# **DOKUMENTASI**

**Nama : Dzaky A. Badawi**

**Email : dzaky.badawi@gmail.com**

# **INSTALASI AMBARI DENGAN CENTOS 6**

## **Persiapan dan Instalasi Tools :**

* WinSCP
* PuTTY
* VMWare
* Sebelum menginstall Ambari, diperlukan melakukan instalasi terhadap 3 buah mesin sistem operasi CentOS 6.4 pada VMware, dengan rincian :
* 1 mesin / 1 node sebagai server (master), dengan jumlah memory minimal 40 GB.
* 2 mesin / 2 node sebagai host (node1, node2), dengan jumlah memory masing-masing minimal 30 GB.

## **Instalasi Apache Ambari 2.4.0**

### **Bagian I – Persiapan**

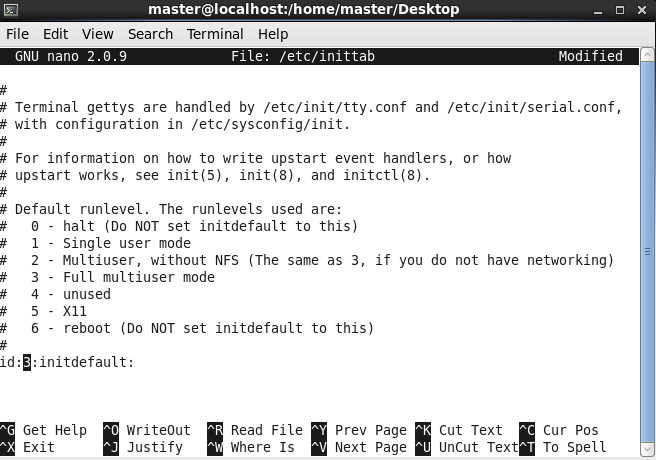
1. Jalankan ketiga mesin CentOS yang telah dibuat sebelumnya, lalu bukalah terminal dan masuk sebagai root dengan perintah “su”, lalu isikan password yang telah dibuat sebelumnya.



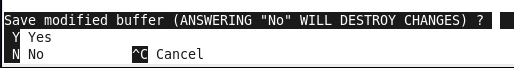
1. Kemudian lakukan konfigurasi pada file inittab, pada mesin master, node1, dan node2. Dengan perintah sebagai berikut :



1. Lalu ubahlah id runlevel menjadi 3 (Full multiuser mode).



1. Kemudian simpanlah konfigurasi file inittab dengan menekan Ctrl+X.

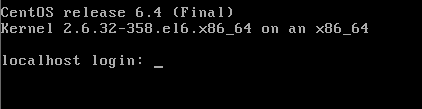




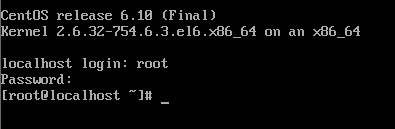
1. Kemudian lakukan reboot dengan perintah “reboot”, pada ketiga mesin tersebut.



1. Jika sudah selesai melakukan reboot, maka tampilan pada node master, node1, dan node2 akan seperti pada gambar berikut :

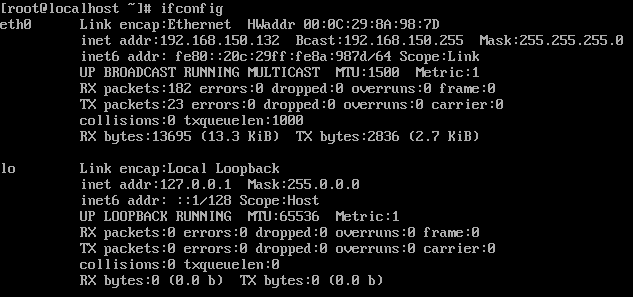


1. Lalu lakukan login sebagai root pada ketiga mesin.



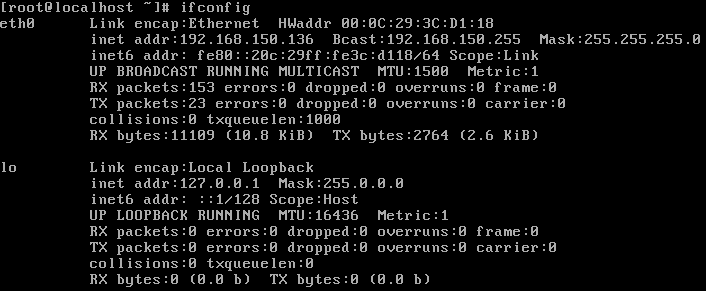
1. Kemudian lakukan pengecekkan alamat IP pada ketiga mesin, dengan perintah ifconfig :
2. Pengecekan alamat IP pada mesin master, didapat memiliki alamat IP yaitu :

**192.168.150.132**.



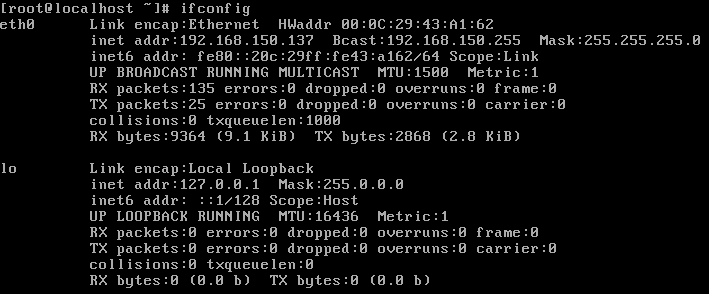
1. Pengecekan alamat IP pada mesin node, didapat memiliki alamat IP yaitu :

**192.168.150.136.**

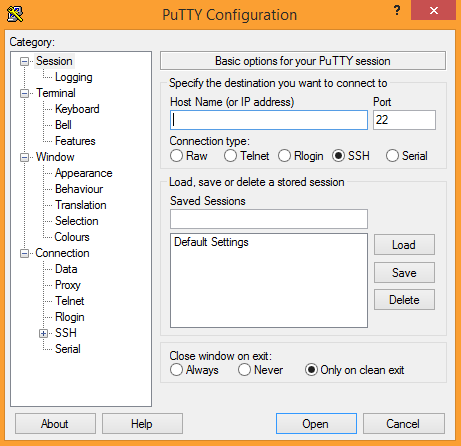


1. Pengecekan alamat IP pada mesin node2, didapat memiliki alamat IP yaitu :

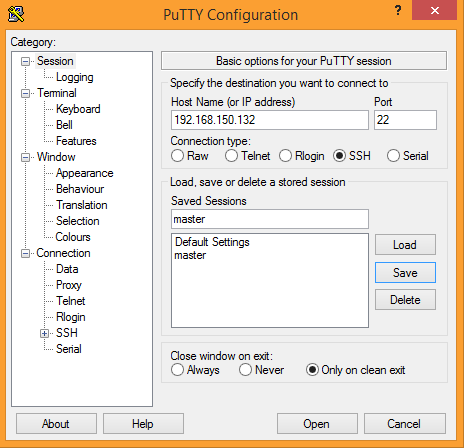
**192.168.150.137**.



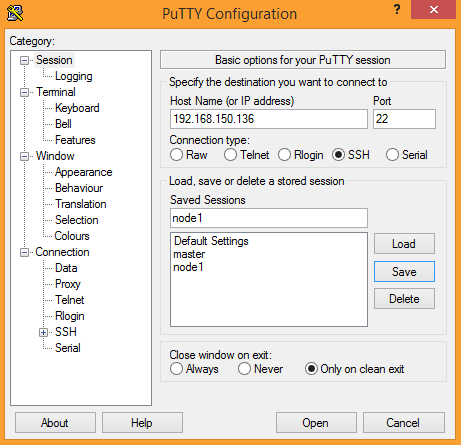
1. Setelah mendapatkan alamat IP ketiga mesin, jalankan aplikasi PuTTY.



