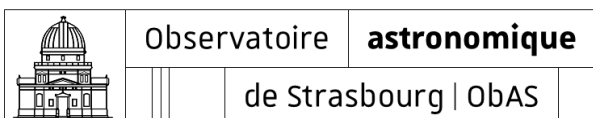


Rapport de stage



Mise en place d'une infrastructure d'authentification centralisée

Jason SECULA



IUT



UNIVERSITÉ
DE LORRAINE

IUT nancy Charlemagne
Informatique

Nancy
Charlemagne

Université de Lorraine
2 ter boulevard Charlemagne
BP 55227
54052 Nancy Cedex

Département informatique

Mise en place d'une infrastructure d'authentification centralisée

Rapport de stage de LP Administration systèmes, réseaux et applications à base de logiciels libres
Entreprise : Observatoire astronomique de Strasbourg

Jason Secula

Tuteur : Christophe Saillard

Année universitaire 2018-2019

Remerciements

Tout d'abord, je tiens à remercier l'ensemble de l'équipe de l'Observatoire astronomique de Strasbourg pour m'avoir accueilli et m'avoir permis d'effectuer mon stage lors de ces trois derniers mois.

Je tiens aussi à remercier mes encadrants, Christophe Saillard et Thomas Keller pour le temps qu'ils m'ont consacré tout au long du stage et le soutien qu'ils m'ont apporté dans mes démarches.

J'aimerais aussi remercier mon référent pendant ce stage, Lucas Nussbaum, pour m'avoir permis d'en apprendre davantage sur le monde du logiciel libre lors de ses cours et d'avoir fait partie des personnes qui m'ont donné envie de poursuivre dans l'administration systèmes.

Finalement, je remercie Philippe Dosch pour l'attention qu'il a pu m'apporter lors de certaines démarches personnelles concernant mon orientation cette année, ainsi que le suivi dont il a fait preuve et sa disponibilité tout au long de la formation.

Table des matières

1. Introduction	8
2. L'observatoire astronomique de Strasbourg	9
2.1 Présentation de l'observatoire	9
2.2 Le service informatique	9
3. Déroulement du stage	10
3.1 Contexte	10
3.2 Problématique	11
3.3 Description du sujet	11
3.4 Environnement de travail	12
3.5 Moyens de documentation	12
3.6 Contacts	12
4. Travail réalisé	13
4.1 Mise en place d'un annuaire LDAP	13
4.1.1 OpenLDAP	13
4.1.2 Le format LDIF	13
4.1.3 FusionDirectory	14
4.1.4 Prise en main et création de l'annuaire OpenLDAP	14
4.1.5 Mise en place d'un outil de gestion pour l'annuaire	15
4.2 Sécurisation de l'annuaire	18
4.2.1 Chiffrement des communications avec LDAPS	19
4.2.2 Politique d'accès aux données	20
4.3 Synchronisation des utilisateurs de l'université	20
4.3.1 L'outil de synchronisation LSC	20
4.3.2 Création d'un schéma et d'attributs personnalisés	21
4.3.3 Synchronisation de l'annuaire	21
4.4 Performances et disponibilité du service d'annuaire	22
4.4.1 Le moteur de réplication syncrepl	23
4.4.2 L'équilibreur de charge HAProxy	23
4.4.3 Corosync et Pacemaker	23
4.4.4 Zabbix	23
4.4.5 Création d'annuaires consumer avec réplication	24
4.4.6 Création d'un cluster HAProxy	26
4.4.7 Supervision avec Zabbix	27

4.5 Authentification des services et des outils	28
4.5.1 Création d'un utilisateur pour l'authentification	28
4.5.2 Configuration de l'authentification GLPI	29
4.5.3 Configuration de l'authentification GRR	30
4.5.4 Configuration de l'authentification Zabbix	31
4.6 Authentification des postes et serveurs.....	31
4.6.1 SSSD	32
4.6.2 Authentification sur les postes et serveurs avec SSSD	32
4.7 Déploiement de la configuration des machines	33
4.7.1 Ansible.....	33
4.7.2 Création de playbook Ansible.....	33
4.7.3 Déploiement avec Ansible	34
5. Conclusion.....	35
6. Bibliographie	36
7. Annexes	37
7.1 Fichier de configuration LSC.....	37
7.2 Fichier de configuration SSSD d'un serveur	46
7.3 Extrait du template Ansible pour la configuration SSSD	47

Index des illustrations

Illustration 1: Schéma représentant le contexte.....	10
Illustration 2: Schéma de l'authentification centralisée	11
Illustration 3: Organisation de l'annuaire	14
Illustration 4: Page de vérification de l'assistant de configuration de FusionDirectory.....	16
Illustration 5: Écran d'accueil de FusionDirectory	17
Illustration 6: Signalement d'un problème de compatibilité dans FusionDirectory	18
Illustration 7: Schéma de synchronisation LSC	22
Illustration 8: Capture d'écran du statut du cluster HAProxy	27
Illustration 9: Capture d'écran de Zabbix montrant les machines supervisés.....	28
Illustration 10: Écran de configuration de l'authentification LDAP de GLPI.....	29
Illustration 11: Capture d'écran de test de la configuration LDAP de GLPI	30
Illustration 12: Capture d'écran de configuration LDAP de GRR	30
Illustration 13: Capture d'écran de configuration LDAP de Zabbix	31
Illustration 14: Exécution d'un playbook Ansible.....	34

1. Introduction

Dans le cadre de ma formation et pour la validation de ma licence professionnelle, j'ai eu l'opportunité d'effectuer un stage en administration systèmes et réseaux au sein du service informatique de l'observatoire astronomique de Strasbourg. Pendant trois mois, j'ai travaillé avec mon tuteur M. Saillard Christophe et son collègue M. Keller Thomas, responsables du service informatique.

Ce stage a particulièrement attiré mon attention parce qu'il est centré sur la mise en place d'une authentification centralisée utilisant la technologie LDAP que nous avons vu brièvement lors des cours de cette année et qui m'a fortement intéressé. Par ailleurs, étant fasciné par l'astronomie, l'observatoire astronomique m'a permis d'en apprendre certaines choses lors de séminaires.

