Arvados - Feature #15064

[Workbench2] Use long-lived cookies to improve login chooser defaults

04/03/2019 04:55 PM - Tom Clegg

Status:	Resolved	**Start date:**	05/14/2019
Priority:	Normal	**Due date:**	
Assigned To:	Peter Amstutz	**% Done:**	100%
Category:	Workbench	**Estimated time:**	0.00 hour
Target version:	2019-05-22 Sprint		

Description

When logging in to an active account on a cluster ("aaaaa"), set long-lived workbench cookies[1] on remote clusters ('"aaaaa"-is-my-home').

When showing the login chooser, if an "aaaaa"-is-my-home cookie[1] exists for a remote site but not the current site, use the indicated home cluster as the default option.

[1] cookie, localstorage, or whatever suitable mechanism

Subtasks:

Task # 15143: Review 15064-wb2-fed-login	**Resolved**
Task # 15207: spec implementation	**Resolved**

Related issues:

Related to Arvados - Feature #15061: Redirect users to log in with correct fe...	**Resolved**	**04/18/2019**

Associated revisions

Revision 9794db8c - 05/13/2019 02:00 PM - Peter Amstutz

Migrate arvbox api application.yml settings to config.yml

To set workbench2 url. refs #15064

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision 7488f45a - 05/13/2019 02:01 PM - Peter Amstutz

Add workbench2Url to discovery document refs #15064

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision fc5cb3c6 - 05/15/2019 02:30 PM - Peter Amstutz

Merge branch '15064-wb2-fed-login' refs #15064

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision 53a66ff4 - 05/15/2019 02:32 PM - Peter Amstutz

Remove debug logging refs #15064

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

Revision cd10c61b - 05/17/2019 08:27 PM - Peter Amstutz

Fix federated login, only create iframes for local user refs #15064

Arvados-DCO-1.1-Signed-off-by: Peter Amstutz <pamstutz@veritasgenetics.com>

History

#1 - 04/03/2019 04:56 PM - Tom Clegg

- Related to Feature #15061: Redirect users to log in with correct federated identity added

#2 - 04/03/2019 04:58 PM - Peter Amstutz

I think this needs to use local storage? I don't think we can use cookies on Workbench2 which is hosted by a static server (no cookies) and communicates with the API server (which doesn't use cookies).

#3 - 04/03/2019 06:18 PM - Tom Clegg

- Description updated

#4 - 04/03/2019 06:19 PM - Tom Morris

- Target version changed from To Be Groomed to Arvados Future Sprints

- Story points set to 1.0

#5 - 04/24/2019 05:14 PM - Tom Morris

- Related to deleted (Feature #15061: Redirect users to log in with correct federated identity)

#6 - 04/24/2019 05:16 PM - Tom Morris

- Related to Feature #15061: Redirect users to log in with correct federated identity added

#7 - 04/24/2019 05:17 PM - Tom Morris

- Assigned To set to Eric Biagiotti

- Target version changed from Arvados Future Sprints to 2019-05-08 Sprint

#8 - 05/08/2019 01:44 PM - Eric Biagiotti

- Target version changed from 2019-05-08 Sprint to 2019-05-22 Sprint

#9 - 05/08/2019 09:04 PM - Tom Clegg

Cookies/localStorage can only be read by the origin (workbench URL) that created them. Since we don't have/want a server-side component for workbench2, our choices are

1. load the remote workbench2 in an invisible iframe, and have the remote wb2's JS code read the "home cluster ID" from the #hash or ?search part of document.location and save it in a cookie/localStorage
2. do an AJAX request to an endpoint on the remote API/controller, which reads the "home cluster ID" from a request header/parameter and sets a cookie in the http response headers, which can later be retrieved by the remote workbench2 using an API endpoint which copies the cookie from request header to response body

In both cases the remote site must first verify that the "set cookie" request really originated from the home cluster, and that the home cluster is among its (the remote site's) configured remote_hosts. Otherwise, any site on the internet visited by a user would be able to change that user's default login selection.

The second (API) approach also requires the "retrieve cookie" API to recognize and trust its own workbench2 origin. The "retrieve cookie" API is unauthenticated -- after all, the whole point is to get this hint before login.

#10 - 05/09/2019 01:54 PM - Peter Amstutz

Setting cookies (notes) (copied from https://dev.arvados.org/issues/15061#note-26)

https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS?redirectlocale=en-US&redirectslug=HTTP_access_control#Requests_with_credentials

1. API server publishes CORS header "Access-Control-Allow-Credentials: true" (this enables cross-domain cookies)
2. API server adds a "browserhome" endpoint
3. To set browser home, make ajax call to https://api/browserhome?set=xyzab, withCredentials = true, this responds with "Set-Cookie: browserhome=xyzab" (should this be GET or POST?)
4. To get browser home, make ajax call to https://api/browserhome, withCredentials = true, this will read the Cookie and send it back

(I don't think we can use document.cookie because the wb2 document is served on a different domain from the API server.)

#11 - 05/09/2019 02:08 PM - Peter Amstutz

iframe / local storage

Using an invisible iframe:

https://medium.com/@crookse/cross-domain-communication-parent-window-and-child-iframe-aaf90fcb3e26

We would want it to load javascript from the same domain as wb2, but not load the full page.

However this has the same security issue as described for cookies, anybody on the internet could use it to set the preferred cluster in localStorage.

ajax / cookies

Using an API server endpoint, we could make an authorized call with the salted token. The target cluster would verify the user's identity and respond

with a cookie based on the home cluster of the API token (possibly encrypted with api server's rails session secret?)