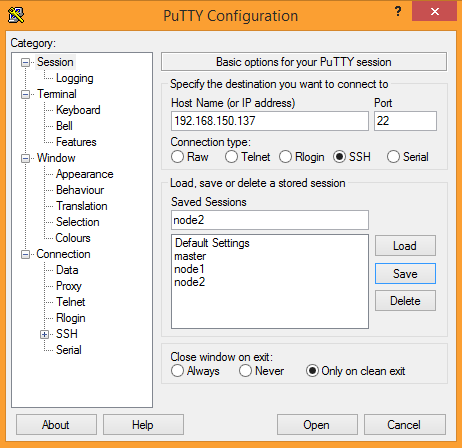
1. Daftarkan alamat IP dari ketiga mesin tersebut :
2. Daftarkan alamat IP dari mesin master yaitu 192.168.150.132, kemudian Save, dan klik Open.



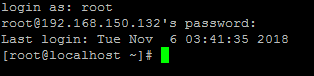
1. Daftarkan alamat IP dari mesin node1 yaitu 192.168.150.136, kemudian Save, dan klik Open.

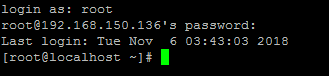


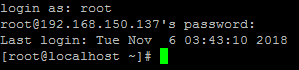
1. Daftarkan alamat IP dari mesin master yaitu 192.168.150.137, kemudian Save, dan klik Open.



1. Jika semua mesin pada puTTY sudah dijalankan, maka lakukan login kembali sebagai root pada ketiga session yang ada :



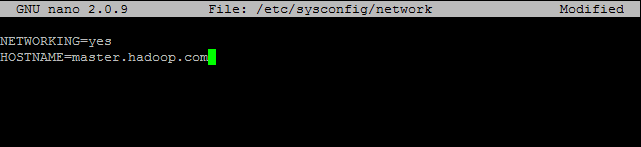




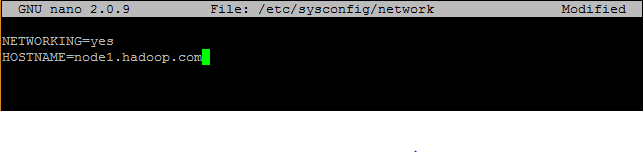
1. Lalu bukalah file network dengan perintah :



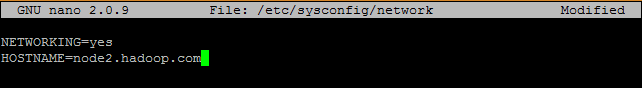
1. Lakukan konfigurasi pada file network di semua mesin :
2. Pada mesin master, berilah nama HOSTNAME=master.hadoop.com



1. Pada mesin node1, berilah nama HOSTNAME=node1.hadoop.com



1. Pada mesin node2, berilah nama HOSTNAME=node2.hadoop.com



Jika sudah melakukan konfigurasi nama HOSTNAME, lalu keluar dengan perintah Ctrl+X, kemudian lakukan penyimpanan dengan menekan “Y”.

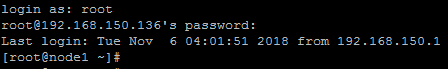
1. Setelah itu lakukan, reboot pada ketiga mesin tersebut :



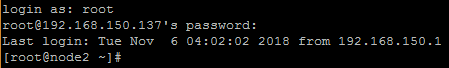
1. Jalankan kembali aplikasi puTTY dan lakukan login kembali sebagai root pada ketiga mesin.
2. Maka dapat dilihat bahwa nama hostname dari mesin master telah berubah menjadi @master.



1. Maka dapat dilihat bahwa nama hostname dari mesin node1 telah berubah menjadi @node1

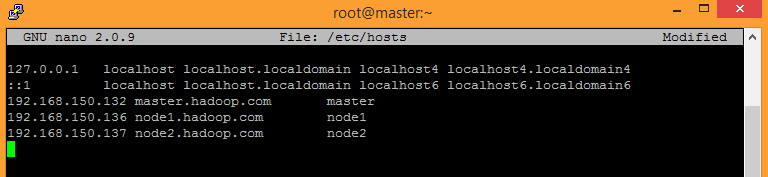


1. Maka dapat dilihat bahwa nama hostname dari mesin node2 telah berubah menjadi @node2



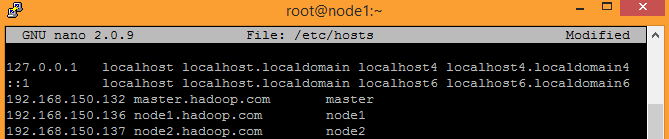
1. Setelah itu bukalah file hosts untuk mendaftarkan host antara mesin, supaya dapat saling terhubung, dengan sebagai berikut :
2. Pada mesin master, daftarkanlah host dengan nama hostname dan alamat IP masing-masing mesin yang ada.





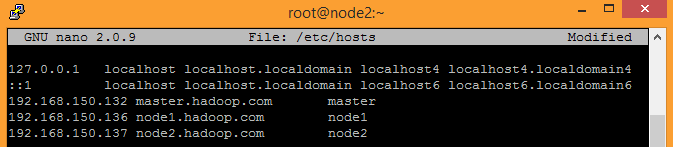
1. Pada mesin node1, daftarkanlah host dengan nama hostname dan alamat IP masing-masing mesin yang ada.



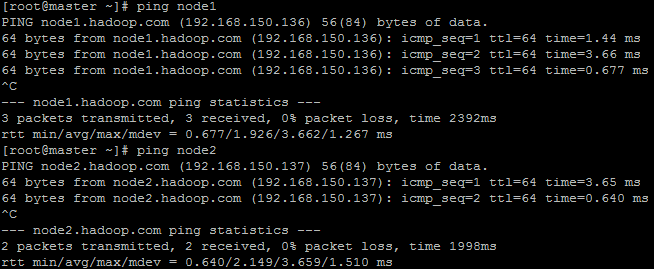


1. Pada mesin node2, daftarkanlah host dengan nama hostname dan alamat IP masing-masing mesin yang ada.

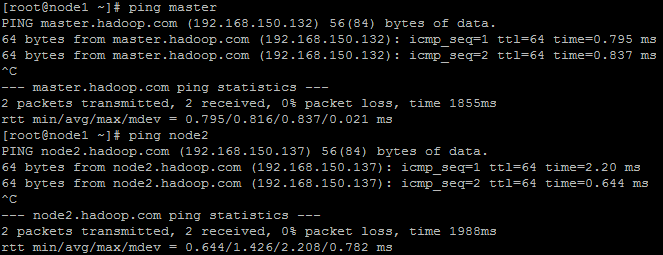




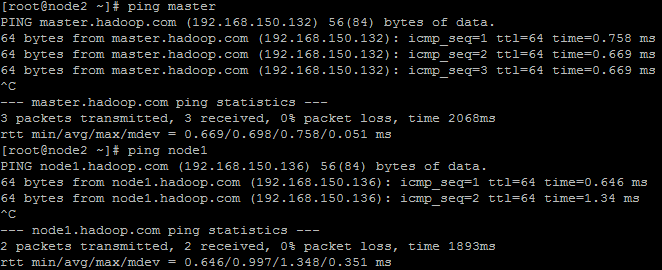
1. Untuk memastikan ketiga mesin sudah saling terhubung maka dapat dilakukan pengecekan dengan melakukan ping antara satu mesin dengan mesin yang lain.
2. Melakukan ping dari mesin master.



