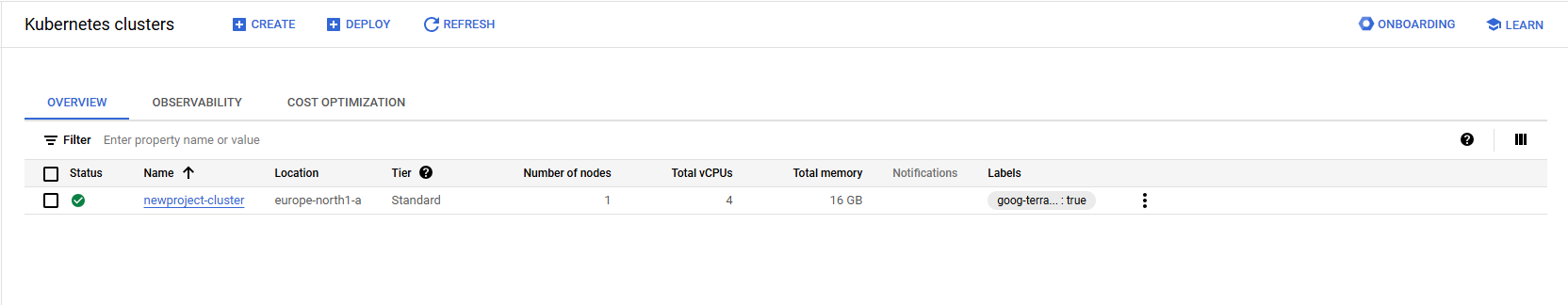
…



…



**Clusterul acum există în Google Cloud.**



# 7.Modificarea resurselor

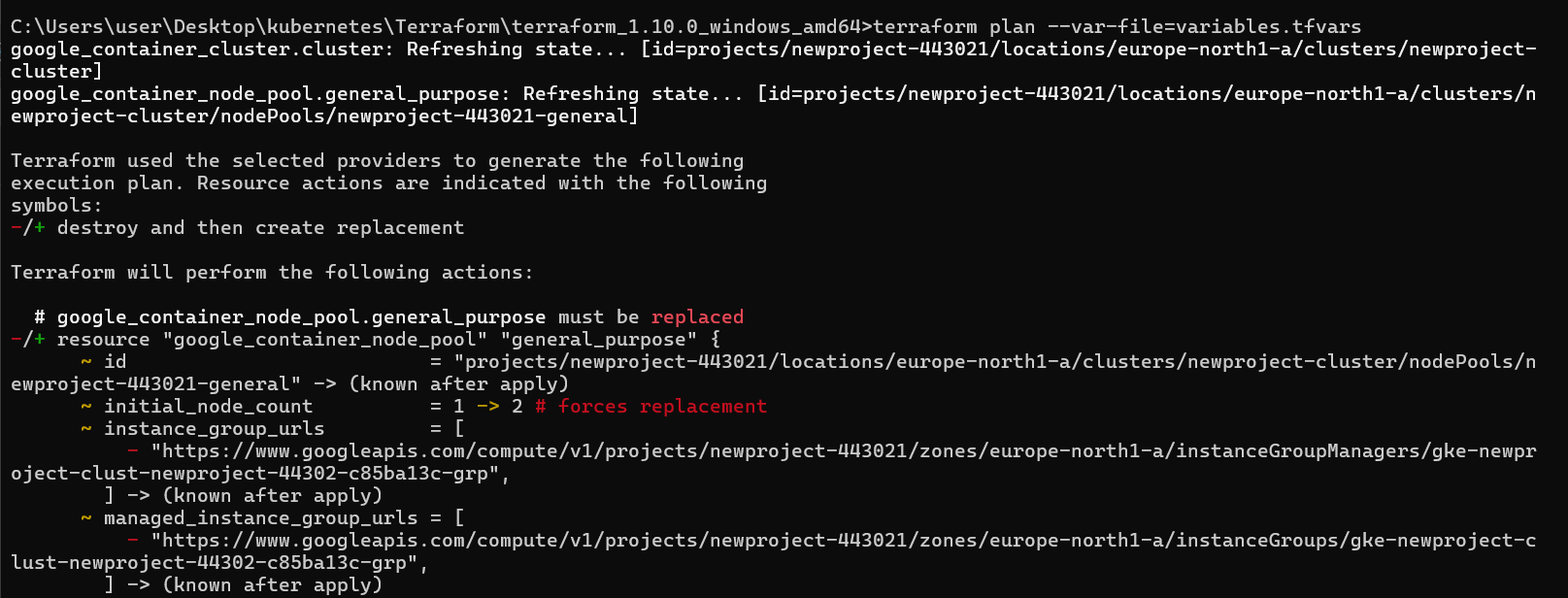
**Facem modificări în fișierul variables.tfvars și repetăm ​​comenzile terraform init, plan, apply așa cum este descris mai sus.**

