**RESPONSI PRAKTIKUM**

**TEKNOLOGI CLOUD COMPUTING LANJUT**

**Docker Swarm**

****

**Disusun Oleh :**

|  |  |
| --- | --- |
| **NAMA** | **: DEVIANA WULANDARI**  **: MARIA ERNITA S**  **: MONIKA DWI A** |
| **NIM** | **: 155410082**  **: 1554101**  **: 155410056** |
| **JURUSAN** | **: TEKNIK INFORMATIKA** |
| **JENJANG** | **: S1** |

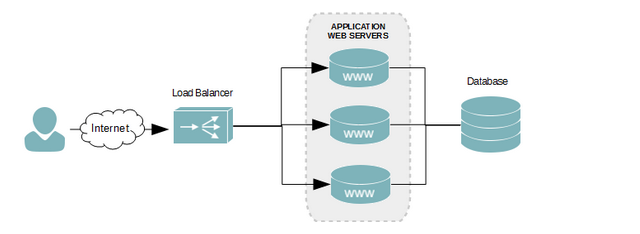
**STMIK AKAKOM**

**YOGYAKARTA**

2018

* **DASAR TEORI**

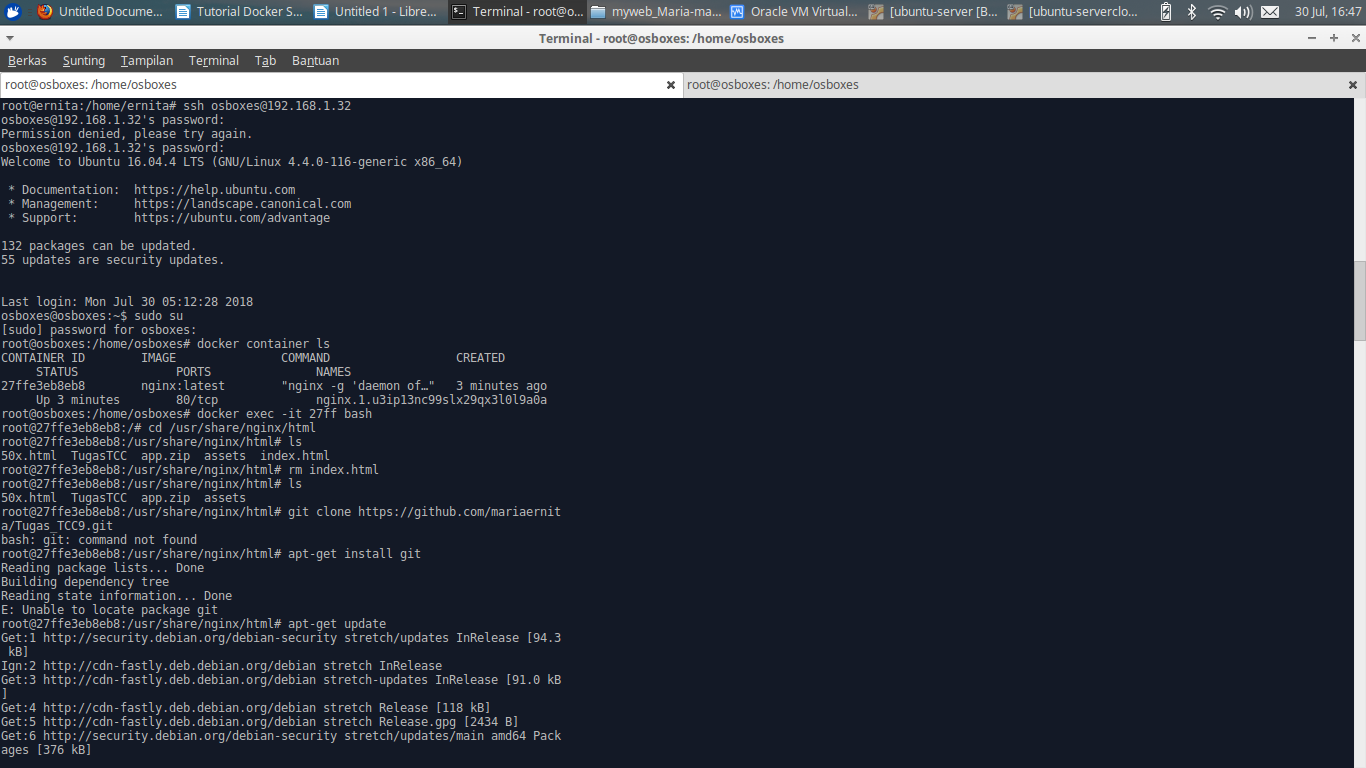
Load Balancing adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan throughput, memperkecil waktu tanggap dan menghindari overload pada salah satu jalur koneksi. Load balancing digunakan pada saat sebuah server telah memiliki jumlah user yang telah melebihi maksimal kapasitasnya. Load balancing juga mendistribusikan beban kerja secara merata di dua atau lebih komputer, link jaringan, CPU, hard drive, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal.



