

New issue

[Jump to bottom](#)

Improve isolation of Che theia and che-machine-exec components #15651

Closed skabashnyuk opened this issue on Jan 10, 2020 · 13 comments

Assignees



Labels

kind/task severity/P1

Projects

Platform-2020-02-18 Platform-2020-03-10

skabashnyuk commented on Jan 10, 2020

Contributor

Is your task related to a problem? Please describe.

Under some conditions, there is a possibility to reach the port of one workspace from another workspace. To improve the isolation of the major Eclipse Che components we would like to.

Describe the solution you'd like

1. Use single pod for jwt proxy and other workspace containers.
2. make machine-exec, theia-remote-runtime and theia endpoints listening to localhost only

Describe alternatives you've considered

n/a

Additional context

n/a

skabashnyuk added kind/task severity/P1 team/platform labels on Jan 10, 2020

skabashnyuk added this to the 7.8.0 milestone on Jan 10, 2020

skabashnyuk added this to To do in Platform-2020-01-28 via automation on Jan 10, 2020

amisevsk commented on Jan 13, 2020

Contributor

How does this affect #11476? Would each pod in #11476 require a JWT container?

skabashnyuk commented on Jan 14, 2020

Contributor Author

How does this affect #11476? Would each pod in #11476 require a JWT container?

I don't know how it affects it. At this moment all our tooling(theia and che-machine-exec) runs in a single pod - no?

skabashnyuk mentioned this issue on Jan 14, 2020

Make it possible to get OpenShift oauth token from theia IDE container #15670

Closed

amisevsk commented on Jan 14, 2020

Contributor

At this moment all our tooling(theia and che-machine-exec) runs in a single pod - no?

Currently yes, but the linked issue is a plan to split workspaces into multiple pods (provided RWX volumes are available) -- some thought should be given how this change would affect future plans. If JWT proxy continues to require ~128Mi of memory, we could be looking at 300-500Mi of overhead if workspaces are split.

skabashnyuk commented on Jan 14, 2020

Contributor Author

@amisevsk you are right. That is a good and nice topic. However, I prefer not to mix them together because #11476 can require some time to implement it. And we would like to fix/improve what we have now already implemented.

1

davidfestal commented on Jan 14, 2020

Contributor

I would say that anyway we should at least allow the JWT proxy to be run inside the POD as a default option.

metlos moved this from To do to In progress in Platform-2020-01-28 on Jan 22, 2020

 **metlos** self-assigned this on Jan 22, 2020


 **metlos** added the `status/in-progress` label on Jan 22, 2020

 **sleshchenko** mentioned this issue on Jan 22, 2020

[devworkspace] Rework WorkspaceRouting to run container inside workspace pod #15786

 Closed

 **skabashnyuk** added this to To do in Platform-2020-02-18 via `automation` on Jan 29, 2020

 **skabashnyuk** removed this from In progress in Platform-2020-01-28 on Jan 29, 2020

 **skabashnyuk** modified the milestones: **7.8.0**, **7.9.0** on Jan 29, 2020


 **skabashnyuk** mentioned this issue on Jan 29, 2020

Platform-2020-02-18 (Sprint: 179) #15869

 Closed

 5 tasks

 **skabashnyuk** moved this from To do to In progress in Platform-2020-02-18 on Jan 29, 2020

 **skabashnyuk** assigned **metlos** and **mshaposhnik** and unassigned **metlos** on Jan 29, 2020

 **metlos** mentioned this issue on Jan 31, 2020

Inject pods into other deployments #15890

 Merged

metlos commented on Feb 10, 2020

Contributor

I could make this work by moving the jwtproxy to the workspace pod and make it proxy the secure servers by resending the traffic to 127.0.0.1 and appropriate secure server port.

While this works, it has at least two consequences:

- doesn't prevent the supposedly secure servers from listening on 0.0.0.0 or any other interface/IP address which would make them accept traffic not proxied by jwtproxy
- The secure servers need to listen on 127.0.0.1 which has not been the requirement so far. In 7.8.0 and prior, a secure server merely needs to listen on the pod IP address.

