



Stacki Tutorial #1: The Basics of Networking in Stacki

Introduction

This series of tutorials is designed to offer short, easy to follow steps to customize your use of Stacki. Each tutorial will offer one set of functionality, and fully explain how that functionality can be achieved in Stacki - the world's fastest open source installer.

Summary

Stacki, as it is initially installed, will quickly install as many machines as you need with intelligent default settings. But it is rare that your environment requires default settings, so the tutorial series will teach you how to customize the installed servers to suit your environment. Since everyone needs networking, and every environment is different, we decided this was a good place to start our tutorial series.

What you will learn

Today's tutorial will walk you through creating networks in the Stacki server, and adding NICs to backend servers that make use of those networks. There is a lot more of networking in Stacki, but we'll get to those other items in future tutorials.

Pre-requisites

This tutorial assumes:

1. You have downloaded Stacki from www.stacki.com
2. You have followed the directions [here](#) to install a Stacki server.
3. You have followed the directions [here](#) to install backend server(s).
4. You have access to a command line on the Stacki server.
5. You have information about the network(s) you want to add to Stacki.

Naming hosts/host groups

Most stacki commands take the name of a host. This can be in any of the following forms:

hostname - Actual host DNS name host-0-0.local – This machine.

hostname* - All hosts whose name match the pattern. **Host*** - All machines whose names start with host.

appliance name - all machines of that appliance type backend – All machines of appliance type backend.

Let's Get Started!

Log in to the Stacki server, and type the following command:

```
stack list network
```

You should see output like this, assuming you've not changed anything yet:

```
[root@stackidon ~]# stack list network
NETWORK      SUBNET      NETMASK      MTU      DNSZONE SERVEDNS
private:     10.1.1.0    255.255.255.0 1500    local      True
public:      192.168.0.0 255.255.255.0 1500    stackiq.com False

[root@stackidon ~]#
```

These are all of the networks Stacki knows about by default. The private network is used for installations, the public so that you can log into the Stacki server remotely, and get updates in the future. This tutorial will show you how to create new networks, and to modify the “public” network of these. Since “private” is used to do installations, we’ll not do anything with that one until a later tutorial, when we have more information about what can be done with networks.

By default, all servers exist on the private network. To see this, type the following...

```
Stack list host interface backend*
```

You should see output like the following:

```
[root@stackidon ~]# stack list host interface backend*
HOST          SUBNET  IFACE  MAC          IP          NETMASK MODULE NAME          VLAN OPTIONS CHANNEL
backend-0-0: private eth0    08:00:27:6e:88:08 10.1.1.253 255.255.255.0 -----
backend-0-0 ----                -----
backend-0-1: private eth0    08:00:27:3a:e0:e4 10.1.1.254 255.255.255.0 -----
backend-0-1 ----                -----

[root@stackidon ~]#
```

Notice that the interface has a lot of information on it. We’ll get to that in a future installment.

To add a network to the list of networks, simply type:

```
stack add network protected 10.10.10.1 255.255.255.0
```

As you likely guessed already, ‘protected’ will be the network name, ‘10.10.10.1’ the base address, and 255.255.255.0 the netmask – meaning the network covers all of the 10.10.10.* network.

And you can check that the network exists by typing:

```
stack list network
```

again.

```
[root@stackidon ~]# stack add network protected 10.10.10.1 255.255.255.0
```

```
[root@stackidon ~]# stack list network
```

NETWORK	SUBNET	NETMASK	MTU	DNSZONE	SERVEDNS
private:	10.1.1.0	255.255.255.0	1500	local	True
protected:	10.10.10.1	255.255.255.0	1500	protected	False
public:	192.168.0.0	255.255.255.0	1500	stackiq.com	False

```
[root@stackidon ~]#
```

Now Stacki knows about the protected network, and we can tell backend machines to attach NICs to it. Let's do so. First, add the interface to the stacki database:

```
stack add host interface backend-0-0 iface=eth1 ip=10.10.10.100 subnet=protected name=prot1
```

This tells Stacki to add eth1 to server backend-0-0, give it ip 10.10.10.100, attach it to subnet protected, and name it prot1 locally.

Now if you type:

```
stack list host interface backend-0-0
```

You'll see our new interface in the list, but if you go to a terminal on backend-0-0, you will not find it listed with ifconfig. That is because Stacki knows about it, but hasn't updated the server yet. To update the server type:

```
Stack sync host network backend-0-0
```

It will tell the host about its new interface, and bring the interface up. If you go to the backend servers' terminal and type ifconfig, you will see eth1 is now there, and it is fully functional (assuming you put it on the right network with a valid IP).

```
[root@stackidon ~]# stack add host interface backend-0-0 iface=eth1 ip=10.10.10.100 subnet=protected name=prot1
```

```
[root@stackidon ~]# stack list host interface backend-0-0
```

SUBNET	IFACE	MAC	IP	NETMASK	MODULE	NAME	VLAN	OPTIONS
private	eth0	08:00:27:6e:88:08	10.1.1.253	255.255.255.0	-----	backend-0-0	----	-----
protected	eth1	-----	10.10.10.100	255.255.255.0	-----	prot1	----	-----

```
[root@stackidon ~]# stack sync host network backend-0-0
```

```
[root@stackidon ~]#
```

Now we can check on the host...

```
[root@stackidon ~]# ssh backend-0-0
Last login: Wed Jul 1 14:00:06 2015
[root@backend-0-0 ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:6E:88:08
          inet addr:10.1.1.253  Bcast:10.1.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:308 errors:0 dropped:0 overruns:0 frame:0
          TX packets:282 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:48737 (47.5 KiB)  TX bytes:38985 (38.0 KiB)
eth1      Link encap:Ethernet  HWaddr 08:00:27:D0:67:B5
          inet addr:10.10.10.100  Bcast:10.10.10.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:55 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5148 (5.0 KiB)  TX bytes:168 (168.0 b)
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:780 (780.0 b)  TX bytes:780 (780.0 b)

[root@backend-0-0 ~]#
```

That's it for this tutorial! You now can access a variety of networks in your environment simply by defining the network and interfaces on the command line.

Stacki Tutorial #1: The Basics of Networking in Stacki

© 2015 StackIQ, Inc. | www.stacki.com