**Astfel, se permite modificarea infrastructurii deja existente (de exemplu, schimbarea numărului de noduri sau a tipului de mașină).**

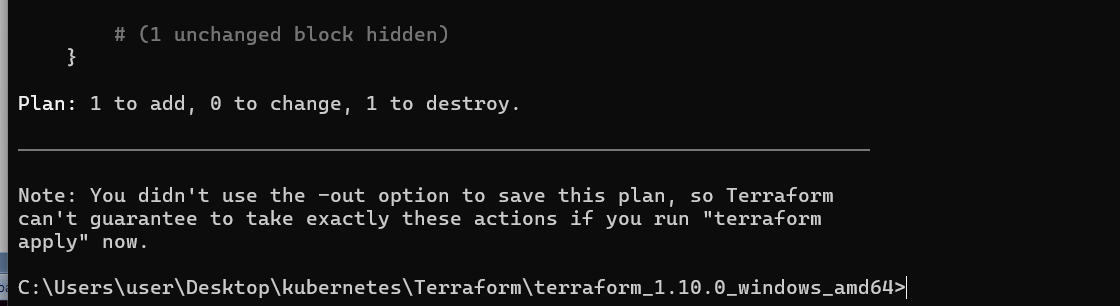
**Dacă aceste valori sunt setate inițiat, atunci ar trebui să fie create 1 nod. Dacă aceste valori sunt setate pe 2, Terraform va crea 2 noduri conform configurației de autoscalare.**



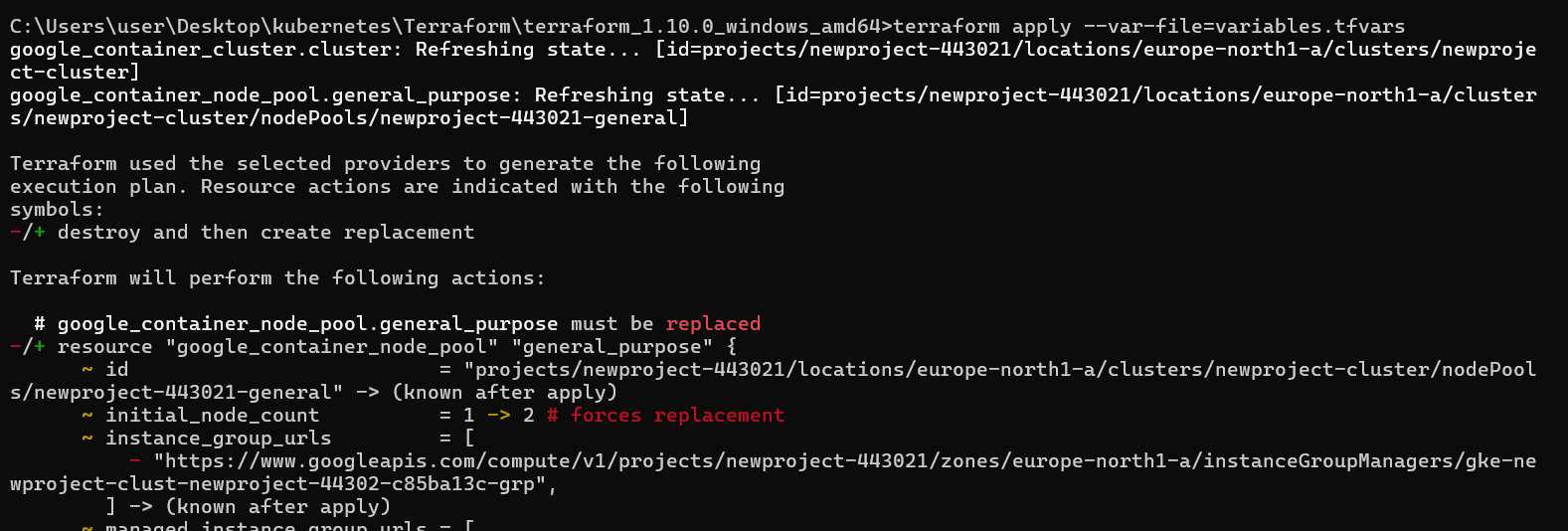
## terraform plan --var-file=variables.tfvars