Ce rapport a pour but de résumer et retransmettre tout le travail que j'ai réalisé et ce que j'ai pu apporter à l'observatoire pendant la durée de mon stage.

Dans un premier temps, je présenterai l'observatoire astronomique de Strasbourg et son service informatique dans lequel j'ai travaillé.

Je détaillerai ensuite le déroulement du stage en commençant par expliquer le contexte, l'environnement de travail, les moyens de documentations utilisés ainsi que les contacts que j'ai pu avoir dans le cadre du stage.

Puis, j'expliquerai de façon chronologique tout le travail réalisé en donnant auparavant une description des technologies utilisés avant d'expliquer la manière dont je les ai utilisés.

Finalement, j'adresserai une conclusion de tout ce que le stage m'a apporté au niveau professionnel et personnel.

2. L'observatoire astronomique de Strasbourg

2.1 Présentation de l'observatoire

L'observatoire astronomique de Strasbourg est un Observatoire des Sciences de l'Univers, une école interne de l'Université de Strasbourg, ainsi qu'une Unité Mixte de Recherche entre l'Université de Strasbourg et le CNRS. L'observatoire héberge le centre de données astronomique de Strasbourg (CDS) labellisé « Infrastructure de recherche » par le Ministère de l'Enseignement Supérieur et de la Recherche.

Composante à part entière de l'Unistra, l'Observatoire astronomique de Strasbourg gère la parcours astrophysique (M2) du Master Physique spécialité Astrophysique, hébergé par la faculté de physique et ingénierie.

Les usagers de l'observatoire sont très variés, il y a le personnel (développeurs, documentalistes, chercheurs, administratifs), des étudiants de Master et des personnes extérieurs (visiteurs). Ce qui fait un total d'environ 100 personnes.

2.2 Le service informatique

J'ai effectué mon stage au sein du service informatique de l'observatoire. Il est responsable de la gestion des infrastructures de l'observatoire. Le parc informatique qu'il gère comprend :

- 60 serveurs (40 serveurs physiques et 20 serveurs virtuels)
- 50 équipements réseaux
- 1 salle de ressources avec 15 postes sous Linux
- Environ 185 postes de travail (80 % sous Linux)

Sur l'ensemble de ces équipements, le service informatique gère aussi plusieurs outils et services (accès VPN, support avec GLPI, réservation de salles et d'équipement avec GRR, supervision avec Zabbix...) mis à la disposition des usagers de l'observatoire.

3. Déroulement du stage

Dans cette section, j'aborde les aspects du stage qui vont me permettre d'introduire le travail que j'ai réalisé. Premièrement, le contexte dans lequel le stage se déroule et la description de la problématique. Puis, la description du sujet de stage et de l'environnement de travail. Ensuite, les différents moyens de documentation que j'ai utilisé ainsi que les personnes avec lesquelles j'ai été en contact lors de doutes ou problèmes.

3.1 Contexte

Les usagers des services de l'observatoire comprennent le personnel de l'observatoire, des étudiants de Master, des stagiaires ainsi que des visiteurs. Chaque usager dispose d'un accès avec un identifiant à divers outils et services internes à l'observatoire. Ces outils et services disposent chacun d'une base de données locale d'utilisateurs. Les identifiants présents dans ces bases de données ne sont pas forcément identiques d'une base à l'autre. En plus de cela, le personnel et les étudiants disposent d'un identifiant personnel unique à l'Université de Strasbourg pour utiliser les services de l'Université (ENT, messagerie, accès wifi...) au sein de l'observatoire. Pour l'ensemble de ses services, l'université utilise un annuaire LDAP regroupant les informations de ses utilisateurs et pouvant les authentifier sur chacun des services.

