|  |
| --- |
| **LAPORAN PRAKTIKUM VIRTUALISASI KOMPUTER**  **PEMBUATAN DAN SETUP POD MENGGUNAKAN DEPLOYMENT KUBERNETES** |
|  |
| **Agus Pranata Marpaung**  **13323033**  **DIII TEKNOLOGI KOMPUTER** |
| **INSTITUT TEKNOLOGI DEL**  **FAKULTAS VOKASI** |

**Judul Praktikum**

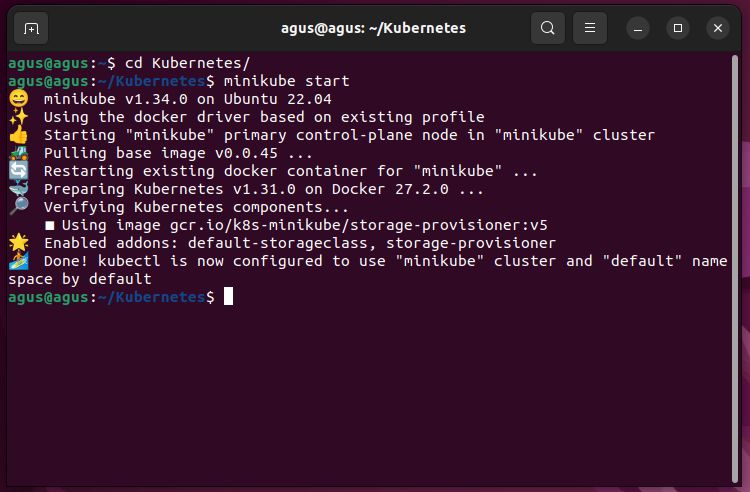
|  |  |  |
| --- | --- | --- |
| **Minggu/Sesi** | : | XIV/2 |
| **Kode Mata Kuliah** | : | 4332103 |
| **Nama Mata Kuliah** | : | VIRTUALISASI KOMPUTER |
| **Setoran** | : | Jawaban dalam bentuk *softcopy* |
| **Batas Waktu Setoran** | : | *Kamis, 28 November 2024 jam 21:30* |
| **Tujuan** | : | 1. Mahasiswa mampu membuat dan melakukan setup pada pod menggunakan Deployment Kubernetes. |

**Petunjuk**

**Praktikum**

1. Pada praktikum sebelumnya, kita sudah mengenal apa itu pod, dan sekarang kita akan mencoba untuk membuat pod secara manual. Namun terlebih dahulu kita bisa menjalankan minikube dikarenakan defaultnya minikube tidak akan berjalan saat boot.

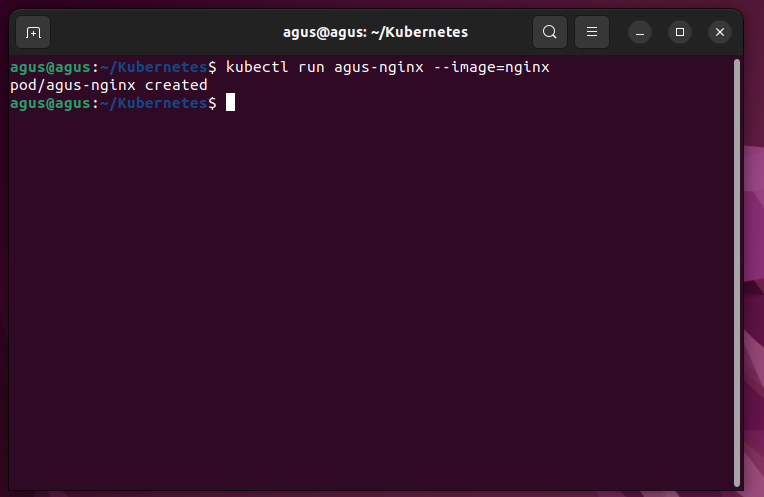
minikube start



1. Agar Kita bisa menjalankan minikube secara otomatis saat boot Kita bisa membuat script pada service minikube tersebut. Kita bisa melakukan eksplor melalui internet.

