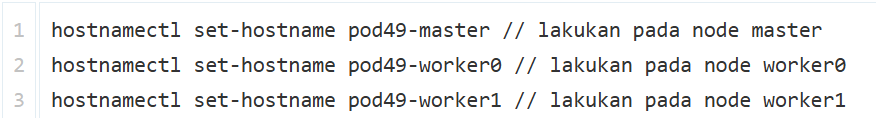
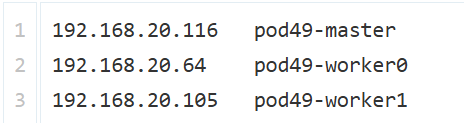
**Instalasi dan Konfigurasi buat tutorial install 1 kubernate dengan 2 docker pada sistim operasi linux**

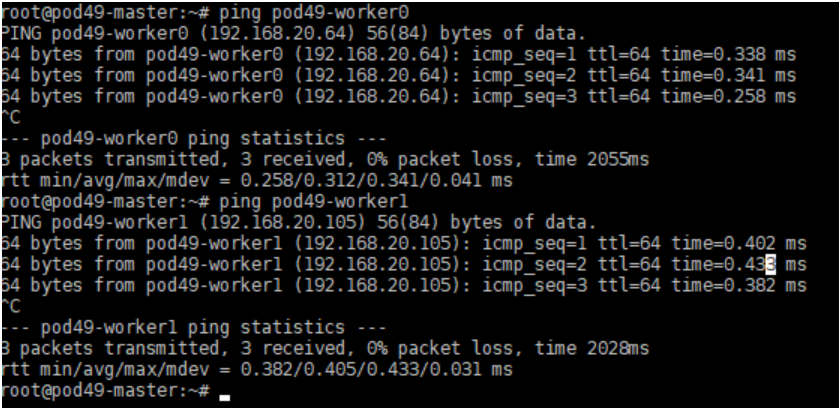
1. Set hostname dan tambahkan baris pada /etc/hosts pada masing-masing server.



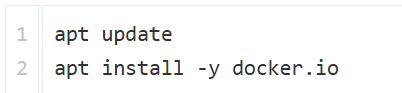
Masukkan baris berikut pada masing-masing server.



Verifikasi dengan test ping menggunakan hostname.



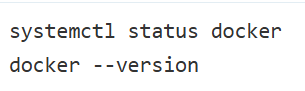
1. Install docker (Container Runtime) pada masing-masing node/server. Login ke masing-masing node dan coba update paket beserta install docker.



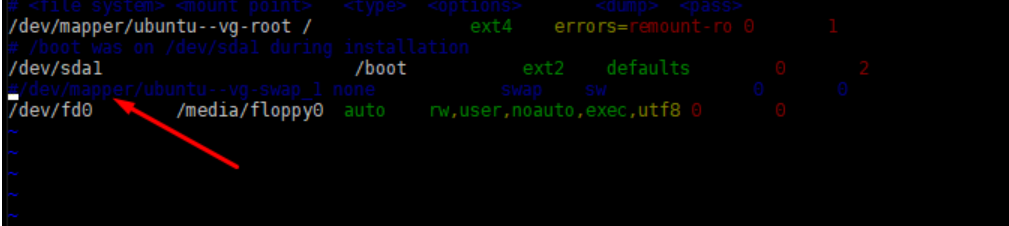
Aktifkan service docker dengan perintah:



Pastikan status service docker aktif, Anda juga bisa menggunakan perintah berikut untuk cek versi dockernya.



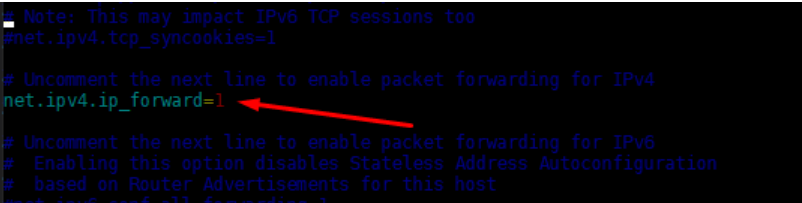
1. Non-aktifkan swap dan aktifkan IP forwarding masing-masing node/server. Untuk menon-aktifkan swap, edit file /etc/fstab dan komen (beri tanda #) pada baris yang mencakup partisi dan file swap.



Simpan dan keluar, lalu jalankan perintah berikut untuk menon-aktifkan swap secara cepat.

