

Foreman - Bug #10112

Allow user to define BMC driver / options per hardware model

04/10/2015 07:54 AM - Lukas Zapletal

Status: New	
Priority: Normal	
Assignee: Lukas Zapletal	
Category: BMC	
Target version:	
Difficulty: easy	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link: 1206361	Red Hat JIRA:
Pull request:	
Description Issue #7543 introduced options hash for BMC operations. We should allow users to define those per-host level. Using hardware model seems to be the best fit.	
Related issues:	
Related to Smart Proxy - Bug #10102: BMC query fails with 'undefined method d...	New 04/09/2015
Related to Smart Proxy - Bug #7543: allow additional rubyipmi connection opti...	Closed 09/19/2014
Has duplicate Foreman - Bug #18091: BMC management doesn't work with Dell's i...	Duplicate 01/16/2017

History

#1 - 04/10/2015 07:54 AM - Lukas Zapletal

- Related to Bug #10102: BMC query fails with 'undefined method downcase for nil:NilClass added

#2 - 04/10/2015 07:56 AM - Lukas Zapletal

- Project changed from Smart Proxy to Foreman

- Category set to BMC

- Difficulty set to easy

#3 - 04/10/2015 08:02 AM - Dominic Cleal

- Related to Bug #7543: allow additional rubyipmi connection options to be passed through added

#4 - 04/10/2015 08:02 AM - Dominic Cleal

- Description updated

#5 - 04/11/2015 01:36 PM - Jérôme Vizcaino

Please keep a per-host configuration for this. Per model would work unless the same model is using different BMC firmware versions.

#6 - 04/13/2015 04:26 AM - Lukas Zapletal

Good point, so the current plan is:

- Use Hardware model (new database column) when set.
- Use Host parameter named "ipmidriver" when set.
- Use Global setting "ipmi_driver" when set.
- Don't send it (and use default "lan20" set by rubyipmi) instead - will work with Foreman 1.9+).

We should modify the error message to instruct the user to set this appropriately when the operation fails.

Also we can pre-populate Hardware model table from now on. The very fist pair would be: "PowerEdge R720" and "lan20" which I have tested on Friday.

#7 - 04/13/2015 04:33 AM - Dominic Cleal

Lukas Zapletal wrote:

- Use Host parameter named "ipmidriver" when set.

Avoid specially named host parameters I'd say, it should be a first class attribute if it's meant to be used.

- Use Global setting "ipmi_driver" when set.
- Don't send it (and use default "lan20" set by rubyipmi" instead - will work with Foreman 1.9+).

I'd suggest doing one or the other to reduce the complexity, e.g. stick to rubyipmi's defaults instead of introducing another global setting.

#8 - 04/18/2015 04:35 AM - Jérôme Vizcaino

I'd suggest doing one or the other to reduce the complexity, e.g. stick to rubyipmi's defaults instead of introducing another global setting.

Don't you think explicit is better than implicit ? I mean, rubyipmi changed their default behavior recently and this caused me headaches (BMC would stop working all of a sudden).

I think explicit default IPMI driver set in Foreman would make things clearer and would avoid such problem in the future.

Btw, you can set "lan20" driver for all the Dell PowerEdge family products starting from 11th Gen:

- M610, M620, M420, M630, (and maybe M610x too but I don't know how the system reports its model)
- R610, R620, R630
- R710, R720, R730

#9 - 08/18/2015 09:45 PM - Corey Osman

I switched the default driver in rubyipmi from auto to lan20 because there were more people having issues with the auto detection mechanism which works ok but not great. However, now that the default is switched to lan20 everyone with really ancient hardware is forced to specify lan15. I didn't think this was a big issue, because of the huge amount of security flaws with IPMI 1.0 and IPMI1.5 that nobody would ever run it in production. But a few people run this old hardware in their home labs. As a stop gap we could add a setting called default_ipmi_driver to the bmc smart proxy and use that setting whenever one is not specified in the options hash. Folks with older hardware would at least be able to get by until this ticket is completed.

This would work similar to the default provider: https://github.com/theforeman/smart-proxy/blob/develop/modules/bmc/bmc_api.rb#L267

If you set the driver-type value to auto, the rubyipmi will always try the lan driver first, fail and then the lan20 driver. This has a side affect executing twice the IPMI commands being called. So if your going to set these options in the database I would set auto for anything you are unsure about such as old hardware, new hardware set to lan20 or nothing and the default lan20 would be used. Additionally my comments above about using a smart proxy default setting could override Rubyipmi's default if needed. I have seen several instances where folks want a different privilege type because the server factory made it that way. So being able to configure this on a per server model / host level would be pretty nice and catch a few more edge cases.

Also I think we can auto detect IPMI version info too.

```
# > dmidecode --type 38
# dmidecode 2.10
SMBIOS 2.5 present.
```

```
Handle 0x0049, DMI type 38, 18 bytes
IPMI Device Information
  Interface Type: KCS (Keyboard Control Style)
  Specification Version: 2.0
  I2C Slave Address: 0x10
  NV Storage Device: Not Present
  Base Address: 0x00000000000000CA2 (I/O)
  Register Spacing: Successive Byte Boundaries
```

#10 - 01/16/2017 06:56 AM - Dominic Cleal

- Has duplicate Bug #18091: BMC management doesn't work with Dell's iDRAC7 added

#11 - 01/16/2017 06:57 AM - Dominic Cleal

- Subject changed from Allow user to define BMC options per hardware model to Allow user to define BMC driver / options per hardware model