

How do I find out why my Docker service is in Pending state?

raarts

I'm playing Docker Swarm, and using a `docker-compose.yml` with `docker deploy`. All services get deployed except for one, which stays in Pending state. I have added a constraint that ties this service to one of the nodes. My question is not so much about this particular problem, but more about how to troubleshoot. The Docker documentation mentions possible causes for a service to be in pending state, but none of those apply (constraint problem, resource drainage).

Can I see the docker swarm 'thought process' somewhere? What is it thinking?

Edit: should have made it more clear that I am using the new, 1.12-introduced, swarm option.

asked Jan 13 '17 at 18:41



I will explain how to debug when the service does not start as expected in the docker swarm mode.

First of all, get a task ID with `docker service ps <service-name>`.

Next, it is useful to check meta data with `docker inspect <task-id>`. In particular, Error message before container start is in the status field, and then reconfirm whether it was started with the intended parameters.

If the task has container ID, it was abnormally exited after starting the container, so check the log of the container with `docker logs <container-id>`

I hope this would be some of help.

answered Jan 15 '17 at 6:35



[docker run swarm](#) has a `--debug` option which can tell you more.

See docker swarm [issue 2341](#) or [docker issue 24982](#) to see that option used to debug pending states.

For instance:

```
(unknown): 192.168.99.106:2375 (node2 ip)
L ID:
L Status: Pending
L Containers: 0
L Reserved CPUs: 0 / 0
L Reserved Memory: 0 B / 0 B
```

L Labels:
L Error: Cannot connect to the Docker daemon. Is the docker daemon running on this host?..

answered Jan 14 '17 at 6:15



Your Answer

- Links
- Images
- Styling/Headers
- Lists
- Blockquotes
- Code
- HTML
- [advanced help »](#)

In most cases, a plain URL will be recognized as such and automatically linked:

Visit <https://area51.stackexchange.com/> regularly!
Use angle brackets to force linking: Have you seen `<https://superuser.com>?`

To create fancier links, use Markdown:

Here's `[a link](http://www.example.com/)`! And a reference-style link to `[a panda][1]`.
References don't have to be `[numbers][question]`.

```
[1]: https://notfound.stackexchange.com/
[question]: https://english.stackexchange.com/questions/11481
```

You can add tooltips to links:

Click `[here](https://diy.stackexchange.com "this text appears when you mouse over")`!
This works with `[reference links][blog]` as well.

```
[blog]: https://stackoverflow.blog/ "click here for updates"
```

Images are exactly like links, but they have an exclamation point in front of them:

`![a busy cat](https://cdn.sstatic.net/Sites/stackoverflow/img/error-lolcat-problemz.jpg)`
`![two muppets][1]`

```
[1]: https://i.imgur.com/I5DFV.jpg "tooltip"
```

The word in square brackets is the alt text, which gets displayed if the browser can't show the image. Be sure to include meaningful alt text for screen-reading software.

Be sure to use text styling sparingly; only where it helps readability.

This is italicized, and so
is `_this_`.

****This is bold****, just like `__this__`.

You can *****combine***** them
if you `___really have to___`.

To break your text into sections, you can use headers:

A Large Header
=====

Smaller Subheader

Use hash marks if you need several levels of headers:

```
# Header 1 #  
## Header 2 ##  
### Header 3 ###
```

Both bulleted and numbered lists are possible:

```
- Use a minus sign for a bullet  
+ Or plus sign  
* Or an asterisk
```

```
1. Numbered lists are easy  
2. Markdown keeps track of  
   the numbers for you  
7. So this will be item 3.
```

```
1. Lists in a list item:  
   - Indented four spaces.  
     * indented eight spaces.  
   - Four spaces again.  
2. You can have multiple  
   paragraphs in a list items.
```

Just be sure to indent.

```
> Create a blockquote by  
> prepending ">" to each line.  
>  
> Other formatting also works here, e.g.  
>  
> 1. Lists or  
> 2. Headings:  
>  
> ## Quoted Heading ##
```

You can even put blockquotes in blockquotes:

```
> A standard blockquote is indented  
> > A nested blockquote is indented more  
> > > You can nest to any depth.
```

To create code blocks or other preformatted text, indent by four spaces or surround with groups of backticks:

```
    This will be displayed in a monospaced font. The first four spaces  
    will be stripped off, but all other whitespace will be preserved.
```

```
```
```

Markdown and HTML are turned off in code blocks:

```
<i>This is not italic</i>, and [this is not a link](https://example.com)
```
```

To create not a block, but an inline code span, use backticks:

The ``` character is just a shortcut for ``window.jQuery``.

If you want to have a preformatted block within a list, indent by eight spaces:

```
1. This is normal text.
```

2. So is this, but now follows a code block:

```
Skip a line and indent eight spaces.  
That's four spaces for the list  
and four to trigger the code block.
```

If you need to do something that Markdown can't handle, use HTML. Note that [we only support a very strict subset of HTML!](#)

Strikethrough humor is `<strike>funny</strike>`.

Markdown is smart enough not to mangle your span-level HTML:

```
<b>Markdown works *fine* in here.</b>
```

Block-level HTML elements have a few restrictions:

1. They must be separated from surrounding text by blank lines.
2. The begin and end tags of the outermost block element must not be indented.
3. Markdown can't be used within HTML blocks.

```
<pre>  
  You can <em>not</em> use Markdown in here.  
</pre>
```

Not the answer you're looking for? Browse other questions tagged [docker](#) [docker-swarm](#) or [ask your own question](#).