This is solvable in two ways IMHO:

1. We just document that if an endpoint is `secure: true`, the backing server needs to listen solely on 127.0.0.1 and listening on any more IP addresses essentially enables unsecured access to the backing server.
2. We add a new attribute to the endpoint, something like `private: true` which tells Che to proxy access to 127.0.0.1. If such attribute was false, Che would deploy a service for the backing server and put a jwtproxy in front of that service, just like it does at the moment. Having `private: false` leaves the backing server open to unsecured access, which would have to be documented. We'd keep the current behavior though, so plugins/devfiles could opt in to this feature gradually.

@skabashnyuk @sleshchenko @l0rd WDYT?

sleshchenko commented on Feb 12, 2020

Member

We add a new attribute to the endpoint, something like `private: true` which tells Che to proxy access to 127.0.0.1.

`private: true` does not solve an issue you described with processes which listen to 0.0.0.0

doesn't prevent the supposedly secure servers from listening on 0.0.0.0 or any other interface/IP address which would make them accept traffic not proxied by jwtproxy right?

we add a new attribute to the endpoint, something like `private: true`

It's a bit confusing to have public and private attributes at the same time, like

```
attributes:
  public: true // I need URL for this endpoint
  secure: true // I need to make sure that nobody accesses my server without token with URL
  private: true // I need to make sure that nobody accesses my server without token within the cluster
```

So, my personal +1 for

We just document that if an endpoint is `secure: true`, the backing server needs to listen solely on 127.0.0.1 and listening on any more IP addresses essentially enables unsecured access to the backing server.

in case `private: true` does not help to solve issues with processes which listen to 0.0.0.0...

l0rd commented on Feb 12, 2020

Contributor

I am +1 for solution n.1

 This was referenced on Feb 13, 2020

Make note of assumptions on the secure endpoints eclipse-che/che-docs#1075

➔ Merged

Make theia hostname configurable eclipse-che/che-theia#626

➔ Merged

This was referenced on Feb 13, 2020

Make che-machine-exec listen only on localhost by default eclipse-che/che-machine-exec#80

🔒 Closed

Secure the che-machine-exec and che-theia plugins eclipse-che/che-plugin-registry#378

➔ Merged

metlos commented on Feb 14, 2020

Contributor

Implemented in #15890, eclipse-che/che-plugin-registry#378 and eclipse-che/che-theia#626.

metlos closed this as completed on Feb 14, 2020

Platform-2020-02-18 automation moved this from In progress to Done on Feb 14, 2020

metlos commented on Feb 14, 2020

Contributor

Note that the PR in che-docs that clarifies the assumptions about the secure servers is still open: [eclipse-che/che-docs#1075](#)

nickboldt commented on Feb 18, 2020

Contributor

Based on discussion today, current status is:

- fixed/working for multiuser mode, will be in 7.9.0 release.
- broken/ not working for singleuser mode, which does not impact CRW 2.1 (no singleuser mode available)
- Once 7.9.0 is live, fix will be reverted in master and a more complete solution will be done to support both single and multiuser modes in Che 7.10, with backport (if maintenance release needed) to 7.9.1 and included in CRW2.1

nickboldt commented on Feb 19, 2020

Contributor

This issue is "closed" and has label "in progress".

So I'm reopening it.

nickboldt reopened this on Feb 19, 2020

skabashnyuk removed this from the 7.9.0 milestone on Feb 20, 2020

skabashnyuk added this to To do in Platform-2020-03-10 via automation on Feb 20, 2020

nickboldt mentioned this issue on Feb 20, 2020

It's not possible to start workspace in Single-User Che anymore #16053

🔒 Closed

📋 23 tasks

metlos mentioned this issue on Feb 25, 2020

Change the default jwtproxy image to the latest one. #16128

➔ Merged

metlos mentioned this issue on Mar 4, 2020

Make single user work with secure components exposed through localhost only #16227

➔ Merged

metlos commented on Mar 6, 2020


Contributor

#16053 is implemented so this is now complete IMHO.



metlos closed this as completed on Mar 6, 2020

Platform-2020-03-10 automation moved this from To do to Done on Mar 6, 2020



 metlos removed the **status/in-progress** label on Mar 6, 2020

Assignees

-  metlos
-  mshaposhnik

Labels

kind/task severity/P1

Projects

No open projects

2 closed projects ▾

Milestone

No milestone

Development

No branches or pull requests

8 participants

