## 1. Jump Server Prepareation

1) Create a directory called “/home/daisy\_install”.

# mkdir /home/daisy\_install

2) Create Daisy configuration file in above directory:

# touch /home/daisy\_install/daisy.conf

3) Fill daisy.conf with the following contents as example:

|  |
| --- |
| [DEFAULT]  #The mangement ip of daisy  #When Daisy will be installed in the virtual machine, this option is required.  daisy\_management\_ip=10.43.177.190  [BACKEND]  #Default backend types of daisy, including tecs, zenic, proton.  #If you want to create a cluster with more than one backend,  #all backend names should be provided for this configuration item,  #such as, default\_backend\_types=tecs,zenic,proton,kolla  default\_backend\_types=kolla  [PXE]  #Set to 'yes' if you want to build a PXE server, otherwise to 'no'.  build\_pxe=no  #the nic name, to build a PXE server on this nic.  eth\_name=  #The ip value of PXE server  ip\_address=99.99.1.5  #the net mask of PXE server  net\_mask=255.255.255.0  #The start value of PXE client ip range  client\_ip\_begin=99.99.1.50  #The end value of PXE client ip range  client\_ip\_end=99.99.1.150 |

4) Modify the above example daisy.conf according to your environment, two config items which mainly need to be changed are:

daisy\_management\_ip is your jump server IP.

default\_backend\_types is the preferred backend which is used for deploying your distributed system. Currently only “kolla” is supported for deploying OpenStack.

5) Make sure your jump server can access the Internet.

## 2. Target Host Prepareation

In this demo, we do not support discovery node over PXE, IPMI and Operating System Installation. So please Install CentOS7 on each target hosts beforehand. Daisy only support CentOS7 as target OS currently.

Currenlty, Daisy restricts that all interfaces for a specific network plane must have exactly the same name. For example, if interface1 on host1 and interface2 on host2 are both used for one specific network plane, then interface1 and interface2 MUST have the same name, such as, they are both named “eth0”. This is a temporary restriction which will be removed in the near future. If you do not have interface which meet this requirement, please see the Appendix A.

All target hosts must have at least 2 physic network interfaces.

## 3. Install Daisy on Jump Server

1) Get the latest installdaisy\_el7\_noarch.bin demo artifact from www.daisycloud.org.

2) Execute it like the following:

|  |
| --- |
| # ./installdaisy\_el7\_noarch.bin  Verifying archive integrity... All good.  Uncompressing tecs............................  ====================================  ZTE DAISY Installation Wizard  ====================================  1. install  2. upgrade  3. clean  4. help  5. exit  Please select an operation: 1 |

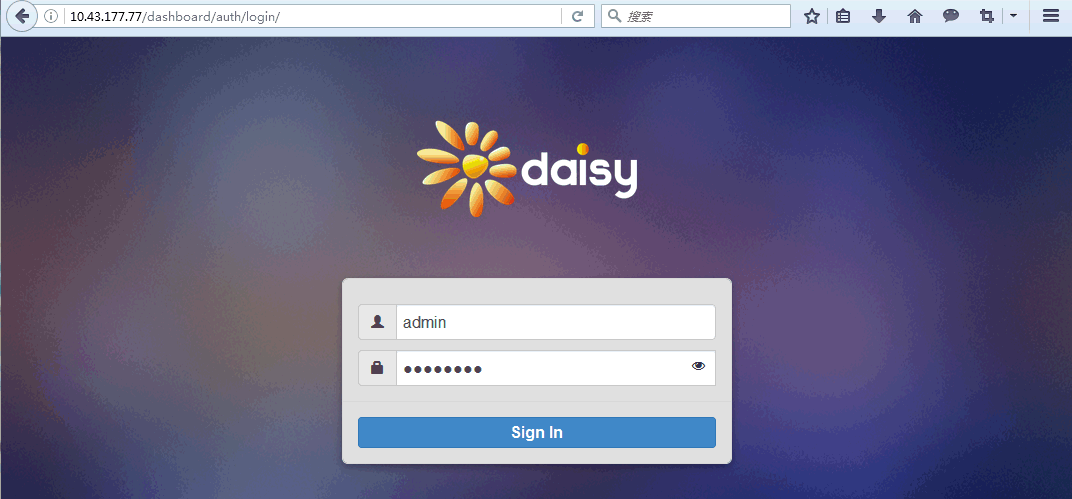
Choose 1 to install Daisy and wait until the installation finished.

## 4. Deploy OpenStack using Daisy

1) Log in to the Daisy Web UI running on your jump server

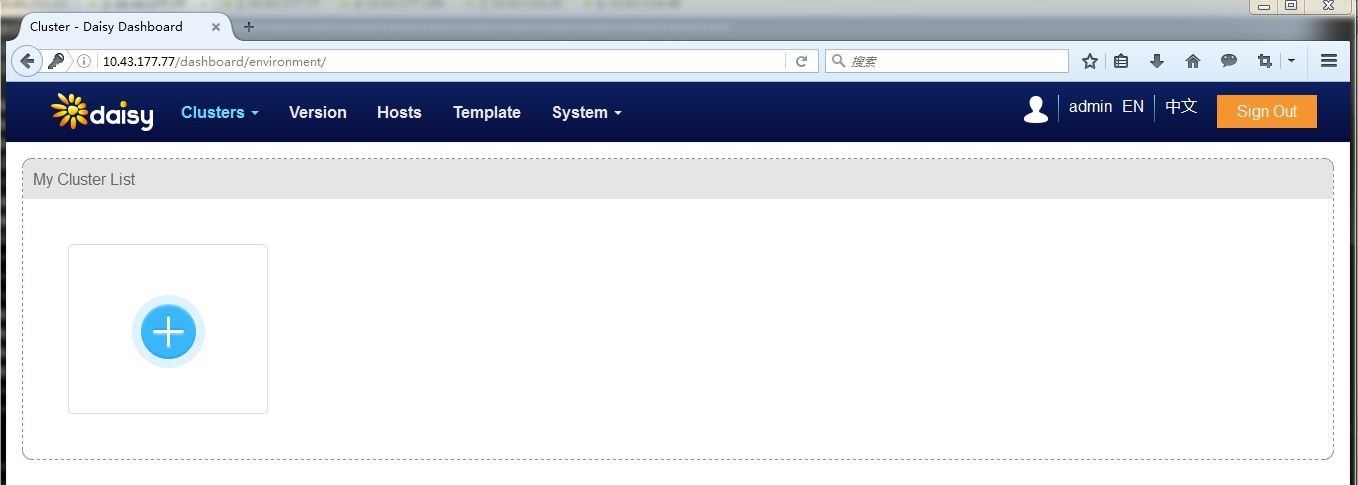
Choose your preferred internet browser and access the Web UI by just simply using your jump server’s IP as the URL. The following example takes 10.43.177.77 as the IP of jump server.

