**UNIVERSITATEA DE STAT DIN MOLDOVA**

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**DEPARTAMENTUL DE INFORMATICĂ**

**GRĂDINARU MĂDĂLINA**

**Cloud Computing**

**Lucrare de Laborator nr.1:**

**“Inițiere în K8s”**

Profesor: Aurelia Prepeliță

**CHIȘINĂU – 2024**

Cuprins

[Instalarea kubectl, minikube 3](#_Toc177201728)

[Partea I. Comenzi de bază K8s 3](#_Toc177201729)

[Lista comenzilor de bază K8s 3](#_Toc177201730)

[Exemple de execuție a comenzilor K8s 4](#_Toc177201731)

[Partea II. Comenzi suplimentare Lab. 1 8](#_Toc177201732)

[Serviciul NodePort: 8](#_Toc177201733)

[Spații de Nume: 8](#_Toc177201734)

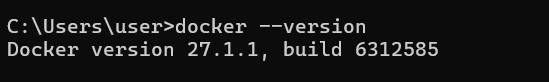
[Fișiere de configurare YAML/JSON: 8](#_Toc177201735)

[Rularea podurilor în mod interactiv: 8](#_Toc177201736)

[Concluzii 9](#_Toc177201737)

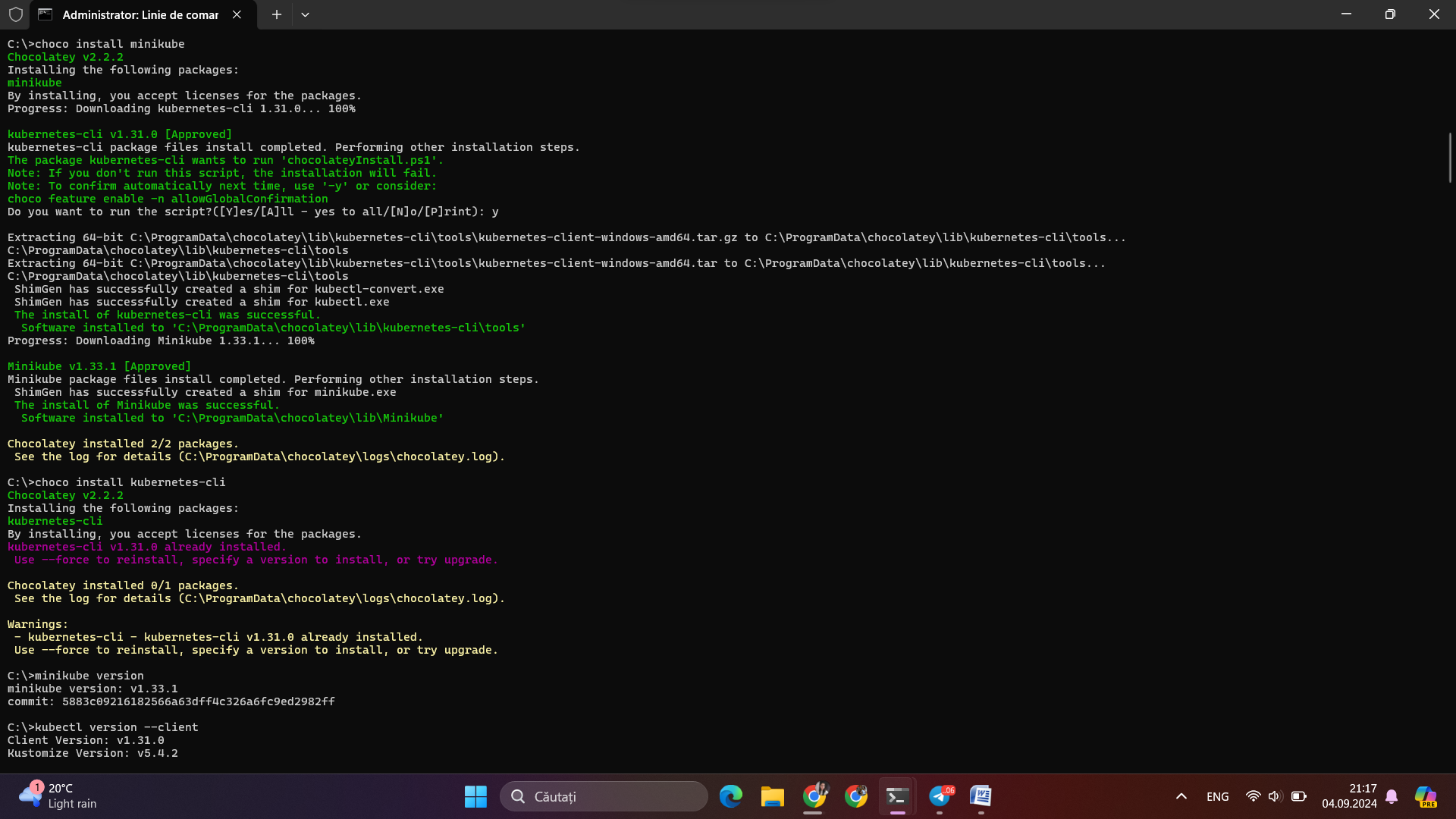
[Bibliografie 9](#_Toc177201738)

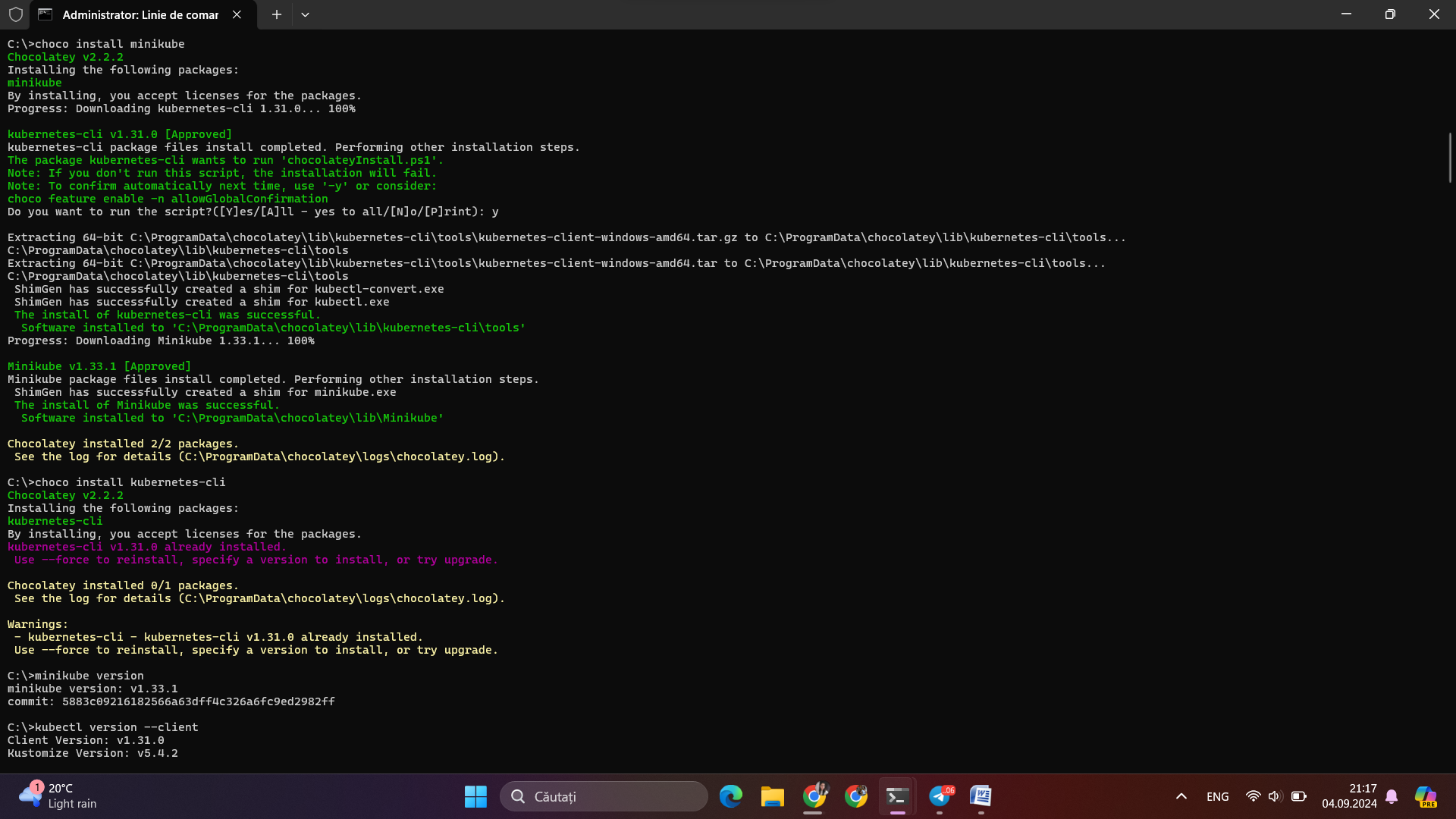
## Instalarea Docker:



## Instalarea Minikube și Kubernetes folosind Chocolatey:

**Chocolatey** este un manager de pachete pentru Windows care facilitează instalarea, actualizarea si gestionarea aplicațiilor și programelor de pe sistem.

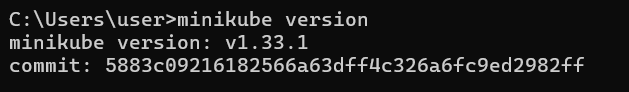




# Partea I

## Lista de comenzi utilizate în Kuberetes:

1. **minikube version**

****

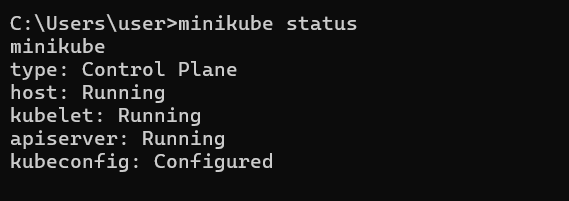
* Comanda `**minikube version**` afișează versiunea curentă a minikube instalată pe sistem. În acest caz, versiunea minikube este `**v1.33.1**`, iar **commit**-ul asociat este `**5883c09216182566a63dff4c326a6fc9ed2982ff**`, care reprezintă un anumit moment din dezvoltarea minikube, indicând exact versiunea codului sursă care a fost utilizată pentru a construi acea versiune specifică a aplicației.

1. **minikube start –** istanțiază un cluster kubernetes local

****

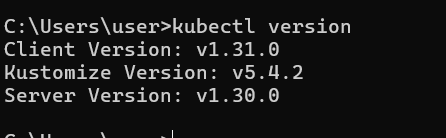
* Afișează **versiunea** curentă de minikube și **SO**.
* Notifică despre disponibilitatea unei **versiuni** mai recente.
* Oferă opțiunea de a dezactiva **notificările** de actualizare.
* Folosește driverul **Docker** bazat pe profilul existent.
* **Inițiază nodul principal** al control-plane-ului în clusterul minikube.
* Descarcă **imaginea** de bază necesară.
* Repornește **containerul** Docker existent.
* Pregătește Kubernetes pe Docker.
* Verifică componentele Kubernetes.
* Activează addon-uri `storage-provisioner` și `default-storageclass`.
* Finalizează configurarea și setează `kubectl` pentru a folosi clusterul "minikube" și spațiul de nume "default".

1. **minikube status -** afișează starea curentă a componentelor minikube.

****

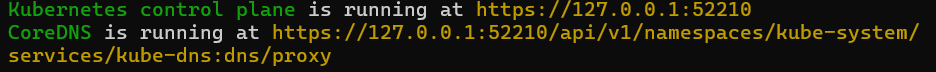
* **type: Control Plane-** Indică faptul că minikube este configurat ca nod de control (Control Plane- acesta este master node) în cluster.
* **host: Running-** Confirmă că gazda minikube este activă și funcționează.
* **kubelet: Running-** Afișează că serviciul kubelet, care se ocupă cu gestionarea containerelor pe nod, este activ.
* **apiserver: Running-** Confirmă că serverul API Kubernetes este funcțional.
* **kubeconfig: Configured-** Indică faptul că fișierul kubeconfig este configurat pentru a se conecta la clusterul minikube.

1. **kubectl version-** permite verificarea versiunii kubectl.

****

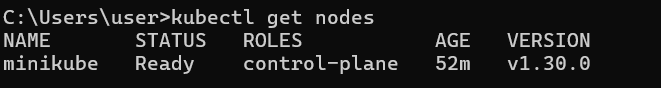
* **Client Version: v1.31.0-** Afișează versiunea clientului kubectl instalat pe sistemul local.
* **Kustomize Version: v5.4.2-** Indică versiunea Kustomize integrată în kubectl, un instrument pentru gestionarea resurselor Kubernetes.
* **Server Version: v1.30.0-** Arată versiunea serverului Kubernetes (Kubernetes API server) care rulează în clusterul minikube.

1. **kubectl cluster-info –** permite vizualizarea detaliilor despre cluster.



* **Kubernetes control plane is running at https://127.0.0.1:52210:** Afișează adresa URL la care control plane-ul Kubernetes este accesibil. În acest caz, este localizat pe `localhost` (127.0.0.1) la portul `52210`.
* **CoreDNS is running at https://127.0.0.1:**52210**/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy:** Oferă URL-ul unde serviciul CoreDNS – Control Plane este accesibil

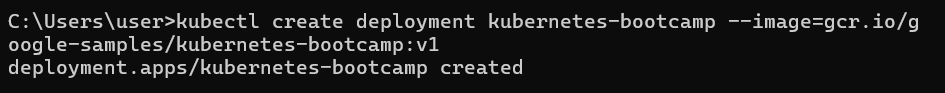
1. **kubectl get nodes -**  vizualizam nodurile

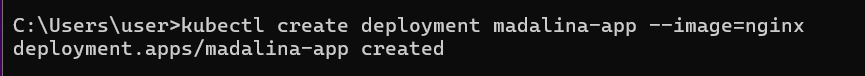
****

* **NAME:** Numele nodului. În acest caz, nodul se numește minikube.
* **STATUS:** Starea nodului. Ready indică faptul că nodul este activ și gata să preia sarcini.
* **ROLES**: Rolurile nodului. control-plane sugerează că nodul îndeplinește rolul de control plane în cluster.
* **AGE**: Vârsta nodului, adică timpul scurs de la crearea acestuia, adică 52m.
* **VERSION**: Versiunea Kubernetes care rulează pe nod, adică versiunea v1.30.0.

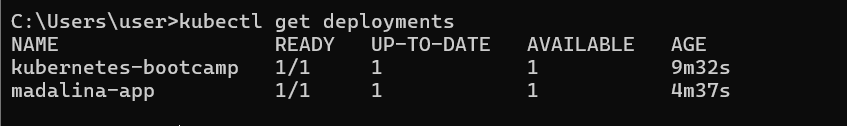
1. **kubectl create deployment {name} --image={app image location**} – creează o implementare

**Deployment** - este un obiect care gestionează și coordonează distribuția și actualizarea aplicațiilor containerizate. Acesta definește modul în care aplicațiile sunt lansate și menținute într-un cluster Kubernetes și asigură că aplicațiile sunt întotdeauna disponibile, în starea dorită.



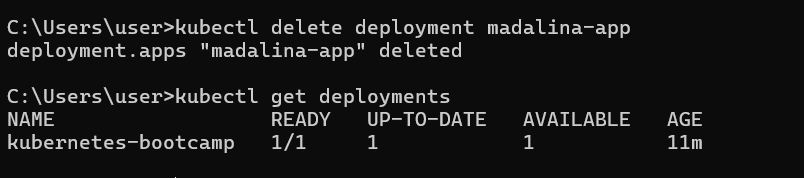


- Comanda este utilizată pentru a crea un deployment în Kubernetes,care definește cum să ruleze și să gestioneze un set de poduri ale unei aplicații, asigurând că un număr specific de instanțe rămân active și disponibile.



