Grouper TIER instrumentation collector POC

There is a collector POC in grouper-misc will collect information sent from the grouper instrumentation daemon

This is running on the demo server, stores data in a relational database, in this case MySQL.

Discovery

The discovery runs on Amazon AWS S3

- https://s3.amazonaws.com/edu.internet2.tierinstrumentationcollector.discovery0/tierInstrumentationCollector_discovery.json
- https://s3-us-west-1.amazonaws.com/edu.internet2.tierinstrumentationcollector.discovery1/tierInstrumentationCollector_discovery.json

```
{
  serviceEnabled: true,
  endpoints: [
    {
       uri: "https://grouperdemo.internet2.edu/tierInstrumentationCollector/tierInstrumentationCollector/v1
/upload"
    }
  ]
}
```

Configuration

Configure the tierInstrumentationCollector.properties

```
# Grouper instrumentation collector uses Grouper Configuration Overlays (documented on wiki)
# By default the configuration is read from grouper.base.properties
# (which should not be edited), and the grouper.properties overlays
# the base settings.  See the grouper.base.properties for the possible
# settings that can be applied to the grouper.properties

# attribute metadata
# type needs to be one of: boolean_type, floating_type, integer_type, string_type, timestamp_type
# e.g. TierInstrumentationCollectorAttributeType
# {
#   reportFormat: 1,
#   component: "grouper",
#   institution: "Penn",
#   version: "2.3.0",
#   patchesInstalled: "api1, api2, api4, ws2",
#   wsServerCount: 3,
#   platformLinux: true,
#   uiServerCount: 1,
#   pspngCount: 1,
#   provisionToLdap: true,
#   registrySize: 12345678,
#   transactionCountMemberships: 1234567,
#   transactionCountPrivileges: 1234,
#   transactionCountPermissions: 123
# }
tic.componentName.grouper.entryVersion.1.attributeName.patchesInstalled.type = string_type
tic.componentName.grouper.entryVersion.1.attributeName.wsServerCount.type = integer_type
tic.componentName.grouper.entryVersion.1.attributeName.platformLinux.type = boolean_type
tic.componentName.grouper.entryVersion.1.attributeName.uiServerCount.type = integer_type
tic.componentName.grouper.entryVersion.1.attributeName.pspngCount.type = integer_type
tic.componentName.grouper.entryVersion.1.attributeName.provisionToLdap.type = boolean_type
tic.componentName.grouper.entryVersion.1.attributeName.registrySize.type = integer_type
tic.componentName.grouper.entryVersion.1.attributeName.transactionCountMemberships.type = integer_type
tic.componentName.grouper.entryVersion.1.attributeName.transactionCountPrivileges.type = integer_type
tic.componentName.grouper.entryVersion.1.attributeName.transactionCountPermissions.type = integer_type

########################################
## TIER Instrumentation Collector Config
########################################
# servlet url to use if the servlet isnt running (e.g. for tests)
# e.g. http://localhost:8090/tierInstrumentationCollector/tierInstrumentationCollector
tierInstrumentationCollector.servletUrl = https://grouperdemo.internet2.edu/tierInstrumentationCollector
/tierInstrumentationCollector
```

Configure the grouper.client.properties

```
########################################
## JDBC settings
########################################


# default database connection name
grouperClient.jdbc.defaultName = default


# the part between jdbc. and the last . is the name of the connection, in this case "default"
# e.g. mysql:           com.mysql.jdbc.Driver
# e.g. p6spy (log sql): com.p6spy.engine.spy.P6SpyDriver
#   for p6spy, put the underlying driver in spy.properties
# e.g. oracle:          oracle.jdbc.driver.OracleDriver
# e.g. hsqldb:          org.hsqldb.jdbcDriver
# e.g. postgres:        org.postgresql.Driver
# e.g. mssql:           com.microsoft.sqlserver.jdbc.SQLServerDriver
grouperClient.jdbc.default.driver = com.mysql.jdbc.Driver


# e.g. mysql:           jdbc:mysql://localhost:3306/grouper
# e.g. p6spy (log sql): [use the URL that your DB requires]
# e.g. oracle:          jdbc:oracle:thin:@server.school.edu:1521:sid
# e.g. hsqldb (a):      jdbc:hsqldb:dist/run/grouper;create=true
# e.g. hsqldb (b):      jdbc:hsqldb:hsql://localhost:9001/grouper
# e.g. postgres:        jdbc:postgresql://localhost:5432/database
# e.g. mssql:           jdbc:sqlserver://localhost:3280
grouperClient.jdbc.default.url = jdbc:mysql://localhost:3302/tierInstrumentationCollector?
CharSet=utf8&useUnicode=true&characterEnco
ding=utf8&autoCommit=false
grouperClient.jdbc.default.user = tic_user
grouperClient.jdbc.default.pass = **********
```

Testing

Test this TierInstrumentationCollectorRestLogicTest

```
  /**
   *
   */
  public static void runClient() {
    try {
      HttpClient httpClient = new HttpClient();
      PostMethod postMethod = new PostMethod("https://grouperdemo.internet2.edu/tierInstrumentationCollector
/tierInstrumentationCollector/v1/upload");
      String json = "{reportFormat: 1, component: \"grouper\", institution: \"Penn\", version: \"2.3.0\", "
          + "patchesInstalled: \"api1, api2, api4, ws2\", wsServerCount: 3, platformLinux: true, uiServerCount:
1, "
          + "pspngCount: 1, provisionToLdap: true, registrySize: 12345678, transactionCountMemberships:
1234567, "
          + "transactionCountPrivileges: 1234, transactionCountPermissions: 123}";
      postMethod.setRequestEntity(new StringRequestEntity(json, "application-json", "UTF-8"));
      int result = httpClient.executeMethod(postMethod);
      System.out.println(result);
    } catch (Exception e) {
      throw new RuntimeException("Error", e);
    }
  }
```