Please log in with user name “admin” and uses “keystone” as the default password.

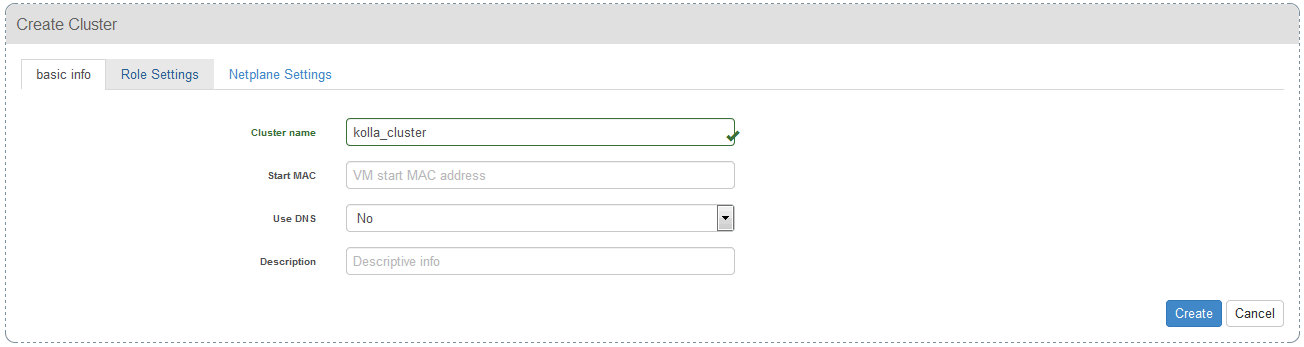


2) Create a new Cluster

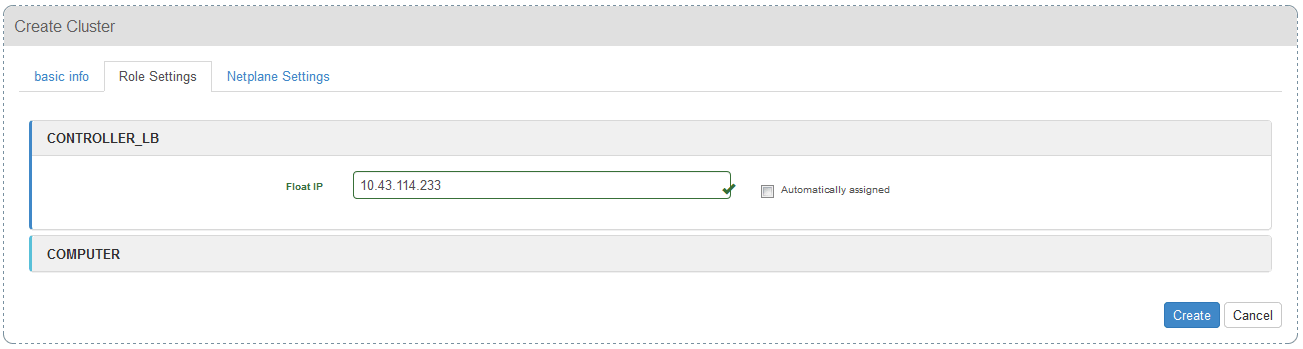
Create a new Cluster by clicking the big “+” icon in the “My Cluster List” frame:



Literally setup the “basic info”, “Role Settings” and “Netplane Settings”. As a demo, please choose “No” for “Use DNS”.

