



## Stacki Tutorial #6: Server Management With Remove

### Introduction

In the course of managing servers, things inevitably change. Routes, SSH keys, even hosts come and go in the network, and whatever management systems are in use need to be able to incorporate these changes.

Stacki implements the ability to permanently remove items that are not longer relevant with the remove commands. This tutorial will go over some of the more frequently used commands, and show you how to use them in managing the overall system.

Many of these commands will appear in various places amongst other task-focused tutorials, this tutorial is aimed at the “cleanup” task, and thus covers them together.

### 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.

### Pre-requisites

This tutorial assumes:

1. You have downloaded Stacki from [www.stacki.com](http://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 at least one server that you can use remove to make changes to – including removing and re-adding the server. If you do not have one currently configured, try the “install” tutorial to get one up and running.

## Let's Get Started!

We want to start large and then get to the smaller ones. While this makes the tutorial more difficult to follow along with, it does allow us to hit all of the warnings/advisories that go along with removing servers, and we can help you get set up such that accidental removal is a problem that can easily be recovered.

If you execute *stack list host*, you should get a list of the hosts installed on the system. It should look something like this:

```
[root@donrh7 ~]# stack list host
HOST          RACK RANK CPUS APPLIANCE DISTRIBUTION RUNACTION INSTALLACTION
donrh7:       0    0   1  frontend default      os      install
backend-0-0:  0    0   1  backend  default      os      install
cocoamaster357: 0    1   1  backend  default      os      install
[root@donrh7 ~]#
```

An explanation of removal, and why it should be done with caution.

Removing a host from Stacki deletes the host and all records about the host from the database. This does not mean it wipes the server being removed, in fact it specifically does not change the server itself or what is installed on the server. What it does do is stop managing the host, setting it off on its own like any hand-configured server.

This is in keeping with the philosophy of the Stacki team – that destruction should be limited and expected, even when requested. So repurposing the machine will mean doing whatever it needs by hand or with some other tool.

One thing to be warned of, removing a host from the database seems like a simple thing you can do with impunity because Stacki makes it so easy to re-add the server. The thing is that any customizations that you made to the install via the command line will have to be redone if you ever remove and re-add a server.

If using spreadsheet configuration, reapplying the spreadsheet will configure things correctly again, and then the server can be booted and reinstalled to bring it back under management. But things done on the command line – like adding routes – will have to be redone via the command line.

What we recommend is to keep all of those command-line changes together in a script that can then be checked into version control. If the number of machines being managed is larger, it might be possible that a site-specific configuration would be a better way to add routes and perform other multi-machine operations. See the Wiki for information on how to set up and manage site-specific customizations.

Now that we've covered the background and the warnings, we'll get to the easy part – removing a server from Stacki. Simply enter the command *stack remove host [hostname]*  
There is no output when this command completes successfully.

```
root@stackihw ~]# stack list host
```

HOST	RACK	RANK	CPUS	APPLIANCE	DISTRIBUTION	RUNACTION	INSTALLACTION
stackihw:	0	0	1	frontend	default	os	install
backend-0-1:	0	1	4	backend	default	os	install
backend-0-2:	0	2	1	backend	default	os	install
puppet:	0	3	4	backend	default	os	install

```
[root@stackihw ~]# stack remove host backend-0-2
```

```
[root@stackihw ~]# stack list host
```

HOST	RACK	RANK	CPUS	APPLIANCE	DISTRIBUTION	RUNACTION	INSTALLACTION
stackihw:	0	0	1	frontend	default	os	install
backend-0-1:	0	1	4	backend	default	os	install
puppet:	0	3	4	backend	default	os	install

```
[root@stackihw ~]#
```

So easy it's dangerous.

Let's say that you want to keep the system as it is currently configured, but remove it from management, and keep your options open. There is one way this can be accomplished with minimal change that does not delete the server from the database.

Normally when a server hits DHCP, Stacki looks in the database to see what to tell it to do. If the boot action in the database is 'os', then it runs the OS from disk. If the boot action is 'install', then it (re)installs the server.

```
[root@stackihw ~]# stack list host boot
```

HOST	ACTION
stackihw:	-----
backend-0-1:	install
puppet:	os

```
[root@stackihw ~]# stack remove host boot backend-0-1
```

```
[root@stackihw ~]# stack list host boot
```

HOST	ACTION
stackihw:	-----
backend-0-1:	-----
puppet:	os

With the “boot” action cleared, whenever the server in question boots up, Stacki will not respond to the DHCP request, and assuming you have BIOS configured to try hard disk, it will boot from hard disk. In this manner a machine can be temporarily repurposed and wiped/installed with a completely different set of software without removing it and all of its customizations from the Stacki database.

## Modifying Network Routes

Network routing changes over time, and the networks that a machine needs access to change also. Perhaps a machine is used as an SQL submission point, and later the submission point is moved, or a machine due to a single application is required to have access to the Internet through the corporate network, but once that application is moved or the method of gaining access changes, routing needs change.

It is relatively simple to clean up routing issues like these in Stacki. Once you’re certain that the route is no longer needed (as opposed to changed, which you can explore the options for in the second [Managing Networking Tutorial](#))

First, determine where the route is defined.

If it is applied to all nodes, and you want to remove it from this global approach, use the following:

```
stack remove route [address=string]
```

Where address is the identifying address in the routing table.

To remove the route from only a single host, use

```
stack remove host route [hostname] [address=string]
```

That will remove the route from the Stacki database. Let’s take a look.

```
root@stackidon ~]# stack remove host route backend-0-0 address = 0.0.0.0
[root@stackidon ~]# stack sync host network backend-0-0
```

Note that you need to issue a *stack sync host network [hostname]* to propagate changes out to the server.

## Removing Attributes

The [Attribute Tutorial](#) shows how to list and modify attributes. Because deleting attributes is a very dangerous thing, we chose to include Remove attributes here, rather than in the other tutorial.

The problem with attributes is that there are a ton of them. If you do `stack list attr`, you will see a large list, and while there is some overlap, `stack list host attr [hostname]` will give you a mostly different list.

The risk with attribute deletion is that Stacki needs most of the attributes that it defines, and you should only delete them with a thorough understanding of what the attribute does within the Stacki system.

For attributes that your organization has created (with `stack add attr` and the host/appliance versions of this command), deletion is a different matter. Presumably, those who created the attribute know when it is safe to delete it.

For those attributes, simply type one of the following commands:

```
stack remove attr attr=[attrname]
stack remove host attr [hostname] attr=[attrname]
stack remove appliance attr [appliance_name] attr=[attrname]
```

They simply remove the attribute from the database. This usually – though not always – means the removal will have a practical impact the next time you build and install an effected server.

We stress again, you can make a mess of your installation by removing random attributes. Make certain removing an attribute is the best solution to your problem, and that you track your changes so they can be reversed if things seem wrong. Finally, know what the attribute is for, and what the impacts of deleting the attribute will be.

While this document is for StackIQ Boss (Cluster Manager was Boss' name before it got cool), it does offer descriptions of attributes, many of which the two systems have in common.

[PDF CM User Guide.](#)

## Summary

Using the various remove commands your systems can be cleaned up relatively easily, preventing cruft from building up in the management system, and allowing for unique customizations based on removal as well as addition. We just recommend that caution is used when removing values from the database.