



Stacki Tutorial #5: Working With Attributes

Introduction

Using Stacki to install a whole collection of servers is as easy as installing Stacki on one of the servers, running insert-ethers, and booting other servers on the network.

But of course, Stacki has to make all sorts of assumptions for this to work. While you'll get a working set of systems, they are not customized to your environment.

In this tutorial, we'll cover Attributes, one of the key pieces of the system that allows massive customization of server installations. Taken with [Tutorial #2](#) attributes can allow for a massive level of customization.

We have already used attributes in the previous tutorials, this tutorial is aimed at showing just how much control is available, and providing a link to the list of attributes with their meanings.

What you will learn

This short tutorial will focus on disabling the host firewall on backend servers. Specifically, when done with this tutorial, you will be able to:

- List attributes globally, by appliance, and by host.
- Research attributes and their meanings.
- Set the value of an attribute at each level.
- Make certain the attributes take effect.

There are many attributes available, and this quick-start tutorial is not intended to offer a comprehensive list. That list exists already, and can be found in [this document](#).

How Stacki organizes attributes

Attributes in Stacki are discrete bits of data relevant to a server, all servers of a given appliance type, or the overall system. Each type of attribute can be modified independently, and they are amalgamated to give a complete picture of the server to be installed. Some bits of data are directive indicators for Stacki, and some are actual data to be used during Kickstart generation to customize the installation.

(Continued...)

To make that last bit clear, here are two attributes at the host level that illustrate the difference. To get the value of an attribute, we use one of the following commands:

```
stack list attr
stack list appliance [appliance]
stack list host [hostname] attr
```

Additionally, each of these commands takes an optional argument `attr=[attribute name]` to return just a single value.

We have used the *nukedisks* attribute in the installation tutorial. Here it is, along with the *hostname* attribute.

```
[root@donrh7 ~]# stack list host attr backend-0-0 attr=nukedisks
HOST          SCOPE ATTR      VALUE SOURCE
backend-0-0:  ----- nukedisks false H

[root@donrh7 ~]# stack list host attr backend-0-0 attr=hostname
HOST          SCOPE ATTR      VALUE SOURCE
backend-0-0:  ----- hostname backend-0-0 I
```

Notice that the first value – *nukedisks* – is set to false. That means the next time we re-install backend-0-0, the disks will not be reformatted except the primary OS partition.

Notice with the second value – *hostname* – the value is set to backend-0-0, which is the host name that will be assigned the next time we sync config or reinstall.

A thing to note, if you wish to obtain a collection of values – like all of the kickstart values, use:

```
stack list attr | egrep Kickstart
```

Changing Values

After retrieving values and determining which ones are of interest, the next step is to modify them. This is easy in Stacki, as long as you know the attribute name, and whether you need it to be a global, appliance, or host change.

```
Global:
stack set attr attr=[name] value=[new setting]

Appliance:
stack set appliance attr [appliance] attr=[name] value=[new setting]

Single host:
stack set host attr [hostname] attr=[name] value=[new setting]
```

Examples:

Note: The following are examples only. Immediately after changing them, we reverted them to original values to maintain stability on demo systems. Use caution when choosing which values to modify.

```
[root@donrh7 ~]# stack set attr attr=firewall value=false
[root@donrh7 ~]# stack list attr attr=firewall
SCOPE ATTR      VALUE SOURCE
----- firewall false G
```

```
[root@donrh7 ~]# stack set appliance attr backend attr=managed value=false
[root@donrh7 ~]# stack list appliance attr backend
APPLIANCE SCOPE ATTR      VALUE
backend:  ----- dhcp_filename pxelinux.0
backend:  ----- dhcp_nextserver 10.1.1.1
backend:  ----- kickstartable yes
backend:  ----- managed false
backend:  ----- node backendv
```

```
[root@donrh7 ~]# stack set host attr backend-0-0 attr=nukedisks value=true
[root@donrh7 ~]# stack list host attr backend-0-0 attr=nukedisks
HOST      SCOPE ATTR      VALUE SOURCE
backend-0-0: ----- nukedisks true H
```

In all of these examples, we modified the default values, and then showed the result. Notice that the appliance “show the result” did not filter the results by attribute name. Since the list of attributes associated with the appliance is so small, this was just an example of using the more general list.

It's about the sync.

One thing that needs to be pointed out is that none of these changes take effect until Stacki has been told to implement them. That's why our examples can change values at will and not adversely impact the overall set of systems. Because the changes you make on the Stacki command line are to the database, so the Stacki server knows what you want the systems to look like. Until those changes are pushed out to the server they impact, the servers will continue to operate as they did before the changes.

So how does an update get forced? There are two ways.

The most certain way to guarantee that the host and Stacki's database are in sync is to reinstall the host. It only takes a minute, and when Stacki generates the Kickstart files, it will use the current database values for everything.

For most changes, the various sync commands will take care of pushing the changes out. Primarily,

```
stack sync config [hostname]
```

Also useful are the firewall and networking sync commands

```
stack sync host firewall [hostname] restart=[true|false]
```

```
stack sync host network [hostname] restart=[true|false]
```

For both of these commands, restart defaults to true.

There are some attributes in global that have no direct relevance to hosts – like the public IP address of the Stacki server. Syncing will have no effect for these items. Others are more directives than attributes – like nukedisks. It is a directive to wipe partitions on the next install, and syncing the config will have no effect for those items either – though a reinstall of the host in question will kick nukedisks off.

Summary

Looking at how a host is configured in Stacki can mostly be achieved through looking at the host list, and attributes by host. If you use `stack list host attr`, you will see a consolidated list of Globals, as replaced by Appliance, as replaced by Host attributes. Giving you a complete picture of what attributes are being used for the host.

Setting attributes is easy also, in fact it is so easy that caution is advised. Knowing what is being changed and what the likely impacts are is important.

A reinstall later, and the server now behaves as the organization expects. And total time invested is pretty small, while the process is repeatable.

Stacki Tutorial #5: Working With Attributes

© 2015 StackIQ, Inc. | www.stacki.com