Creating the tools volume

NOTE: *Version 5.5.2 has incorporated formal support or the tools volume into the standard Managed Persistence mechanism. If you are installing 5.5.2 or upgrading to it, please skip directly to the* **Version 5.5.2 Update** *section below.*

Starting with version 5.5.1, Anaconda Enterprise has been given the ability to harness customized versions of in-browser IDEs such as VSCode, RStudio, Zeppelin, and Jupyter. These tools are expected to be hosted on special volume mounted at the `/tools` mount point, provisioned as a standard AE5 external file share.

This document provides instructions for setting up this volume. Because all of the tools live in this volume, these instructions only need to be followed once.

While any file share that meets the criteria listed in the next section will be acceptable, we offer two recommendations below to re-use existing volumes effectively.

The latest approved version of this document in PDF form can always be found at this link: tools-volume.pdf

General requirements

- We recommend that the volume have at least 10GB of space. The precise needs will depend on the number of tools and extensions installed.
- It must be accessible to all AE5 nodes.
- It must be group writable by a fixed group ID (GID). Any value of the GID, including 0, is acceptable.
- For Gravity-based clusters, it must be an NFS volume.
- For BYOK8s clusters, you can use NFS or any `PersistentVolume` with `ReadWriteOnce` or `ReadWriteMany` semantics.
- Because the volume will be written to only during installation or maintenance, it is reasonable to favor read performance over write if such a choice is available.

Once the volume is created, following the instructions provided in our documentation to add this volume to AE5. Some key aspects that must be correct:

- The mount point must be `/tools`.
- For a basic `nfs:` mount, make sure the `groupID:` value is set to the known group ID that has write access to the volume.

- For a `pvc:` mount, the `groupID:` can either be included in this section or in the `PersistentVolume` specification itself, using a `pv.beta.kubernetes.io/gid` annotation (more information here).
- During normal operation, we will set `readOnly: true` to ensure that users cannot accidentally modify installed tools. But during the installation process, we will set `readOnly: false`.
- Interrupting or removing access to the the volume from the cluster is extremely disruptive. In particular, all sessions, deployments, and jobs will have to be stopped and re-created. For this reason, we strongly recommend selecting a volume that will remain available for the life of the cluster.

Recommendation: managed persistence

If you are using the new Managed Persistence feature of Anaconda Enterprise 5, we strongly recommend re-using the MP volume to host the tools directory as well. To do so, simply create a new directory `tools` alongside the existing directories: `projects`, `environments`, and `gallery`. Give it the same ownership and permissions given these other directories.

For instance, suppose your persistence specification

```
persistence:
   projects:
     pvc: anaconda-persistence
     subPath: projects
```

Then the specification for the tool volume in the `volumes:` section will look something like this:

```
volumes:
   /tools:
     pvc: anaconda-persistence
     subPath: tools
     readOnly: true
```

One reason that we strongly recommend this approach is that it will be compatible with improvements coming in 5.5.2. In this version, the `tools` volume will be `managed` alongside `projects`, `environments`, and `gallery`. This will simplify installing and updating new tools; e.g., by eliminating the need to manually toggle between read-only and read-write mode.

Recommendation: system storage

If you can install an NFS service on the master node of a Gravity-based AE5 cluster, you can simply leverage the existing `/opt/anaconda` volume. To prepare the volume for this purpose, follow these steps:

1. Install the NFS server package for your host operating system, start the service, and configure it to automatically start on reboot.
2. Create a directory `/opt/anaconda/tools`, and give it the same permissions as `/opt/anaconda/storage`. Note the UID and GID of the directory, which will be used below.
3. Create an entry in the /etc/exports file which exports this directory to all AE5 nodes. We recommend using the `all_squash` option, and set `anonuid` and `anongid` to be equal to the UID and GID set in step 2. For example, your `/etc/exports` line might look like this:

```
/opt/anaconda/tools 10.138.148.*(rw,async,all_squash,anonuid=1000,anongid=1000)
```

