PSPNG at Penn

Note: active directory has a CN limit of 64, so at U Penn we are switching to bushy provisioning. See this doc

Make sure you are pointing to an AD domain name with active/standby load balancing or to the primary node. Or there could be CNF conflict groups in AD created.

Maintain

Run the incremental job from gsh

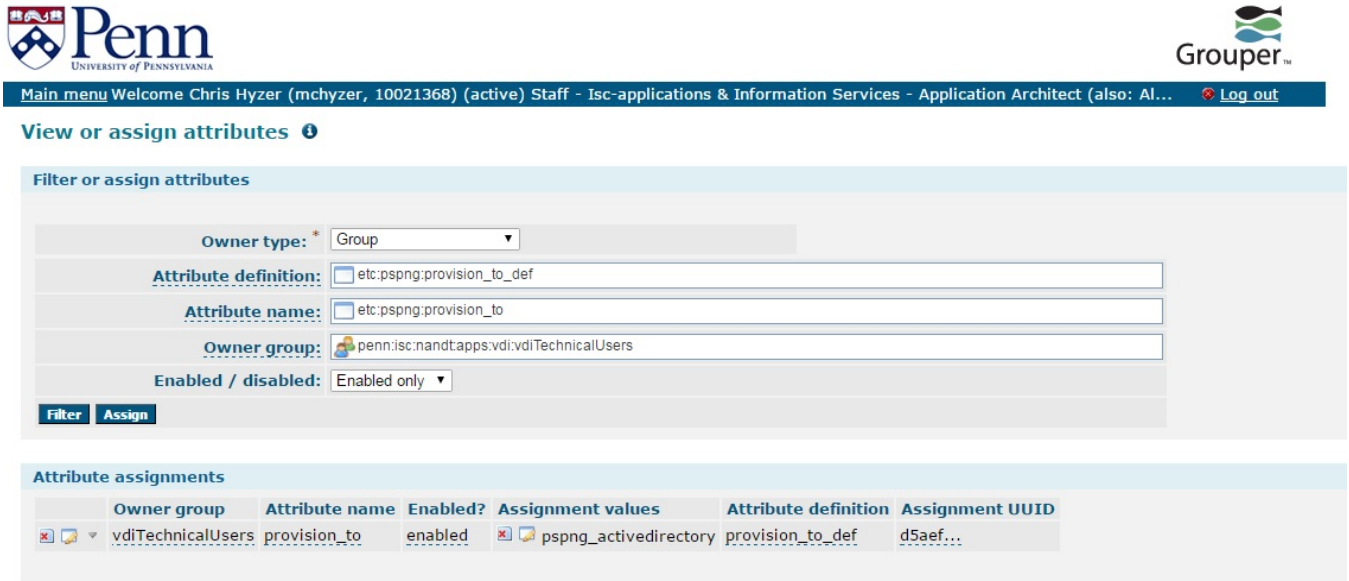
```
loaderRunOneJob("CHANGE_LOG_changeLogTempToChangeLog");
loaderRunOneJob("CHANGE_LOG_consumer_pspng_activedirectory");
```

Run full refresh from GSH (I think this works)

```
loaderRunOneJob("CHANGE_LOG_changeLogTempToChangeLog");
loaderRunOneJob("PSP_FULL_SYNC.runAtStartup");
```

Mark a group / folder as provisionable to AD

Assign an attribute at a group or folder level with a value of the config for AD



Setup

At Penn initially we would like to sync up a Grouper group with an AD group. Later on we might want to sync up more groups. We will use a flat namespace in AD.

In Grouper we have a group:

```
penn:isc:nandt:apps:vdi:vdiTechnicalUsers
```

In AD we have an OU for groups:

```
OU=Grouper,OU=LocalAuth,DC=kite-dev,DC=upenn,DC=edu
```

The group to go into AD should be:

```
CN=penn:isc:nandt:apps:vdi:vdiTechnicalUsers,OU=Grouper,OU=LocalAuth,DC=kite-dev,DC=upenn,DC=edu
```

