Grouper book - Key questions to consider

The first step is to consider some key questions.

Section contents

Database

Grouper requires an SQL database to store group and subject data. The quick start comes with an HSQL database which is fully functional, but not considered production quality as it lacks essential functionality in areas such as maintenance and backup. If possible, you should use one of the following supported databases as they are tested with Grouper:

- Postgres (preferred)
- Oracle
- MySql (has performance issues)

Other databases supported by Hibernate cannot be used

Each database is equal as far as Grouper is concerned, with no functionality tied to, or excluded from, any specific database. You choice will be based on factors such as:

- Which database you are most experienced in/familiar with?
- Does your institution have a preference?
- Do you have existing maintenance/backup procedures in place for any of the databases?
- Do you require high availability? Different databases have different ways of achieving this
- If you are installing a database from scratch, and have no existing preference, then do you prefer open source (MySQL and Postgres) or commercial (Oracle)?

Grouper does not require a dedicated database server. It is quite happy to coexist on a server with other databases, however, you will need to consider capacity requirements of all the databases. Should this be your approach you will need to build in testing and monitoring specifically for this.

For record, if an existing database exists that is properly looked after by someone else, then my preference is for that. If I'm responsible for the database then my preference is for Postgres.

Host to run the Grouper Daemon

Grouper requires a daemon container. The code that runs (regardless of how it is started), is written in Java and requires a Java virtual machine in which to run. You will therefore need a server-type machine which will always be on (this could be a virtual machine), with Java installed on it. The load placed on the machine by Grouper will depend on the number of groups and memberships you have, the frequency with which they change and how often they are queried. This should also form a key part of your testing. As a rule-of-thumb, the daemon/API requires fewer resources than the database. You can have multiple daemons and they scale linearly (the database needs to support it though).

Application server

Grouper requires you to run the UI container. This should be separate from the daemon container and WS container. The WS container is optional though most institutions run the WS container. The UI and WS can have multiple and they scale linearly. You need a sticky load balancer for the UI and any load balancer (round robin is recommended) for the WS.

Data sources

Grouper stores data about groups in its database (called the registry). It also stores references to subjects which have privileges within groups (such as membership). In the quick start the subjects which represent people are also stored in the registry, but this is only for demonstration purposes. Grouper is designed to reference subjects in external data sources. Normally this will be an existing data source storing data about your users. The data stored will include a unique, unchanging identifier and fields that can be searched when looking for a user. Grouper supports any data source which can be queried using JDBC on JNDI. These include networked databases which listen on a port, directories which support Lightweight Directory Access Protocol (LDAP), NIS and DNS servers. Once commonly used database which is not properly query-able is Microsoft Access. This is not a networked database as a file must be opened in order to access the database, rather than a connection made to a service listening on a port.

Assuming you have data available in at least one supported data source you need to consider further questions. Typically subjects in the Grouper registry represent either people or groups. However, there is no restriction which means that subjects which are neither people not groups cannot be used. You may wish to use objects representing computers as subjects. Grouper will enable you to do this, but you may find that the data that you need to provide so that the object can be loaded as a subject using the Subject API does not necessarily fit naturally into the field names available. This is currently being considered by the Grouper development team.

Once you have identified the objects you wish to make expose as subjects, and located them in an addressable data source, you will have one or more subject data sources. Grouper supports multiple subject data sources, but bear in mind that the more sources Grouper needs to query, the more computationally intensive operations such as searches and data retrieval will become. This will in turn impact on performance. Furthermore, remember that your users will probably see any problem in the data or the speed with which it is retrieved as being a problem with Grouper rather than a subject data source. Grouper can only be as good as the slowest subject data source, and this may be unacceptably slow. It is planned that Grouper will cache sufficient data from subject sources to ensure that common operations will not suffer as a result of slowness, or multiplicity of subject data sources. However, you may wish to plan to consolidate data into a single data source and then configure Grouper to use that as the subject source. Grouper will make no effort to match and de-duplicate subject data retrieved from multiple subject sources, so consolidation may be a requirement should your data contain:

- Data for the same person, split across multiple data sources. For example, a student may also be a part-time members of staff, working in a library, and so legitimately have a record in a student records database and an employee database
- Duplicates within a single data source. This may be a result of the way data is modelled within an application. For example, it could be that a person exists as an applicant record in a Student Records application, and as a separate enrolled record - this would be a problem if you wished to include both applicants and enrolled students as subjects

Data consolidation falls outside the scope of this book, but is a very common requirement in Identity and Access Management and so will almost certainly have a larger scope that just Grouper.

Data consumers

Building and populating groups in Grouper is all very well, but you need to consider what they are going to be used for, and how. It is quite possible to query Grouper to find out a subject's group memberships, or to query a group to list the members, and connectors exist for applications such a Shibboleth to do just that (need to check this). However, many applications, both open source and commercial, would require additional development before they would be able to query Grouper. Many of these applications are already capable of querying a repository through a standard interface, and typically this is a directory that can be manipulated and queried using Lightweight Directory Access Protocol. Grouper has the ability to create and maintain groups in any LDAP-compliant directory, including Microsoft Active Directory, OpenLDAP, Novell eDirectory, Apache Directory server and IBM Tivoli directory. The chances are that you have identified one of these as your subject source anyway. Grouper also has a Webservices interface that can be used for queries (this will be covered later). An alternative approach is to provision groups from Grouper into internal storage within an application. This is an entirely valid approach, but may require some development.

Structuring your group hierarchy

As you saw in the quick start, Grouper organisers Groupers into folders. These folders are useful for 2 reasons:

- They can confer a logical structure on your groups, making them easier to manage
- They are used to assign management privileges. For example, a departmental administrator can easily be given rights over all groups in that department if they are all collected in a single folder. All that's required is for the administrator to be given rights over the folder

From this point onwards I'll assume that you've considered these questions. You may not have final answers yet, but the key issues will be clear in your mind as we explore Grouper further.