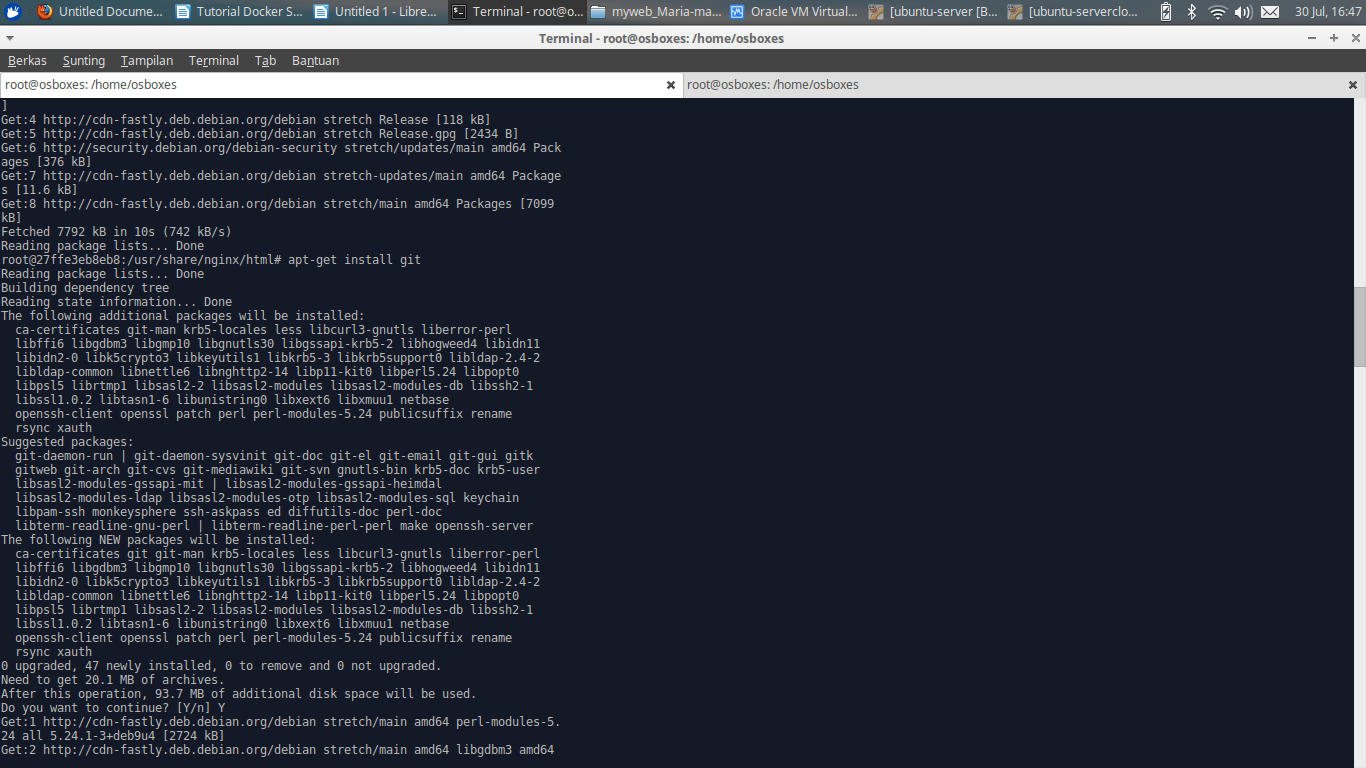
Adapun yang perlu dipahami kembali dari pentingnya menggunakan Load balancing untuk website atau aplikasi berbasis web lainnya diantaranya :