1. Configure pspng (backup grouper-loader.properties first, then add this)

```
#note the URL should start with ldap: or ldaps: if it is
SSL.
#It should contain the server and port (optional if not default), and
baseDn,
#e.g. ldaps://ldapserver.school.edu:636/dc=school,
dc=edu
ldap.pennKiteAd.url = ldaps://someServer.upenn.edu:
636
#optional, if
authenticated

ldap.pennKiteAd.user = someUserName
#optional, if authenticated, note the password can be stored encrypted in an external
file
ldap.pennKiteAd.pass = ************



####################################

##
PSPNG

####################################

changeLog.consumer.pspng_activedirectory.class = edu.internet2.middleware.grouper.pspng.
PspChangelogConsumerShim
changeLog.consumer.pspng_activedirectory.type = edu.internet2.middleware.grouper.pspng.
LdapGroupProvisioner
changeLog.consumer.pspng_activedirectory.quartzCron = 0 * * * * ?
changeLog.consumer.pspng_activedirectory.ldapPoolName = pennKiteAd
changeLog.consumer.pspng_activedirectory.grouperIsAuthoritative = true
changeLog.consumer.pspng_activedirectory.isActiveDirectory = true
changeLog.consumer.pspng_activedirectory.memberAttributeName = member
changeLog.consumer.pspng_activedirectory.memberAttributeValueFormat = ${ldapUser.getDn()}
changeLog.consumer.pspng_activedirectory.groupSearchBaseDn = OU=Grouper,OU=LocalAuth,DC=kite,DC=upenn,
DC=edu
changeLog.consumer.pspng_activedirectory.allGroupsSearchFilter = objectclass=group
changeLog.consumer.pspng_activedirectory.singleGroupSearchFilter = (&(objectclass=group)
(gidNumber=${idIndex}))
changeLog.consumer.pspng_activedirectory.groupCreationLdifTemplate = dn: cn=${group.name}||cn: ${group.
name}||objectclass: group||gidNumber: ${group.idIndex}
changeLog.consumer.pspng_activedirectory.userSearchBaseDn = DC=kite,DC=upenn,DC=edu
changeLog.consumer.pspng_activedirectory.userSearchFilter = employeeID=${subject.id}
changeLog.consumer.pspng_activedirectory.userSearchAttributes = dn,cn,uid,mail,samAccountName, uidNumber,
objectclass,employeeID
changeLog.consumer.pspng_activedirectory.groupSearchAttributes = cn,gidNumber,samAccountName,objectclass
# This happens in the background, so should usually be enabled, and should _definitely_
# be enabled when new provisioners are added
changeLog.psp.fullSync.runAtStartup = true
```

2. Test LDAP connectivity via GSH, run a simple filter that returns a string

```
try {LdapSessionUtils.ldapSession().list("pennKiteAd", "OU=UnivOfPennsylvania,DC=kite,DC=upenn,DC=edu",
LdapSearchScope.SUBTREE_SCOPE, "(CN=mchyzer)", GrouperUtil.toArray(GrouperUtil.toList("objectClass",
"cn", "employeeid"), String.class), null); } catch (Exception e) {e.printStackTrace();}
```