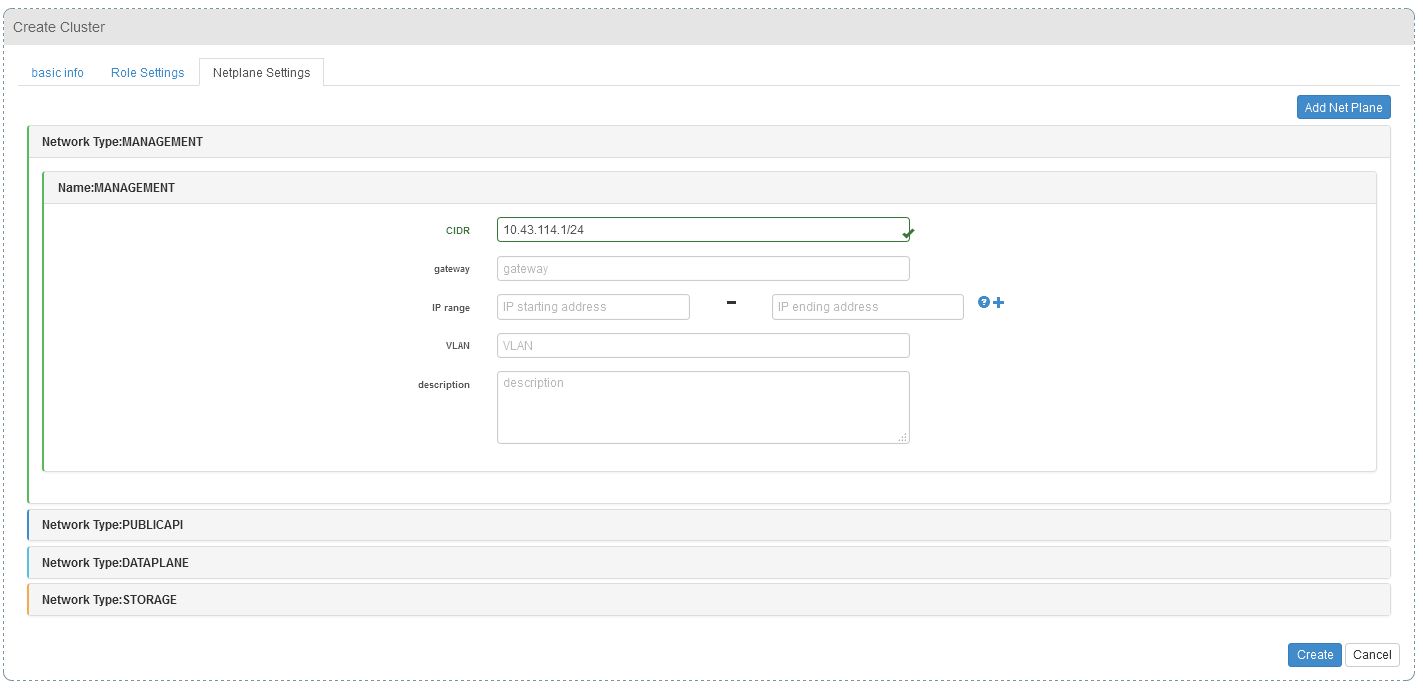


Each host has to be assigned to at least one role, such as to be controller node or a compute node. Currently we only support two roles, CONTROLLER\_LB and COMPUTE. The suffix \_LB for controller means that Daisy will apply load balance high-availability resolution(by using keepalived) on those controller hosts which role are CONTROLLER\_LB. The Float IP (a.k.a. VIP) is an unused IP on your network that will float between the hosts running keepalived for high-availability.



Currently, only three different net planes are supported by Daisy, MANAGEMENT is for accessing the VIP from internet, PUBLICAPI os for vm to access the internet and DATAPLANE is for the east-west tenant network flow.

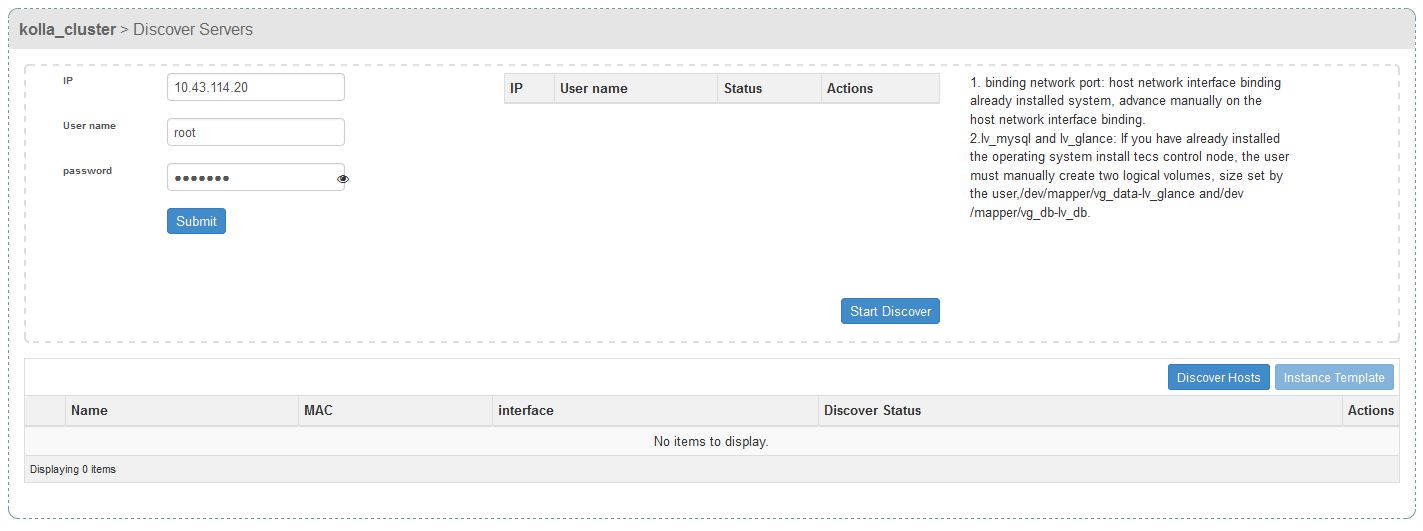
Do not use STORAGE netplane for this demo.



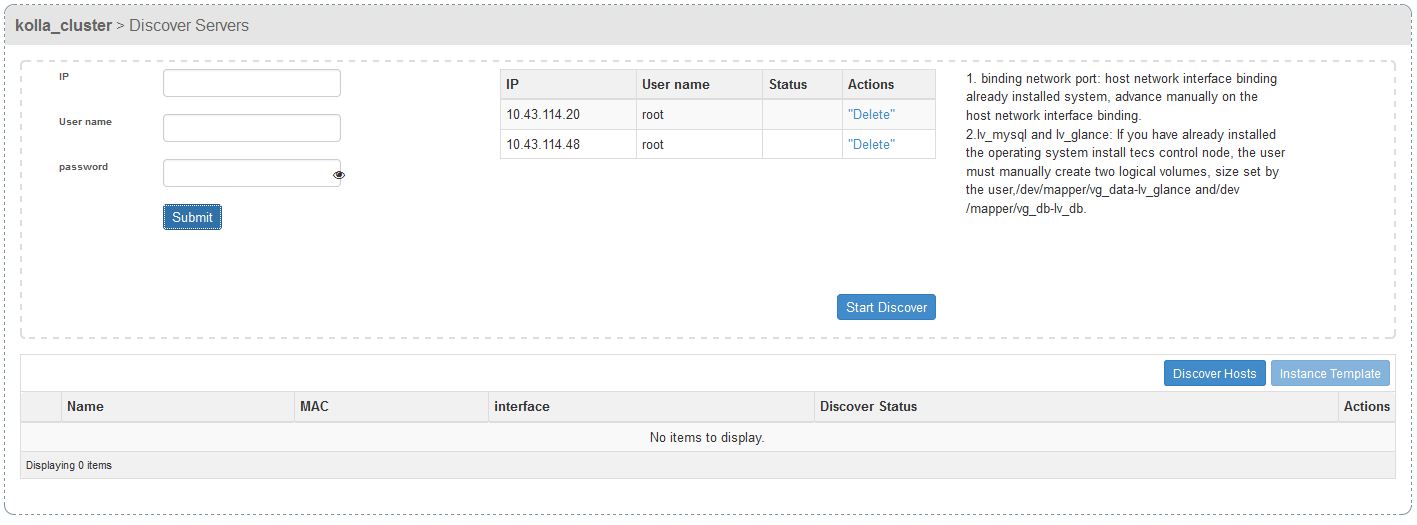
Click “Create” button to create the cluster and the UI will change to “Discover Servers” page automatically.