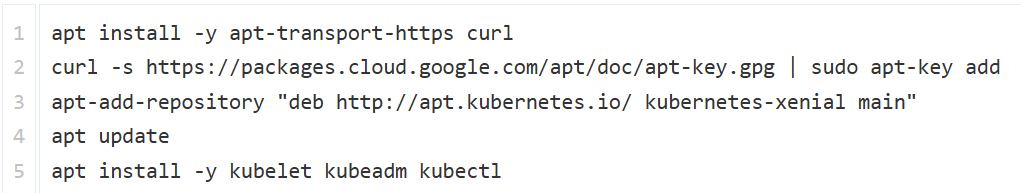


Selanjutnya, aktifkan IP Forwarding secara permanen. Edit file /etc/sysctl.conf, lalu cari baris net.ipv4.ip\_forward=1 dan hilangkan tanda pagar. Setelah itu, simpan dan jalankan perintah berikut:





1. Install Kubectl, kubelet dan kubeadm pada masing-masing node/server



1. Inisialiassi Kubernetest Cluster menggunakan Kubeadm (dari master node)

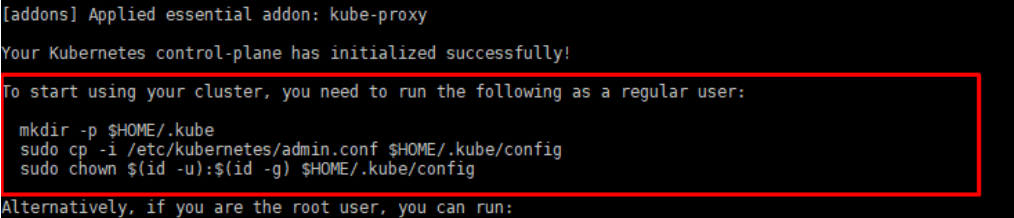
Login ke node master dan jalankan perintah berikut:



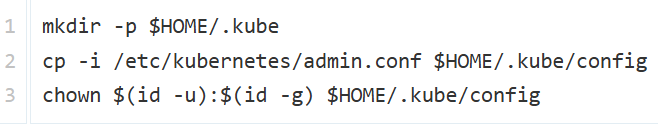
Karena saya menggunakan VCPu 1 core, maka akan muncul pesan error penggunaan vCPU kurang mencukupi (minimal 2 core). Hal tersebut bisa diantisipasi dengan menambahkan flag --ignore-preflight-errors=NumCPU.



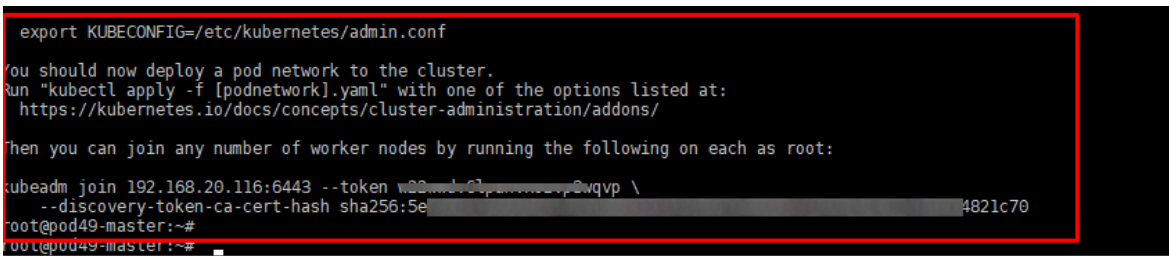
Jika berhasil maka akan muncul informasi perintah mulai join cluster seperti berikut:

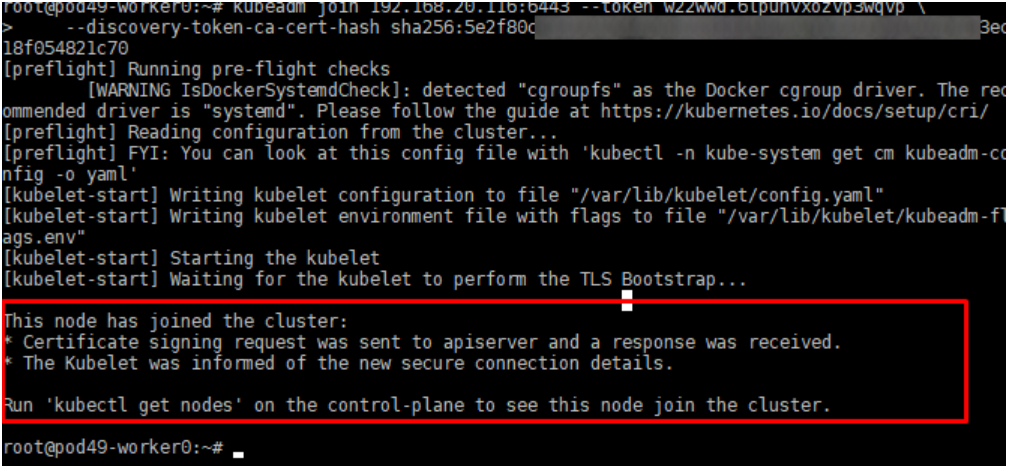


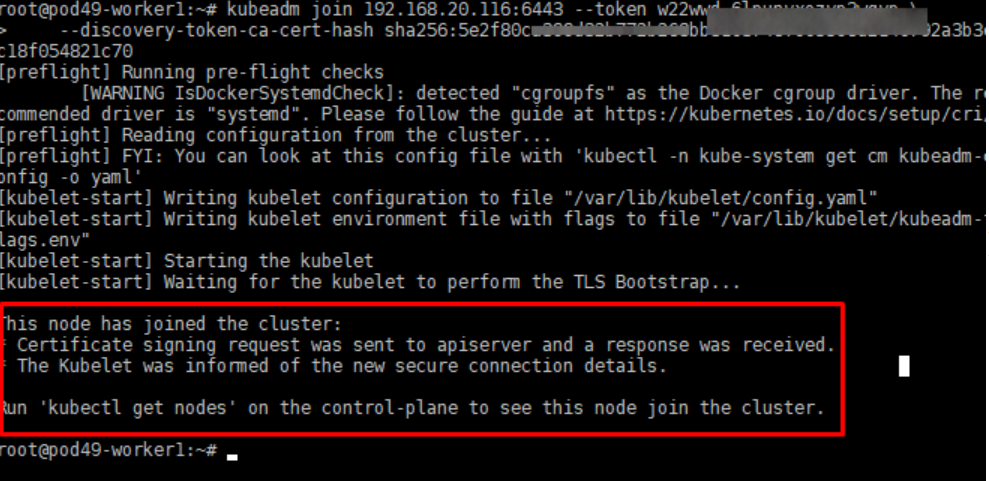
Jalankan perintah berikut pada node master.



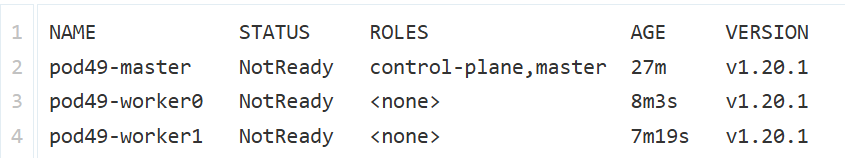
Setelah itu, jalankan perintah berikut ini pada masing-masing node worker untuk join ke node master.







Kemudian, dari sisi node master cek status node apakah sudah ready atau belum dengan perintah: kubectl get nodes.

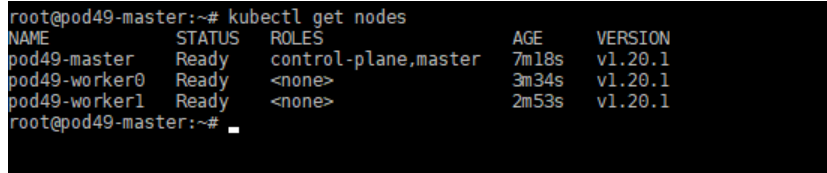


Pada keterangan diatas terlihat join cluster kubernetes sudah berhasil, namun untuk status masih belum ready. Untuk membuat status node menjadi ready, kita harus deploy Container Network Interface (CNI) yang berbasis add-ons Pod network seperti calico, kube-router, dan weave-net. Add-ons Pod network tersebut berfungsi untuk membuat Pod berhubungan satu sama lain.