When the user wants to log in, the wb2 would load the same endpoint, the browser would implicitly provide the cookie, the API server would read the cookie and respond with the preferred home cluster.

#12 - 05/09/2019 02:29 PM - Tom Clegg

More detailed example of the first option from note-9:

After authenticating to an active local account, or activating a previously inactive local account, workbench2 should:

1. for each entry in remoteHosts in the current cluster's discovery doc:
    1. retrieve the remote host's discovery doc to find its workbench2 url[1]
    2. create a salted token for the remote cluster
    3. make an iframe width="1" height="1" src="https://workbench2.remote.example/#setLoginHint=v2/a-b-c/defg" (where v2/a-b-c/defg is the salted token)
2. window.localStorage.setItem("loginHint", currentUserResponse.uuid)

At startup, workbench2 should:

1. check whether document.location.hash matches ^#setLoginHint=.* -- if not, just continue normal operation
2. use the provided token to get /arvados/v1/users/current from the local cluster, which will recognize it as a salted token and call the home cluster to verify it -- if this fails, log an error to the console for debugging, and stop (the page is in an invisible iframe anyway so showing an error on the page isn't useful)
3. window.localStorage.setItem("loginHint", currentUserResponse.uuid)

When showing the login chooser, workbench2 should:

1. check window.localStorage.getItem("loginHint"); if not empty, use the first 5 chars of the UUID as the default selection for the login chooser

[1] looks like we need to add this to the discovery doc; currently we only have workbenchUrl → Workbench1.ExternalURL

#13 - 05/10/2019 01:56 PM - Peter Amstutz

- Assigned To changed from Eric Biagiotti to Peter Amstutz

#14 - 05/14/2019 01:50 PM - Peter Amstutz

15064-wb2-fed-login @ commit:f35559bdd3569d836d084524ff64ed050e3c926e

Uses invisible iframe to load federated workbench2 instances using the /fedtoken route which sets the token but does not run the rest of the application.

The /fedtoken route stores credentials in local storage for that site so when user visits other workbench they won't have to log in at all.

The preferred cluster (last logged in cluster) is also set as the "home cluster" in local storage. This affects the default selection on the login screen.

Also updated search to indicate which clusters are being searched, and direct the user to do the search from the home cluster in order for multi-site search to work.

I haven't written any tests. I should probably spend some time learning how to do that.

#15 - 05/14/2019 08:19 PM - Eric Biagiotti

After clicking "Log in to x33ma with user from x1vn7" on cluster2 after logging into WB2 on cluster1, I get redirected to https://undefined/login?remote=x33ma&return_to=https://172.17.0.3:3001/token, which has an undefined home cluster.

My setup:

cluster1: x1vn7, https://172.17.0.2:3001
cluster2: x3mma, https://172.17.0.3:3001

Local storage while logged into cluster1:

homeCluster:"x1vn7"
apiToken:"v2/x1vn7-gj3su-y1m025skorrigrg/5udgomh9djj72wu2bt8aehawmsujskoeeg4zc0ybarfz906z9"

Local storage at the WB2 login page on cluster2:

homeCluster:"x1vn7"
apiToken:"v2/x1vn7-gj3su-y1m025skorrigrg/d376d54a7c603c964bd881dc4e2c22f0dd5f3861"

Some additional comments:

- In initAuth, there is some console logging that seems like debug code.
- In initAuth, can logout()(dispatch, getState, services); be dispatch(logout()) instead?

#16 - 05/14/2019 08:23 PM - Peter Amstutz

Eric Biagiotti wrote:

> After clicking "Log in to x33ma with user from x1vn7" on cluster2 after logging into WB2 on cluster1, I get redirected to https://undefined/login?remote=x33ma&return_to=https://172.17.0.3:3001/token, which has an undefined home cluster.
>
> My setup:
>
> cluster1: x1vn7, https://172.17.0.2:3001
> cluster2: x3mma, https://172.17.0.3:3001

You have both 'x33ma' and 'x3mma' which looks like a typo somewhere? But maybe it should be able to handle typos better.

#17 - 05/14/2019 08:47 PM - Eric Biagiotti

Peter Amstutz wrote:

> Eric Biagiotti wrote:
>
> > After clicking "Log in to x33ma with user from x1vn7" on cluster2 after logging into WB2 on cluster1, I get redirected to https://undefined/login?remote=x33ma&return_to=https://172.17.0.3:3001/token, which has an undefined home cluster.
> >
> > My setup:
> >
> > cluster1: x1vn7, https://172.17.0.2:3001
> > cluster2: x3mma, https://172.17.0.3:3001
>
> You have both 'x33ma' and 'x3mma' which looks like a typo somewhere? But maybe it should be able to handle typos better.

Whoops, that was a typo in my note. My cluster configs are using x33ma.

Looking at the redux state for auth when on the WB2 cluster2 login, I see:

- localCluster: "x33ma"
- homeCluster: "x1vn7"
- remoteHosts:
  - x33ma: "172.17.0.3:8000"

So remoteHosts[homeCluster] returns undefined in auth-services.ts - login.

#18 - 05/15/2019 02:03 PM - Eric Biagiotti

After getting the federation set up correctly, this works well and LGTM. Thanks!

#19 - 05/15/2019 03:15 PM - Peter Amstutz

- Status changed from New to Resolved

#20 - 05/21/2019 10:27 PM - Tom Morris

- Release set to 15