1. Setelah itu kita bisa membuat pod secara manual dengan menjalankan *command* berikut.

kubectl run nginx --image=nginx



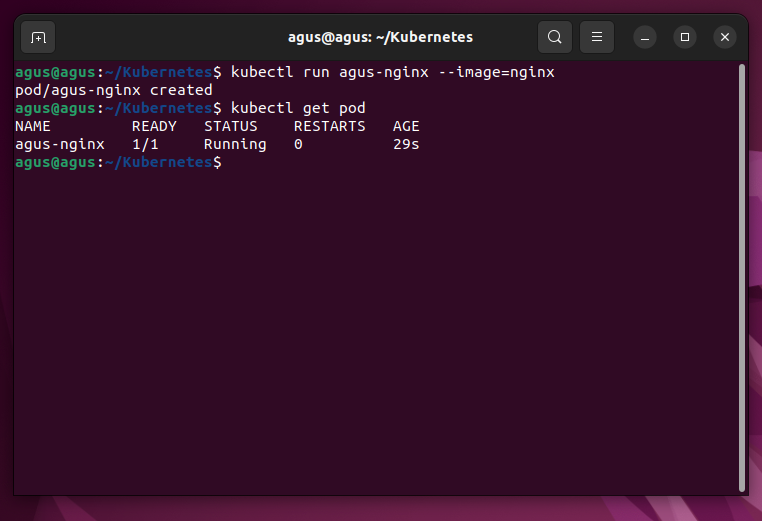
**Keterangan:**

**virkom-nginx** : nama pod yang akan dibuat

**--image=nginx** : image nginx yang digunakan untuk membuat pod

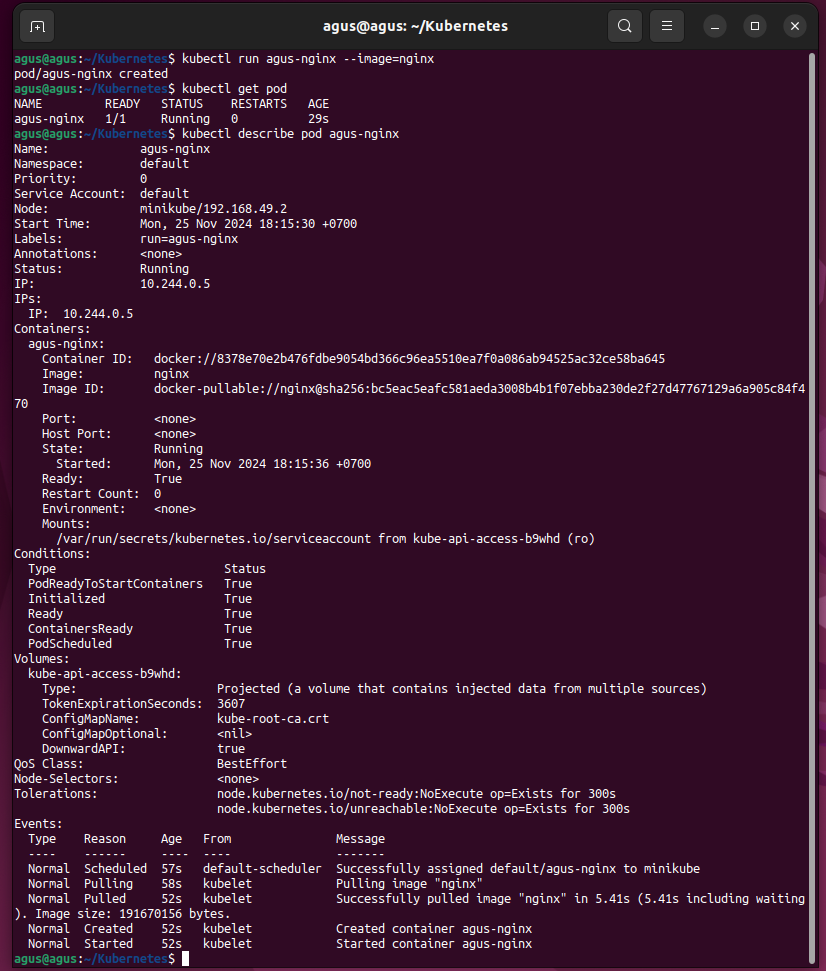
1. Lalu Kita bisa melihat status pod itu sendiri dengan menjalankan *command* berikut.