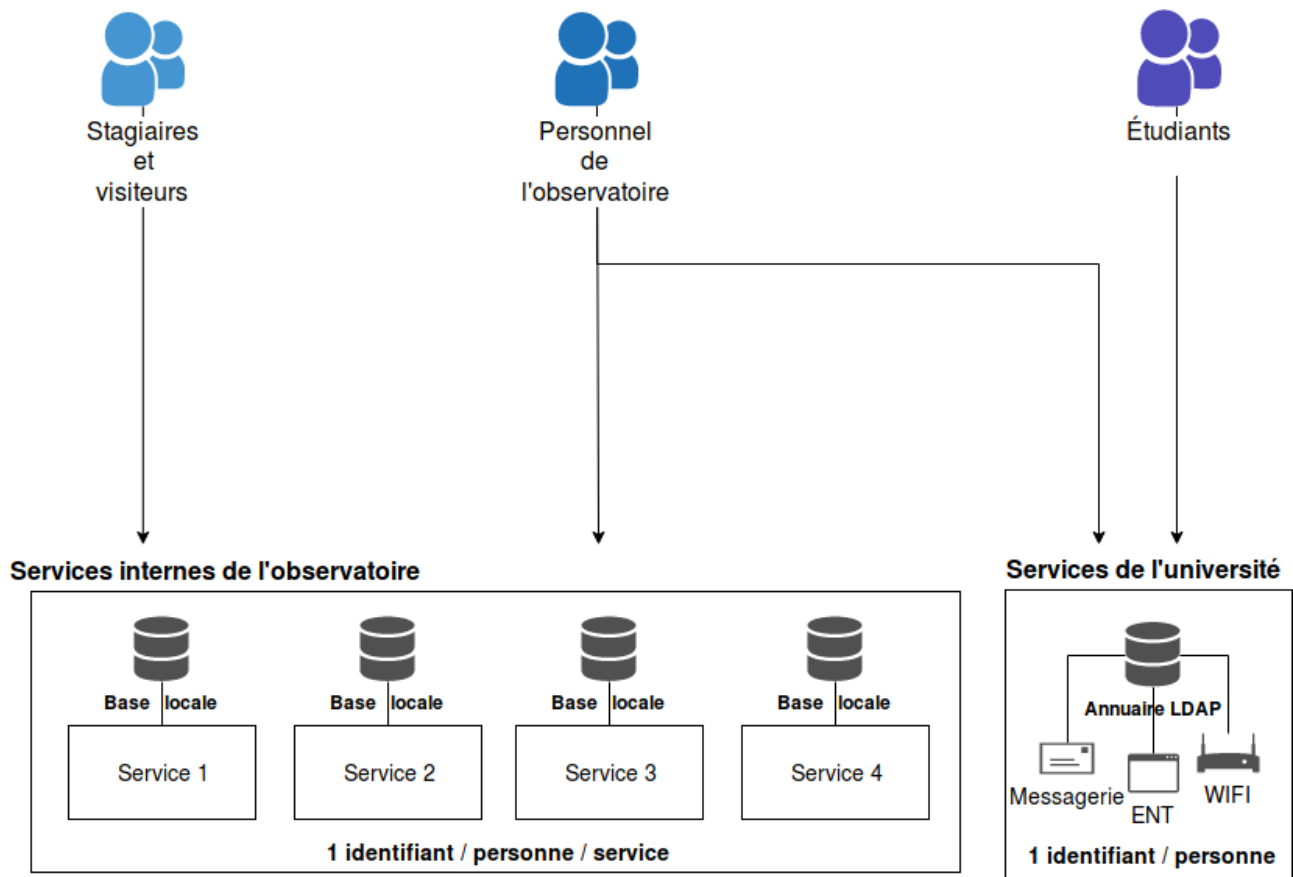


Illustration 1: Schéma représentant le contexte

3.2 Problématique

La gestion des identifiants sur chacun des outils et services de l'observatoire devient de plus en plus difficile avec l'augmentation du nombre de ses services ainsi que le nombre d'utilisateurs. L'utilisation des outils et services par

les usagers devient aussi problématique avec la présence de différents identifiants à mémoriser pour l'ensemble des outils et services.

3.3 Description du sujet

Le stage proposé est centré sur la mise en place d'une infrastructure d'authentification centralisée utilisant la technologie LDAP. L'infrastructure d'authentification doit avoir les identifiants des usagers de l'observatoire synchronisé avec les identifiants existant sur l'annuaire de l'Université de Strasbourg. Il doit aussi permettre la création manuelle d'utilisateurs locaux à l'observatoire et ne disposant pas d'identifiant à l'université (stagiaire, visiteurs). L'objectif final est de pouvoir authentifier les usagers sur les services et outils de l'observatoire ainsi que sur les postes et serveurs à l'aide des identifiants présents dans l'annuaire.

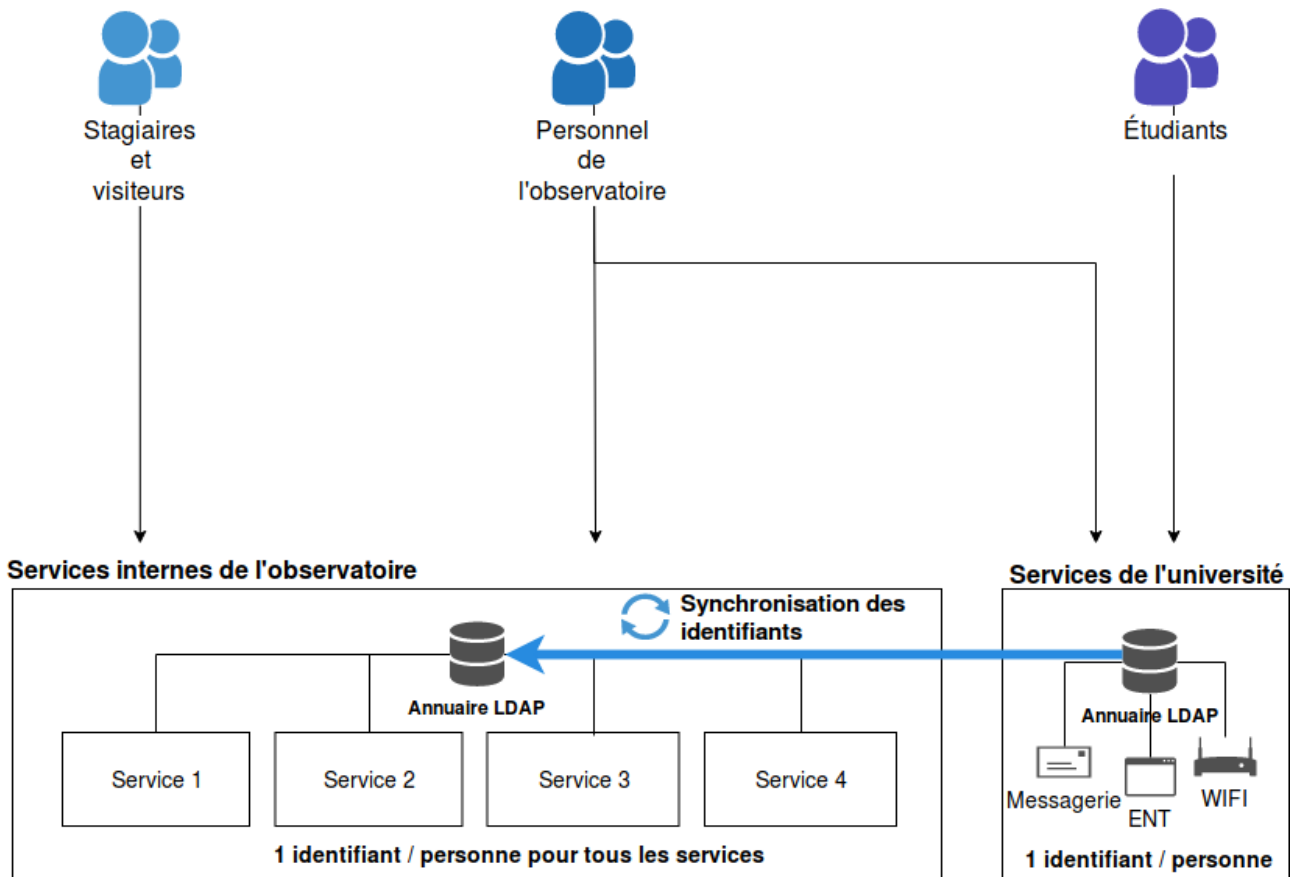


Illustration 2: Schéma de l'authentification centralisée

3.4 Environnement de travail

Durant la durée du stage, j'ai rejoint le bureau du service informatique situé dans le bâtiment sud de l'observatoire. Ce bureau est occupé par mon tuteur M. Saillard Christophe et son collègue M. Keller Thomas, responsables du service informatique. Ceci m'a été très bénéfique puisque j'avais la possibilité de leur poser des questions si besoin concernant les aspects spécifiques du stage. Par ailleurs, ils ont mis à ma disposition un espace de travail avec un poste sur lequel j'ai installé une version de la distribution Linux Ubuntu et où j'étais administrateur.

