Subject: Multiple veth on Multiple bridges with vlans over bonded ethernet
Posted by ctellechea on Wed, 18 Jun 2008 12:51:37 GMT
View Forum Message <> Reply to Message

Greetings,

Spasibo, It seems this might be covered in Russian but I unfortunately am unable to understand the language

http://forum.openvz.org/index.php?t=tree&goto=28107& &srch=veth+vnet.conf#msg_28107

I have setup an openvz environment with two bonded GigE interfaces + vlans + Bridging, with veth interfaces.

I have it all working having coded up some perl (shell scripting frustrates me past the simplest of imlementations) that gets called by vzctl via the vnet.conf hook that you have provided.

Through strace, trial and error, and a bunch of print statements in various files, I have discovered that vzctrl calls the script pointed to by /etc/vz/vnet.conf once per each veth interface definition defined within the /etc/vz/conf/$veth.conf for each VE.

My script slurps up and parses the conf file and grabs the INETIF assignment as well as an additional variable which I inserted into the openvz conf file, for each vlan that I have created, since by using a bridge interface calling convention that includes the vlan name, I can arrange, by vlans order that I set the vlans ids in the conf file veth definition, and vzctl order that I create the veth interfaces (through vzctl as stored in the conf file) I can reliably assign the correct interface, to the correct bridge, and in the order that makes the VE network configuration come up and vzctl seamlessly upon a reboot VEID star/stop veth vzctl.

My goal is to make the system sufficiently flexible that I can drive it, for different vlan <--> bridge mappings for different classes of VEs.

Unfortunately, my script is made much more complex than it could be if vzctl would export not only the VEID, as it does, but also the veth interface name that it has just created, to the environment, so that I would not have to create a number of tests within my script to prevent the script from attempting to up nonexistent interfaces, or add already added interfaces to a bridge.

Or, can you guys suggest a simpler yet effective way to do this?

Thanks for all of your great work on this software

PS: I'll gladly contribute the perl after I have it cleaned up.