1. Waktu respon adalah manfaat terbesar untuk meningkatkan kecepatan akses website saat dibuka. Dengan dua atai lebih server yang saling berbagi bebam lalu lintas web, masing-masing akan berjalan lebih cepat karena beban tidak berada pada 1 server saja. Ini berarti ada lebih banyak sumber daya untuk memenuhi permintaan halaman website.
2. Dengan Load balancing akan mewarisi sedikit redudansi. Sebagai contoh, jika website kita berjalan seimbang di 3 server dan salah satu server bermasalah, maka dua server lainnya dapat terus berjalan dan pengunjung website kita akan menyadari downtime apapun.

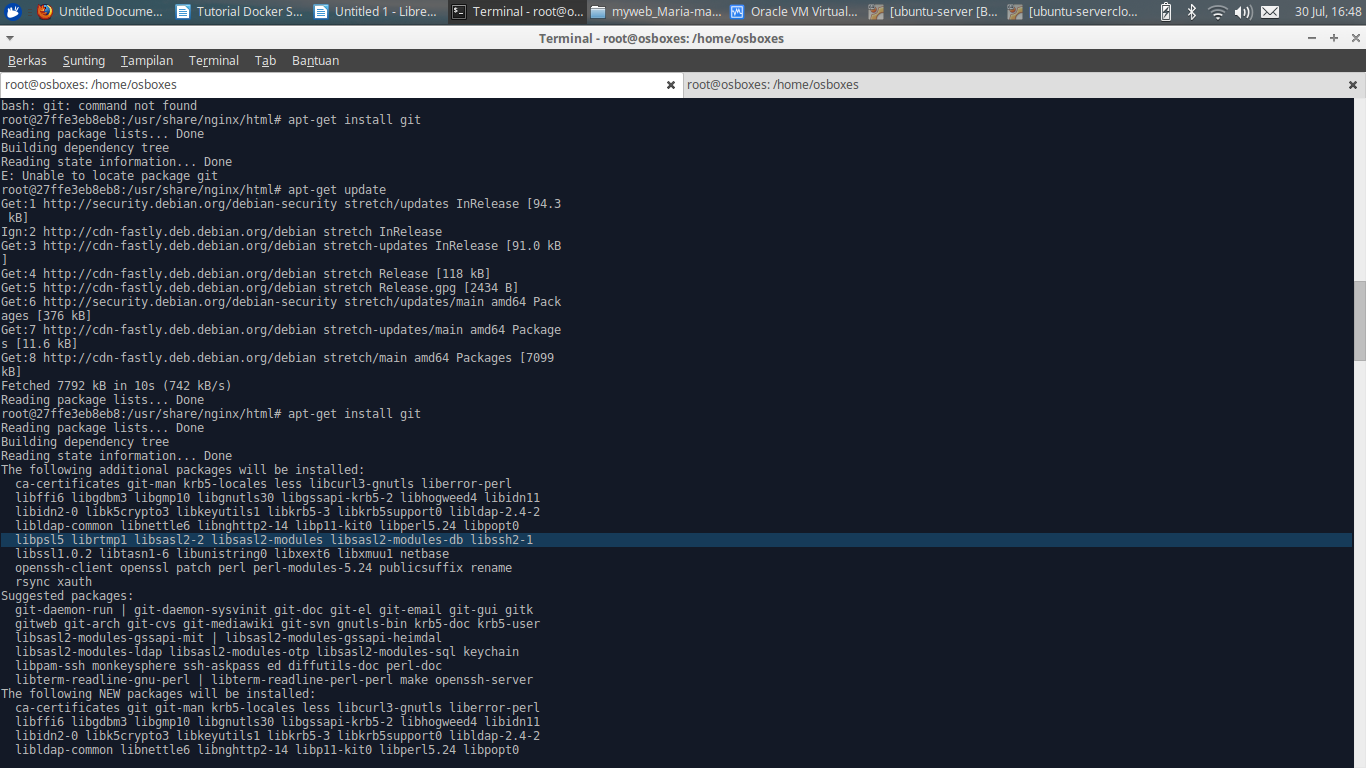
* **PRAKTIK**



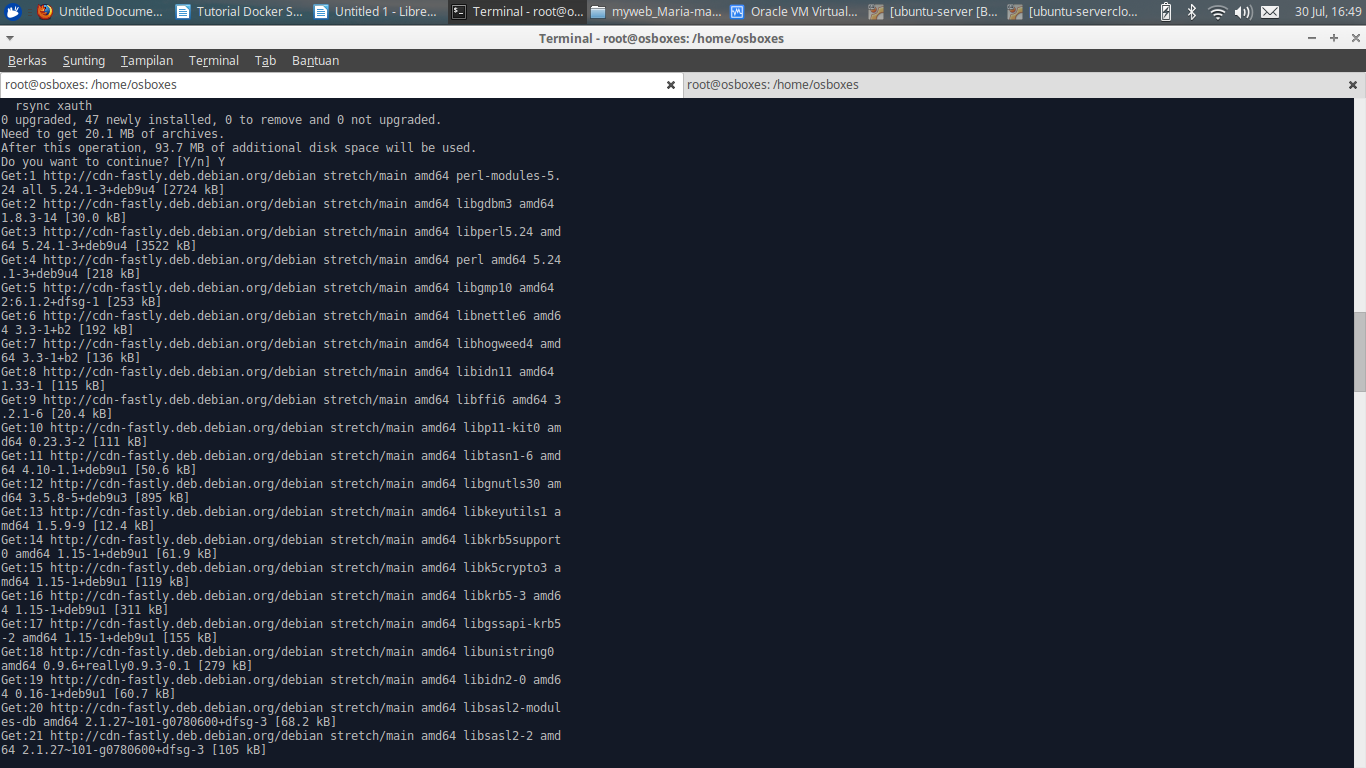
Pembahasan : **sudo su** -> Sebuah perintah dalam command-line Linux. Apabila memiliki akses root,maka sudo akan melakukan perintah sebagai superuser. Di dalam program diatas terdapat nama root yaitu dengan **@osboxes.** kita juga melihat sebuah Container, Name, serta informasi lain yang telah kita buat dengan menggunkan perintah **docker container ls.** Didalam command diatas terdapat perintah **ls** yang artinya index.html tidak memiliki image yang bisa dieksekusi,karena sebelumnya image index.html sudah di buat kemudian di hapus kembali dengan menggunkan perintah **rm index**.**html** . di dalam program diatas juga kita meng **Clone**  git dari <http://github.com/mariaernita/Tugas_TCC9.git> yang artinya disini kita mengambil salah satu tugas tang berada di akun github milik mariaernita. Kemudian setelah kita clone maka kita harus menginstall git terlebih dahulu dengan menggunakan perintah **apt-get install git**