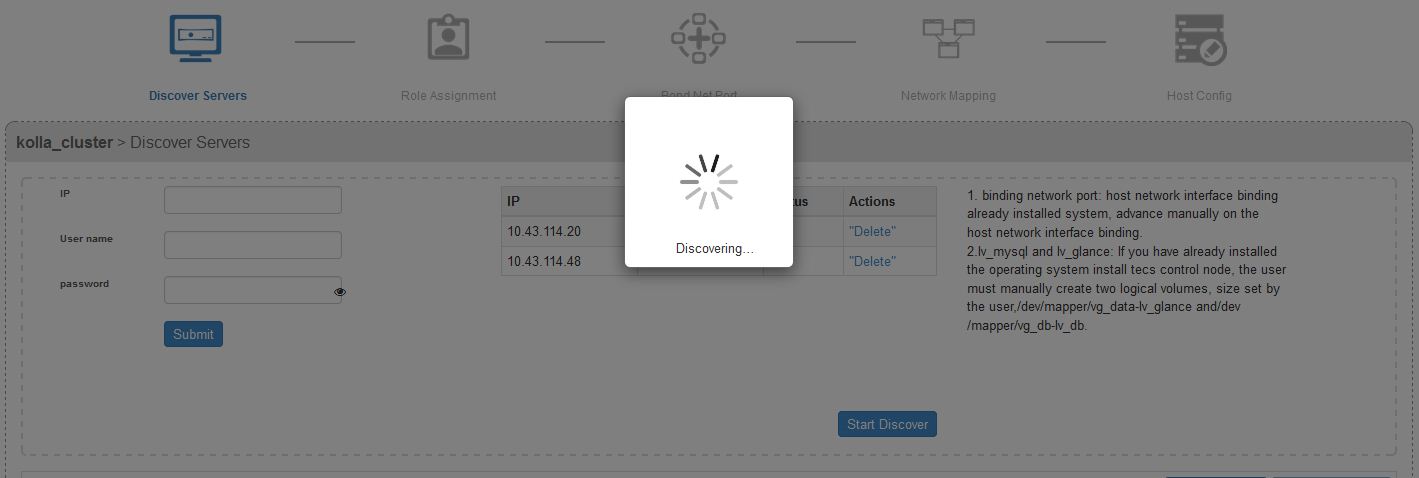
After clicking the “Discover Hosts” Button, input your target host’s IP (for example 10.43.114.20), ssh username (root) and password, then by clicking the “Submit” button.



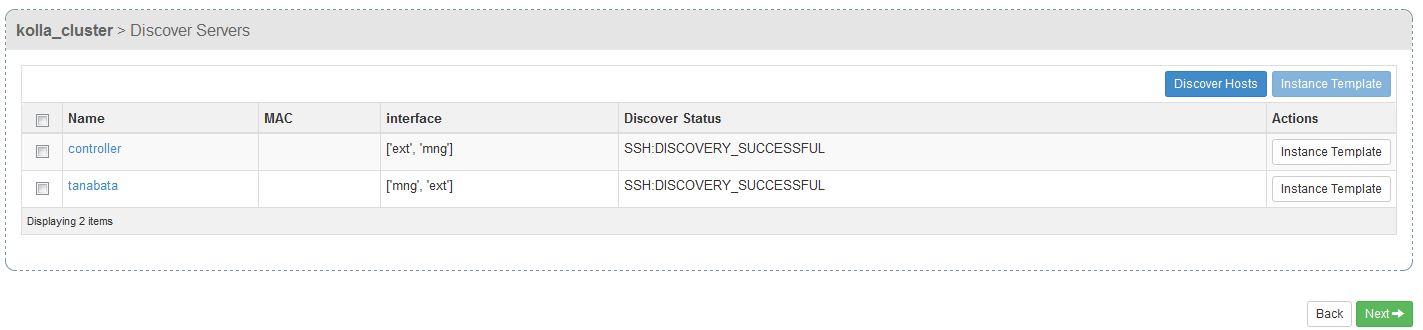
After clicking the “Submit” button, the to-be-discovered hosts information will be displayed in a table. Here is the final table which contents two to-be-discovered target hosts that according to your input.



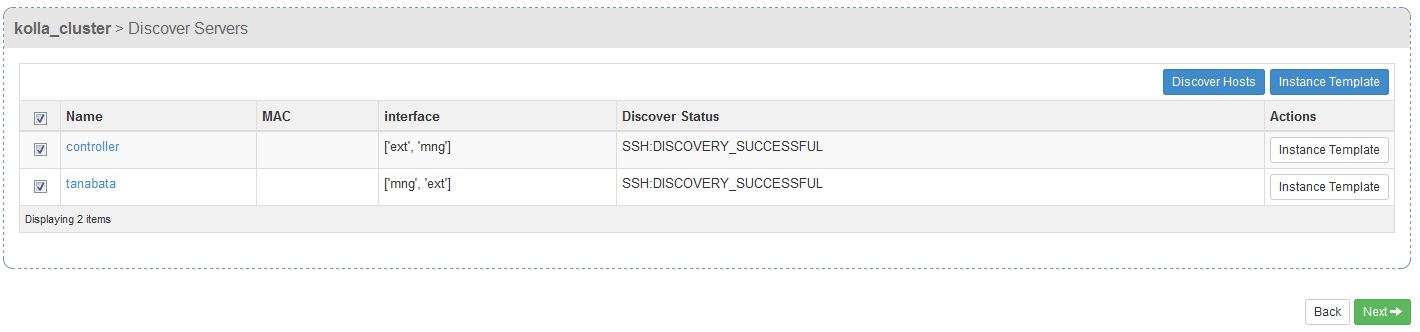
Click the “Start Discover” button, the non-PXE discovery process will start for those target hosts.