kubectl get pod

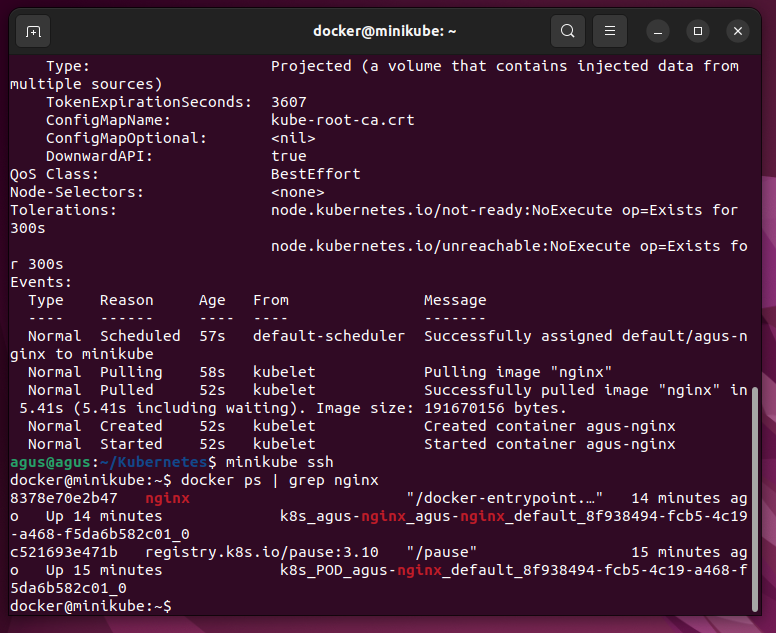


1. Kita juga bisa melihat informasi lengkap mengenai pod yang Kita buat dengan menjalankan *command* berikut

kubectl describe pod virkom-nginx



1. Lalu, Kita sekarang coba mengecek apakah terdapat container nginx pada node yang sudah Kita buat sebelumnya. Namun Kita terlebih dahulu login kedalam node yang sudah Kita buat sebelumnya dan menjalankan *command* berikut.



Maka akan terdapat 2 container, yaitu container nginx dan juga container pause yang berguna untuk memanajemen pod dan juga memberikan identitas unik pada pod.

1. Kemudian Kita coba untuk masuk kedalam container nginx dengan menjalankan *command* berikut.



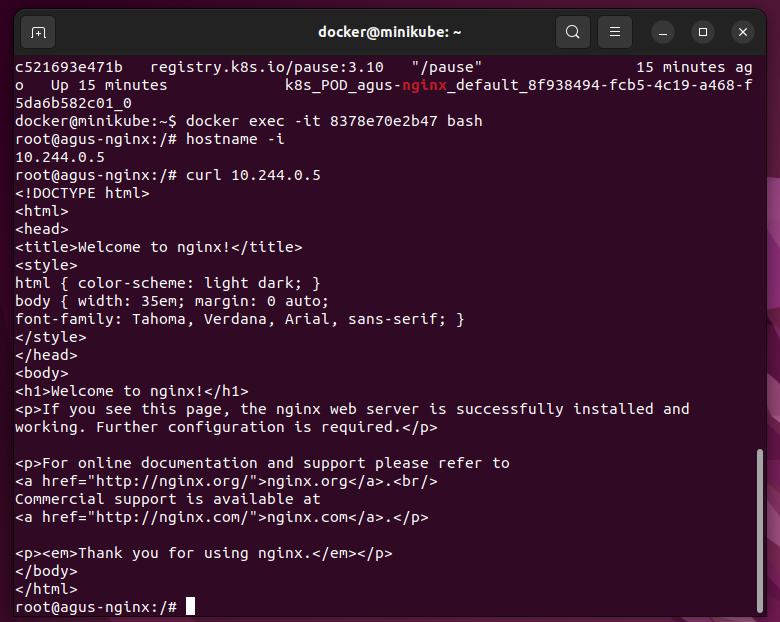
1. Setelah masuk, Kita akan mencoba untuk mengakses web server nginx. Untuk itu Kita perlu mengetahui apa IP Address dari container tersebut dengan menjalankan *command* berikut.

hostname -i



1. Lalu Kita sekarang coba mengakses web server dengan menjalankan *command* berikut.

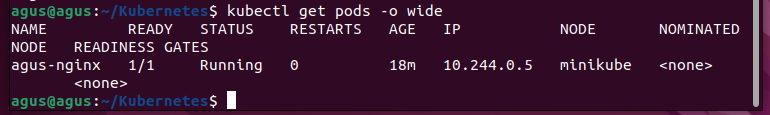
curl 10.244.0.5



Dari gambar diatas, dapat disimpulkan bahwa web server berjalan dengan baik.