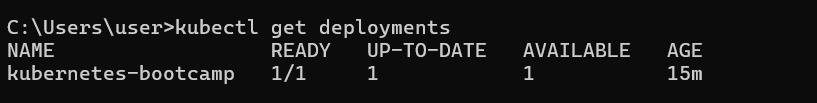
- commanda afișează implementările create anterior

1. **kubectl delete deployment {deployment name}-** șterge o implementare

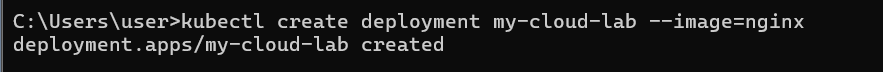
****

- Kubernetes șterge deployment-ul madalina-app împreună cu toate podurile și resursele asociate acestuia.

1. **kubectl get deployments –** listează implementările



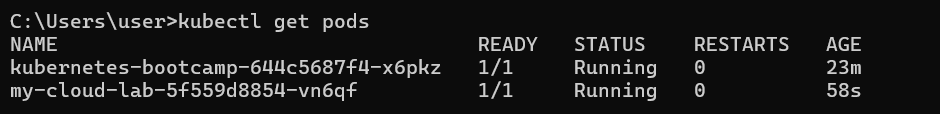
- în momentul de față a rămas o singura implementare creată anterior, așa cum una din ele a fost ștearsă.



-am mai creat o implementare pentru a avea mai multe exemple de examinat.

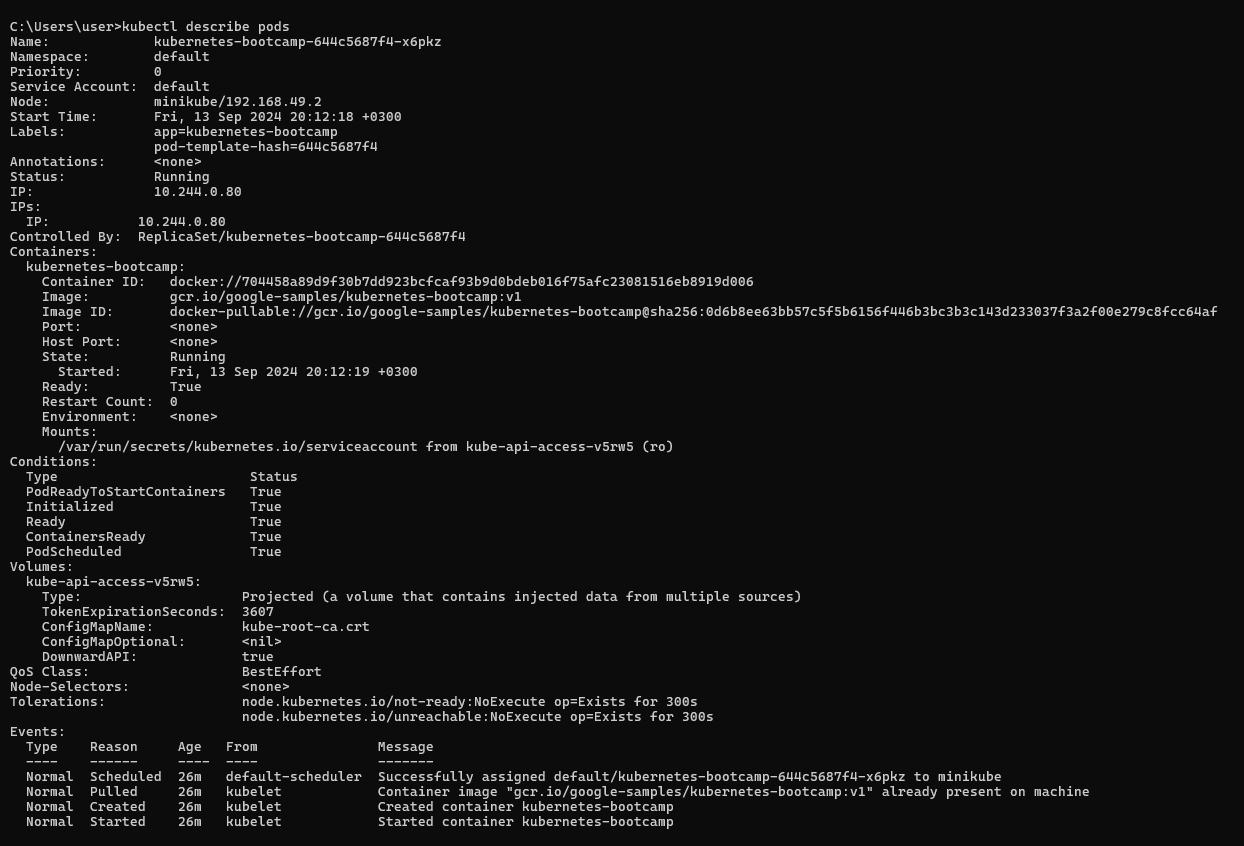
1. **kubectl get pods –** afișează pod-urile

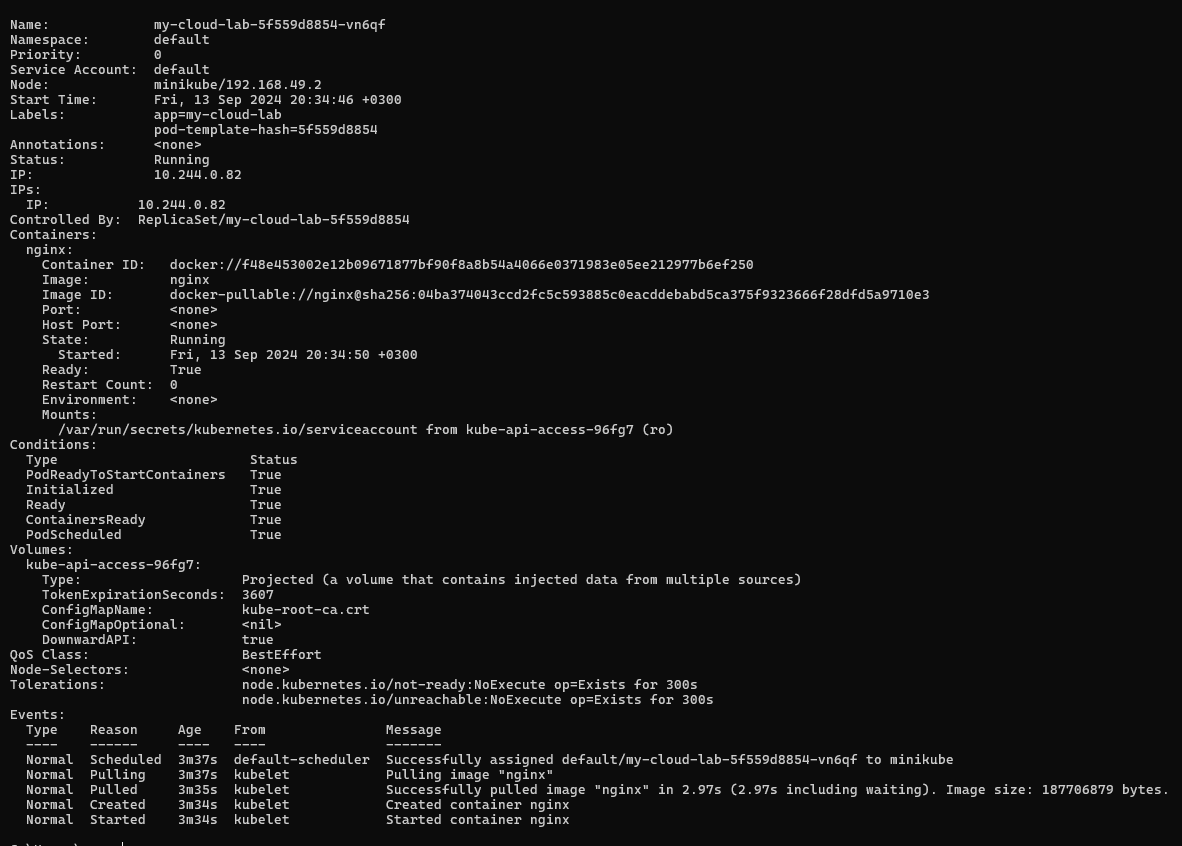
Un **Pod** în Kubernetes este cea mai mică unitate de lucru care reprezintă un set de unul sau mai multe containere ce rulează împreună pe același nod, partajând aceleași resurse de rețea și stocare. Toate containerele dintr-un Pod au același IP și pot comunica între ele direct, dar fiecare container rulează un proces independent. Pod-urile sunt efemere și se recreează automat dacă suferă o eroare sau sunt șterse, ceea ce le face componente esențiale pentru rularea și gestionarea aplicațiilor containerizate.