Au fur à mesure du stage, mon tuteur a mis à ma disposition des machines virtuelles sous VMware. Pour administrer ces machines à distance, j'avais un accès à l'interface de virtualisation vSphere ainsi qu'un accès SSH. Cela m'a permis d'avoir un contrôle total sur les machines pour effectuer mon travail.

3.5 Moyens de documentation

Lors du stage, j'ai consulté les documentations des sites officiels des outils que j'ai utilisés, ainsi que des « how-to » présents sur internet expliquant la mise en place ou la configuration de certains des outils. J'avais aussi à ma disposition des magazines Linux dont un expliquant la configuration d'un outil spécifique que j'ai utilisé (Ansible).

3.6 Contacts

Pour m'aider dans l'exécution de mes tâches ou me conseiller lors du stage, j'étais au quotidien en présence de mon tuteur de stage Christophe Saillard et de son collègue Thomas Keller qui ont répondu à mes questions techniques concernant la mise en place de l'annuaire LDAP.

Par la suite, j'ai pu rencontrer l'administrateur de l'annuaire LDAP de l'Université de Strasbourg, Alain Zamboni. Lors d'un entretien, il m'a apporté beaucoup d'éclaircissement sur la méthode à utiliser pour synchroniser les données entre l'annuaire de l'université et l'annuaire que j'étais en train de mettre en place. Il m'a aussi suggéré certains outils que je pouvais utiliser et m'a fourni un accès en lecture à l'annuaire de l'université avec un identifiant pour l'observatoire. Enfin, il m'a présenté un document qui me donnait certaines informations techniques sur l'annuaire de l'université ainsi le type de donnée s'y trouvant ce qui m'a été d'une très grande aide pour sélectionner les données à récupérer.

Puis en dehors du personnel de l'observatoire ou de l'université, j'ai pris contact avec les développeurs de certains outils en soumettant des « issues » sur des plateformes comme GitHub ou Gitlab et aussi par mail pour demander de l'aide ou signaler des problèmes dans l'utilisation des outils.

4. Travail réalisé

Cette section a pour objectif de détailler tout le travail que j'ai réalisé pendant la durée du stage ainsi que les technologies que j'ai utilisées pour effectuer mes tâches. Pour chaque partie, je vais d'abord introduire les technologies que j'ai utilisées avant de décrire le travail réalisé.

4.1 Mise en place d'un annuaire LDAP

J'ai utilisé la solution libre OpenLDAP pour mettre en place un annuaire LDAP sur un serveur avec la distribution Ubuntu. Pour gérer l'annuaire, j'ai utilisé le format LDIF et j'ai mis en place un outil de gestion d'annuaire LDAP du nom de FusionDirectory.

4.1.1 OpenLDAP

OpenLDAP est une implémentation libre du protocole LDAP par la fondation OpenLDAP. Il permet de réaliser un annuaire capable de stocker des informations de n'importe quelle nature et de les ranger de manière hiérarchique. Il est en général principalement utilisé pour l'authentification des utilisateurs d'un service ou d'une application de manière centralisée.

4.1.2 Le format LDIF

Le format LDIF (LDAP DATA Interchange Format) est un format standardisé d'échange de données conçu au début des années 1990 par Gordon Good, il permet de représenter les données contenues dans un annuaire LDAP. Il y est aussi possible de représenter des opérations sur les données. Les enregistrements au format LDIF sont représentés comme un ensemble de couples d'attributs et de valeurs. Chaque enregistrement doit être séparé d'un autre par une ligne vide.

```
1 dn: uid=alice,ou=people,o=annuaire,dc=obas-ldap
2 uid: alice
3 uidNumber: 1003
4 gidNumber: 1003
5 cn: alice
6 loginShell: /bin/bash
7 homeDirectory: /home/alice
8 userPassword: {SSHA}VYeXUL7smgZ/u+JOqrUgVpNVEPXCRtfk
9 objectClass: inetOrgPerson
10 objectClass: posixAccount
```

Exemple de création d'une utilisatrice du nom d'Alice en utilisant le format LDIF

4.1.3 FusionDirectory

FusionDirectory est un outil libre écrit en PHP de gestion des données pour les annuaires LDAP. Il inclut des fonctionnalités spécifiques à l'Enseignement Supérieur et Recherche et possède une interface beaucoup plus simple d'utilisation que le format LDIF. Ses usages principaux sont pour des actions quotidiennes (ajouter/éditer/supprimer un utilisateur, une machine ou des droits Sudo). Il est possible d'y ajouter des plugins supplémentaires afin d'étendre le type de donnée que l'outil peut gérer. Ces types de données peuvent être

spécifiques à une application (AutoFS, Dovecot, Squid...) ou spécifiques à l'enseignement supérieur (schéma Supann).

4.1.4 Prise en main et création de l'annuaire OpenLDAP

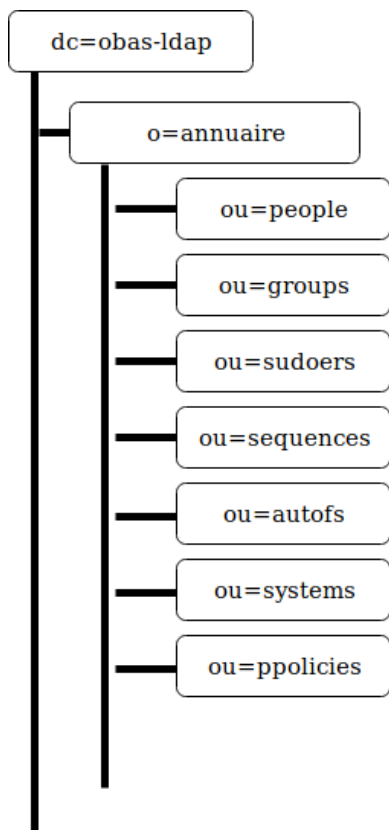


Illustration 3: Organisation de l'annuaire

J'ai commencé par me renseigner sur le logiciel OpenLDAP à partir de la documentation officielle disponible sur internet. Par la suite, j'ai décidé de commencer la réalisation d'un annuaire LDAP avec des utilisateurs et des groupes dans un environnement de test. Pour la création de l'environnement de test, j'ai utilisé les outils Vagrant et VirtualBox que j'ai installé sur mon poste. Cela m'a permis de me familiariser au fonctionnement d'OpenLDAP et au format LDIF avant de pouvoir commencer la mise en production d'un annuaire sur une machine virtuelle déployée sur un hyperviseur de l'observatoire.