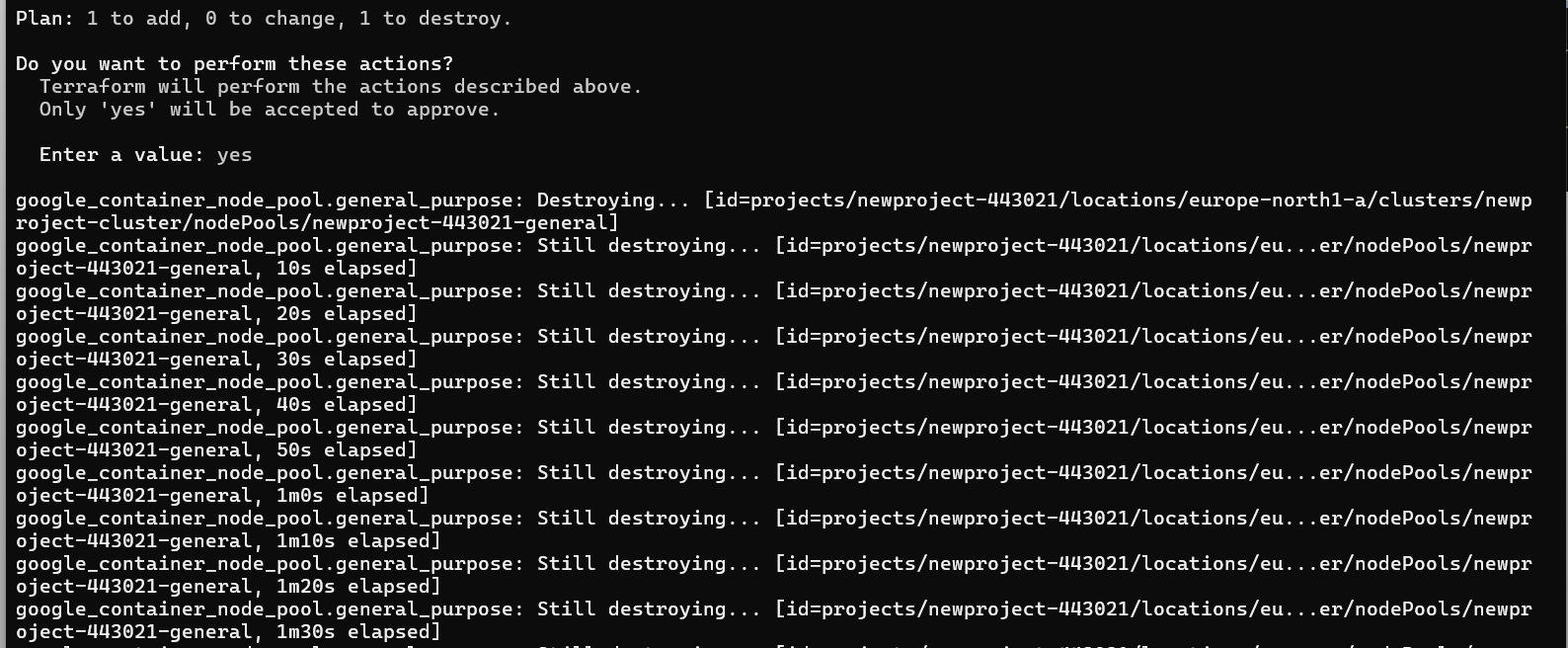
…



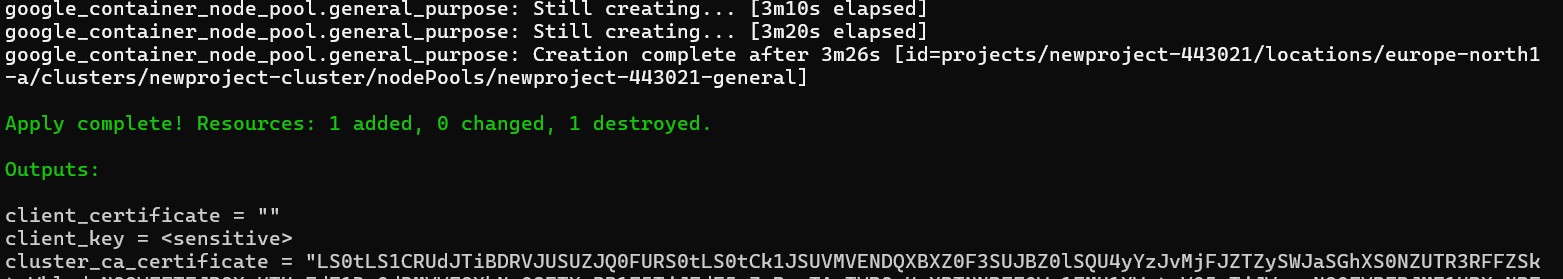
## terraform apply --var-file=variables.tfvars