****

- Comanda afișează o listă a tuturor podurilor din cluster, inclusiv starea acestora - Running, numărul de reporniri - 0 și durata de funcționare de 23 min și 58 sec. Aceasta permite utilizatorilor să monitorizeze și să verifice starea curentă a podurilor pentru a se asigura că aplicațiile rulează corect și că resursele sunt disponibile.

1. **kubectl describe pods –**afișează descrierea pod-urilor

****

****

- Comanda include o descriere detaliată a podurilor, cum ar fi  
**Informații Generale:**

**Name:** Numele podului, cum ar fi kubernetes-bootcamp-644c5687f4-x6pkz și my-cloud-lab-5f559d8854-vn6qf.

**Namespace:** Spațiul de nume în care se află podul, de obicei default dacă nu ai specificat altul.

**Priority:** Prioritatea podului (0 în mod implicit).

**Service Account:** Contul de serviciu asociat podului (de obicei default).

**Node:** Numele nodului pe care rulează podul și adresa IP a acestuia.

**Stare și Configurare:**

**Start Time**: Data și ora la care podul a început să ruleze.

**Labels:** Etichetele asociate podului, care ajută la organizarea și filtrarea resurselor.

**Annotations:** Anotările atașate podului, care sunt informații adiționale.

**Status:** Starea curentă a podului (de exemplu, Running).

**Detalii Container:**

**Containers:** Lista containerelor din pod, inclusiv:

**Container ID:** ID-ul containerului, specific pentru Docker.

**Image:** Imaginea containerului utilizată (de exemplu, nginx).

**Image ID:** ID-ul imaginii de container.

**State:** Starea containerului (de exemplu, Running).

**Ready:** Indică dacă containerul este gata (True sau False).

**Restart Count:** Numărul de reporniri ale containerului.

**Condiții și Volumes:**

**Conditions:** Condiții ale podului care trebuie să fie îndeplinite (de exemplu, PodReady, ContainersReady).

**Volumes:** Detalii despre volumele montate în pod, cum ar fi volumele proiectate pentru accesul la datele Kubernetes.

**Evenimente:**

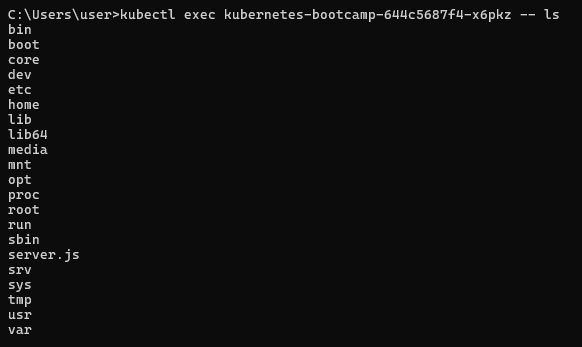
Înregistrări ale evenimentelor recente legate de pod, inclusiv programarea, tragerile de imagini, crearea și pornirea containerelor.

1. **kubectl logs {pod name} –** permite vizualizarea logărilor pe pod-ul respectiv.

**C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{AF4F7233-D913-4E60-991F-2C26618154EA}.png**

- logul arată că aplicația Kubernetes Bootcamp a început să ruleze la data 13-09-2024 și ora 17.29 pe podul respectiv.

1. **kubectl exec {pod name}** - execută comenzi direct pe container odată ce Pod-ul este pornit



- Comanda ls a fost executată în containerul din podul kubernetes-bootcamp-644c5687f4-x6pkz și a returnat o listă de directoare și fișiere găsite în directorul rădăcină (/) al containerului.

1. **kubectl exec -ti {pod name} bash** - începe o sesiune bash în containerul podului

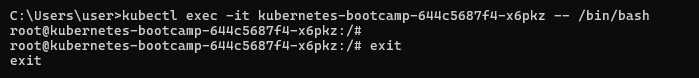
C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{9221FF36-FF0F-486D-B8B2-F9F97F364956}.png

**root:** Indică faptul că ai accesat un shell cu privilegii de utilizator root în container.

**kubernetes-bootcamp-644c5687f4-x6pkz:** Numele podului în care se află containerul.

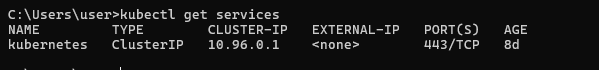
**:/# :** Promptul shell-ului indică directorul curent (/ în acest caz) și că shell-ul este pregătit să primească comenzi.

1. **exit -** închidem tipul de conexiune al containerului



1. **kubectl get services-** listează toate serviciile din clusterul Kubernetes.

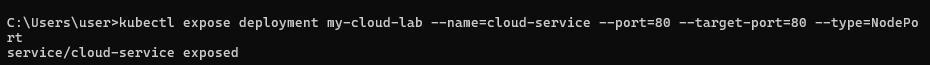
Un **Service** în Kubernetes este un mecanism care permite accesul constant și fiabil la aplicații containerizate, indiferent de schimbările sau fluctuațiile din cluster. Acesta acționează ca un punct de intrare stabil pentru un grup de Pod-uri care îndeplinesc aceeași funcție, distribuind cererile către ele în mod echilibrat



- Comanda afișează informații despre serviciile din cluster, inclusiv numele, tipul, IP-ul intern, porturile și vârsta serviciului, în acest caz arătând serviciul **kubernetes** cu tipul **ClusterIP**, IP-ul **10.96.0.1** și portul **443/TCP**.

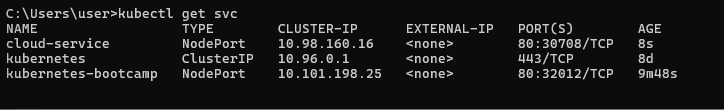
1. **kubectl expose deployment**- crează un serviciu Kubernetes care expune un deployment în cluster.

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{6B1DE908-A5BF-40C2-9D9D-3E6EFB487939}.png

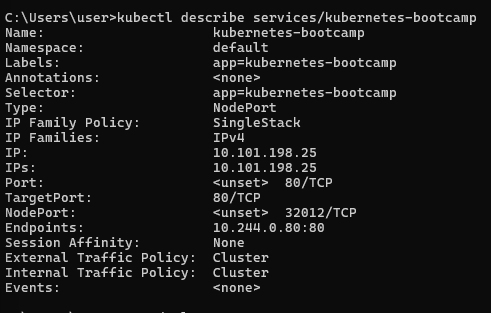


- Comanda a creat servicii de tip **NodePort(toate nodurile se deschid pe acelasi port)** care expune deploymentul **kubernetes-bootcamp și my-cloud-app** pe portul **80**, permițând accesul din afara clusterului. Aceste servicii va redirecționa traficul pe portul **80** al fiecărui nod din cluster către portul 80 al containerului din deploymentul specificat.

- afișează tipurile de resurse acceptate de cluster



1. **kubectl describe services/my-service-** ofera detalii suplimentare despre configurarea serviciului, inclusiv etichetele, selectorii și altele.

