
Subject: CT console feature

Posted by [kir](#) on Thu, 23 Feb 2012 14:39:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

I am glad to announce CT console feature.

Available in RHEL6-based kernel since 042stab048.1, this feature is pretty simple to use. Use `vzctl attach CTID` to attach to this container's console, and you will be able to see all the messages CT init is writing to console, or run `getty` on it, or anything else.

Please note that the console is persistent, i.e. it is available even if a container is not running. That way, you can run `vzctl attach` and then (in another terminal) `vzctl start`. That also means that if a container is stopped, `vzctl attach` is still there.

Press `Esc .` to detach from the console.

The feature will be available in up-coming `vzctl-3.0.31`. I have just made a nightly build of `vzctl` (version 3.0.30.2-18.git.a1f523f) available so you can test this. Check <http://wiki.openvz.org/Download/vzctl/nightly> for information of how to get a nightly build.

For anyone interested in user-space implementation of this, check this git commit:

<http://git.openvz.org/?p=vzctl;a=commitdiff;h=a1f523f59a6e321ce2cc6dd42d0f5a660a712339>

Subject: Re: CT console feature

Posted by [Tim Small](#) on Thu, 23 Feb 2012 16:18:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 23/02/12 14:39, Kir Kolyshkin wrote:

> I am glad to announce CT console feature.

Very cool - been on my wish list for a while!

> Available in RHEL6-based kernel since 042stab048.1, this feature is
> pretty simple to use. Use `vzctl attach CTID` to attach to this
> container's console, and you will be able to see all the messages CT
> init is writing to console, or run `getty` on it, or anything else.

>

> Please note that the console is persistent, i.e. it is available even
> if a container is not running. That way, you can run `vzctl attach` and
> then (in another terminal) `vzctl start`. That also means that if a
> container is stopped, `vzctl attach` is still there.

>

> Press Esc . to detach from the console.

Cool. Just a thought on this escape sequence... ipmitool and ssh (amongst others) use ~ after a carriage return as the command sequence, and this is sort-of a defacto standard...

OpenSSH..

tim@ermintrude:~\$ ~?

Supported escape sequences:

~. - terminate connection (and any multiplexed sessions)

[....]

~? - this message

~~ - send the escape character by typing it twice

(Note that escapes are only recognized immediately after newline.)

ipmitool

Special escape sequences are provided to control the SOL session:

~. Terminate connection

~^Z Suspend ipmitool

~^X Suspend ipmitool, but don't

restore tty on restart

~B Send break

~~ Send the escape character by

typing it twice

~? Print the supported escape sequences

Note that escapes are only recognized immediately after newline.

Both allow you do change the escape character with a command line switch.

Any chance of adopting this "standard" instead of something else?

Any thoughts?

Cheers,

Tim.

--

South East Open Source Solutions Limited

Registered in England and Wales with company number 06134732.

Registered Office: 2 Powell Gardens, Redhill, Surrey, RH1 1TQ

VAT number: 900 6633 53 <http://seoss.co.uk/> +44-(0)1273-808309

Subject: Re: CT console feature

Posted by [kir](#) on Thu, 23 Feb 2012 20:38:46 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 02/23/2012 08:18 PM, Tim Small wrote:

> On 23/02/12 14:39, Kir Kolyshkin wrote:

>> I am glad to announce CT console feature.

> Very cool - been on my wish list for a while!

>

>> Available in RHEL6-based kernel since 042stab048.1, this feature is

>> pretty simple to use. Use `vzctl attach CTID` to attach to this

>> container's console, and you will be able to see all the messages CT

>> init is writing to console, or run `getty` on it, or anything else.

>>

>> Please note that the console is persistent, i.e. it is available even

>> if a container is not running. That way, you can run `vzctl attach` and

>> then (in another terminal) `vzctl start`. That also means that if a

>> container is stopped, `vzctl attach` is still there.

>>

>> Press `Esc .` to detach from the console.

> Cool. Just a thought on this escape sequence... `ipmitool` and `ssh`

> (amongst others) use `~` after a carriage return as the command sequence,

> and this is sort-of a defacto standard...

>

> OpenSSH..

>

>

> `tim@ermintrude:~$ ~?`

> Supported escape sequences:

> `~.` - terminate connection (and any multiplexed sessions)

> `[....]`

> `~?` - this message

> `~~` - send the escape character by typing it twice

> (Note that escapes are only recognized immediately after newline.)

>

>

> `ipmitool`

>

> Special escape sequences are provided to control the SOL session:

> `~.` Terminate connection

> `~^Z` Suspend `ipmitool`

> `~^X` Suspend `ipmitool`, but don't

> restore tty on restart

> `~B` Send break

> `~~` Send the escape character by

> typing it twice

> `~?` Print the supported escape sequences

> Note that escapes are only recognized immediately

> after newline.

>
>
> Both allow you do change the escape character with a command line switch.
>
> Any chance of adopting this "standard" instead of something else?
>
> Any thoughts?

See, the problem here is you usually access your node through ssh, and if we will use the same sequence than ssh will close.

Having said that, I welcome patches that

- make accept the sequence only after a newline
- make an option and/or config parameter to change Esc to something else

>
> Cheers,
>
> Tim.
>

Subject: Re: CT console feature
Posted by [Solar Designer](#) on Thu, 23 Feb 2012 21:15:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kir,

Thanks for your work on this!

On Fri, Feb 24, 2012 at 12:38:46AM +0400, Kir Kolyshkin wrote:
> See, the problem here is you usually access your node through ssh, and
> if we will use the same sequence than ssh will close.

Not necessarily. The person will need to type ~. (or ~~, etc.) to get to the desired nesting level. This is already the case with ssh in ssh (which is usually a bad idea for security reasons, but still), as well as with certain other tools in a ssh session.

> Having said that, I welcome patches that
> - make accept the sequence only after a newline

Isn't this the same as making the sequence "<newline> ESC ." (or "<newline> ~ .")?

> - make an option and/or config parameter to change Esc to something else

Sounds like a fine feature.

Alexander

Subject: Re: CT console feature
Posted by [Tim Small](#) on Fri, 24 Feb 2012 08:40:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 23/02/12 17:12, Kirill Kolyshkin wrote:

> Using ~. would be problematic since it interferes with e.g. ssh.
>

Hmm, on reflection, I can see "ESC ." possibly causing accidental escapes when using vi? A reasonably common vi sequence is to insert a body of text, and then repeat that insert. Wouldn't it be better to make the escape ~ as-standard, and allow changing to escape too? I can't think of a time when I've accidentally triggered the ~ sequence in ssh...

> Patches to allow changing escape sequence and to only recognise it after a newline are welcome.
>

I'd be willing to contribute such a patch, but I have no spare time for the next few months (I'm putting an extra storey on my house, and I don't have much of a roof at the moment - so that's priority A1 for the foreseeable).

Cheers,

Tim.

--

South East Open Source Solutions Limited
Registered in England and Wales with company number 06134732.
Registered Office: 2 Powell Gardens, Redhill, Surrey, RH1 1TQ
VAT number: 900 6633 53 <http://seoss.co.uk/> +44-(0)1273-808309

Subject: Re: CT console feature
Posted by [ragubhat](#) on Sat, 03 Mar 2012 04:04:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Kir announces:

>> Use vzctl attach CTID to attach to this container's console, and you
>> will be able to see all the messages CT init is writing to console

Real cool feature, thanks for your work on improving vzctl and friends.

--

Regards

Raghavendra Bhat

Help stop Net Censorship in India

Stomp down on Censorship of Google, Yahoo, Facebook and Twitter!