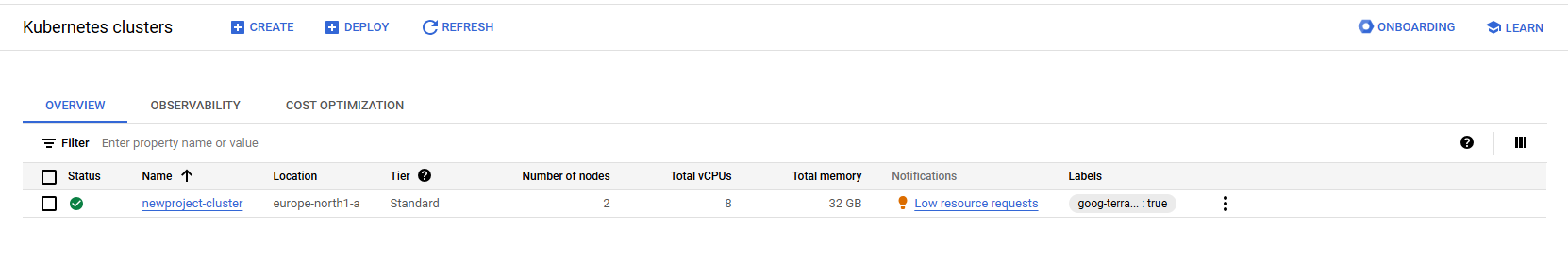


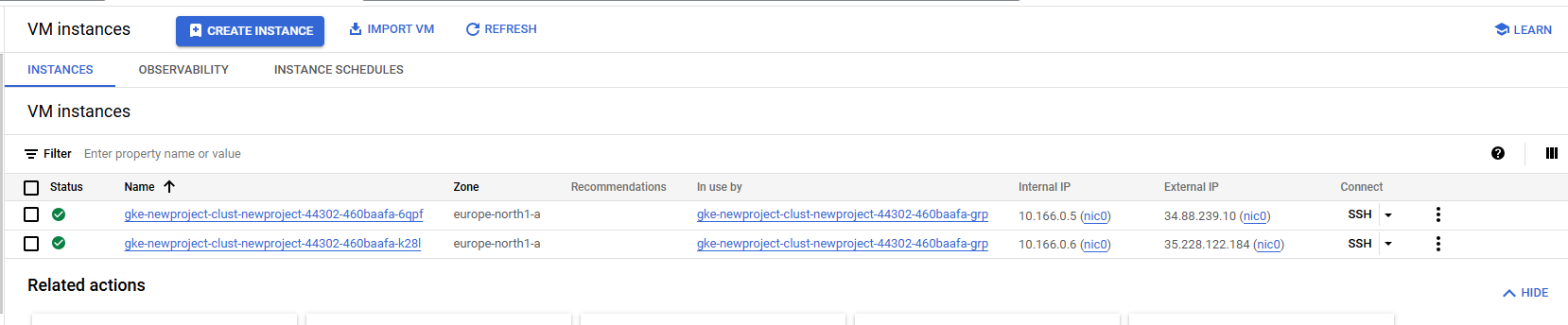
…



…

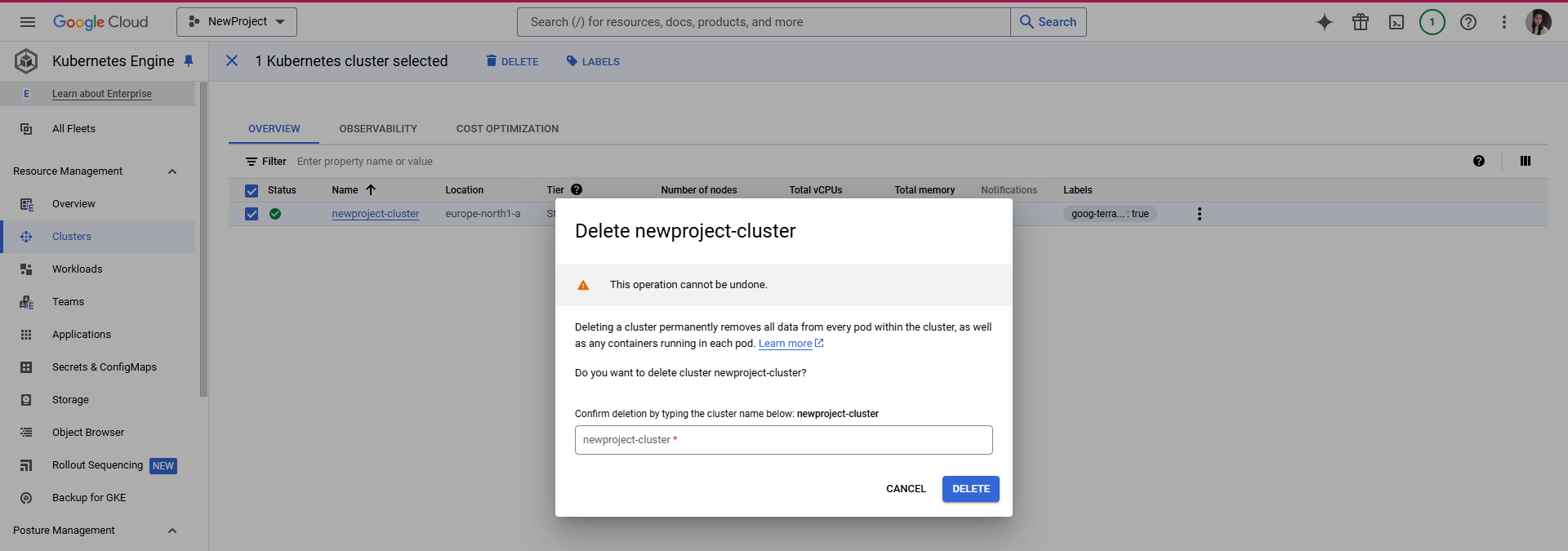