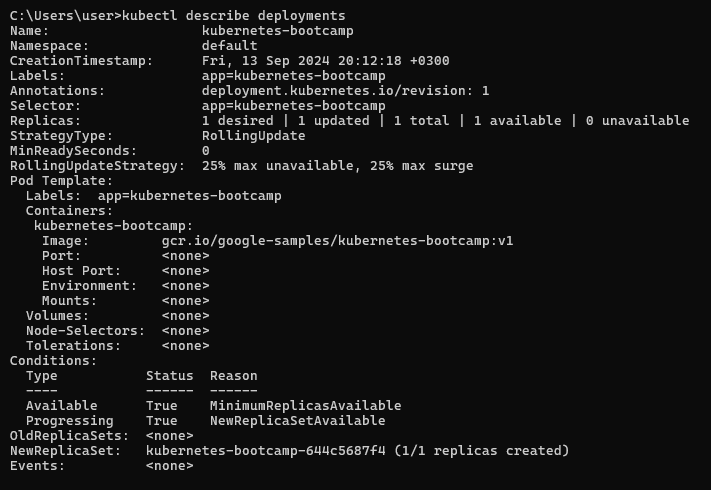


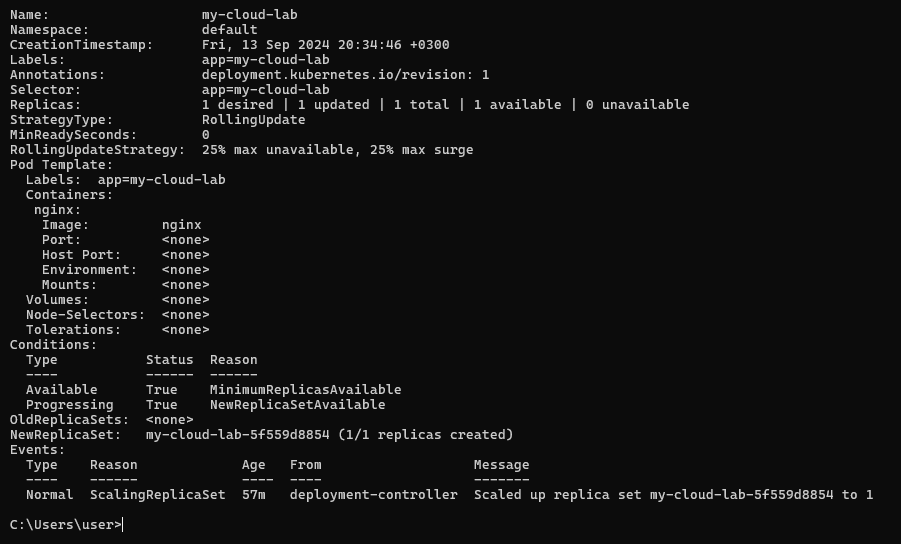
TargePort- portul containerului pe care se apeleaza aplicatia.

Endpoints –adresa ip al posului unde sta containerul nostru

- Comanda oferă detalii despre serviciul kubernetes-bootcamp. Aceasta arată că serviciul este de tip **NodePort**, ascultă pe portul **80** și redirecționează traficul către portul 80 al containerelor. Serviciul are un IP de cluster de **10.101.198.25** și este expus pe portul **32012** pe nodurile clusterului. Endpointurile servesc la adresa **10.244.0.80:80**, iar politica de trafic extern și intern este setată pe **Cluster**.

1. **kubectl describe deployments** – afișează informația detaliată despre deployment.





- **kubernetes-bootcamp**

**Nume**: kubernetes-bootcamp

**Namespace**: default

**Stare**: 1 replica dorită, 1 actualizată, 1 totală și disponibilă, fără indisponibile.

**Strategie**: RollingUpdate

**Pod Template**: Folosește imaginea gcr.io/google-samples/kubernetes-bootcamp:v1.

**ReplicaSet Nou**: kubernetes-bootcamp-644c5687f4.

**Condiții**: Disponibil și Progresând.

**Evenimente**: Nu sunt raportate evenimente recente.

**my-cloud-lab**

**Nume**: my-cloud-lab

**Namespace**: default

**Stare**: 1 replica dorită, 1 actualizată, 1 totală și disponibilă, fără indisponibile.

**Strategie**: RollingUpdate, cu 25% maxim disponibil și 25% maxim surge.

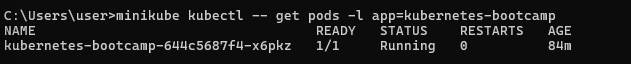
**Pod Template**: Folosește imaginea nginx.

**ReplicaSet Nou:** my-cloud-lab-5f559d8854.

**Condiții**: Disponibil și Progresând.

**Evenimente**: ReplicaSet-ul my-cloud-lab-5f559d8854 a fost scalat la 1 replica.

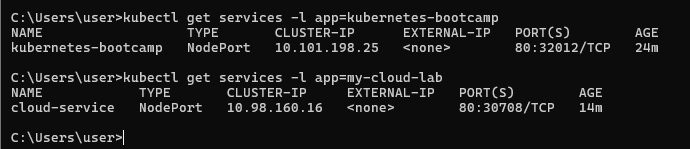
1. **kubectl get pods -l app=my-service** - listează toate pod-urile care au o etichetă specifică



- Comanda arată că podul kubernetes-bootcamp-644c5687f4-x6pkz este activ, funcționează corect fără reporniri și are o vârstă de 84 de minute.

kubectls label pod – adaugam etichete (cate sunt nevoie) aspura podului -

1. **kubectl get services -l app=my-service** - listează serviciile care au o etichetă specifică



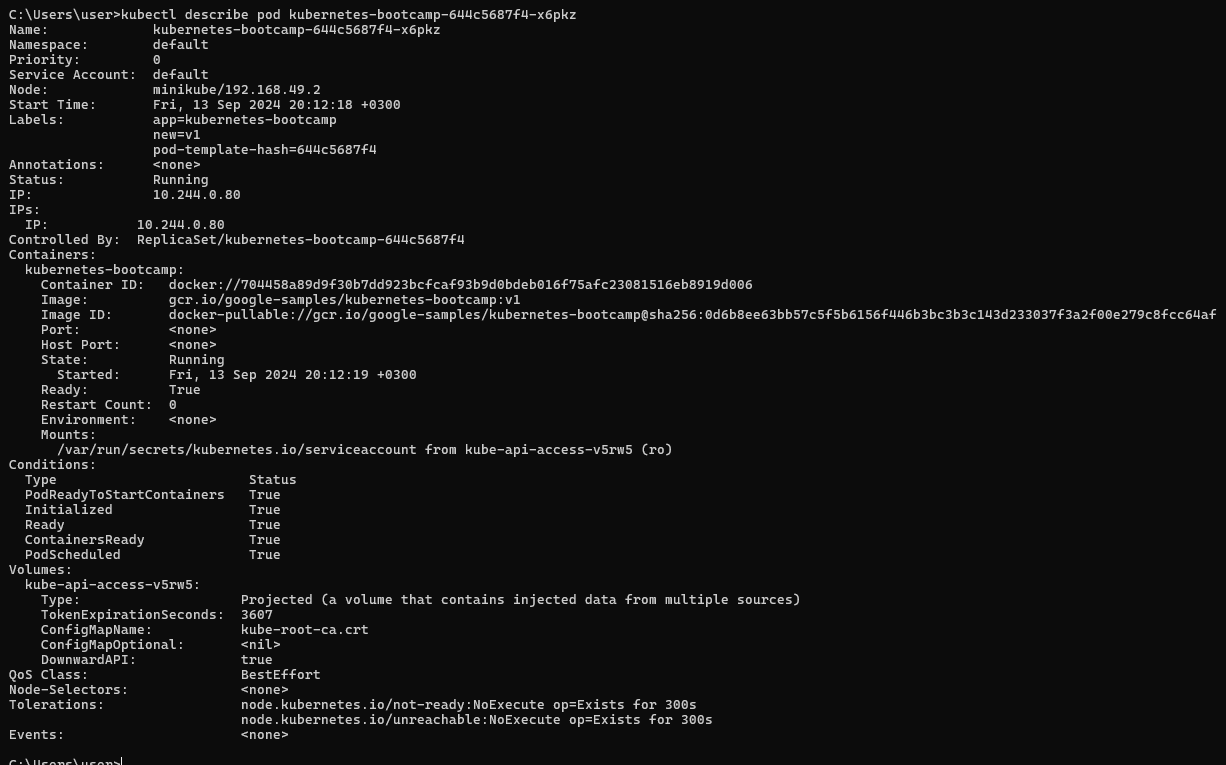
- Comenzile afișează serviciile `**kubernetes-bootcamp**` și `**cloud-service**`, cu tipul `**NodePort**` și porturile **80:32012/TCP** și **80:30708/TCP**, respectiv. Serviciile sunt disponibile **fără IP extern** și au fost create de **24** și, respectiv, **14** minute.

