major.io

Words of wisdom from a systems engineer

———

———

Search this website …

———

Setting up a telnet handler for OpenStack Zuul CI jobs in GNOME 3

July 22, 2016 By Major Hayden — 5 Comments

The OpenStack Zuul system has gone through some big changes recently, and one of those changes is around how you monitor a running CI job. I work on OpenStack-Ansible quite often, and the gate jobs can take almost an hour to complete at times. It can be helpful to watch the output of a Zuul job to catch a problem or follow a breakpoint.

New Zuul

In the previous version of Zuul, you could access the Jenkins server that was running the CI job and monitor its progress right in your browser. Today, you can monitor the progress of a job via telnet. It's much easier to use and it's a lighter-weight way to review a bunch of text.

Some of you might be saying: "It's 2016. Telnet? Unencrypted? Seriously?"

Before you get out the pitchforks, all of the data is read-only in the telnet session, and nothing sensitive is transmitted. Anything that comes through the telnet session is content that exists in an open source repository within OpenStack. If someone steals the output of the job, they're not getting anything valuable.

I was having a lot of trouble figuring out how to set up a handler for `telnet://` URL's that I clicked in Chrome or Firefox. If I clicked a link in Chrome, it would be passed off to `xdg-open`. I'd press OK on the window and then nothing happened.

Creating a script

First off, I needed a script that would take the URL coming from an application and actually do something with it. The script will receive a URL as an argument that looks like `telnet://SERVER_ADDRESS:PORT` and that must be handed off to the `telnet` executable. Here's my basic script:

```bash
#!/bin/bash

# Remove the telnet:// and change the colon before the port
# number to a space.
TELNET_STRING=$(echo $1 | sed -e 's/telnet:\/\///' -e 's/:/ /')

# Telnet to the remote session
/usr/bin/telnet $TELNET_STRING

# Don't close out the terminal unless we are done
read -p "Press a key to exit"
```

I saved that in `~/bin/telnet.sh`. A quick test with localhost should verify that the script works:

```
$ chmod +x ~/bin/telnet.sh
$ ~/bin/telnet.sh telnet://127.0.0.1:12345
Trying 127.0.0.1...
telnet: connect to address 127.0.0.1: Connection refused
Press a key to exit
```

Linking up with GNOME

We need a `.desktop` file so that GNOME knows how to run our script. Save a file like this to `~/.local/share/applications/telnet.desktop`:

```
[Desktop Entry]
Version=1.0
Name=Telnet
GenericName=Telnet
Comment=Telnet Client
Exec=/home/major/bin/telnet.sh %U
Terminal=true
Type=Application
Categories=TerminalEmulator;Network;Telnet;Internet;BBS;
MimeType=x-scheme/telnet
X-KDE-Protocols=telnet
Keywords=Terminal;Emulator;Network;Internet;BBS;Telnet;Client;
```

Change the path in `Exec` to match where you placed your script.

We need to tell GNOME how to handle the `x-scheme-handler/telnet` mime type. We do that with `xdg` utilities:

```
$ xdg-mime default telnet.desktop x-scheme-handler/telnet
$ xdg-mime query default x-scheme-handler/telnet
telnet.desktop
```

Awesome! When you click a link in Chrome, the following should happen:

- Chrome will realize it has no built-in handler and will hand off to `xdg-open`
- `xdg-open` will check its list of mime types for a telnet handler
- `xdg-open` will parse `telnet.desktop` and run the command in the `Exec` line within a terminal
- Our `telnet.sh` script runs with the `telnet://` URI provided as an argument
- The remote telnet session is connected

Share this post:

Twitter   Google   LinkedIn   Reddit   Email   Print

Tagged With: fedora, gnome, linux, networking, openstack

Comments

tosky says
July 22, 2016 at 2:50 pm

This is not GNOME-only, as you are using the Desktop Entry Standard: https://specifications.freedesktop.org/desktop-entry-spec/latest/

For the record, if you have any KDE Frameworks 5-based application, the kio library contains and installs already an handler for telnet (through the simple CLI program ktelnetservice5 which by default runs konsole, but it can be changed).

Reply

Major Hayden says
July 22, 2016 at 2:52 pm

major.io

Words of wisdom from a systems engineer

Who am I?
icanhazip FAQ
Résumé
Keybase
RSS

Search this website …

Thanks for the tip about the KDE applications, tosky. That could make things a little easier if the list of required dependencies isn't terribly long.

Reply

> tosky says
> July 22, 2016 at 3:02 pm
>
> There are not "KDE applications" as in the past anymore :)
> They are all Qt applications which happens to use some of the Frameworks libraries:
> https://api.kde.org/frameworks/index.html
> kio is one of the complex libraries (transparent local and remote file access) and many (complex) applications ends up using it.
>
> Reply

Sam J Sharpe says
July 23, 2016 at 12:31 am

The / separator for sed is what people use by default, but when doing sed against things with slashes in them, it's better to change it – thus `sed -e 's/telnet:////'` becomes `sed -e 's#telnet://##'` and avoids the need to escape.

Reply

Ihar Hrachyshka says
April 6, 2017 at 3:56 pm

Great work. I needed to modify it a bit to make it work for ipv6 addresses:

https://github.com/booxter/configs/blob/master/bin/telnet.sh

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

---

major.io

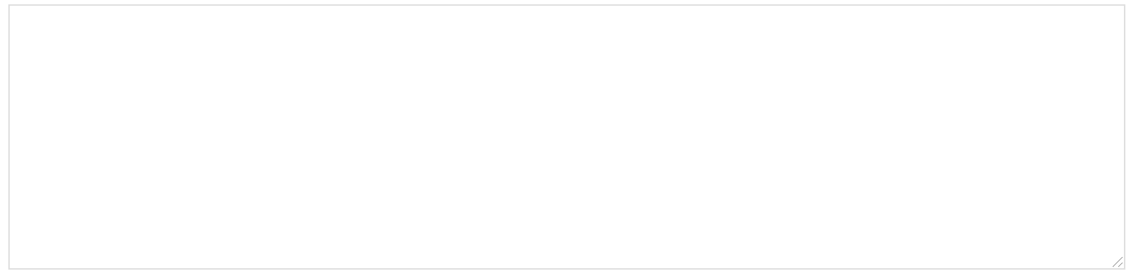Words of wisdom from a systems engineer

———

Who am I?
icanhazip FAQ
Résumé
Keybase
RSS

———

Search this website …

———

major.io

Words of wisdom from a
systems engineer

Who am I?
icanhazip FAQ
Résumé
Keybase
RSS

Search this website …

Name *

Email *

Website

☐

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.