The discovery process will be completed in about a minute and the result should be like the following. If anything failed, please try to click the “Start Discover” again.



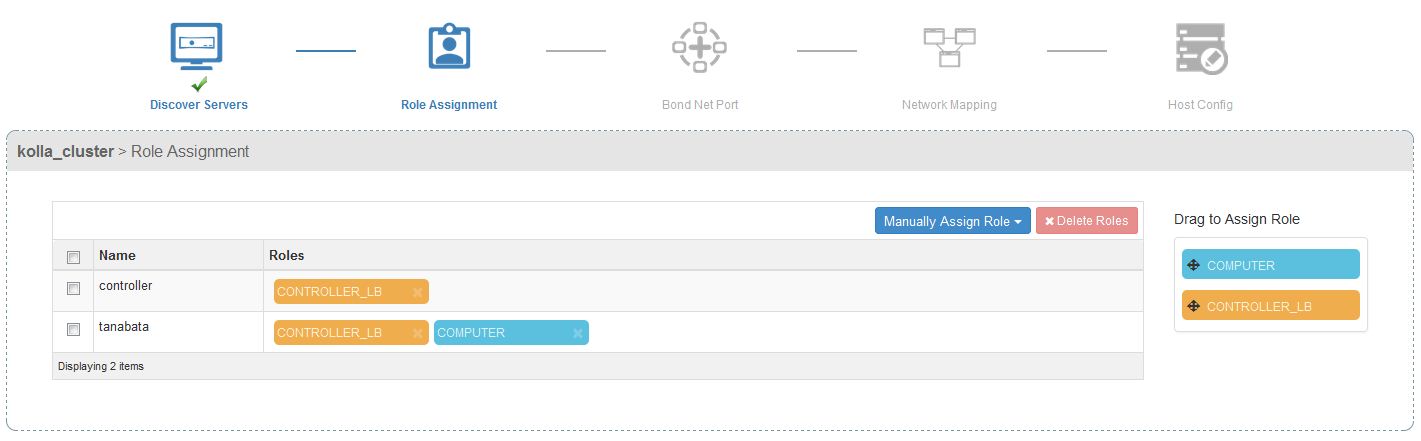
You can decide which host can join the cluster by checking the check box of a host. Like the following, we decide to add both hosts into our new cluster. Click “Next” after you made the decision.



The UI now switch into “Role Assignment” stage where you can assign roles to each host. In this example, we assign CONTROLLER\_LB role to both of our hosts, but assign “COMPUTER” role to tanabata only.

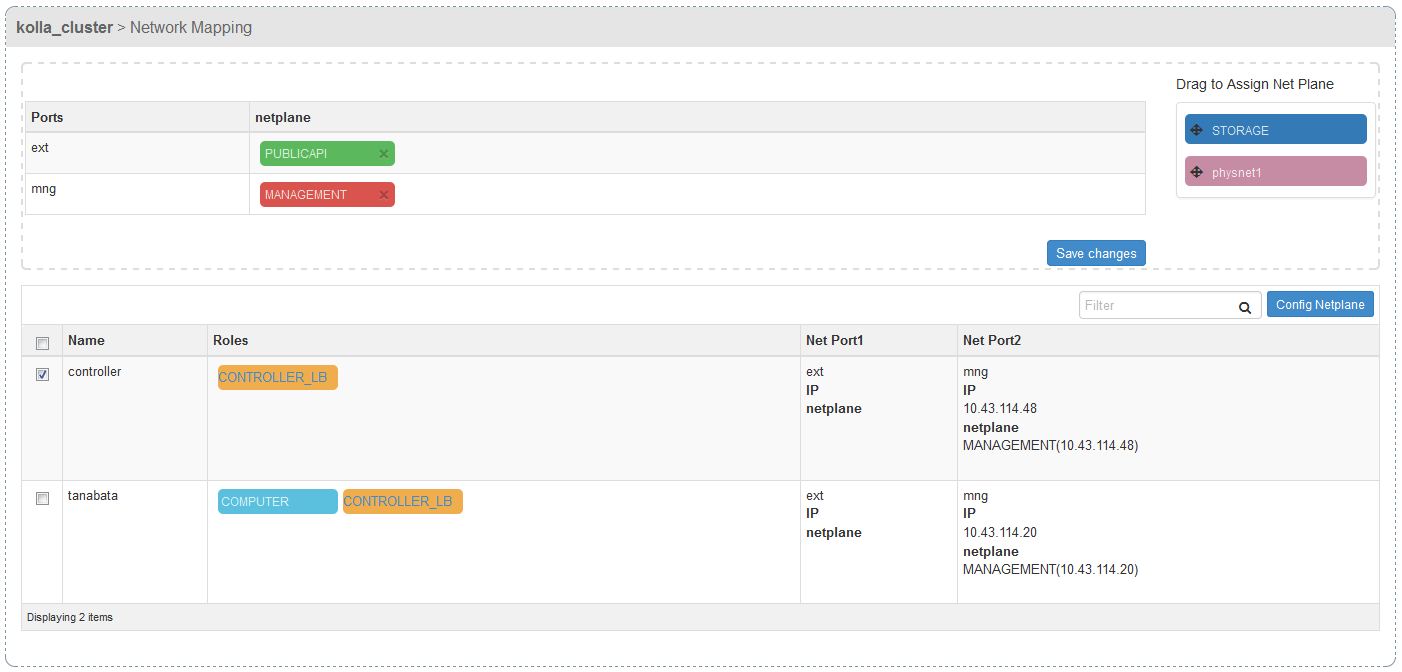
|  |
| --- |
| Note: Actually “COMPUTER” role should be name as “COMPUTE” which is more precise. |

You can simply drag the role icon and drop it on the host to make the assignment.