4. Activate this new export by running the command `exportfs -a` as root.

With a volume such as this, the volume specification might look as follows, but of course with a different server address and possibly a different `groupID`.

```
volumes:
   /tools:
     groupID: 1000
     nfs:
       path: /opt/anaconda/tools
       server: 10.138.148.187
       readOnly: true
```

Completing the volume addition

As instructed in our documentation, certain system pods must be restarted once a new volume is added to the ConfigMap. Specifically, those instructions call for a restart of both the `workspace` and `deploy` pods. However, because this volume is only useful for user sessions, we can in fact restart only the `workspace` pod:

```
kubectl get pods | grep ap-workspace | \
```

```
    cut -d ' ' -f 1 | xargs kubectl delete pod
```

Once the workspace pod has stabilized, create a new project in AE5, using the R project type. Launch a session using either Jupyter or JupyterLab, and open a terminal window. Manually confirm that the directory `/tools` exists. If you have set `readOnly: false` in preparation for installation, make sure the directory is writable.

Removing the volume

Removing the `/tools` volume, once it has begun to be used, is very disruptive. In particular, removing the volume will interrupt all active user sessions, deployments, and scheduled jobs that were created with the volume in place. (Indeed, this is the case for any shared volume.) For this reason, we recommend leaving it in place even if its content is removed.

If the volume *must* be removed, here are the steps required.

1. Shut down *all* sessions at the AE5 level.
2. Terminate all deployments and jobs, including scheduled jobs. It is *not* sufficient to simply pause them; they must be re-created.
3. Edit the ConfigMap and remove the `/tools` volume from the `volumes:` section.
4. Restart the workspace and deploy pods.
5. Re-create any deployments and jobs at the AE5 level.

Because of the complexity of this operation, it is wise to consider scheduling time with Anaconda's support team to assist.

Version 5.5.2 Update

With Version 5.5.2, we have elected to incorporate formal support for the `/tools` volume directly into our managed peristence functionality. Specifically, `tools` is now a formal entry in the persistence configuration alongside `projects`, `environments`, and `gallery`. This approach greatly simplifies the process of managing the installation of additional IDEs. In particular, AE5 controls the read-write status of `tools` the same way as it does for `environments` and `gallery`, simplifying the management of this volume.

If you are performing a fresh installation of 5.5.2, please follow our improved installation instructions. You wil be able to activate managed persistence during the installation process.

If you are upgrading a cluster that does not have managed persistence, complete the upgrade to 5.5.2 first *before* activating managed persistence.

If you are upgrading a cluster with managed peristence, complete the upgrade first. This will preserve your existing managed persistence configuration. Once this is complete, it is straightforward to add the `tools` volume to your `peristence:` section. For example, suppose your persistence configuration looks like this:

```
persistence:
   projects:
     pvc: anaconda-persistence
     subPath: projects
   ...
```

to add support for the `tools` volume, simply add another section like so:

```
persistence:
   tools:
     pvc: anaconda-persistence
     subPath: tools
   projects:
     pvc: anaconda-persistence
     subPath: projects
   ...
```

Once the change has been made, restart the `workspace` pod so that all future sessions will be given access to the `tools` volume.

If you are upgrading a cluster with an existing tools volume, complete the upgrade to 5.5.2 first. You can continue to use the volume with no further modification. However, we do recommend migrating your configuration, so that the managed persietence framework can "adopt" your existing tools volume. To do so, you must move the volume specification `volumes:` section of the ConfigMap to the `persistence:` section. For instance, suppose your `volumes:` configuration looks like this:

```
volumes:
   /tools:
     pvc: anaconda-persistence
     subPath: projects
     readOnly: true
```

The new configuration removes this entry from `volumes:` and adds it to the `persistence:` section, like so:

```
persistence:
   tools:
     pvc: anaconda-persistence
     subPath: tools
   projects:
     ...
```

Once you have saved the changes to your ConfigMap, restart both the `workspace` and `deploy` pods so that the changes take effect. Some additional notes:

- You can make this change even if your tools volume is different than your projects, environments, and/or gallery volume.
- Do not include the `readOnly:` flag in the `persistence` section. AE5.5.2 will mount the tools volume as read-only for your normal users, and read-write for your storage manager (typically the `anaconda-enterprise` user).