1. Melakukan ping dari mesin node1.



1. Melakukan ping dari mesin node2.



1. Jika semua mesin sudah saling terhubung, kemudian salinlah text berikut :

echo "echo never > /sys/kernel/mm/redhat\_transparent\_hugepage/enabled" >> /etc/rc.local

echo "echo never > /sys/kernel/mm/redhat\_transparent\_hugepage/defrag" >> /etc/rc.local

echo "vm.swappiness = 10" >> /etc/sysctl.conf

chkconfig iptables off

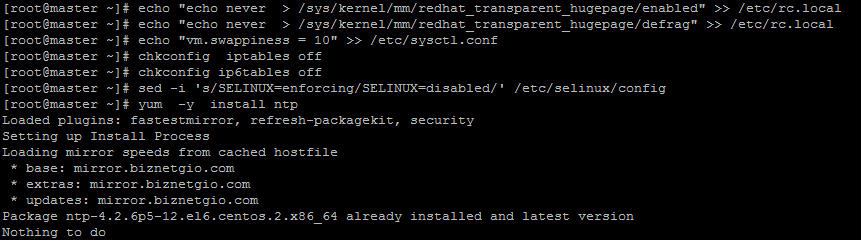
chkconfig ip6tables off

sed -i 's/SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config

yum -y install ntp

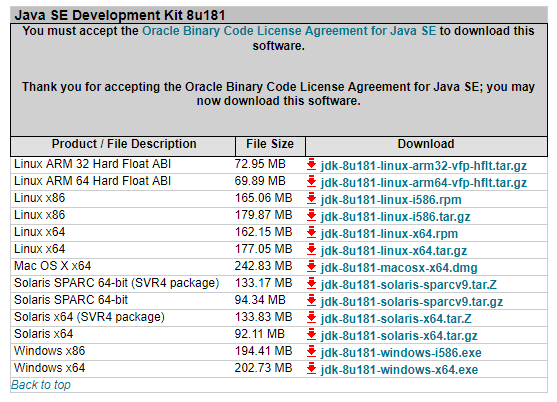
chkconfig ntpd on

Lalu tuliskan perintah di atas pada terminal pada ketiga mesin yaitu master, node1, dan node2. Seperti pada gambar berikut :

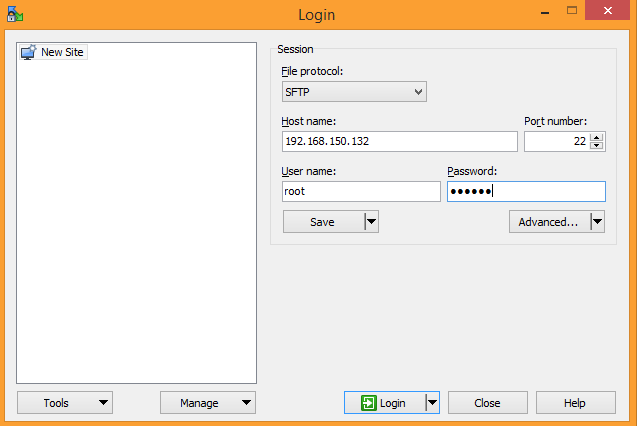


1. Download file JDK pada tautan berikut :

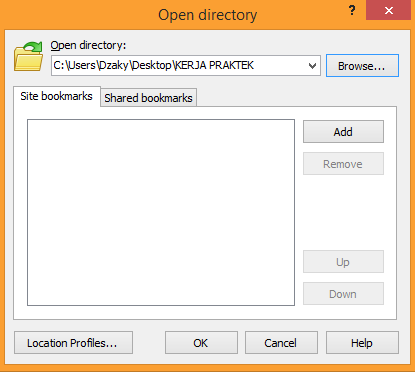
[***https://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html***](https://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase8-2177648.html)



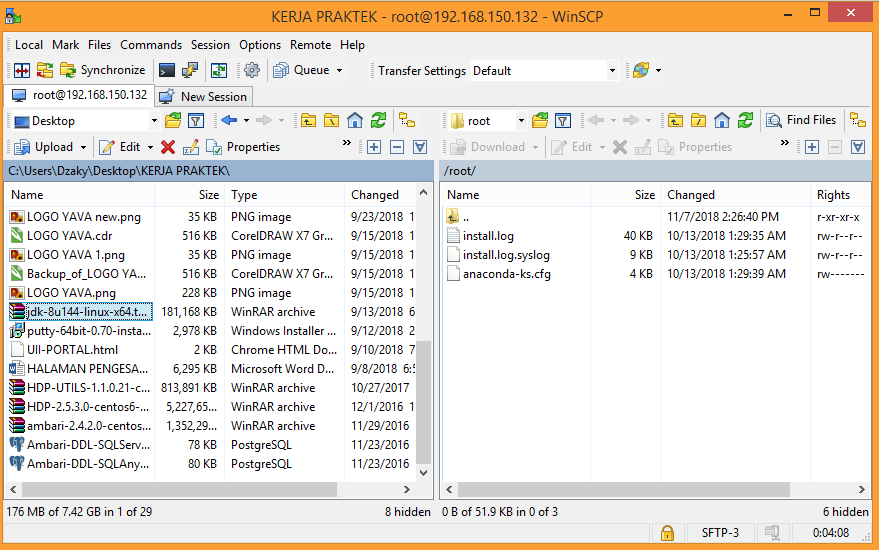
1. Apabila proses download telah selesai, bukalah aplikasi winSCP untuk menyalin file JDK yang telah didownload sebelumnya pada ketiga mesin yang digunakan yaitu master, node1, dan node2. Dengan langkah-langkah sebagai berikut :
2. masukkan alamat IP mesin master beserta username dan password yang digunakan, seperti pada gambar berikut :



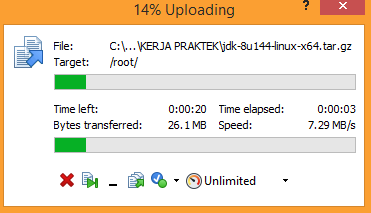
1. Lalu bukalah folder atau direktori tempat dimana file JDK yang telah didownload sebelumnya.



“Klik Browese > Pilih Folder tempat dimana file JDK disimpan > Klik OK”



1. Lakukan penyalinan file JDK kepada folder /root/ pada mesin master.



1. Setelah selesai melakukan penyalinan file, kita dapat melihat file yang sudah disalin pada direktori /root/ di mesin master dengan perintah ls, seperti pada gambar di bawah :



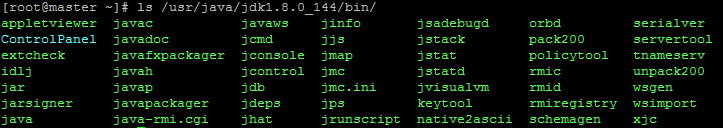
1. Kemudian buatlah direktori dengan perintah mkdir –p /usr/java/



1. Lakukan unzip atau ekstrak file jdk dan letakan hasil ekstrak file tersebut pada direktori /usr/java/ yang telah dibuat dengan perintah berikut :

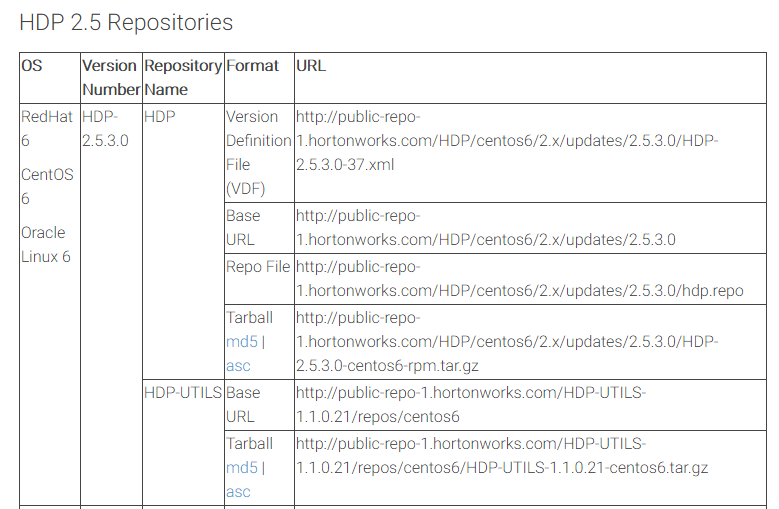


1. Lakukan hal yang sama pada mesin node1 dan node2.
2. Jika pada semua mesin selesai melakukan ekstrasi file JDK, maka kita dapat melihat isi dari direktori /usr/java/, seperti pada gambar berikut :