1. Untuk mendapatkan informasi lebih detail mengenai pod yang sedang berjalan di Kubernetes cluster, Kita bisa menjalankan *command* berikut.

kubectl get pods -o wide



1. Setelah Kita sudah selesai membuat single pod, Kita bisa menghapus pod tersebut dengan menjalankan *command* berikut.

kubectl delete pod virkom-nginx

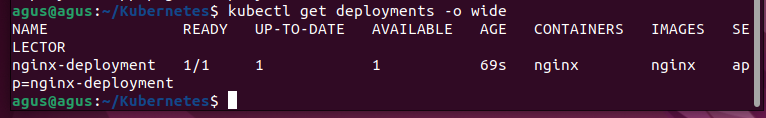


1. Setelah terhapus, sekarang Kita akan diminta untuk membuat beberapa pod yang dimana Kita bisa melakukan custom konfigurasi, seperti menambahkan jumlah pod dan juga mengurangi jumlah pada pod tersebut. Kita bisa membuat sebuah Deployment untuk melakukan hal tersebut. Untuk membuat Deployment, Kita bisa menjalankan *command* berikut.

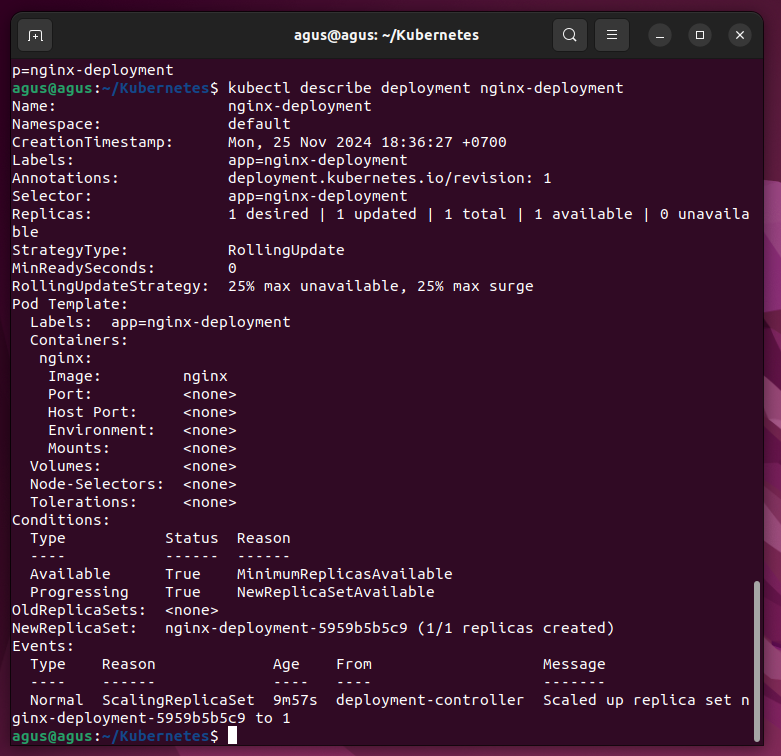
kubectl create deployment nginx-deployment --image=nginx



1. Lalu Kita bisa melihat status dari deployment yang sedang berjalan lebih rinci dengan menjalankan *command* berikut.

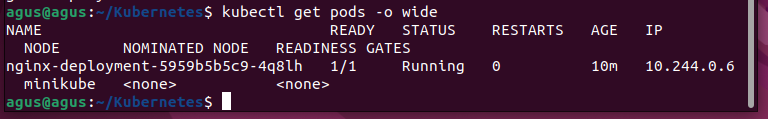


1. Kita juga bisa melihat rincian informasi mengenai deployment tertentu dengan menjalankan *command* berikut.



Pada gambar tersebut Kita melihat bahwa hanya ada 1 pod yang terbentuk.