1. **kubectl label pod {pod\_name} new=v1** - aplicam o nouă etichetă pod-ului

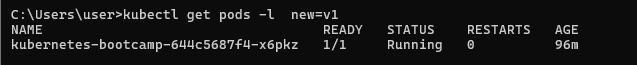
C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{0F8ABAB6-23E1-4711-ADB2-E99F1E83AAB7}.png

- Comanda a adăugat eticheta `**new**` cu valoarea `**v1**` la podul `kubernetes-bootcamp-644c5687f4-x6pkz`.

1. **kubectl describe pods {pod\_name}-** afișeaza etichetele pod-ului specificat.



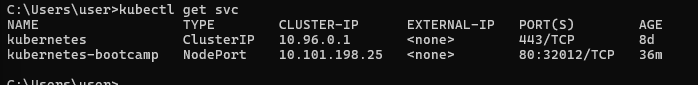
1. **kubectl get pods -l new=v1** – listează lista de poduri ce au noua etichetă



- Comanda afișează podul kubernetes-bootcamp-644c5687f4-x6pkz, care are eticheta **new=v1**, indicând că acest pod este activ și funcționează corect de 96 de minute.

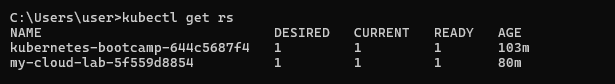
1. **kubectl delete service my-new-service** – stergerea unui serviciu

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{ABB2F04A-F664-4DAC-9CF4-4FD2FAD42BF0}.png



1. **kubectl get rs** – verifică crearea ReplicaSet de implementare.

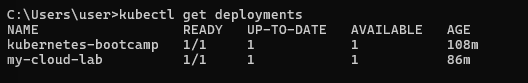
Un **ReplicaSet** în Kubernetes este un obiect care asigură faptul că un anumit număr de replici (instanțe) ale unui Pod sunt întotdeauna rulate într-un cluster. Rolul său principal este să mențină numărul dorit de Pod-uri disponibile, creând sau ștergând Pod-uri după necesitate pentru a păstra constant acest număr.



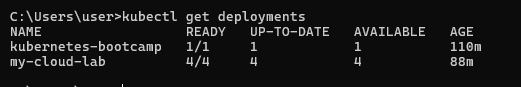
- **kubernetes-bootcamp-644c5687f4:** Este un ReplicaSet asociat cu deploymentul **kubernetes-bootcamp**, având 1 replică dorită, 1 replică actuală și 1 replică pregătită, care funcționează de 103 minute.

**my-cloud-lab-5f559d8854**: Este un ReplicaSet asociat cu deploymentul **my-cloud-lab**, având 1 replică dorită, 1 replică actuală și 1 replică pregătită, care funcționează de 80 de minute.

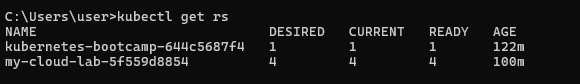
1. **kubectl scale deployment/my-cloud-lab --replicas=4** – ajustează numărul de replici ale unui Deployment în Kubernetes

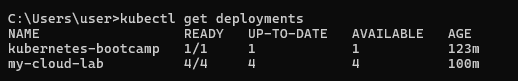


C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{10B0A069-C65C-429B-9C05-57676EEF4DF7}.png

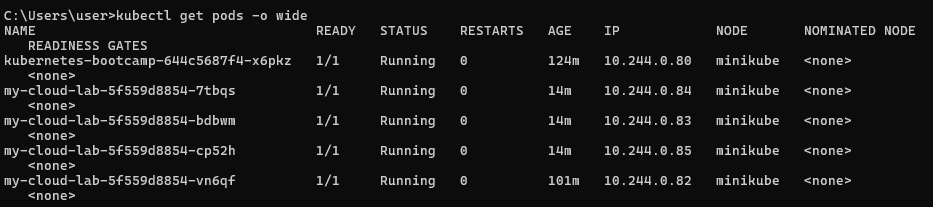


- Comanda a fost utilizată pentru a schimba numărul de replici ale deploymentului my-cloud-lab la **4**. După această operațiune, comanda kubectl get deployments confirmă că deploymentul my-cloud-lab are acum 4 replici, toate fiind disponibile și actualizate.



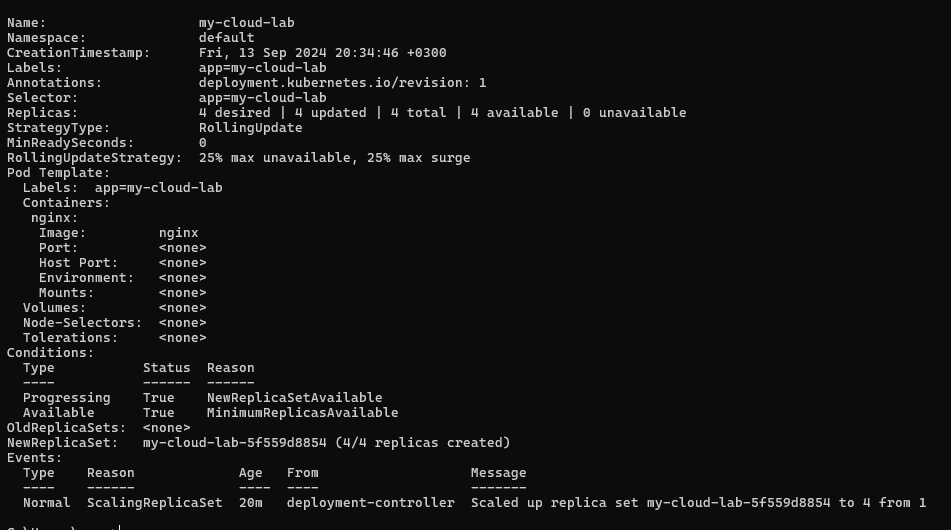


1. **kubectl get pods -o wide** – verifica numarul de pod-uri (modificat)



- În exemplu, sunt listate patru poduri pentru deploymentul my-cloud-lab, toate având statusul Running, cu IP-uri diferite și toate rulând pe nodul minikube

1. **kubectl describe deployments** - obține detalii amănunțite despre Deployments specifici, inclusiv istoricul evenimentelor și statusul actual.