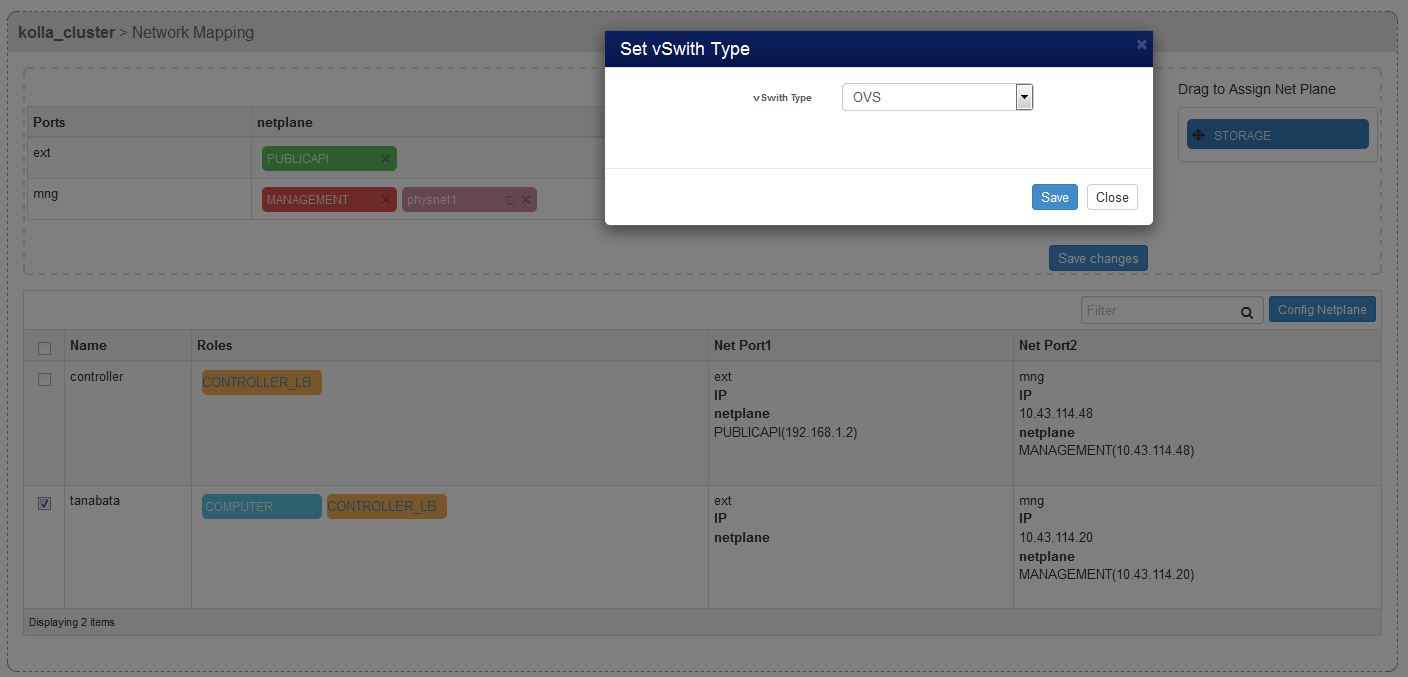


After assigning role, we skip the “Bond Net Port” step and do “Network Mapping” configuration for each host. By checking the check box of a specific host, a table will appear for you to map network plane for each port(or interface) of the host. In this example, “ext” is used as the PUBLICAPI port and “mng” is configured to be the MANAGEMENT interface. “ext” and “mng” is the interface name on the host. You may have those interfaces named differently such as “eth0” or “enp9s1”.

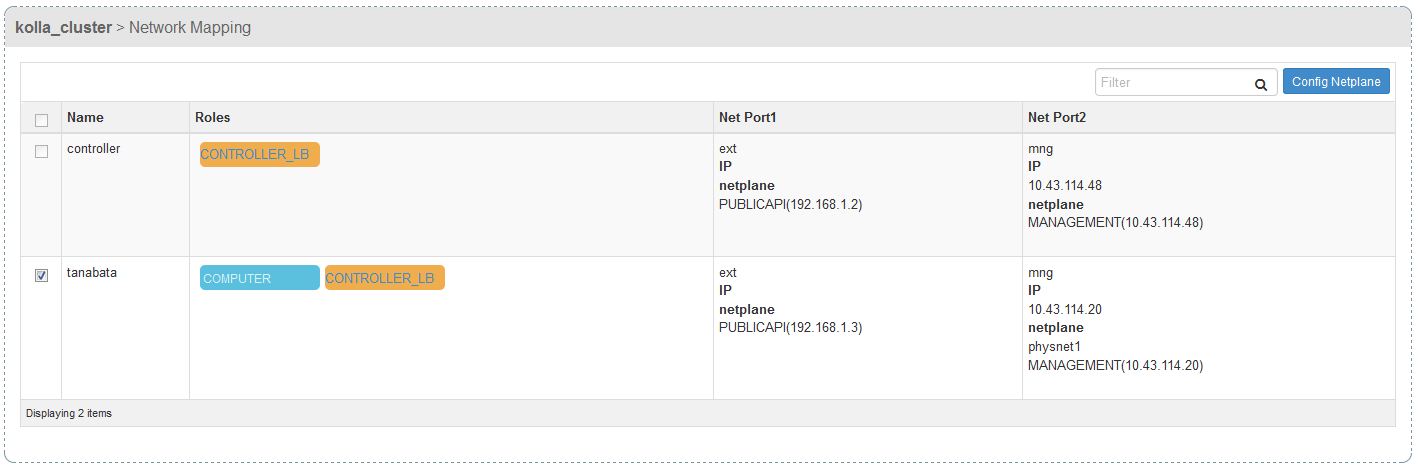
For CONROLLER\_LB hosts, the PUBLICAPI and MANAGEMENT net planes must be configured.



For CONROLLER\_LB hosts, the physnet1(DATAPLANE) and MANAGEMENT net planes must be configured. Please choose OVS(Open VSwitch) as the Virtual switch type when configuring the physnet1(DATAPLANE).