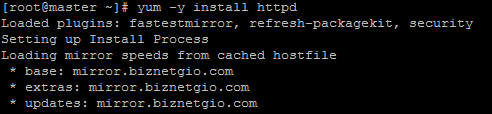


### **Bagian II - Instalasi Ambari**

1. Download file <https://docs.hortonworks.com/HDPDocuments/Ambari-2.6.2.0/bk_ambari-installation/content/hdp_25_repositories.html>



1. Kemudian lakukanlah instalasi httpd jika belum pernah menginstall sebelumnya, dengan perintah berikut :



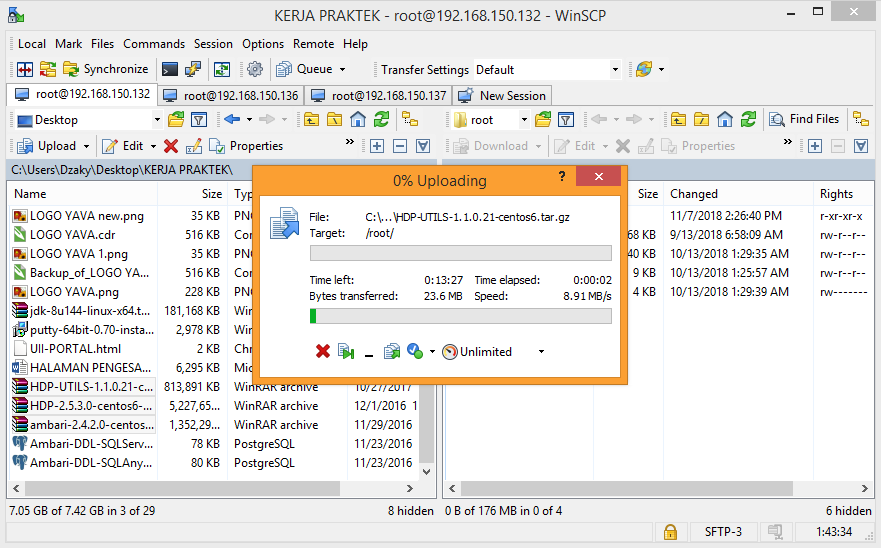
Kemudian ketikkan perintah berikut :



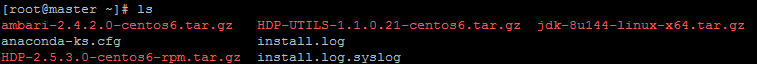
1. Setelah itu jalankan service httpd dengan perintah berikut :



1. Salin 2 file yang telah di download pada direktori /root/ di ketiga mesin yaitu, master, node1 dan node2.



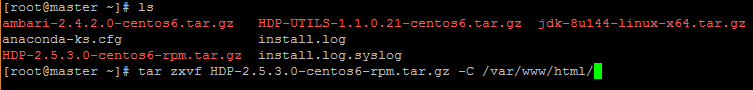
Kemudian Lakukan pengecekan file yang telah disalin dengan perintah ls, untuk memastikan file sudah berhasil disalin :



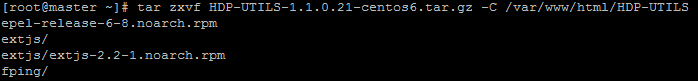
1. Lakukan ekstrak file ambari-2.4.2.0 ke dalam folder /var/www/html/



1. Kemudian lakukan ekstrasi file HDP-2.5.3.0, ke dalam folder /var/www/html/.



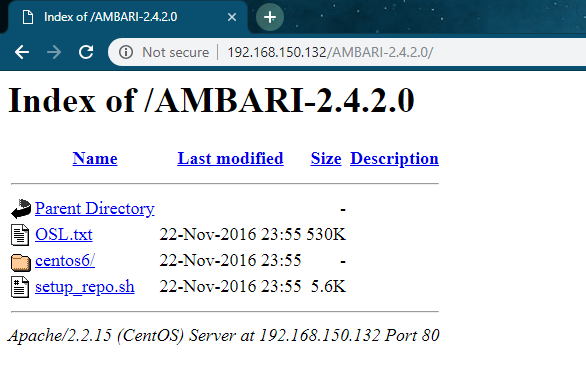
1. Buatlah folder “HDP-UTILS” dengan perintah mkdir, pada path /var/www/html/, kemudian lakukan ekstrasi file HDP-UTILS-1.1.0.21-centos6.tar.gz seperti gambar berikut :



1. Kemudian lakukan reboot :



1. Setelah selesai reboot, bukalah browser lewat komputer utama anda dan ketikkan url dengan alamat IP mesin master seperti gambar berikut :



Lalu salinlah path url berikut : <http://192.168.150.132/AMBARI-2.4.2.0/centos6/2.4.2.0-136/>.

1. Gantilah direktori pada terminal mesin master,



1. Lalu di dalam direktori tersebut, buatlah file dengan nama ambari.repo dengan perintah nano.



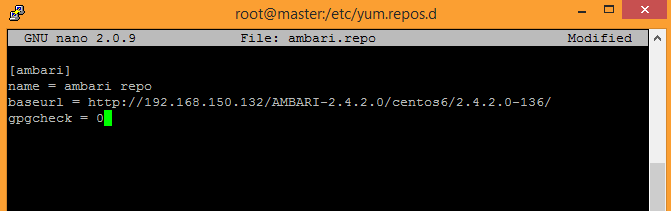
[ambari]

name = ambari repo

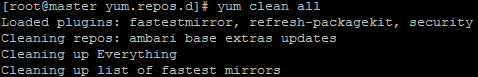
baseurl = http://**[Alamat IP Master]**/AMBARI-2.4.2.0/centos6/2.4.2.0-136/

gpgcheck = 0

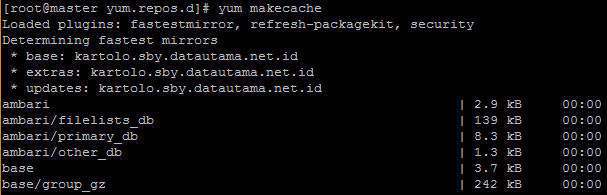
Kemudian ketiklah tulisan pada kotak di atas dan di dalam file ambari.repo tersebut, seperti pada gambar di bawah dan sesuaikan dengan alamat IP mesin master/servernya, kemudian simpan.



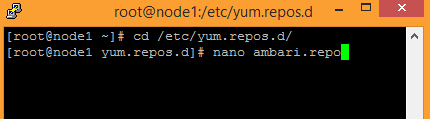
1. Lalu ketiklah perintah yum clean all :

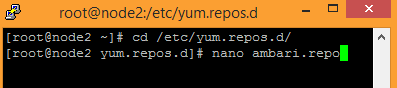


1. Ketiklah perintah yum makecache :



1. Buatlah file ambari.repo kembali, pada mesin node1 dan node2 di direktori /etc/yum.repos.d/ :





Lalu di dalam file ambari.repo tersebut, salinlan teks di bawah, kemudian simpan.

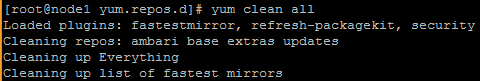
[ambari]

name = ambari repo

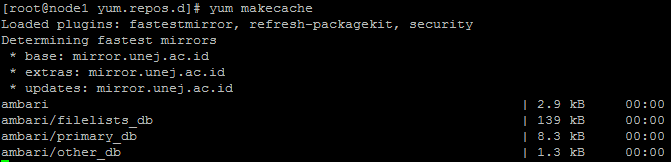
baseurl = http://**[Alamat IP Master]**/AMBARI-2.4.2.0/centos6/2.4.2.0-136/

gpgcheck = 0

Kemudian, ketikkan perintah yum clean all pada terminal mesin node1 dan node2 :



Jika proses clean all sudah selesai, maka ketiklah perintah yum makecache pada terminal mesin node1 dan node2 :



1. Kemudian pada terminal mesin master atau server, ketikkan perintah cd ~ untuk kembali ke direktori /root/ :



1. Lakukan instalasi ambari-server hanya pada mesin master dengan perintah seperti pada gambar di bawah, dan tunggulah beberapa saat hingga proses instalasi selesai.