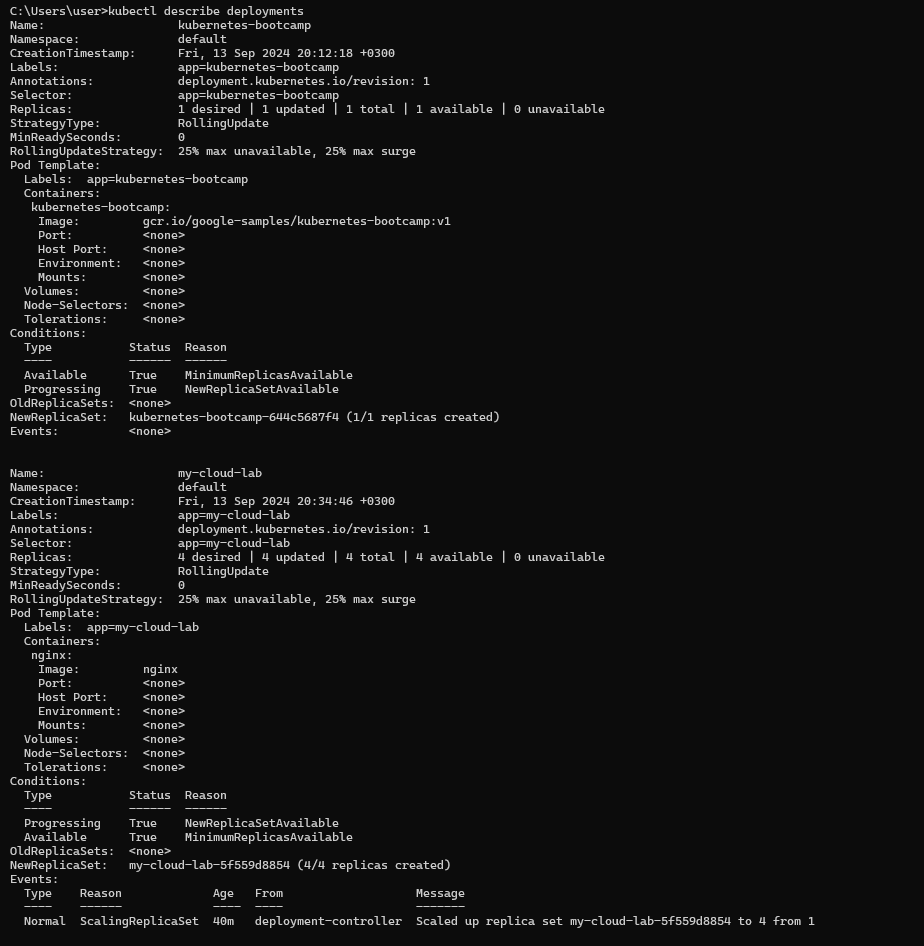
- Descrierea comenzii pentru `my-cloud-lab` arată că deploymentul a fost creat cu 4 replici dorite și toate cele 4 sunt actualizate și disponibile. Strategie de actualizare este `RollingUpdate`, iar evenimentele recente indică faptul că numărul de replici a fost scalat de la 1 la 4 cu succes.

1. **kubectl describe services** – afișează o descriere a serviciilor existente.



- Comanda oferă detalii despre două servicii. Primul, **kubernetes**, este un serviciu de tip **ClusterIP** utilizat pentru accesul intern al API-ului Kubernetes, având IP-ul 10.96.0.1 și portul 443/TCP. Al doilea, **kubernetes-bootcamp**, este un serviciu de tip **NodePort** care expune aplicația pe portul 80 și utilizează portul 32012 pe noduri, având IP-ul 10.101.198.25 și direcționând traficul către portul 80 al pod-ului corespunzător.

1. **kubectl describe deployment** – afișează versiunea actuală a imaginii a aplicației (verificam campul **Image**)



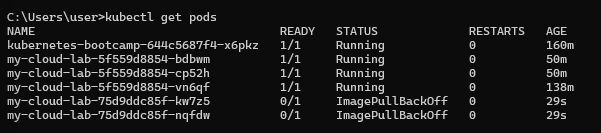
- **kubernetes-bootcamp**: Imaginea utilizată este **gcr.io/google-samples/kubernetes-bootcamp:v1**. Aceasta specifică că se folosește o imagine din Google Container Registry (GCR) cu tag-ul v1, care conține aplicația kubernetes-bootcamp.

-**my-cloud-lab**: Imaginea utilizată este **nginx**. Aceasta specifică că se folosește imaginea oficială nginx de pe Docker Hub, care este o imagine standard pentru serverul web Nginx.

1. **kubectl rollout undo { numele de implementare }-** este utilizată pentru a reveni la o versiune anterioară a unei implementări.

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{4630497B-30F9-4B02-8C88-33D67EA69644}.png

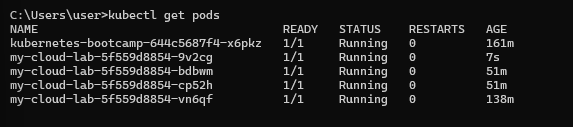
- Această comandă actualizează imaginea containerului nginx din deployment-ul my-cloud-lab la o cale invalidă (**invalid-image-path**). Ca urmare, Kubernetes începe să înlocuiască instanțele existente ale pod-urilor cu noi pod-uri folosind imaginea invalidă.



- Rezultatul arată că noile pod-uri create din cauza actualizării imaginii au statutul **ImagePullBackOff**, ceea ce indică faptul că imaginea nu poate fi descărcată (deoarece este invalidă).

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{25C01EB5-F979-446E-ACF9-CECBDAF2D7F6}.png

- Comanda efectuează un rollback la versiunea anterioară a deployment-ului my-cloud-lab. Aceasta restaurează imaginea veche și restabilește configurația inițială a deployment-ului pentru a aduce din nou pod-urile în stare funcțională.



- Rezultatul arată că pod-urile au revenit la starea **Running**, inclusiv noile pod-uri create în timpul rollback-ului. Acest lucru confirmă că rollback-ul a fost efectuat cu succes și că imaginea veche a fost restabilită

# Partea II

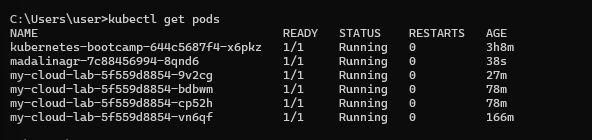
## Comenzi suplimentare pentru Lab. 1:

**Serviciul NodePort**

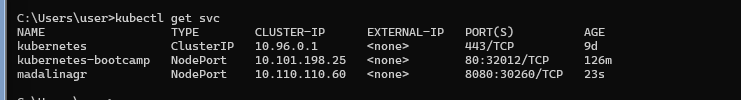
Un **Serviciu NodePort** în Kubernetes este un tip special de Service care expune aplicația rulată într-un cluster Kubernetes la adresele IP externe, permițând accesul din afara clusterului la aplicație.

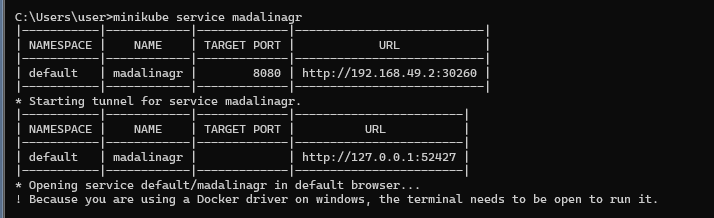
* Creați un serviciu NodePort

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{3CB0C32E-698E-4DE4-B6FB-EE7441BC5692}.png

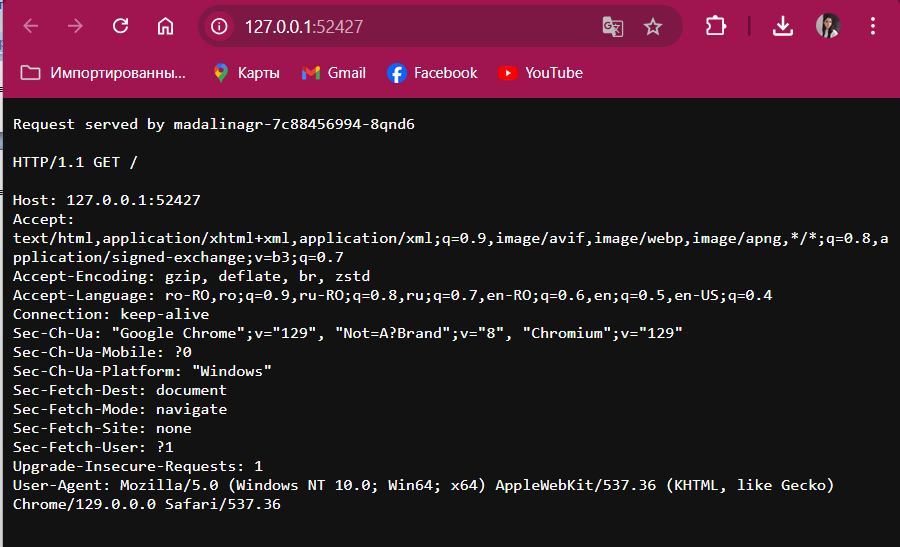


C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{AD2D246B-C4D1-4D53-8CE5-C7EED6B5CF7C}.png