1. Untuk memastikan apakah hanya 1 pod yang terbentuk, Kita bisa menjalankan *command* berikut.



1. Dikarenakan masih hanya 1 pod yang terbentuk, Kita perlu menambahkan jumlah replica dari deployment dengan menjalankan *command* berikut.

kubectl scale deployment nginx-deployment --replicas=3



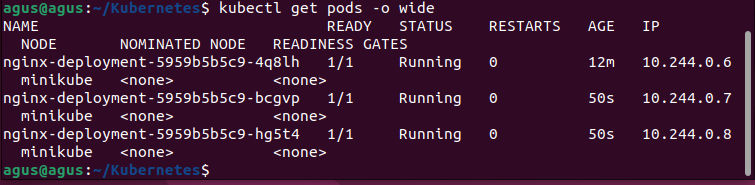
**Keterangan:**

**scale deployment** :mengubah jumlah replica dari suatu deployment

**nginx-deployment** :nama deployment yang ingin diubah jumlah replica dari deployment tersebut

**--replicas=3** : menentukan jumlah replica yang diinginkan, yaitu 3

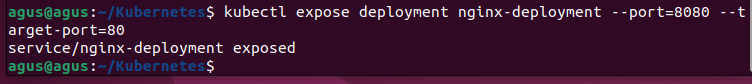
1. Lalu Kita verifikasi apakah sudah bertambah.



1. Setelah itu, Kita diminta untuk membuat service untuk terhubung ke deployment tertentu menggunakan IP Address tertentu seperti membuat cluster IP. Cluster IP ini akan berperan untuk menghubungkan deployment tertentu dan mendistribusikan load ke berbagai pod yang berbeda di dalam cluster. Untuk itu, Kita bisa menjalankan *command* berikut.

**Note: Cluster IP tidak berfungsi diluar dari Kubernetes cluster.**

kubectl expose deployment nginx-deployment --port=8080 --target-port=80



**Keterangan:**

**expose deployment** :mengekspos deployment ke luar cluster

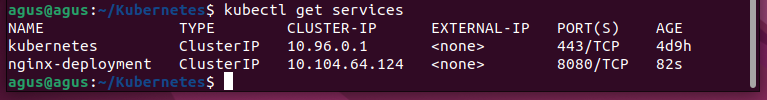
**nginx-deployment** :nama deployment yang ingin diekspos

**--port=8080** :port yang akan menerima request dari luar cluster, yaitu 8080

**--target-port=80** :port yang berada di dalam deployment untuk menerima request, yaitu 80

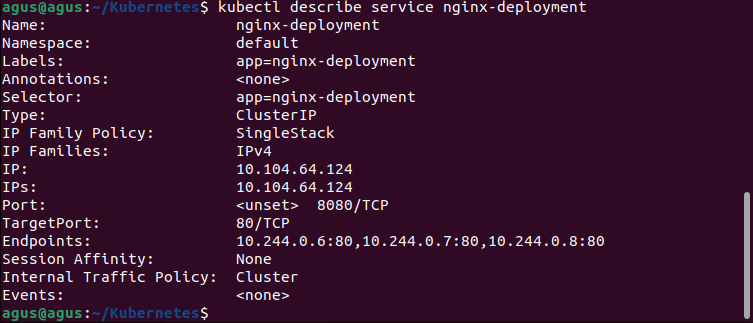
1. Lalu Kita lihat list untuk servicenya.

kubectl get services



1. Kita bisa melihat informasi detail mengenai service dari deployment yang sudah Kita buat sebelumnya.

kubectl describe service nginx-deployment



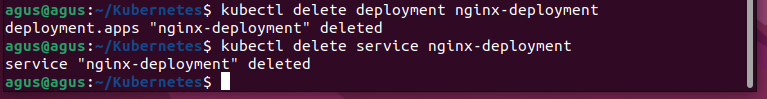
1. Kemudian Kita akan mencoba untuk terhubung ke deployment yang sudah Kita buat menggunakan cluster IP yang tersedia. Namun Kita terlebih dahulu masuk kedalam node yang sudah Kita buat sebelumnya dan menjalankan *command* berikut.

curl 10.104.64.124:8080



Pada gambar diatas, terdapat keluaran yang disediakan oleh 1 dari 3 pod yang tersedia.

1. Lalu Kita bisa menghapus deployment dan service yang sudah Kita buat sebelumnya dengan menjalankan *command* berikut.



1. Selesai!