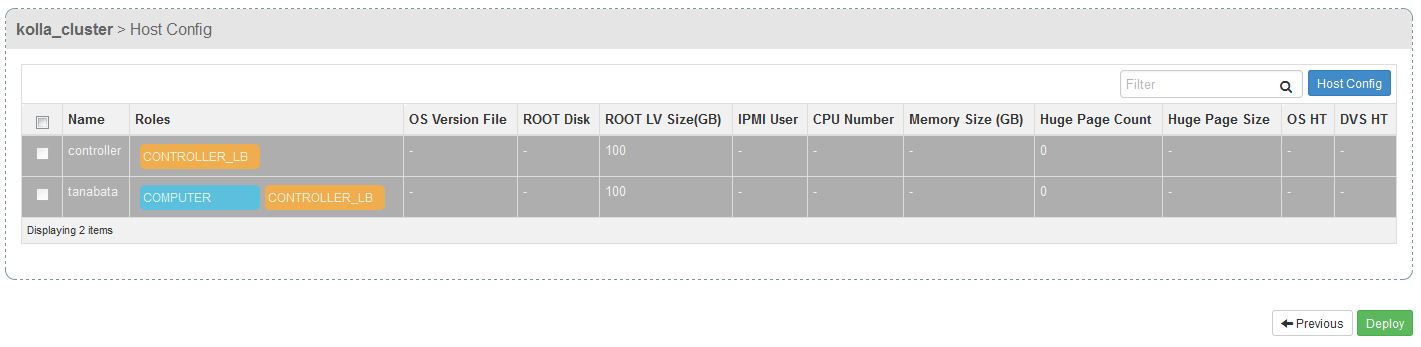


Here is the final network mapping configuration for this example.

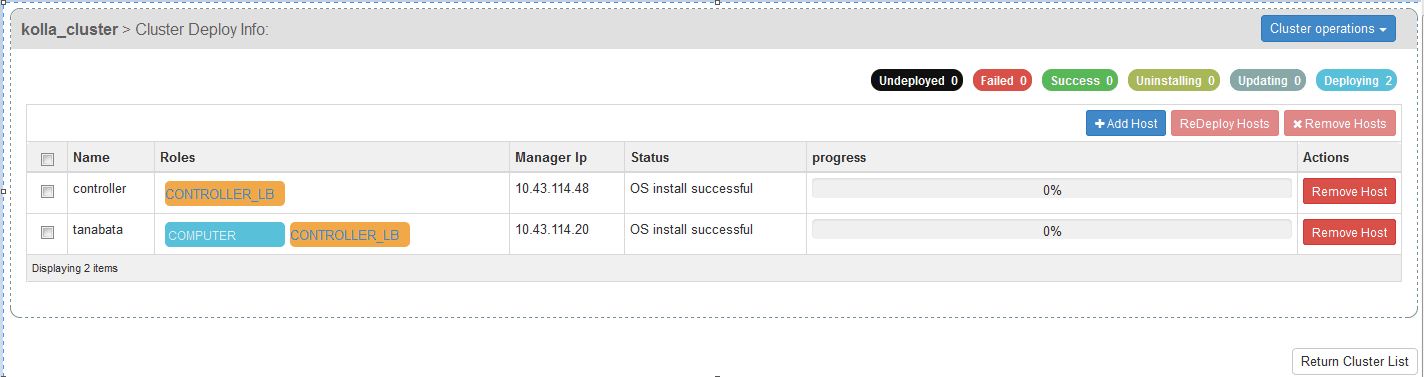


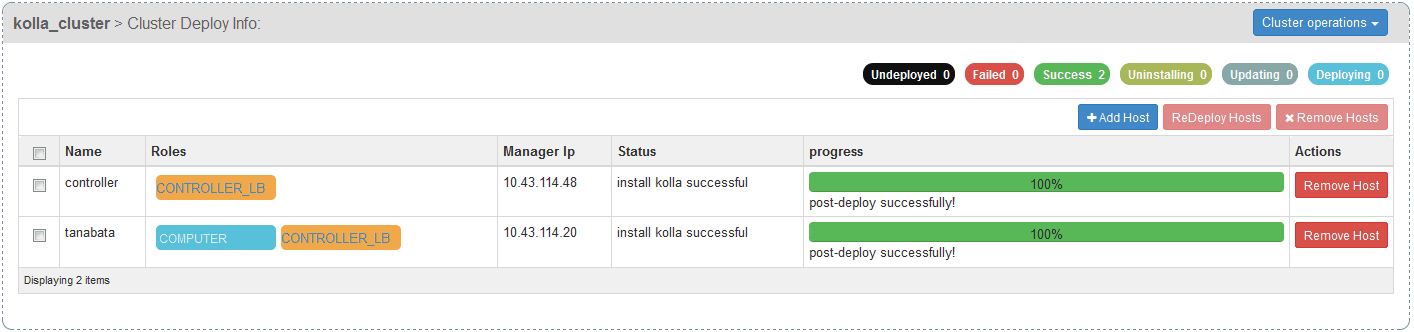
|  |
| --- |
| Note: Remind again, Currenlty, Daisy restricts that all interfaces for a specific network plane must have exactly the same name. As in this example, all “ext” are used for PUBLICAPI, all “mng” are used for MANAGEMENT. |

Finally, click the “Deploy” button on the “Host Config” page to start the deploying process. There is nothing we can do on the “Host Config” page currently.