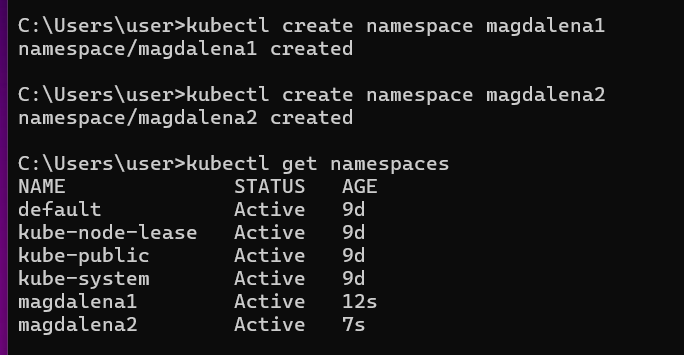
* Apelați aplicația utilizând serviciul NodePort



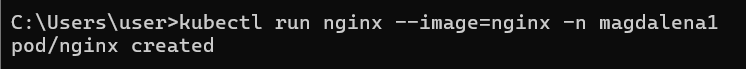
**Spațiu de Nume**

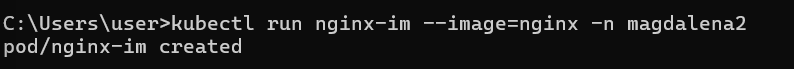
- acestea sunt utilizate pentru a **organiza** și **izola** resursele într-un cluster, permițând gestionarea separată a acestora pentru diferite echipe, aplicații sau medii. Ele asigură că resursele nu se amestecă între ele.

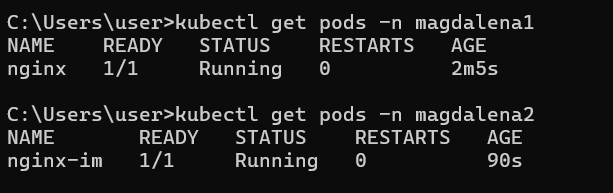
Creați 2 spații de nume.



* Creați 2 poduri în 2 spații de nume diferite.







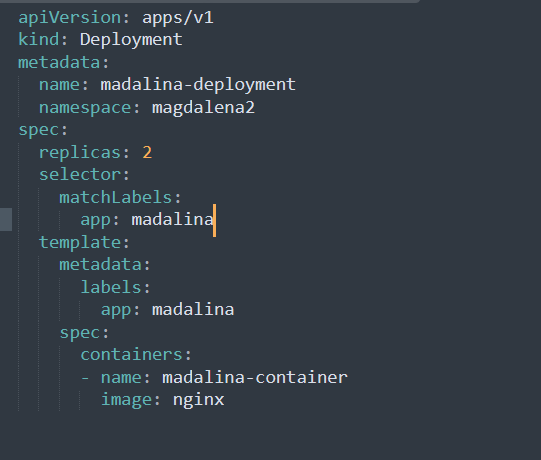
**Fișiere YAML**

* Creați un pod/deployment/serviciu în mod declarativ folosind un fișier de configurare.

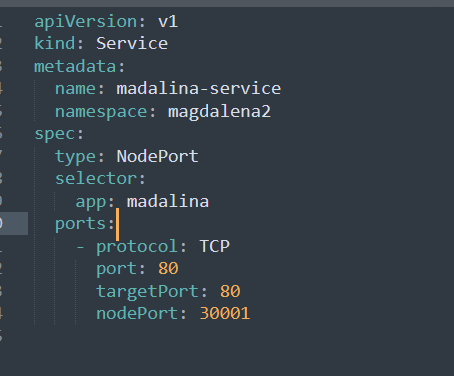
**Crearea unui pod**

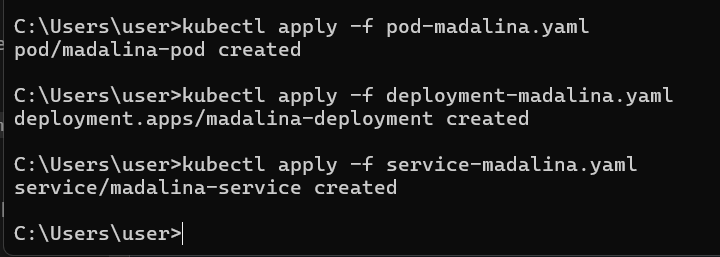


**Crearea unui Deployment:**



**Crearea unui Serviciu:**

****

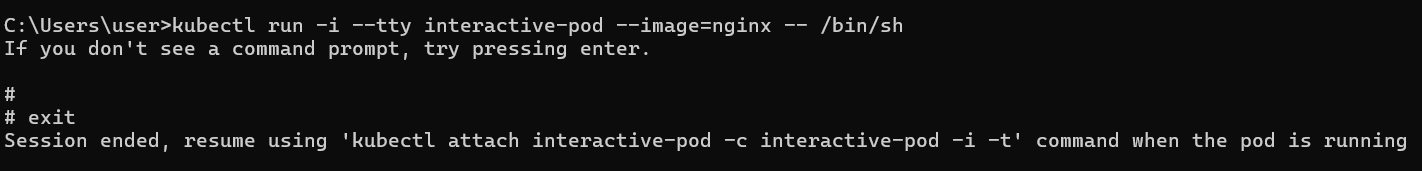
****

**Rularea unui pod în mod interactiv**

* Rularea în mod interactiv a unui pod (2 comenzi)

Comanda **kubectl run**:

-Aceasta creează și rulează un pod simplu folosind o imagine Docker specificată. Este rapidă pentru testare și rularea unui singur container.



**kubectl run:** Creează și rulează un pod.

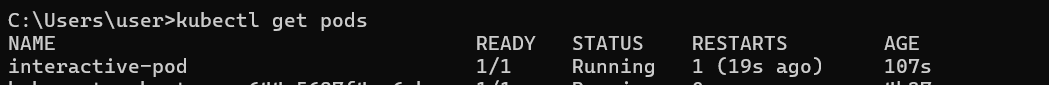
**-i**: Permite interactivitatea.

**--tty:** Alocă un terminal virtual.

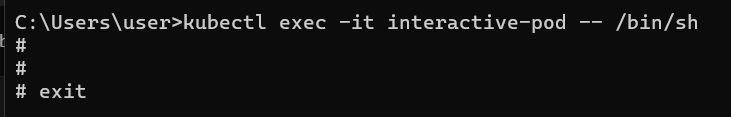
**interactive-pod:** Numele podului.

**--image=nginx:** Folosește imaginea nginx.

**-- /bin/bash:** Rulăm un shell Bash în container.



* Comanda **kubectl exec**:

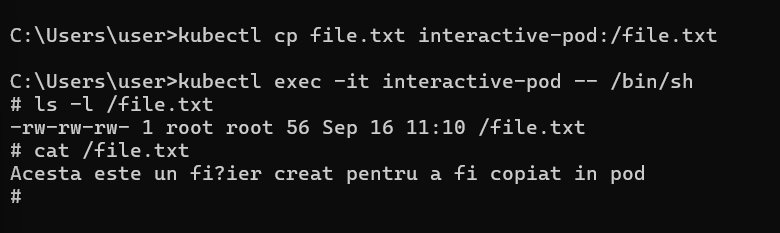


1. - Aceasta va deschide un terminal interactiv într-un pod existent numit interactive-pod.

**Copierea fișierelor pe pod**

C:\Users\user\AppData\Local\Packages\MicrosoftWindows.Client.CBS_cw5n1h2txyewy\TempState\ScreenClip\{9EB03ACA-F741-4739-B326-1FAEBB67F979}.png

- am creat un fi;ier file.txt in directorul current, cu următorul conținut “Aceste este un fisier creat pentru a fi copiat in pod”.



- utilizand metoda interactiva **exec** am rulat podul nostru **interactive-pod** cu o linie de comanda shell si cu **ls** (listare) am afisat fisierele pe care le contine pod-ul resectiv. Cu **cat**, am afisat de asemenea si continutul fisierului.