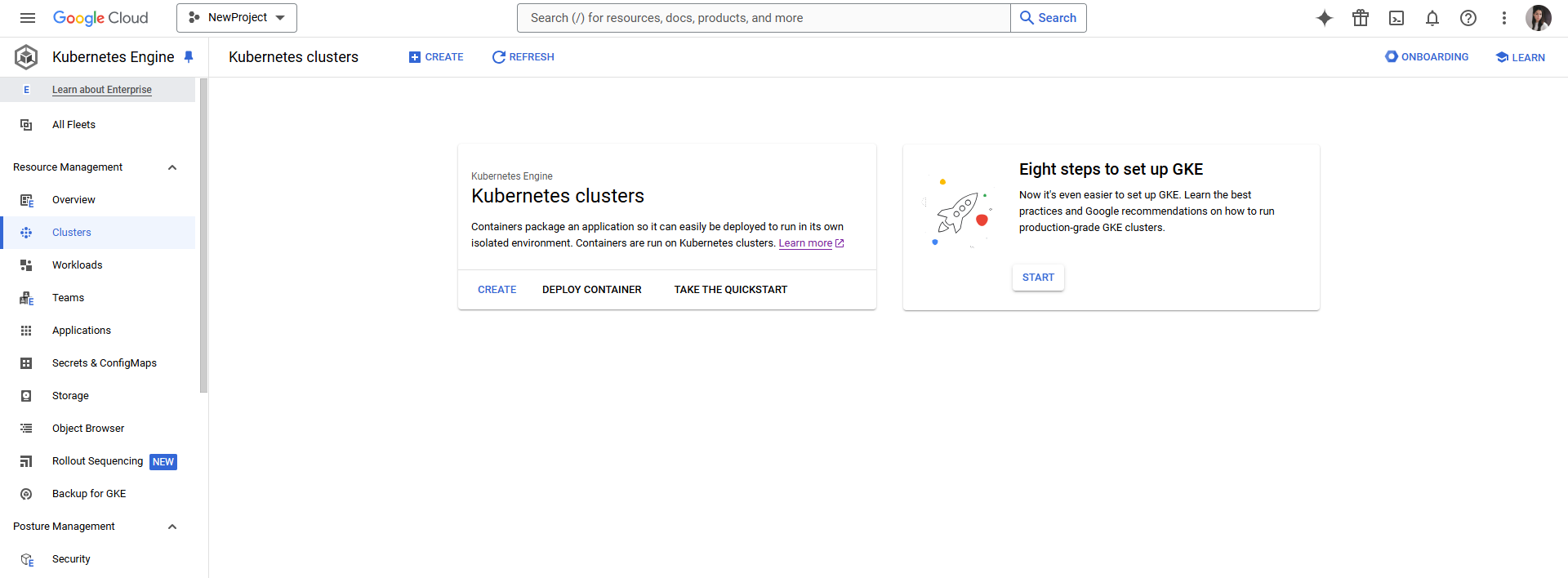
**Deja putem observa că avem un cluster cu 2 noduri**  