3. Note, had to create the attributes for pspng

```
GrouperSession grouperSession = GrouperSession.startRootSession();
long gshTotalObjectCount = 0L;
long gshTotalChangeCount = 0L;
long gshTotalErrorCount = 0L;
StemSave stemSave = new StemSave(grouperSession).assignName("etc:pspng").
assignCreateParentStemsIfNotExist(true).assignDescription("Location for pspng-management objects.").
assignDisplayName("etc:pspng");
stem = stemSave.save();
gshTotalObjectCount++;
if (stemSave.getSaveResultType() != SaveResultType.NO_CHANGE) { System.out.println("Made change for
stem: " + stem.getName()); gshTotalChangeCount++;}
System.out.println(new java.util.Date().toString() + " Done with folders, objects: " +
gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
System.out.println(new java.util.Date().toString() + " Done with groups, objects: " +
gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
System.out.println(new java.util.Date().toString() + " Done with composites, objects: " +
gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
AttributeDefSave attributeDefSave = new AttributeDefSave(grouperSession).assignName("etc:pspng:
do_not_provision_to_def").assignCreateParentStemsIfNotExist(true).assignToGroup(true).assignToStem(true).
assignAttributeDefType(AttributeDefType.type).assignMultiAssignable(true).assignMultiValued(false).
assignValueType(AttributeDefValueType.string);
AttributeDef attributeDef = attributeDefSave.save();
gshTotalObjectCount++;
if (attributeDefSave.getSaveResultType() != SaveResultType.NO_CHANGE) {System.out.println("Made change
for attributeDef: " + attributeDef.getName()); gshTotalChangeCount++;}
AttributeDefSave attributeDefSave = new AttributeDefSave(grouperSession).assignName("etc:pspng:
provision_to_def").assignCreateParentStemsIfNotExist(true).assignToGroup(true).assignToStem(true).
assignAttributeDefType(AttributeDefType.type).assignMultiAssignable(true).assignMultiValued(false).
assignValueType(AttributeDefValueType.string);
AttributeDef attributeDef = attributeDefSave.save();
gshTotalObjectCount++;
if (attributeDefSave.getSaveResultType() != SaveResultType.NO_CHANGE) {System.out.println("Made change
for attributeDef: " + attributeDef.getName()); gshTotalChangeCount++;}
System.out.println(new java.util.Date().toString() + " Done with attribute definitions, objects: " +
gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
System.out.println(new java.util.Date().toString() + " Done with role hierarchies, objects: " +
gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
attributeDef = AttributeDefFinder.findByName("etc:pspng:do_not_provision_to_def", false);
if (attributeDef != null) { int changeCount = attributeDef.getAttributeDefActionDelegate().
configureActionList("assign"); gshTotalObjectCount+=1; if (changeCount > 0) {
gshTotalChangeCount+=changeCount; System.out.println("Made " + changeCount + " changes for actionList of
attributeDef: etc:pspng:do_not_provision_to_def");  } } else { gshTotalErrorCount++;  System.out.println
("ERROR: cant find attributeDef: 'etc:pspng:do_not_provision_to_def'"); }
attributeDef = AttributeDefFinder.findByName("etc:pspng:provision_to_def", false);
if (attributeDef != null) { int changeCount = attributeDef.getAttributeDefActionDelegate().
configureActionList("assign"); gshTotalObjectCount+=1; if (changeCount > 0) {
gshTotalChangeCount+=changeCount; System.out.println("Made " + changeCount + " changes for actionList of
attributeDef: etc:pspng:provision_to_def");  } } else { gshTotalErrorCount++;  System.out.println
("ERROR: cant find attributeDef: 'etc:pspng:provision_to_def'"); }
System.out.println(new java.util.Date().toString() + " Done with attribute actions, objects: " +
gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
System.out.println(new java.util.Date().toString() + " Done with attribute action hierarchies, objects:
" + gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known
errors (view output for full list): " + gshTotalErrorCount);
Subject subject = SubjectFinder.findByIdAndSource("GrouperSystem", "g:isa", false);
```

```
if (subject == null) { gshTotalErrorCount++; System.out.println("Error: cant find subject: g:isa:
GrouperSystem");  }
Privilege privilege = Privilege.listToPriv("stemAdmins", false);
Stem stem = StemFinder.findByName(grouperSession, "etc:pspng", false);
if (privilege != null) { if (subject != null) { if (stem != null) { boolean changed = stem.grantPriv
(subject, privilege, false);   gshTotalObjectCount++;  if (changed) { gshTotalChangeCount++;  System.out.
println("Made change for stem privilege: " + stem.getName() + ", privilege: " + privilege + ", subject:
" + GrouperUtil.subjectToString(subject)); } } else { gshTotalErrorCount++; System.out.println("ERROR:
cant find stem: 'etc:pspng'"); } }  }
System.out.println(new java.util.Date().toString() + " Done with memberships and privileges, objects: "
+ gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
AttributeDef attributeDef = AttributeDefFinder.findByName("etc:pspng:do_not_provision_to_def", false);
if (attributeDef != null) {  AttributeDefNameSave attributeDefNameSave = new AttributeDefNameSave
(grouperSession, attributeDef).assignName("etc:pspng:do_not_provision_to").
assignCreateParentStemsIfNotExist(true).assignDescription("Defines what provisioners should not process
a group or groups within a folder. Since the default is already for provisioners to not provision any
groups, this attribute is to override a provision_to attribute set on an ancestor folder. ").
assignDisplayName("etc:pspng:do_not_provision_to");  AttributeDefName attributeDefName =
attributeDefNameSave.save();  gshTotalObjectCount++;  if (attributeDefNameSave.getSaveResultType() !=
SaveResultType.NO_CHANGE) {gshTotalChangeCount++;  System.out.println("Made change for attributeDefName:
" + attributeDefName.getName()); }   } else { gshTotalErrorCount++;  System.out.println("ERROR: cant
find attributeDef: 'etc:pspng:do_not_provision_to_def'"); }
AttributeDef attributeDef = AttributeDefFinder.findByName("etc:pspng:provision_to_def", false);
if (attributeDef != null) {  AttributeDefNameSave attributeDefNameSave = new AttributeDefNameSave
(grouperSession, attributeDef).assignName("etc:pspng:provision_to").assignCreateParentStemsIfNotExist
(true).assignDescription("Defines what provisioners should process a group or groups within a folder").
assignDisplayName("etc:pspng:provision_to");  AttributeDefName attributeDefName = attributeDefNameSave.
save();  gshTotalObjectCount++;  if (attributeDefNameSave.getSaveResultType() != SaveResultType.
NO_CHANGE) {gshTotalChangeCount++;  System.out.println("Made change for attributeDefName: " +
attributeDefName.getName()); }   } else { gshTotalErrorCount++;  System.out.println("ERROR: cant find
attributeDef: 'etc:pspng:provision_to_def'"); }
System.out.println(new java.util.Date().toString() + " Done with attribute names, objects: " +
gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
System.out.println(new java.util.Date().toString() + " Done with attribute name hierarchies, objects: "
+ gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
System.out.println(new java.util.Date().toString() + " Done with attribute definition scopes, objects: "
+ gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
Set attributeAssignIdsAlreadyUsed = new HashSet();
System.out.println(new java.util.Date().toString() + " Script complete: total objects, objects: " +
gshTotalObjectCount + ", expected approx total: 8, changes: " + gshTotalChangeCount + ", known errors
(view output for full list): " + gshTotalErrorCount);
```

4. dsf