

Foreman - Bug #35682

PR #9238 breaks specific provision scenarios

10/27/2022 09:25 AM - Maximilian Sesterhenn

<div>Status:New</div> <div>Priority:Normal</div> <div>Assignee:</div> <div>Category:Templates</div> <div>Target version:</div> <div>Difficulty:</div> <div>Triaged:No</div> <div>Bugzilla link:</div> <div>Pull request:</div>	<div>Fixed in Releases:</div> <div>Found in Releases:3.3.1, 3.4.0</div> <div>Red Hat JIRA:</div>
<div>Description</div> <div>Hello,</div> <div>we've upgraded our Foreman instance from 3.3.0 through 3.3.1 to 3.4.0.</div> <div>After upgrading, we've discovered that we're now unable to provision machines.</div> <div>Let me shortly introduce you to our infrastructure:</div> <div>We're using a single instance of Foreman (now 3.4.0) based on el8.</div> <div>DHCP is provided by an external, redundant service, which detects the client type based on their DHCP vendor class identifier.</div> <div>Based upon this detection, clients get DHCP options either for PXE boot or HTTP boot or different provisioning mechanisms (like ZTP for switches).</div> <div>This works reliably and has not changed.</div> <div>We're provision only kickstart based distributions (in our tests RockyLinux 8).</div> <div>In all cases (PXE and HTTPBoot) we use Grub2 on UEFI machines.</div> <div>PXE or HTTPBoot are affected both.</div> <div>All of our subnets are configured with boot mode static in Foreman.</div> <div>The drawback of our scenario is that in the early stages of the provisioning process (Bootloader stage, while anaconda is not running yet) the machines has a different IP than assigned and expected by Foreman.</div> <div>This was no issue in the past since Foreman was able to detect the machine by it's mac address.</div> <div>After discovering that the provisioning is broken now, we starting searching for relevant changes in the past updates.</div> <div>We found issue #34975 in Foreman 3.3.1.</div> <div>As far as I understood the idea in this issue was to replace the deprecated anaconda parameters like ks= with inst.ks= for releases greater or equal to el9 / fedora 33.</div> <div>This was covered in PR #8485.</div> <div>However there was also PR #9238 merged.</div> <div>This PR changes the logic in kickstart_kernel_options to use different parameters in the grub config based on the boot mode setting in the subnet object that is assigned to the machine.</div> <div>In case of our "static" network, the sendmac options were removed and the option static=1 was added to the foreman provisioning url.</div> <div>Based upon the issue and the PR, I can not see a reason for this.</div> <div>We observe two different consequences:</div> <div>- VMs are able to boot, but are not starting the provisioning process, instead the default boot option (in our case the discovery image) is shown.</div> <div>Most likely, this is due to Foreman trying to identify the machine by their IP address and not their mac address anymore.</div> <div>As I said before, because we use an external DHCP service, the IP address of the machine is not the IP address that Foreman expects in the early stages of the provisioning process.</div> <div>When trying to boot, grub2 is erroring out because of a memory issue.</div> <div>- Bare metal machines (in this case a Supermicro 1028U-TNRT+P) do not boot at all. It looks like Grub2 can not parse the menu entries and as there is no single entry, it errors out.</div> <div>Error message is "You need to load the kernel first".</div> <div>When reverting the change of PR #9238 only for a single menu entry, it starts working again, obviously only this single menu entry is</div>	

shown now.

Working kickstart_kernel_options:
BOOTIF=01-5c-6f-69-XX-XX-XX
ks=http://server.example.com:8000/unattended/provision?token=52s5811b-298f-4174-949e-4e66f58e8c70 **kssendmac**
ks.sendmac ip=10.1.101.74::10.1.101.1:255.255.255.0:server.example.com:enp130s0f0np0:none nameserver=10.1.101.1

Not working kickstart_kernel_options:
BOOTIF=01-5c-6f-69-XX-XX-XX
ks=http://server.example.com::8000/unattended/provision?*static=1*&token=59s5811b-298f-4174-949e-4e66f58e8c70
ip=10.1.101.74::10.1.101.1:255.255.255.0:server.example.com::enp130s0f0np0:none nameserver=10.1.101.1

See also the attached files.

What was the reason for the changes in PR [#9238](#)?
Can we revert back to mac-based detection?
Using an external DHCP service and therefore relying on sendmac should be quite common.

To be clear:
The changes in PR [#8485](#) are totally fine, they are not the issue here.

We have introduced a workaround by cloning kickstart_kernel_options and doing the changes there ourselves, but due to the structure of the template hierarchy we also have to clone Kickstart default and I want to avoid that.

Thanks in advance!

Related issues:

Related to Foreman - Bug #34975: ks= kernel parameter in Kickstart default iP... **Closed**

History

#1 - 10/27/2022 09:27 AM - Maximilian Sesterhenn

Kickstart default --> Kickstart default PXEGrub2 was meant, I apologize

#2 - 01/04/2023 11:45 AM - Ewoud Kohl van Wijngaarden

- Related to Bug #34975: ks= kernel parameter in Kickstart default iPXE causes RHEL9 Anaconda failure to start added

#3 - 01/29/2023 11:50 AM - Gerald Vogt

I'll also ran into this problem. See <https://community.theforeman.org/t/kickstart-with-static-ip-fails/32055>

The problem is the ampersand (&) in the generated URL.

According to <https://access.redhat.com/solutions/6810791> it needs to be quoted, i.e. either put the URL into single or double quotes or needs to be escaped with a \

Looking at the [source code](#) it seems easiest to put in quotes, I guess.

For meantime I manually hacked the generated cfg file to escape the ampersand and it's booting fine, then.

Files

not_working_grub.cfg	1.41 KB	10/27/2022	Maximilian Sesterhenn
working_grub.cfg	1.47 KB	10/27/2022	Maximilian Sesterhenn