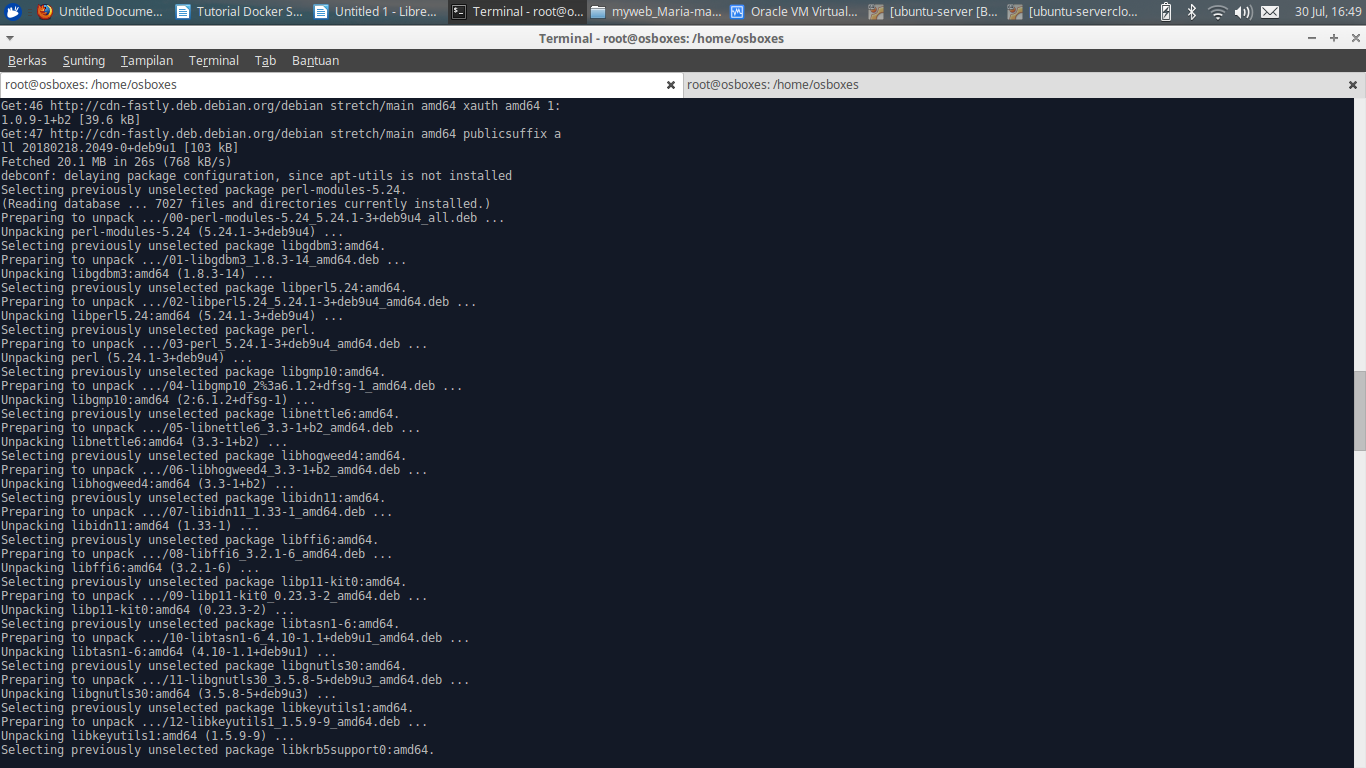
Pembahasan : Command diatas merupakan proses download git yang sedang berjalan. Setelah selesai mendownload akan ada menu pilihan yang harus kita pilih **Y/N?**  Disini kita menuliskan Y karena untuk melakukan proses/langkah selanjutnya.



Pembahasan : Command diatas kesalahan waktu kita mendownload maka kita harus memulai kembali dengan perintah untuk menginstall kembali **git.** Setelah sudah terinstall kita juga harus melakukan update dengan kita menggeikkan perintah **apt-get update** dengan perintah tersebut maka akan mengupdate git yang telah kita install sebelumnya agar tidak mengalami masalah.