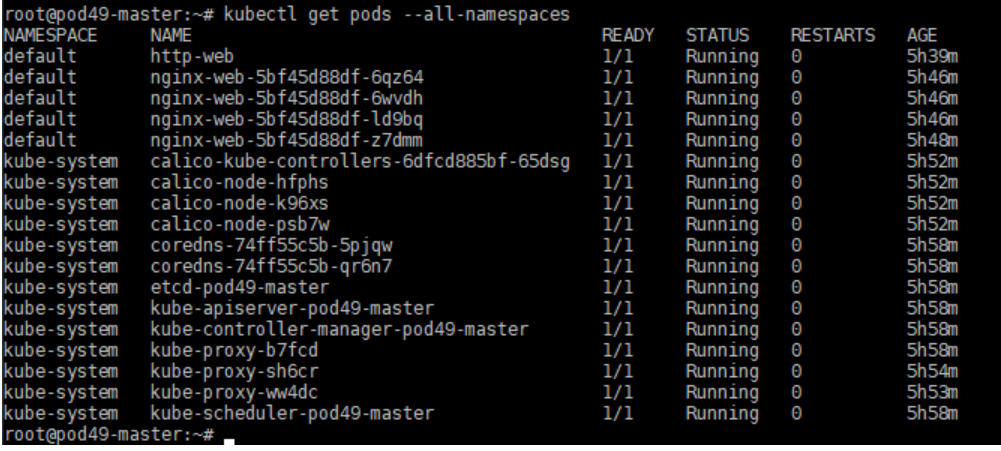
1. Deploy Add-on Pod Network Calico pada node master Login ke node master, lalu jalankan perintah berikut untuk install Add-on Pod Network Calico:



Apabila instalasi sudah selesai, silakan cek kembali status node. Nantinya akan menjadi Ready.



Verifikasi Pod semua namespace apakah sudah bersatus running atau belum.

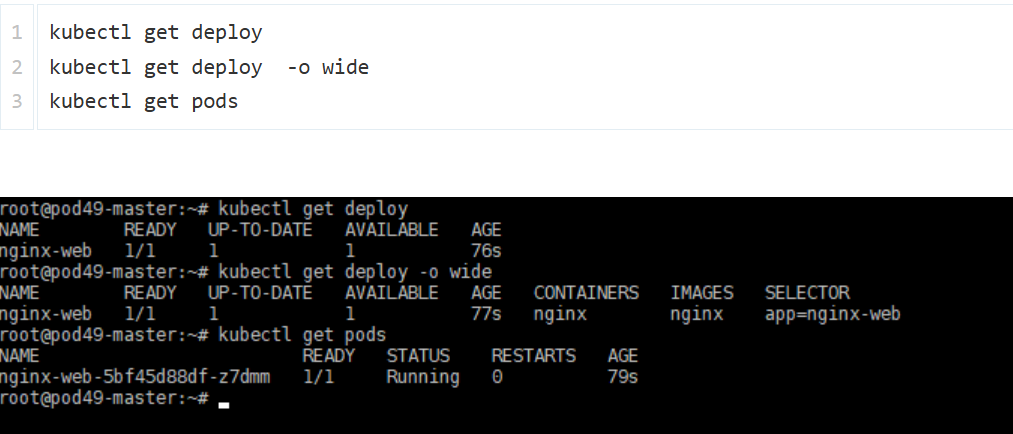


Pada gambar diatas menunjukkan bahwa stutus pod dalam kondisi baik, oleh karena itu bisa melanjutkan untuk mencoba deploy pod, deployment dan service. Hal tersebut berguna untuk cek apakah kondisi cluster berjalan dengan baik atau tidak.

1. Uji coba dan Verifikasi Kubernetes Cluster Coba buat deployment dengan nama **nginx-web** menggunakan image kontainer **nginx** didalam namespace default. Jalankan perintah berikut dari node master.



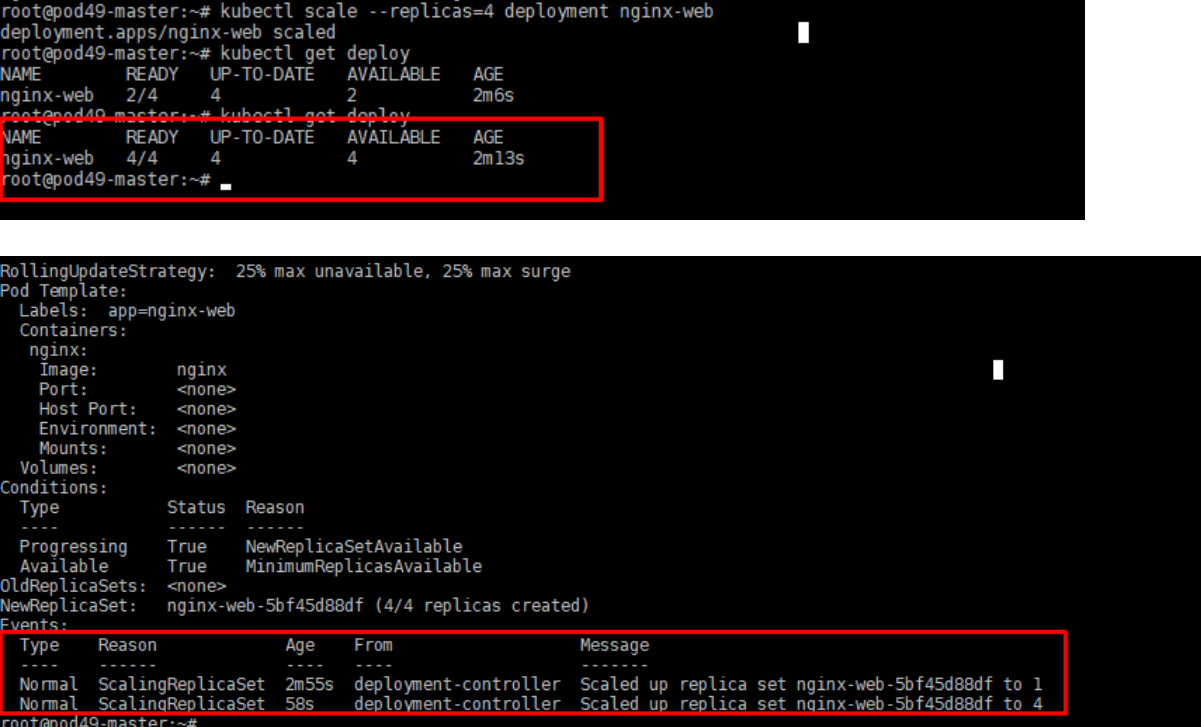
Jalankan perintah berikut untuk verifikasi status deployment.