1. Apabila proses instalasi ambari-server sudah selesai, maka langkah selanjutnya adalah melakukan instalasi ambari-agent pada mesin master, node1 dan node2, dengan perintah seperti pada gambar berikut di semua mesin :





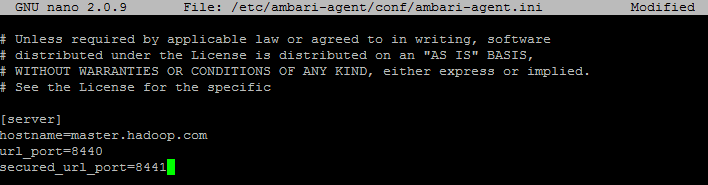


1. Lakukan konfigurasi pada file ambari-agent.ini di mesin master, node1 dan node2.





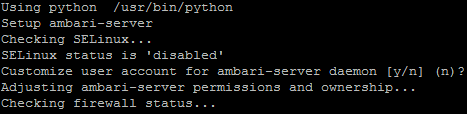
1. Lalu ubahlah isi hostname dengan hostname milik mesin master karena sebagai server, yang memiliki hostaname yaitu master.hadoop.com



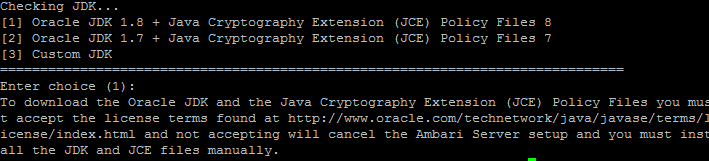
1. Lalu jalankan setup ambari-server.



Pada saat setup, langkah pertama adalah pengecekan SELinux, untuk defaultnya maka klik enter atau pilih n :



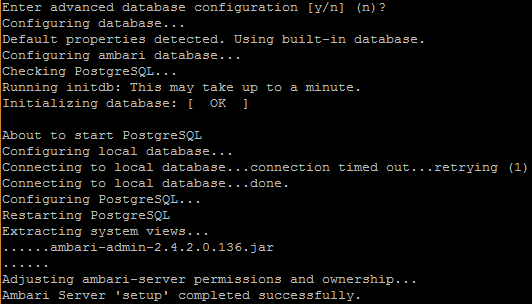
Kemudian pilih pilihan (1) atau enter, untuk memilih JDK secara default.



Kemudian ketik y atau enter, untuk accept agreement.



Klik n, untuk memilih database yang digunakan secara default yaitu PostgreSQL.



1. Jalankan ambari-server pada terminal master dengan perintah seperti berikut :



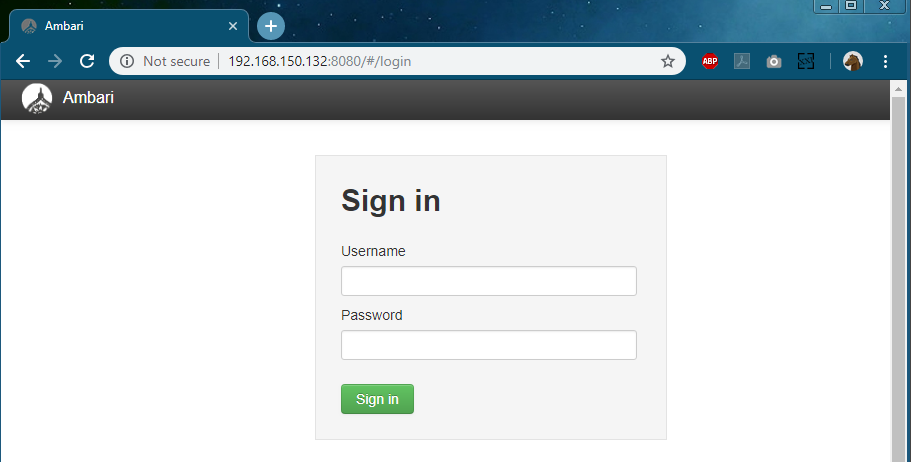
1. Jalankan ambari-agent pada terminal di mesin master, node1, dan node2 dengan perintah “service ambari-agent start” :







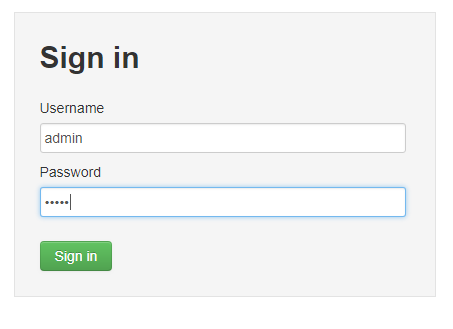
1. Setelah semua ambari-agent dan ambari-server dijalankan, bukalah browser dan masukkan alamat IP dari mesin master yang terinstall dengam ambari-server, seperti pada gambar berikut :



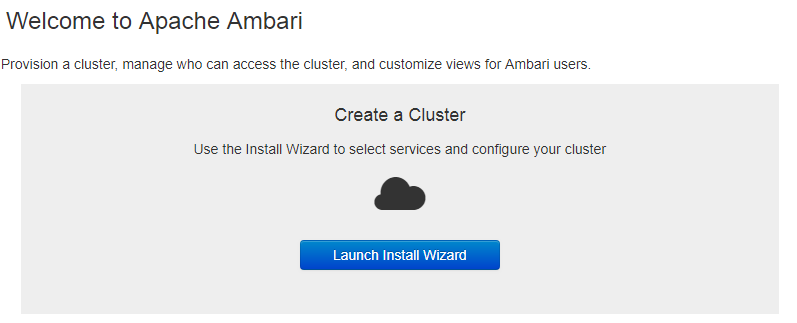
Dengan demikian, cara untuk menginstall proses setup ambari-server dan ambari-agent.

### **Bagian III – Membuat Cluster**

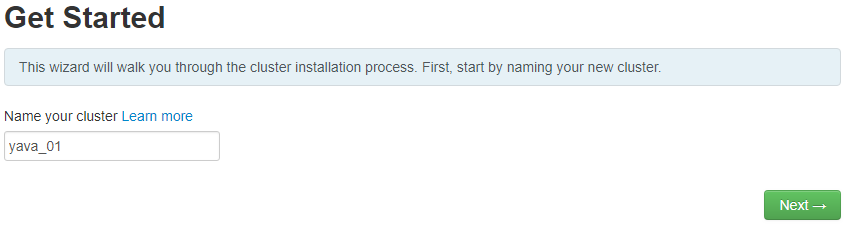
1. Bukalah browser lalu login dengan user yang sudah terbuat pada saat melakukan setup ambari di proses sebelumnya. Masukkan username “admin” dan password “admin”. Kemudian klik “Sign in”



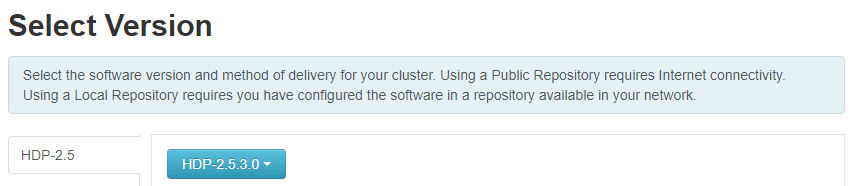
1. Untuk membuat cluster baru, klik tombol “Launch Install Wizard”.



