Backup and Restore

Contents

Backup and Restore

Backup files, regardless of component, are stored on the PHEMI Central server. Backups are stored in the directory / backup.

The most recent backups are stored in /backup/latestOlder backups reside in directories named /backup/*timestamp*, where *timestamp* is a timestamp in Epoch time format.

Back Up the Database

You back up the PHEMI Central database using an Ansible playbook. The playbook backs up data each tenant deployed on your system, including the Accumulo database, MongoDB, and all deployed Data Processing Functions (DPFs), as well as the Ansible configuration.

You access the playbook and run it from the command line of the PHEMI Central Docker container.

1. Log on to the server running PHEMI Central as a user with sudo privileges.
2. From the PHEMI Central server command line, attach to the PHEMI Central container and access its command line.

   ```
   sudo docker attach phemi_central
   ```

   After attaching to the process, you may have to press <Enter> for the container command prompt to appear.
3. Navigate to the directory where the backup script is located.

   ```
   cd ~/agile/ansible/
   ```
4. Run the script.

   ```
   ansible-playbook -u ubuntu -i inventory/template backup-prod.yml
   ```

Restore Data from a Backup

Restore each of Accumulo, MongoDB, Data Processing Functions (DPFs), and Ansible configuration separately.

Before beginning any restore procedure, check the running processes to ensure that no instance of the agile.py process is running.

Restore Accumulo

Accumulo backups preserve the data, Accumulo splits, Accumulo configuration, and the logical time information for the Accumulo table.

You restore Accumulo information using a Python script.

Accumulo information for each PHEMI Central tenant is restored separately.

1. Log on to the server running PHEMI Central as a user with sudo privileges.
2. From the PHEMI Central server command line, attach to the PHEMI Central container and access its command line.

   ```
   sudo docker attach phemi_central
   ```

   After attaching to the process, you may have to press <Enter> for the container command prompt to appear.

3. At the container command line, become the Ubuntu user.

   ```
   su ubuntu
   ```

4. Navigate to the PHEMI Central web application folder.

   ```
   cd /home/ubuntu/agile/agile-web-ui/
   ```

5. For each tenant in your deployment, execute the restore script.

   ```
   python -m utils/backup.accumulo_backup --config
   {{ tenant_id }}.central --backup-dir /backup/timestamp/accumulo
   --restore
   ```

   where *tenant_id* is the identifier of the tenant (for example, tenant_01) and *timestamp* is either latest (for the most recent backup) or a timestamp in Epoch time format.

Restore MongoDB

The backup script uses the mongodump utility to back up MongoDB data. The mongodump utility creates a binary backup.

You use the mongorestore utility to restore from the binary backup.

MongoDB information for each PHEMI Central tenant is restored separately.

1. Log on to the server running PHEMI Central as a user with sudo privileges.
2. From the PHEMI Central server command line, attach to the PHEMI Central container and access its command line.

   ```
   sudo docker attach phemi_central
   ```

   After attaching to the process, you may have to press <Enter> for the container command prompt to appear.
3. At the container command line, become the Ubuntu user, so that you have sudo privileges.

   ```
   su ubuntu
   ```

4. Navigate to the PHEMI Central web application folder.

   ```
   cd /home/ubuntu/agile/agile-web-ui/
   ```

5. For each tenant in your deployment, execute the restore script.

   ```
   mongorestore --host tenant_host --port tenant_port
   -u tenant_username -p tenant_password
   --authenticationDatabase tenant_database /backup/timestamp/mongodb
   ```

   where:

   - *tenant_host* is the name or IP address of the host running the tenant's MongoDB server
   - *tenant_port* is the port being used for MongoDB on the tenant's host
   - *tenant_username* is the username for logging on to the tenant's MongoDB server
   - *tenant_password* is the password for logging on to the tenant's MongoDB server
   - *tenant_database* is the name of the tenant's MongoDB database.
   - *timestamp* is either latest (for the most recent backup) or a timestamp in Epoch time format.

Restore DPFs

A Data Processing Function, or DPF, is an executable piece of code that supplies the instructions for processing raw data to extract meaningful, context-specific information (such as a temperature reading or blood glucose measurement) that can be queried or exported for analysis.

The set of code that makes up a DPF is called a DPF archive. A DPF archive is delivered as a ZIP file archive. It consists of two parts: a manifest file and a code library. To associate a DPF with a data collection, the DPF archive is ``registered`` with the data collection by uploading the DPF archive. Registering the DPF is part of data collection configuration.

DPF archives are backed up into the directory /backup/*timestamp*/dpf, where *timestamp* is either latest (for the most recent backup) or a timestamp in Epoch time format.

Once retrieved, the DPF archive may need to be re-registered with the appropriate data collection in PHEMI Central. Your PHEMI Administrator can do this.

Retrieve an Ansible Playbook

The PHEMI Central platform uses Ansible to set up and configure various components during installation.

The PHEMI Central Ansible Playbooks are backed up into the directory /backup/*timestamp*, where *timestamp* is either latest (for the most recent backup) or a timestamp in Epoch time format.