

[\[Date Prev\]](#)[Date Next] [\[Thread Prev\]](#)[Thread Next] [\[Thread Index\]](#) [\[Date Index\]](#) [\[Author Index\]](#)

[rhos-docs] Fwd: [openstack-dev] [nova] How to connect to a serial port of an instance via websocket?

- *From:* Steve Gordon <sgordon@redhat.com>
- *To:* Don Domingo <ddomingo@redhat.com>
- *Cc:* rhos-docs <rhos-docs@redhat.com>
- *Subject:* [rhos-docs] Fwd: [openstack-dev] [nova] How to connect to a serial port of an instance via websocket?
- *Date:* Thu, 30 Oct 2014 14:57:13 -0400 (EDT)

Hi Don,

This feature would be a great candidate for a RHEL OSP 6 KBase.

Thanks,

Steve

----- Forwarded Message -----

```
> From: "Markus Zoeller" <mzoeller@de.ibm.com>
> To: "OpenStack Development Mailing List (not for usage questions)" <openstack-dev@lists.openstack.org>
> Sent: Thursday, October 30, 2014 9:45:08 AM
> Subject: Re: [openstack-dev] [nova] How to connect to a serial port of an instance via websocket?
```

```
> The cause of this is that the serialproxy was not started. So nobody
> was listening to the port 6083. Validate with:
> $ netstat -nat | grep :608
```

```
>
> tcp 0 0 0.0.0.0:6080          0.0.0.0:*             LISTEN
> tcp 0 0 0.0.0.0:6081          0.0.0.0:*             LISTEN
> tcp 0 0 192.168.122.41:60858 192.168.122.41:5672   ESTABLISHED
> tcp 0 0 192.168.122.41:60859 192.168.122.41:5672   ESTABLISHED
> tcp6 0 0 192.168.122.41:5672  192.168.122.41:60858  ESTABLISHED
> tcp6 0 0 192.168.122.41:5672  192.168.122.41:60859  ESTABLISHED
```

```
> After finding [1] all I had to do was to start this proxy manually with:
> $ nova-serialproxy
```

```
>
> INFO nova.console.websocketproxy [-] WebSocket server settings:
> INFO nova.console.websocketproxy [-] - Listen on 0.0.0.0:6083
> INFO nova.console.websocketproxy [-] - Flash security policy server
> INFO nova.console.websocketproxy [-] - No SSL/TLS support
>                                     (no cert file)
> INFO nova.console.websocketproxy [-] - proxying from 0.0.0.0:6083
>                                     to None:None
```

```
> After executing this command, the `netstat` command from above shows a
> listener for port 6083:
```

```
> $ netstat -nat | grep :608
>
> tcp 0 0 0.0.0.0:6080          0.0.0.0:*             LISTEN
> tcp 0 0 0.0.0.0:6081          0.0.0.0:*             LISTEN
> tcp 0 0 0.0.0.0:6083          0.0.0.0:*             LISTEN
> tcp 0 0 192.168.122.41:60858 192.168.122.41:5672   ESTABLISHED
> tcp 0 0 192.168.122.41:60859 192.168.122.41:5672   ESTABLISHED
> tcp6 0 0 192.168.122.41:5672  192.168.122.41:60858  ESTABLISHED
> tcp6 0 0 192.168.122.41:5672  192.168.122.41:60859  ESTABLISHED
```

```
> By using Sahids websocketclient and the URI I got from the command
> `nova get-serial-console instance1` the connection gets established and
> one will see the login screen (e.g. from cirros).
```

```
>
> I was expecting the nova-serialproxy to start automatically when the
> section [serial_console] has the entry `enabled=True`. Is this a bug or
```

```

> do I have a wrong assumption here?
>
> Thanks again Sahid for your help! Thanks to Solly too, for offering JS
> help!
>
> [1] OpenStack Nova Developer Docs; "Websocket serial Proxy for OpenStack
> Nova serial ports." ;
> http://docs.openstack.org/developer/nova/man/nova-serialproxy.html
>
>
>
> Markus Zoeller/Germany/IBM wrote on 10/30/2014 11:29:22 AM:
>
> > From: Markus Zoeller/Germany/IBM
> > To: "OpenStack Development Mailing List (not for usage questions)"
> > <openstack-dev@lists.openstack.org>
> > Date: 10/30/2014 11:29 AM
> > Subject: [nova] How to connect to a serial port of an instance via
> > websocket?
> >
> > On Wed Oct 29 09:42:52 UTC 2014, Sahid Orentino Ferdjaoui wrote:
> > >
> > > The aim of the feature is exposing an interactive web-based serial
> > > consoles through a websocket proxy. The API returns an URL with a
> > > valid token that should be used with a websocket client to read/write
> > > on the stream.
> > >
> > > Considering the service nova-serialproxy is running and well
> > > configured you can use this simple test purpose client to connect
> > > yourself on the URL returned by the API:
> > >
> > > https://gist.github.com/sahid/894c31f306bebacb2207
> > >
> > > The general idea behind this service is for example to help debugging
> > > VMs when something was wrong with the network configuration.
> >
> > Hi Sahid,
> >
> > thanks for your great example! When I execute it I get the error
> > socket.error: [Errno 111] Connection refused
> > How do I debug this? Did I miss a configuration?
> >
> > ./lazyclient.py
> ws://127.0.0.1:6083/?token=39262891-2588-4872-994b-3be9b7333fd7
> > Traceback (most recent call last):
> > File "./lazyclient.py", line 27, in <module>
> >     ws.connect()
> > File "/usr/local/lib/python2.7/dist-packages/ws4py/client/
> > __init__.py", line 209, in connect
> >     self.sock.connect(self.bind_addr)
> > File "/usr/lib/python2.7/socket.py", line 224, in meth
> >     return getattr(self._sock,name)(*args)
> > socket.error: [Errno 111] Connection refused
>
>
>
> _____
> OpenStack-dev mailing list
> OpenStack-dev@lists.openstack.org
> http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-dev
>
--
Steve Gordon, RHCE
Sr. Technical Product Manager,
Red Hat Enterprise Linux OpenStack Platform

```

[\[Date Prev\]](#)[\[Date Next\]](#)
[\[Thread Prev\]](#)[\[Thread Next\]](#)
[\[Thread Index\]](#)
[\[Date Index\]](#)
[\[Author Index\]](#)