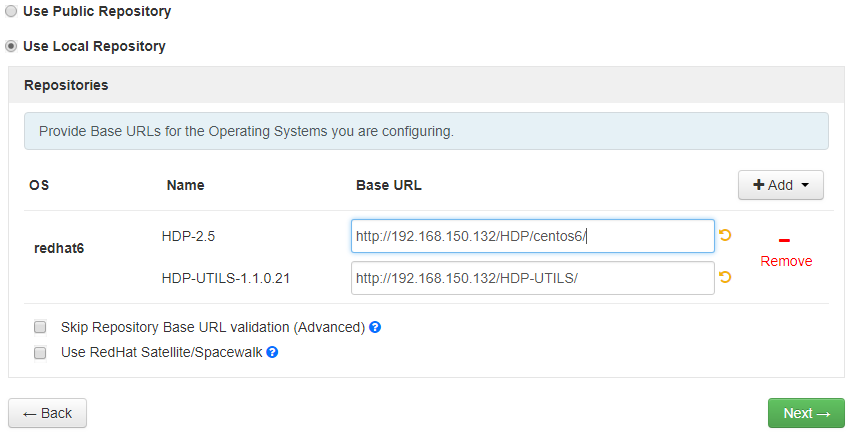
1. Isikan nama cluster yang dibuat misalkan “yava\_01”, kemudian klik next.



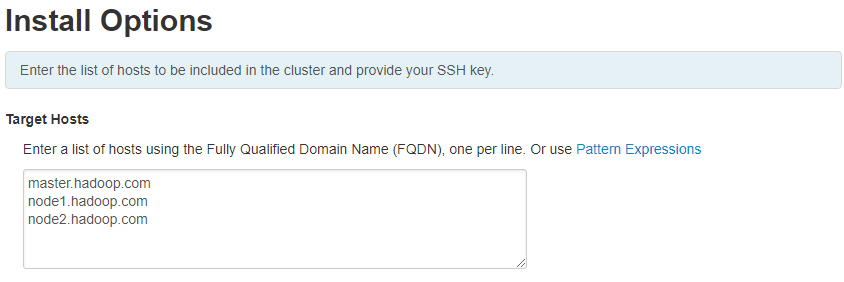
1. Gunakanlah versi HDP-2.5.



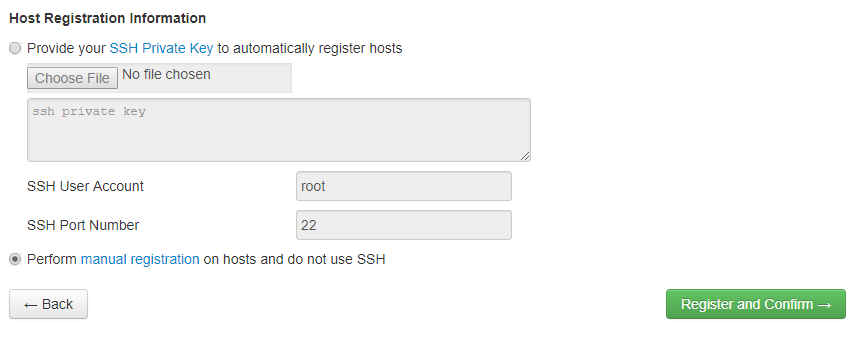
Kemudian klik menggunakan “local repository”, dan remove atau hapus sistem operasi selain redhat6 (karena dalam dokumentasi ini menggunakan OS CentOS 6). Lalu masukkan Base URL, sesuai dengan dimana kita telah meengstrak file HDP dan HDP-UTILS sebelumnya. Kemudian klik next.



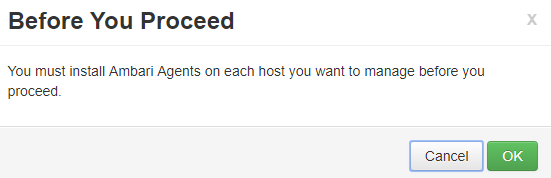
1. Pada tahapan Install Options,isilah target hosts berdasarkan hostname yang sudah dikonfigurasi sebelumnya pada mesin master, node1, dan node2.



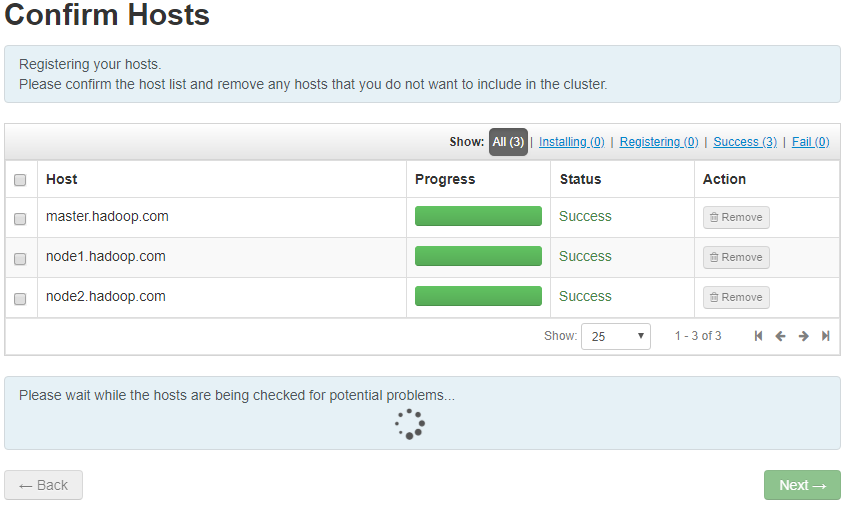
Karena pada proses pembuatan cluster ini juga belum memiliki SSH, maka pilih “Perform manual registration…” lalu klik Register and Confirm 🡪.



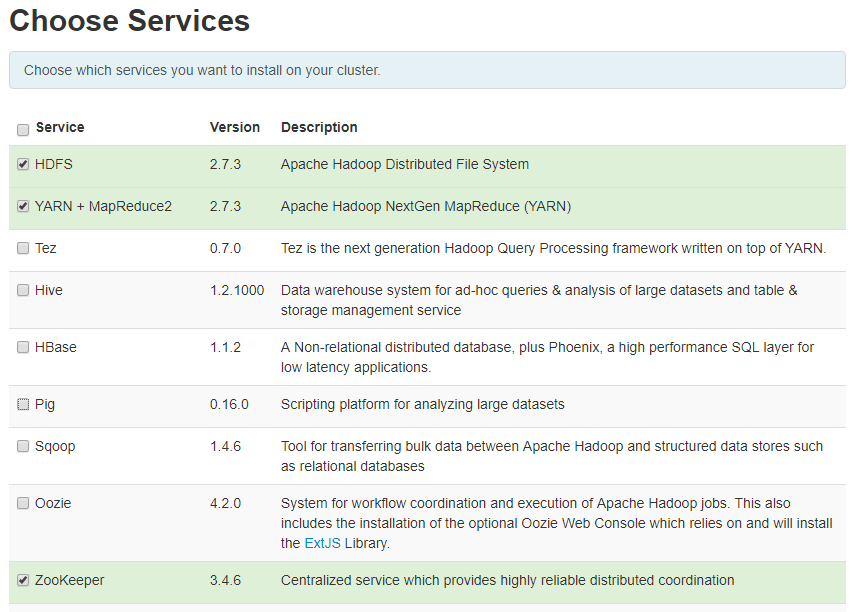
Karena pada proses sebelumnya juga sudah melakukan instalasi ambari-agent maka langsung saja klik OK.