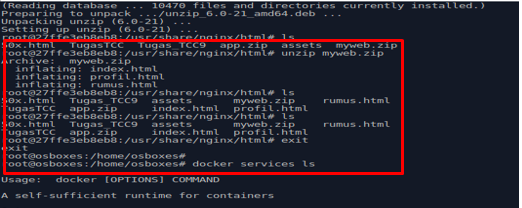
Pembahasan : Tampilan diatas merupaka masih tampilan program yang masih mengupdate **git** yang telah kita install sebelumnya, karena biasanya mengupdate memerlukan waktu yang lumayan lama dan membutuhkan koneksi yang stabil agar tidak terjadi kesalahan terhadap progam yang telah kita install sebelumnya.



Pembahasan : **Tampilan diatas masih menampilkan proses update**



Pembahasan : Setelah mengupdate kita bisa memberikan command pada perintah dengan menggunakan perintah **ls,** dimana perintah ini merupakan perintah untuk menampilkan container yang aktif. Seperti gambar dibawah ini kita membuat **app.zip asset myweb**  seperti dibawah.setelah selesai kita buat maka kita bisa menyimpan file tersebut di dalam **/usr/share/nginx/html** . jika kita ingin melihat informasi nya kita melihat dengan menggunakan perintah **/usr/share/nginx/html# ls** maka akan menmpilkan seluruh informasi yang sedang berjalan. Kemudian jika ingin keluar kita bisa menuliskan perintah **exit**.

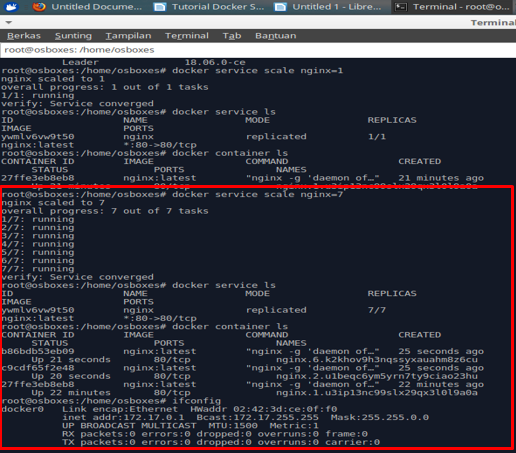




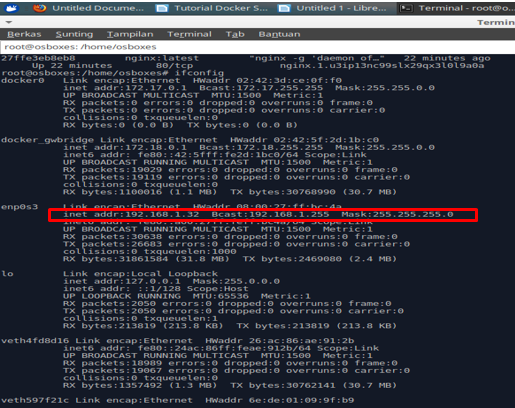
Pembahasan : Tampilan diatas merupakan tamplan dari menampilkan informasi **management command** dan didalamnya kita bisa melihat informasi.