Après m'être renseigné suffisamment sur OpenLDAP et avoir réalisé mes propres tests, je me suis rendu compte rapidement qu'il était important de définir une structure d'organisation générale avant de commencer l'ajout de données.

Pour cela, j'ai créé une organisation « annuaire » qui allait contenir plusieurs unités d'organisations (voir illustration 3) en fonction des données qui y seront stockées (utilisateurs, groupes, machines, droits Sudo...).

J'ai commencé par créer deux unités d'organisations, « people » et « groups ». D'autres unités d'organisations ont été créées au fur et à mesure de la manifestation du besoin de stocker d'autres types de données.

```
1 dn: o=annuaire, dc=obas-ldap
2 o: annuaire
3 objectClass: top
```

```

4  objectClass: organization
5
6  dn: ou=people,o=annuaire,dc=obas-ldap
7  ou: people
8  objectClass: organizationalUnit
9
10 dn: ou=groups,o=annuaire,dc=obas-ldap
11 ou: groups
12 objectClass: organizationalUnit

```

Instructions au format LDIF permettant la création des unités d'organisations de l'annuaire

4.1.5 Mise en place d'un outil de gestion pour l'annuaire

Après la définition de la structure d'organisation de l'annuaire et sa création, il a fallu que je mette en place un outil pour pouvoir gérer l'annuaire lors d'un usage quotidien. Cet outil ne devait pas nécessiter de connaître les subtilités du format LDIF pour son utilisation. Après une suggestion d'Alain Zamboni, je me suis renseigné sur l'outil FusionDirectory que j'ai finalement décidé d'utiliser.

J'ai installé et configuré l'outil FusionDirectory sur la même machine virtuelle que le serveur OpenLDAP. L'installation de l'outil nécessitait d'installer un serveur web et une version de PHP supérieur à la version 5.6 ainsi que plusieurs modules PHP (voir illustration 4) pour son fonctionnement.

Illustration 4: Page de vérification de l'assistant de configuration de FusionDirectory

Une fois le nécessaire installé en utilisant les commandes apt-get et la vérification de la configuration PHP dans le fichier php.ini, je suis passé à la configuration de FusionDirectory à l'aide de son interface web et j'ai suivi les instructions qui étaient données dans l'assistant de configuration.

Après la configuration de l'outil de base, j'ai installé plusieurs plugins en utilisant la commande apt-get et en installant les schémas fournis par ces plugins dans l'annuaire LDAP à l'aide de la commande fourni par l'outil Fusiondirectory (fusiondirectory-insert-schema) qui permet d'insérer les schémas sans passer par une conversion au format LDIF. Ces plugins, permettent de pouvoir gérer les données concernant les droits sudo, les partages autofs, les machines, les politiques de mots de passe, le mélange des groupes POSIX avec les groupes de noms (voir illustration 5).