1. Maka tunggulah beberapa saat hingga semua Host sudah terdaftar, kemudian klik Next.

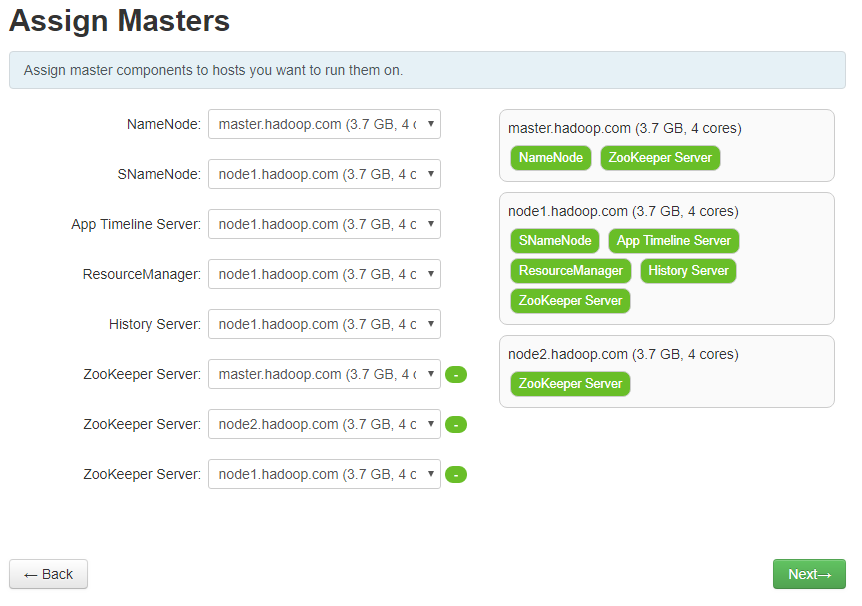


1. Lalu pilihlah services yang akan diinstal, misalkan beberapa service seperti pada gambar berikut :

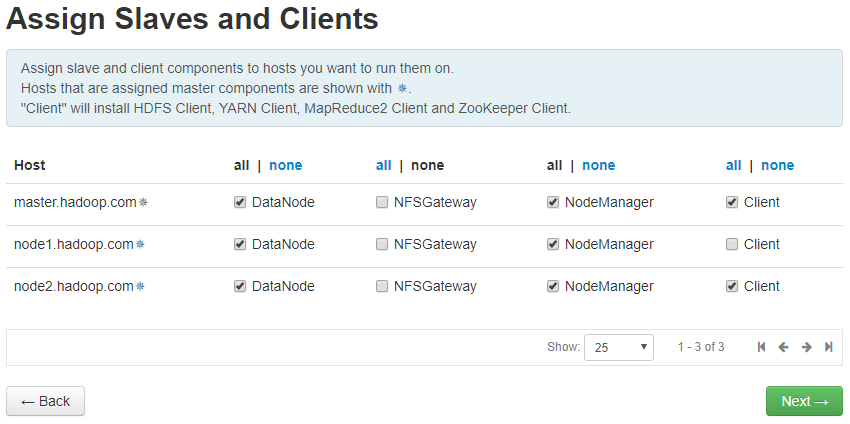


Jika sudah memilh services, maka klik Next.

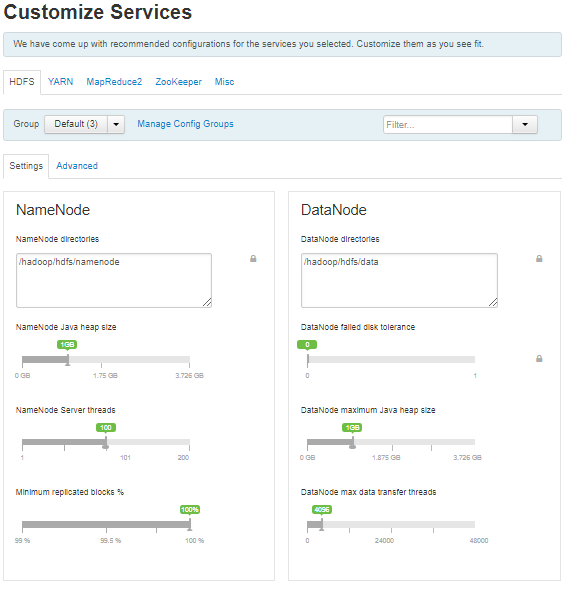
1. Klik next apabila tidak ingin melakukan perubahan.



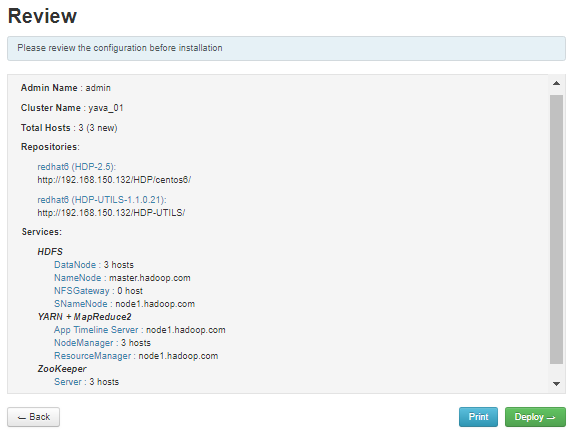
1. Klik next saja, apabila tida ingin melakukan pengaturan terkait pengaturan Slaves dan Clients.



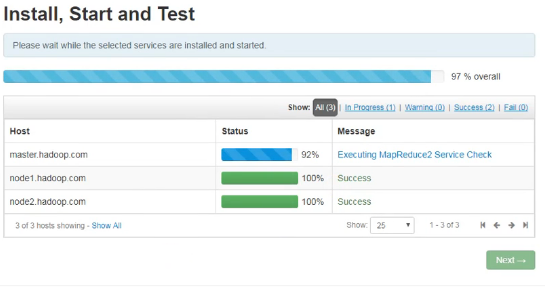
1. Lakukan kustomisasi jika diperlukan, apabila tidak maka klik Next.



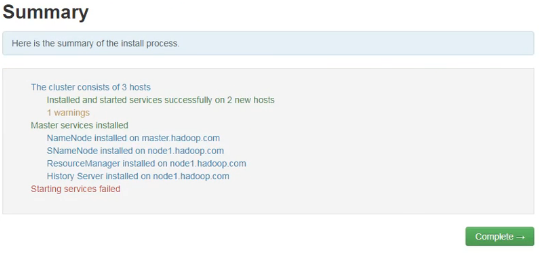
1. Berikut adalah hasil Review dari hasil proses pembuatan Cluster, kemudian klik Deploy untuk memulai proses penerapan Services pada Cluster yang dibuat.



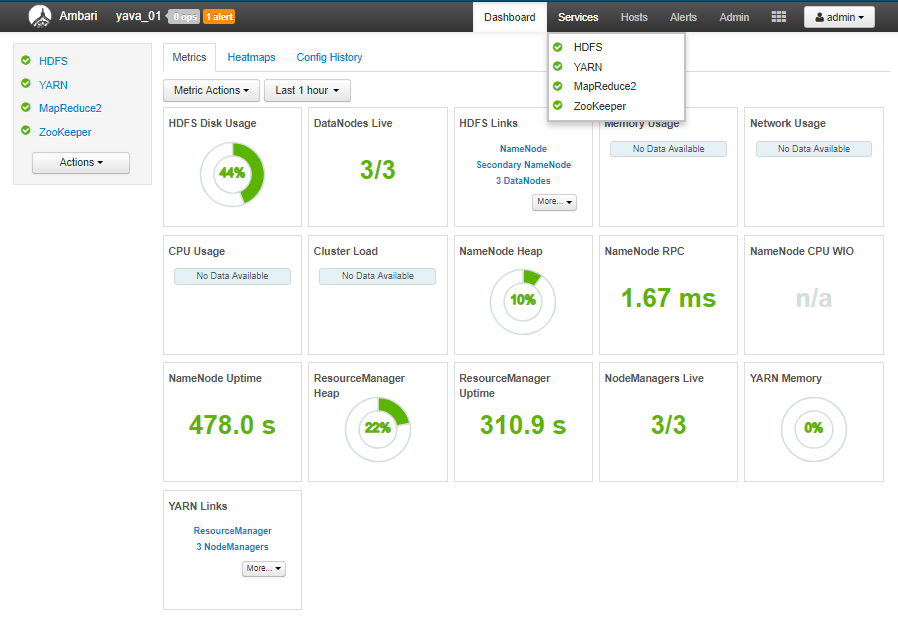
1. Tunggulah beberapa saat untuk melakukan instlasi services terhadap host yang terdaftar, jika sudah selesai maka klik next.