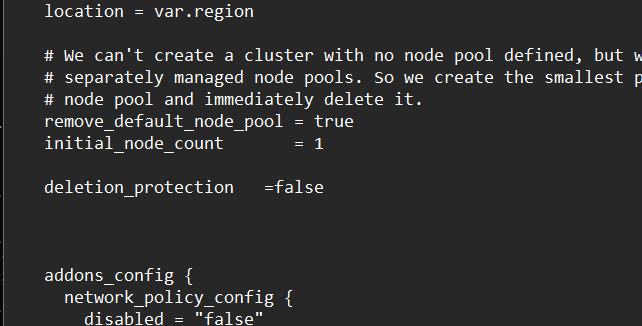
**În secțiunea Compute Engine, de asemenea observăm că avem 2 VM Instances, ce corespund acelor 2 noduri**.  


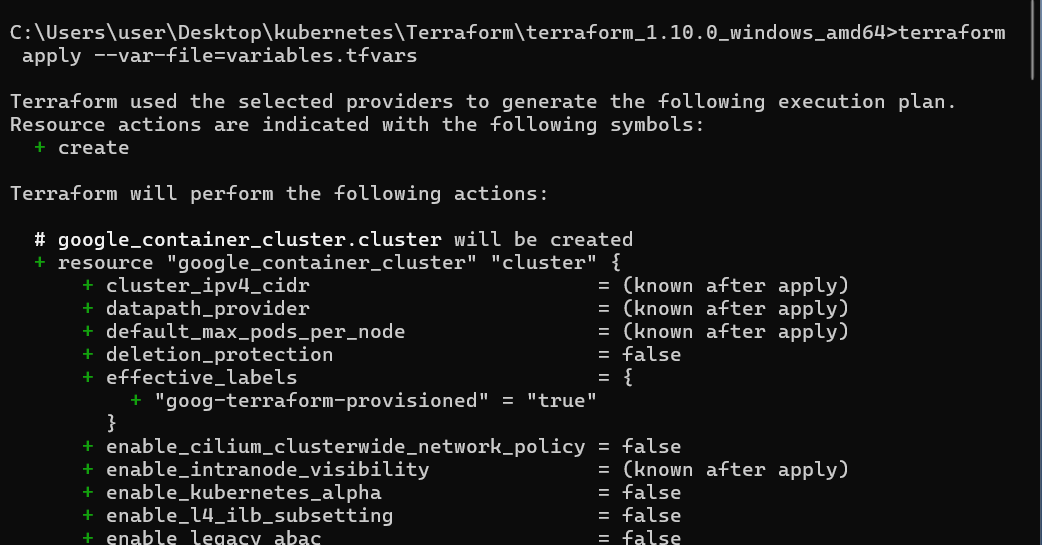
# 8. Ștergem clusterul

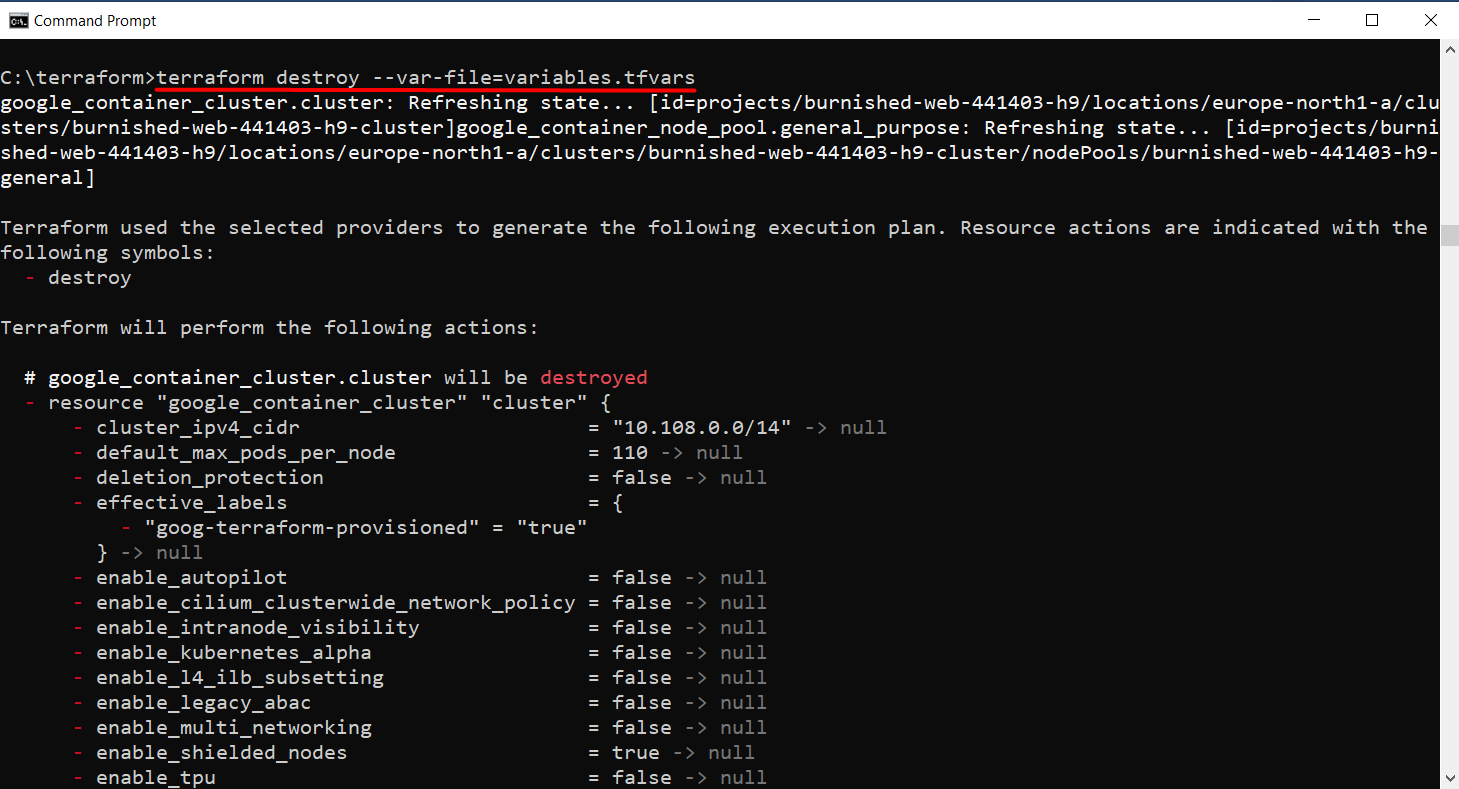
Comanda terraform destroy este folosită pentru a șterge resursele create de Terraform. Este utilă pentru a curăța infrastructura creată după ce nu mai este necesară.





Alternativ  






**Concluzie**  
În această lucrare, am folosit Terraform pentru a crea și gestiona un cluster Kubernetes pe Google Cloud Platform (GCP), automatizând procesul de infrastructură ca cod (IaC). Am început prin instalarea și configurarea Terraform, crearea unui proiect GCP, activarea Kubernetes Engine API și crearea unui cont de serviciu IAM cu permisiuni corespunzătoare. Apoi, am configurat fișierele Terraform pentru a defini variabilele și resursele necesare pentru a crea un cluster Kubernetes și node pool-uri asociate. După aplicarea modificărilor, am observat că au fost create două noduri în loc de unul, ceea ce se datorează configurației autoscalării. Concluzia principală este că o configurare corectă a parametrilor precum min\_node\_count, max\_node\_count și initial\_node\_count este esențială pentru controlul numărului de noduri create în cluster.