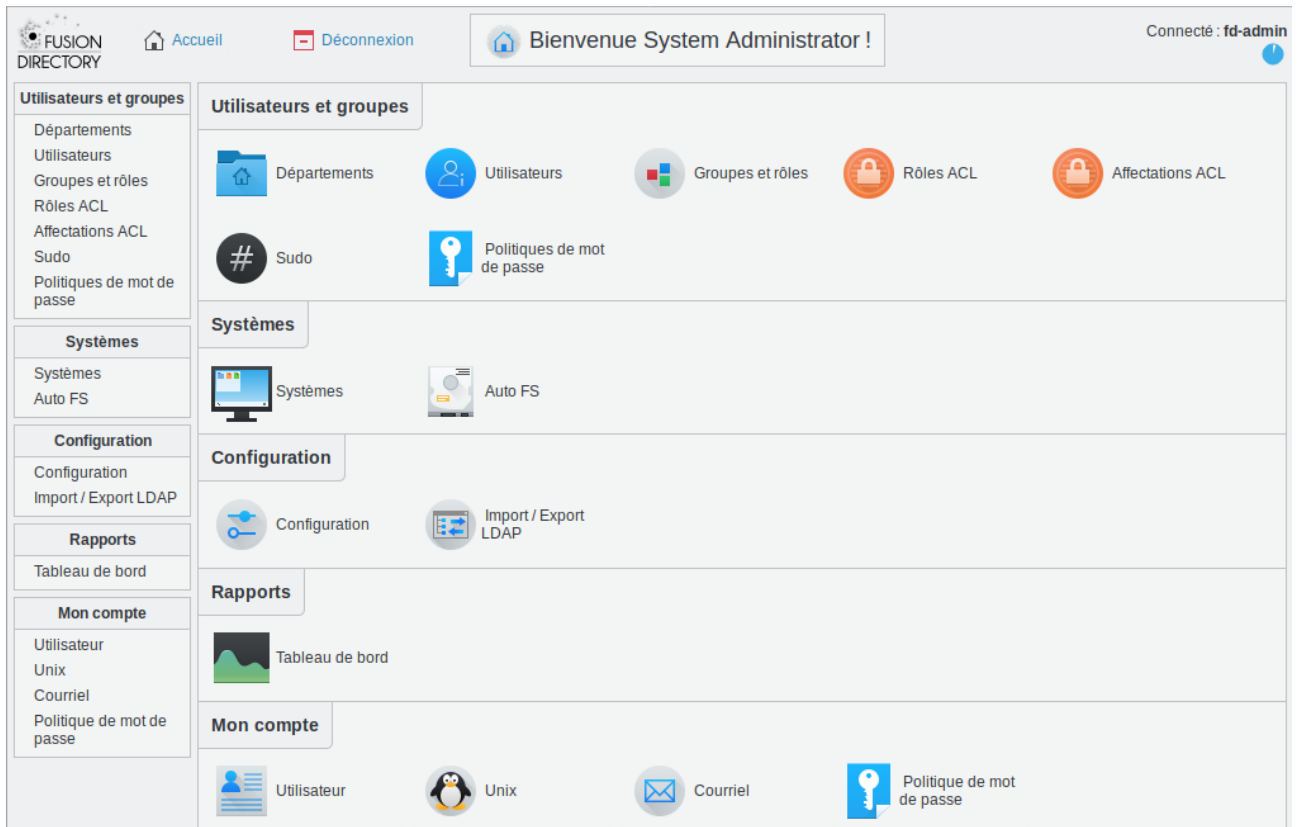



Illustration 5: Écran d'accueil de FusionDirectory

Après la mise en place de l'outil FusionDirectory, j'ai découvert quelques problèmes de compatibilité entre certains plugins provoquant une erreur lors de certaines actions. Après avoir fait quelques tests dans un environnement isolé pour déterminer l'origine du problème, je l'ai signalé aux développeurs de l'outil via une issue sur le site Gitlab de l'équipe de développement (voir illustration 6).

Open Opened 22 hours ago by  Jason Secula

Close issue

New issue

Compatibility error between sudo and mixedgroup plugins

Description

There is a compatibility error between sudo and mixedgroup plugin triggering an error when adding, deleting or editing a group with 2+ users in it already. The error only happens when sudo and mixedgroup plugins are installed altogether. Also it sometimes takes multiples retry to successfully edit the group once the error is showing up.

Distribution Name and Version

Ubuntu 18.04 bionic

FusionDirectory Version

1.3

Plugin with the defect

Sudo & Mixedgroup

PHP version used

7.2


Origin of php packages

Distribution packages

Steps to Reproduce

1. Install both plugins and use rfc2307bis schema so you can use mixedgroup

Todo


Assignee  **bmortier**
@bmortier


Milestone
None




Time tracking
No estimate or tim

Due date
None

Labels
Bugs

Confidentiality
 Not confidenti

Lock issue
 Unlocked

3 participants
  

Notifications

Reference: fusiond

Illustration 6: Signalement d'un problème de compatibilité dans FusionDirectory

4.2 Sécurisation de l'annuaire

Après la fin de la mise en place de l'annuaire, j'ai dû sécuriser les communications entre le serveur et les clients. Pour la sécurisation des communications, j'ai utilisé le chiffrement via le protocole LDAP over SSL (LDAPS). J'ai aussi dû définir une politique d'accès aux données de l'annuaire pour éviter que toutes les données soient accessibles par tous.

4.2.1 Chiffrement des communications avec LDAPS

Pour chiffrer les communications de l'annuaire, j'ai choisi d'utiliser le protocole LDAPS plutôt que le protocole LDAP avec TLS car même s'ils sont très similaires, le protocole LDAP ne démarre pas de connexion sécurisée dès le début de la communication mais par la suite contrairement au protocole LDAPS. Cela peut poser problème dans la mesure où le premier peut envoyer les identifiants en clair si la requête ne précise pas que l'on souhaite démarrer une connexion TLS sécurisée.

Afin de pouvoir activer le protocole LDAPS, j'ai dû faire une demande à mon tuteur pour obtenir un certificat pour le nom de domaine du serveur LDAP (obas-ldap-master). Une fois la clé privée, le certificat du nom

de domaine et le certificat de l'autorité de certification sur le serveur, j'ai modifié les droits d'accès à ces fichiers pour les rendre lisibles par le serveur LDAP en donnant l'accès en lecture au groupe « openldap ».

Ensuite, j'ai rajouté dans la configuration de l'annuaire les chemins pointant vers les fichiers de certificat nécessaire pour établir une connexion TLS sécurisée.

```
1 dn: cn=config
2 changetype: modify
3 add: olcTLSCACertificateFile
4 olcTLSCACertificateFile: /etc/ssl/certs/DigiCertCA.crt
5 -
6 add: olcTLSCertificateFile
7 olcTLSCertificateFile: /etc/ssl/certs/obas-ldap-master_astro_unistra_fr.crt
8 -
9 add: olcTLSCertificateKeyFile
10 olcTLSCertificateKeyFile: /etc/ssl/private/obas-ldap-master.unistrafr.key
```

Configuration au format LDIF des chemins pointant vers les fichiers de certificats

Finalement, j'ai modifié le fichier « /etc/default/slapd » pour ajouter le protocole ldaps et supprimer l'utilisation du protocole ldap non sécurisé.

```
1 # slapd normally serves ldap only on all TCP-ports 389. slapd can also
2 # service requests on TCP-port 636 (ldaps) and requests via unix
3 # sockets.
4 # Example usage:
5 # SLAPD_SERVICES="ldap://127.0.0.1:389/ ldaps:/// ldapi:///"
6 SLAPD_SERVICES="ldaps:/// ldapi:///"
```

Extrait du fichier de configuration /etc/default/slapd

4.2.2 Politique d'accès aux données

Lors de la définition de la politique d'accès aux données, j'ai défini les droits que devait avoir chaque type d'utilisateur sur les données (voir tableau 1).

	Accès mot de passe	Accès au reste des données
Utilisateur authentifié	Lecture/écriture de son propre mot de passe mais pas des autres	Lecture
Utilisateur anonyme	Pas d'accès	Pas d'accès
Utilisateur admin	Tous les accès en lecture et écriture	Tous les accès en lecture et écriture

Tableau 1: Politique d'accès aux données de l'annuaire

Dans un premier temps, j'ai autorisé les utilisateurs authentifiés à lire et écrire le mot de passe pour leur compte seulement. Ils avaient aussi le droit en lecture aux autres données de l'annuaire ne concernant pas un mot de passe.

Puis, j'ai ensuite interdit l'accès aux données de l'annuaire aux utilisateurs anonymes (qui ne sont pas authentifiés). Ces utilisateurs devront d'abord s'authentifier avant de pouvoir faire n'importe quelle action sur l'annuaire (lecture, écriture).

Finalement, l'utilisateur admin a tous les droits en lecture et écriture sur l'annuaire. Ces droits sont ceux par défaut pour l'utilisateur admin (le root DN / super utilisateur).