1. Berikut tampilan Summary atau kesimpulan dari hasil proses instalasi.



1. Setelah proses pembuatan Cluster telah selesai, berikut tampilan Dashboard dari Ambari beserta grafik yang ditampilkan berdasarkan Services yang telah diinstal.



**Referensi :**

[*https://www.youtube.com/watch?v=\_T4Ktq6yZ3Q&list=PLY-V\_O-O7h4dwTtMk77X6JO-AcENbSBr\_*](https://www.youtube.com/watch?v=_T4Ktq6yZ3Q&list=PLY-V_O-O7h4dwTtMk77X6JO-AcENbSBr_)

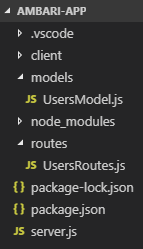
[*https://docs.hortonworks.com/HDPDocuments/Ambari/Ambari-2.4.2.0/index.html*](https://docs.hortonworks.com/HDPDocuments/Ambari/Ambari-2.4.2.0/index.html)

# **PERANCANGAN REST API LOGIN & USER**

## **Persiapan Tools :**

* Visual Studio Code (Text Editor)
* NPM v.6.4.1
* Express JS (Framework)
* MongoDB (Database)
* Postman (API Tester)

1. Struktur Direktori Back-End :



1. client : Direktori untuk membuat Front-End
2. models : Direktori untuk menyimpan model yang menangani data
3. routes : Direktori untuk membuat API
4. Menghubungkan back-end dengan mongoDB :
5. Instalasi module mongoDB : **npm install mongoose --save**
6. Buatlah variable di file server.js untuk memanggil module mongoose

***const mongoose = require("mongoose")***

1. Deklarasikan variable terkait path dan nama database yang akan dibuat seperti sintaks berikut dan letakkan di server.js :

**const mongoURI = 'mongodb://localhost:27017/ambariDB'**

1. Lalu hubungkan back-end dengan mongoDB dengan menuliskan sintaks berikut pada file server.js :

**mongoose.connect(mongoURI, {useNewUrlParser: true})**

**.then(() => console.log("MongoDB Berhasil Terhubung"))**

**.catch(err => console.log(err))**

1. Membuat models :
2. Buatlah direktori /models dan file userModels
3. Buatlah variable di file server.js untuk memanggil module mongoose dan buatlah fungsi schema dari module mongoose
   1. ***const mongoose = require("mongoose")***
   2. ***const schema = mongoose.Schema***
4. Schema untuk tabel users :

***const UserSchema = new Schema({***

***\_id: mongoose.Schema.Types.ObjectId,***

***username: {***

***unique: true,***

***type: String,***

***required: true***

***},***

***password: {***

***type: String,***

***required: true***

***}***

***})***

1. Eksport model users

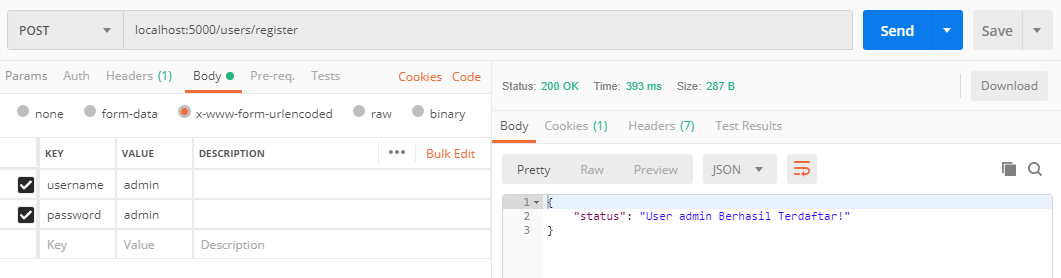
***module.exports = User = mongoose.model('users', UserSchema)***

## **Fungsi API Login & User :**

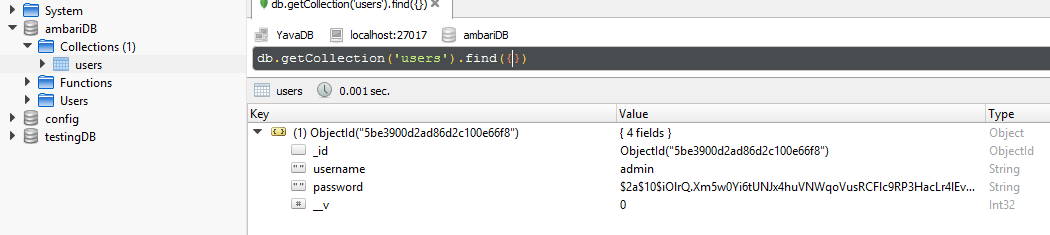
1. Register / Create User

<http://localhost:5000/users/register>

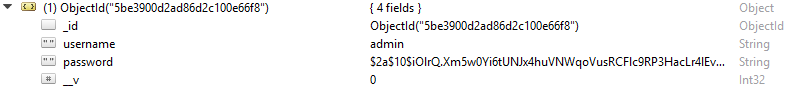
Contoh :



Berhasil ditambah pada database :

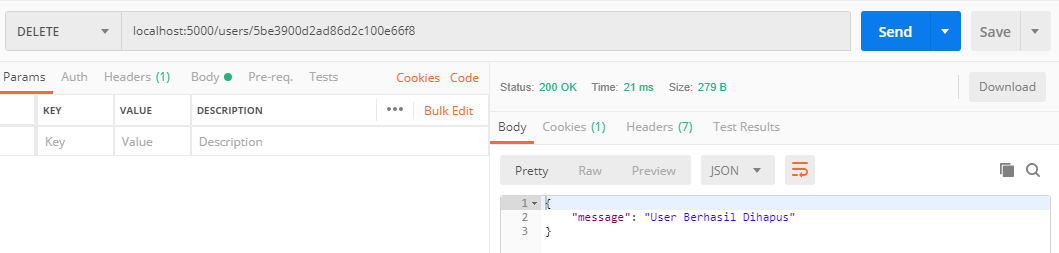


1. Hapus / Delete User berdasarkan \_id

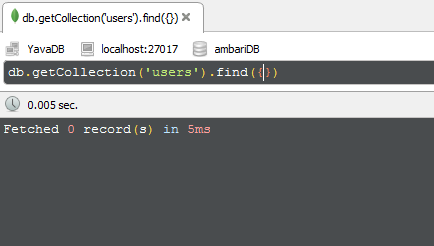


<http://localhost:5000/users/:user>Id

Contoh :



Data user dengan id “5be3900d2ad86d2c100e66f8” berhasil dihapus dari database :

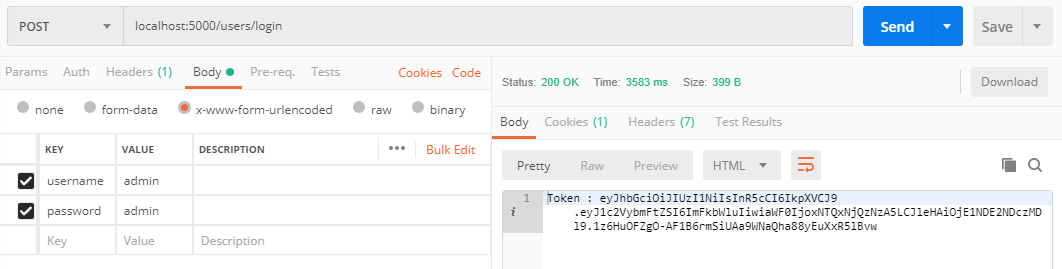


1. Login User

“Jika login berhasil, maka server akan menghasilkan token untuk autentikasi.”

http://localhost:5000/users/login

Contoh :



**Referensi :**

[*https://www.youtube.com/watch?reload=9&v=yJchGDVZFTA*](https://www.youtube.com/watch?reload=9&v=yJchGDVZFTA)