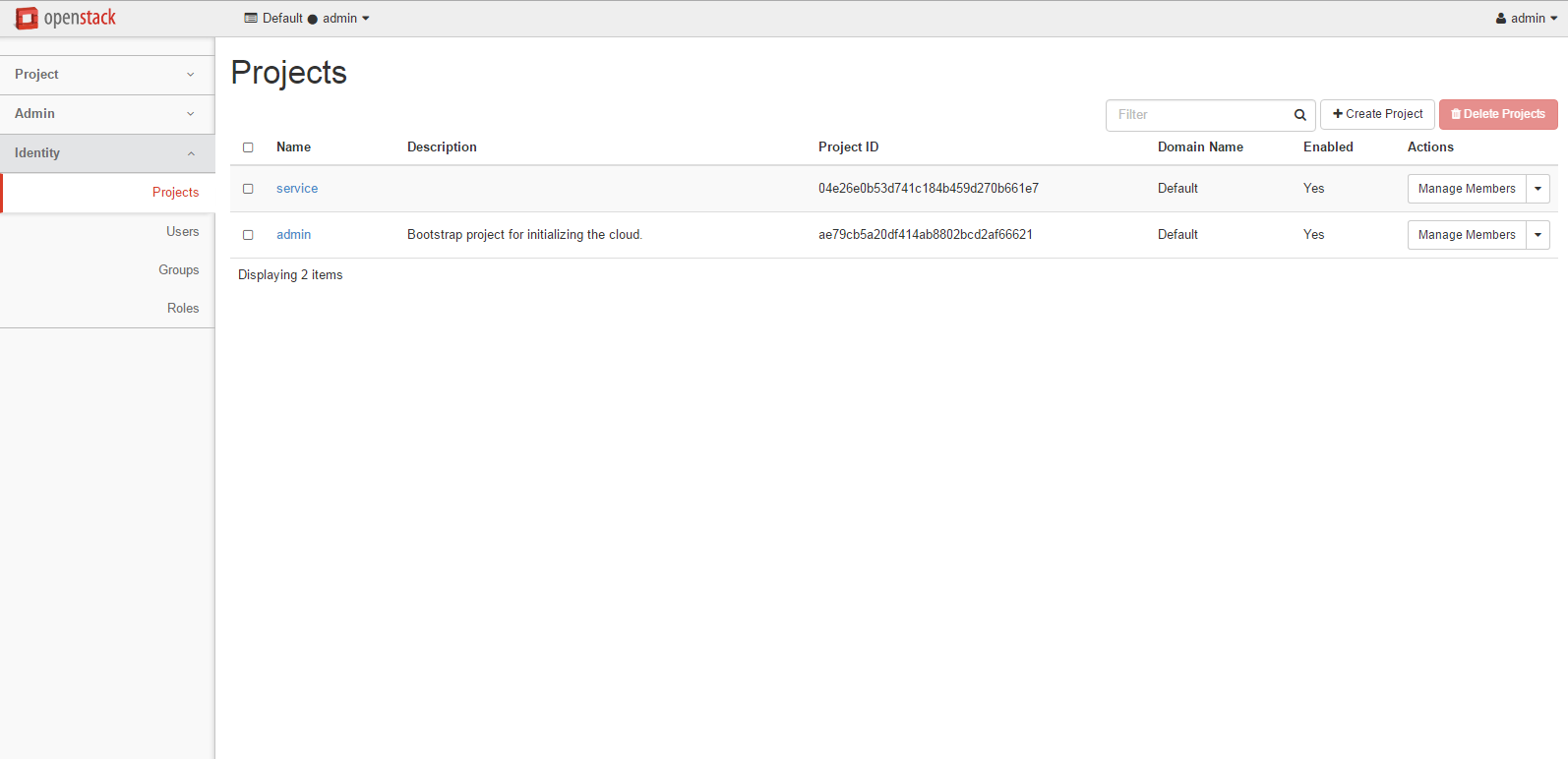
You can monitor the deploying progress on the “Cluster Deploy Info” page. Wait until the progress to be 100%, then your OpenStack cluster is fully prepared.





Test The Deployed OpenStack Environment like what you usually did before. Please use user name “admin” and password “keystone” to log in to OpenStack Horizon.

|  |
| --- |
| Note: If you can not log in to Horizon, Please see Appendix B |



## Appendix A. How to change network interface name on CentOS7

1) Restore old naming convention for linux kernel

First, let's disable the predictable naming rule. For that, you can pass "net.ifnames=0" kernel parameter during boot. This is achieved by editing /etc/default/grub and adding "net.ifnames=0" to GRUB\_CMDLINE\_LINUX variable.

Then run this command to regenerate GRUB configuration with updated kernel parameters.

#grub2-mkconfig -o /boot/grub2/grub.cfg

2) Change udev rule for naming nic device to be on the mac address basis.

create /etc/udev/rules.d/70-persistent-net.rules

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?\*", ATTR{address}=="ec:88:8f:ee:5b:03", ATTR{type}=="1", KERNEL=="eth\*", NAME="ext"

SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?\*", ATTR{address}=="00:19:21:b8:17:39", ATTR{type}=="1", KERNEL=="eth\*", NAME="mng"

3) modify ifcfg-\* file for this device to use the new name.

HWADDR=ec:88:8f:ee:5b:03

TYPE=Ethernet

BOOTPROTO=none

NAME=ext

UUID=af2cf27f-3ced-4f30-b49e-88e46f41365f

ONBOOT=yes

NM\_CONTROLLED=no

4) rename ifcfg-\* file to new name

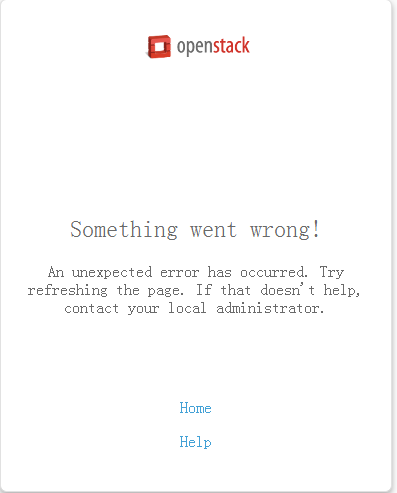
mv ifcfg-enp1s2 ifcfg-ext

5) Finally reboot machine.

6) Done.

## Appendix B. Cannot log in Horizon after deployed

If you cannot log in to Horizon and get error like below when trying to log in, it is probably caused by a bug in the python-django-1.8.13..



This issue was solved in python-django-1.8.14 by patch : <https://github.com/django/django/commit/c8d2120b06e1b82cb0db896aa0f9767b2f14256c> which has not been included in our Docker container image. We are still working on upgrading our Docker container image. So for now, please apply this patch into the container name “horizon” manually and try log in again.