Dans les listes d'accès LDAP, tout ce qui n'est pas explicitement précisé comme étant autorisé est automatiquement refusé.

4.3 Synchronisation des utilisateurs de l'université

Une fois l'annuaire créé, j'ai dû synchroniser les identifiants des usagers de l'observatoire à partir de l'annuaire de l'Université de Strasbourg. Pour cela, j'ai utilisé l'outil « LDAP Synchronization Connector » qui m'avait aussi été suggéré par Alain Zamboni lors d'un entretien.

4.3.1 L'outil de synchronisation LSC

LSC est un outil libre codé en Java qui permet de synchroniser des données entre un serveur LDAP et n'importe quelle source de données, cela inclut n'importe quelle base de données avec un connecteur JDBC, un autre serveur LDAP, des fichiers de données ou des APIs.

4.3.2 Création d'un schéma et d'attributs personnalisés

Avant de pouvoir synchroniser les données, j'ai dû créer un attribut personnalisé pour pouvoir reconnaître un utilisateur synchronisé d'un utilisateur créé manuellement (obs-datasource). Il me fallait aussi un attribut correspondant à l'attribut udsDirectoryId créé par l'université et présent pour chaque identifiant de l'annuaire de l'université. Ce dernier permet à l'université d'identifier un utilisateur en cas de changement d'identifiant et doit être utilisé en tant qu'attribut pivot pour la synchronisation vu qu'il est immuable. La création de schéma personnalisé contenant des classes d'objets et des attributs personnalisés nécessite un numéro unique identifiant l'organisation qui crée cet attribut. Ce numéro nous a été fourni après une requête à l'IANA.

```
1 dn: cn=obs,cn=schema,cn=config
2 objectClass: olcSchemaConfig
3 cn: obs
4 olcAttributeTypes: ( 1.3.6.1.4.1.53904.2.1.1 NAME 'obs-datasource'
5   DESC 'Used to define datasource when syncing the data.'
6   EQUALITY caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch
7   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
8 olcAttributeTypes: ( 1.3.6.1.4.1.53904.2.1.2 NAME 'udsDirectoryId'
9   DESC 'Used to identify a user in case of uid changes.'
10  EQUALITY caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch
11  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE )
12 olcObjectClasses: ( 1.3.6.1.4.1.53904.2.2.1 NAME 'obs-custom'
13   DESC 'Custom attributes used by Observatoire de Strasbourg'
14   AUXILIARY MAY ( obs-datasource $ udsDirectoryId ) )
```

Schéma personnalisé obs au format LDIF

4.3.3 Synchronisation de l'annuaire

Avant de commencer la configuration de la synchronisation des données, j'ai dû faire des recherches sur l'annuaire de l'université pour savoir quelles données (attributs) je devais récupérer pour chaque identifiant. Pour cela, j'ai utilisé le document fourni par Alain Zamboni qui décrivait le contenu de l'annuaire. J'ai aussi utilisé l'outil ldapvi pour parcourir les données de l'annuaire de l'université et voir les données directement.

Après avoir fait une liste des données à récupérer, j'ai installé l'outil LSC et ses dépendances (Java) sur le serveur hébergeant l'annuaire. Pour modifier la configuration de LSC, je devais me référer à la documentation officielle, car les modifications de la configuration n'étaient pas simples à réaliser et demander de bien se renseigner à partir de la documentation avant (voir annexe page 39).

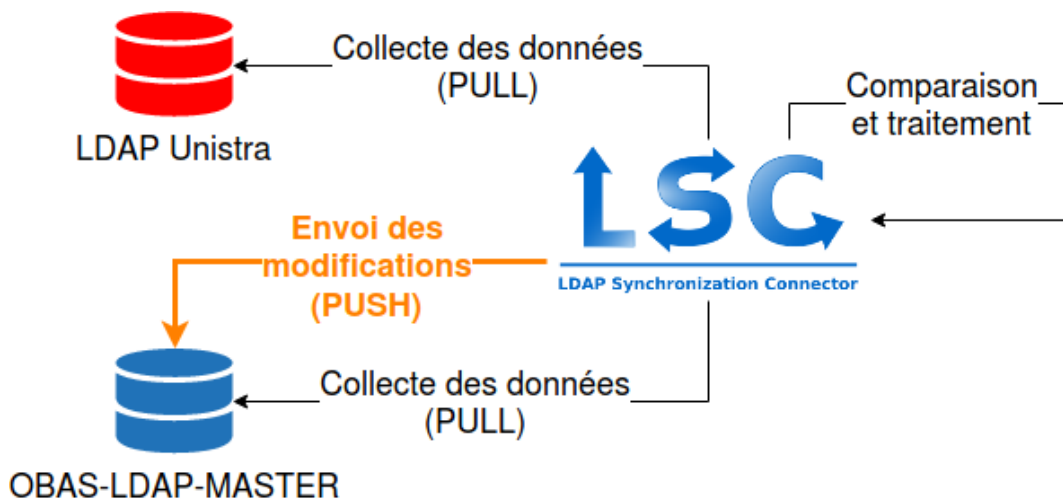


Illustration 7: Schéma de synchronisation LSC

Lors de la synchronisation des données LSC va d'abord collecter les attributs précisés dans son fichier de configuration sur toutes les entrées qu'il va trouver sur les deux annuaires. Ensuite, il va comparer ces données à partir d'un attribut qui sert de pivot entre les deux annuaires. Si des entrées ou des attributs n'existent que sur l'annuaire de l'Unistra, LSC va alors tenter de les rajouter à l'annuaire « obas-ldap-master ».

4.4 Performances et disponibilité du service d'annuaire

Dans un premier temps, pour assurer les performances et la disponibilité du service d'annuaire LDAP, j'ai mis en place des annuaires *consumers*¹ aux données identiques à l'annuaire principal. Ces annuaires *consumers* sont uniquement accessibles en lecture et les données automatiquement répliquées de l'annuaire principal. Pour réaliser cela, j'ai utilisé le moteur de réplication syncrepl inclut dans OpenLDAP.

Par la suite, j'ai mis en place un serveur HAProxy pour assurer l'équilibrage de charge entre les deux annuaires. Cependant après avoir remarqué un point de faiblesse dans l'infrastructure au niveau du serveur HAProxy, j'ai décidé de créer un cluster incluant un autre serveur HAProxy identique au premier afin de limiter les indisponibilités du service d'annuaire.