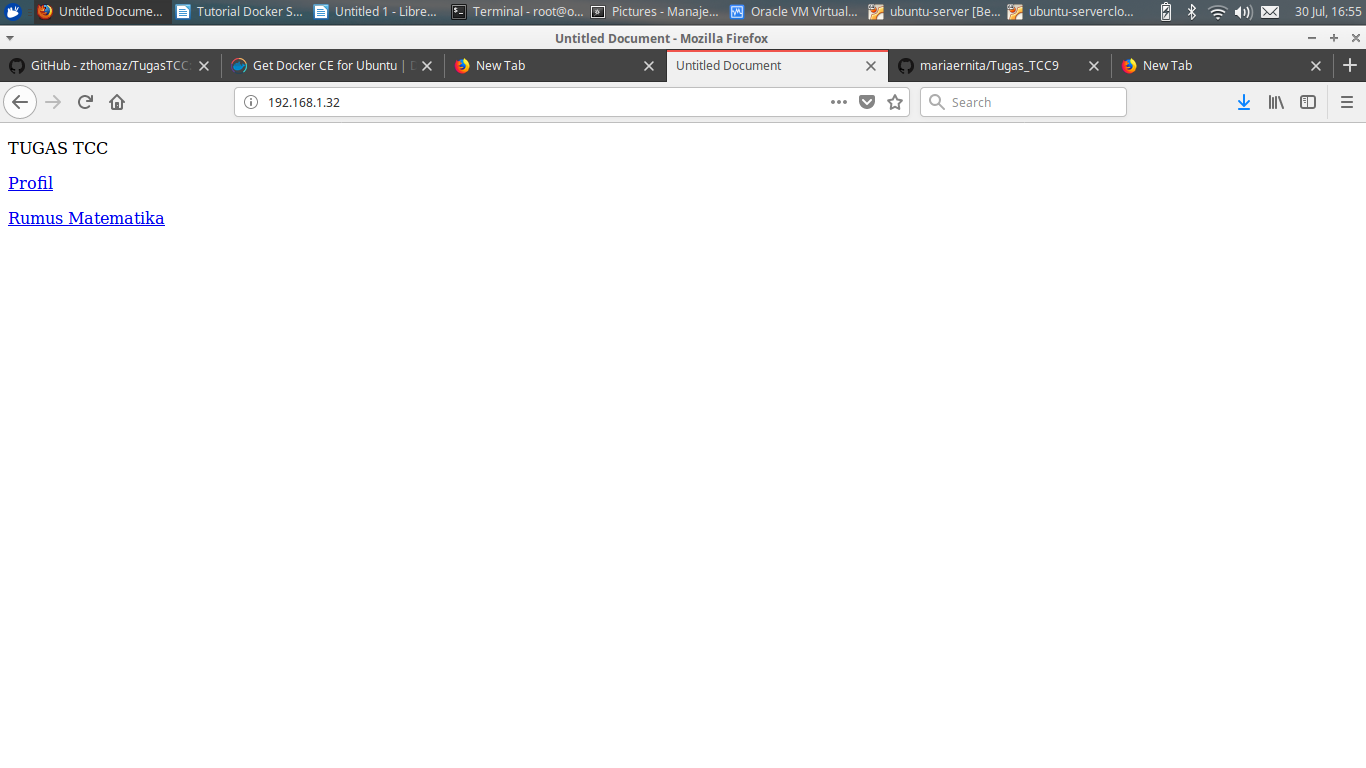
Pemabahasan : **docker service ls ->**  menampilkan seluruh informasi container yang telah aktif yang kita buat sebelumnya pada gambar diatas kita bisa melihat beberapa container yang aktif, sedangkan untuk **docker node ls ->** menampilkan informasi yang lebih kompleks dengan menampilkan status ready atau tidak. Perintah **ls** merupakan perintah untuk menampilkan informasi container yang telah kita berjalan.



Pembahasan : tampilan diatas merupakan tampilan untuk perintah **docker service scale nginx=7** ,setelah proses scaled sampai 7 selesai sudah berstatus running kemudian kita menjalankan dengan perintah **docker service ls** kita dapat melihat informasi docker service yang telah berjalan. Terdapat beberapa informasi container yang berjalan yang bisa kita lihat pada gambar diatas. Dan diatas kita bisa melihat 3 container yang sedang berjalan yang telah kita buat sebelumnya dengan **status, images,command, replicates** yang berbeda-beda. Dan dengan kita mengunakan perintah command **ifconfig** kita bisa melihat informasi apakah ada error di container kita.



Pembahasan : tampilan diatas merupakan tampilan dari running nya perintah ifconfig . **192.168.32** merupakan salah satu link yang bisa kita gunakan untuk mendapatkan hasil dari apa yang telah kita buat di container kita tadi, apakah container yang telah kita buat bisa terbuhung kedalam server ataupun tidak. Jadi kita bisa melihat hasilnya jika kita membuak link tersebut di mozilla, chrome,dll.



Pembahasan : pada tampilan diatas link yang kita akses bisa menampilkan apa yang kita buat sebelumnya di dalam container. Jadi hasilnya container yang telah kita buat mampu terhubung dalam server sehingga tidak terjadi error.