Dari gambar diatas jumlah Pod masih satu dan belum terdapat replikasinya. Apabila ingin scale up bisa menjalankan perintah:



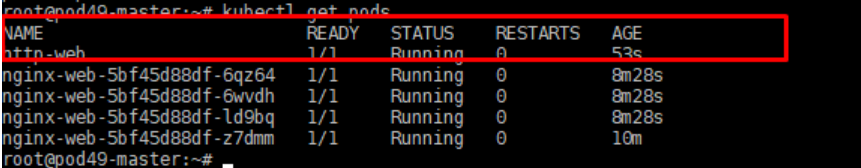
Pastikan jumlah pod adalah 4.



Selanjutnya silakan coba deploy satu pod dengan nama “http-web” dan ekspos dengan nama service “http-service” menggunakan port default 80 dan tipe service NodePort.

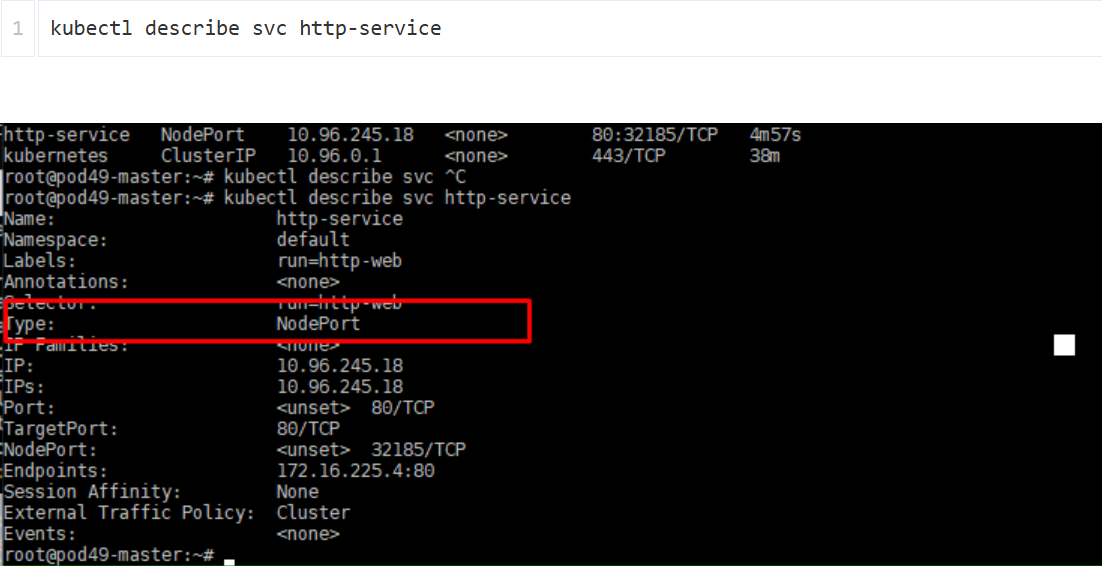


Jalankan perintah kubectl get pods untuk verifikasi status Pod sudah Running atau belum.





Untuk cek service sudah sesuai atau belum bisa menjalankan perintah berikut:



Cek IP atau hostname yang digunakan pod “http-web” lalu silakan akses menggunakan curl via NodePort (32185).