¹Terme pour désigner un serveur OpenLDAP esclave

Finalement, j'ai rajouté la supervision des serveurs et des services de l'infrastructure en utilisant Zabbix et des modules spécifiques aux services.

4.4.1 Le moteur de réplication syncrepl

Le moteur de réplication LDAP « Sync Replication engine » (syncrepl) est un moteur de réplication permettant aux serveurs OpenLDAP consumer de pouvoir maintenir une copie d'un fragment des données d'un serveur OpenLDAP provider². Il permet notamment de pouvoir programmer une mise à jour périodique des données ou une mise à jour dès le changement du contenu des données. Lors d'une mise à jour périodique, c'est le serveur consumer qui va aller chercher les données sur le serveur provider. Dans le cas d'une mise à jour dès la modification, le serveur consumer se mets à écouter le serveur provider pour une mise à jour éventuelle des données envoyé par ce dernier.

4.4.2 L'équilibreur de charge HAProxy

HAProxy est un logiciel libre et open source écrit en C. Son développement a commencé en 2000 par Willy Tarreau, l'une des dernières versions du logiciel date du 16 juin 2019. Il est capable de faire à la fois équilibreur de charge avec haute disponibilité et serveur proxy pour les applications communiquant en TCP ou HTTP. C'est une solution utilisée par un certain nombre de sites web important comme GitHub, Twitter, Stack Overflow, Reddit.

4.4.3 Corosync et Pacemaker

Corosync est un logiciel open source sous licence BSD dont le développement a débuté en 2008. Il permet d'établir un système de communication de groupe avec des fonctionnalités pour la mise en œuvre de la haute disponibilité des applications.

Pacemaker est un outil open source écrit en C supporté par les entreprises Red Hat, SUSE, et Linbit. Il permet de gérer une grappe de machines pour un service de haute disponibilité. Il est chargé de démarrer, arrêter et superviser les ressources du cluster.

Ces deux logiciels sont souvent utilisés ensemble pour fournir un service de haute disponibilité à des applications.

4.4.4 Zabbix

Zabbix est un outil libre de supervision créé par Alexei Vladishev en 1998. Il permet de surveiller l'état de divers services réseau, serveurs et autres matériels réseau via une interface web écrit en PHP. C'est l'outil de supervision utilisé par les administrateurs systèmes et réseaux pour superviser les différents services et serveurs de l'observatoire.

²Terme pour désigner un serveur OpenLDAP maître

4.4.5 Création d'annuaires consumer avec réplication

Pour la création des annuaires consumer, j'ai eu accès à deux autres machines virtuelles où j'ai installé un serveur OpenLDAP sur chacun. Ensuite il a fallu que je m'assure que ces annuaires partageaient les mêmes schémas que l'annuaire provider pour que toutes les données puissent être répliquées.

Une fois le rajout des schémas manquants, j'ai activé le module syncprov sur le provider pour que celui-ci puisse envoyer les données modifiées aux consumers.

```
1 dn: cn=module,cn=config
2 objectClass: olcModuleList
3 cn: module
4 olcModulePath: /usr/lib/ldap
5 olcModuleLoad: syncprov.la
6
7 dn: olcOverlay=syncprov,olcDatabase={1}hdb,cn=config
8 olcOverlay: syncprov
9 objectClass: olcOverlayConfig
10 objectClass: olcSyncProvConfig
```

Instructions LDIF pour activer le module syncprov

Après avoir activé le module syncprov, j'ai créé un identifiant « replicator » dans l'annuaire du provider pour permettre aux consumers de s'y connecter.

```
1 dn: cn=replicator,o=annuaire,dc=obas-ldap
2 cn: replicator
3 sn: Replicator Account
4 userPassword: {SSHA}uRgRtF4K3P4S5FPY65dTOm3SSD8WbSraG2KNLf
5 objectclass: top
6 objectclass: person
```

Instructions LDIF pour créer un utilisateur du nom de replicator

J'ai par la suite modifié l'ACL permettant aux utilisateurs connectés de voir le hash de leur mot de passe en permettant à l'utilisateur replicator de voir ce hash pour pouvoir le répliquer sur les consumers.

```
1 olcAccess: {2}to attrs=userPassword by self write by dn.exact="cn=replicator,o=annuaire,dc=obas-ldap"
2 read by anonymous auth by * none
```

ACL autorisant l'accès à l'attribut de mot de passe des utilisateurs

Finalement, j'ai activé la réplication persistante sur les *consumers* en indiquant les identifiants à utiliser (identifiants de l'utilisateur replicator), l'adresse du provider, le DN où commençait la recherche de données (base) ainsi que les attributs à répliquer.

```
1 dn: olcDatabase={1}hdb,cn=config
2 changetype: modify
```



```
3  replace: olcSyncRepl
4  olcSyncRepl: rid=000
5  provider=ldaps://obas-ldap-master.astro.unistra.fr
6  binddn="cn=replicator,o=annuaire,dc=obas-ldap"
7  bindmethod=simple
8  credentials=motDePasseUtilisateurReplicator
9  searchbase="o=annuaire,dc=obas-ldap"
10 attrs="*,+"
11 type=refreshAndPersist
12 retry="5 5 300 5"
13 timeout=1
```

Instructions LDIF pour activer la réplication sur un serveur consumer

Après la réplication des données réalisée, j'ai appliqué la même configuration de sécurisation que celle de l'annuaire principal aux serveurs *consumers*. Pour la sécurisation des communications, j'ai demandé un certificat pour chacun des serveurs.

4.4.6 Création d'un cluster HAProxy

Dans un premier temps, j'ai installé sur deux machines, un serveur HAProxy avec une configuration identique. J'ai vérifié le fonctionnement des configurations en faisant des requêtes LDAP vers chaque serveur en attendant une réponse.

```
1  frontend ldaps-front
2      mode      tcp
3      bind      0.0.0.0:636 ssl crt /etc/ssl/private/obas-ldap.pem ca-file /etc/ssl/certs/DigiCertCA.crt verify
4  optional
5      description  LDAP Service
6      option      tcplog
7      option      tcpka
8      timeout client 3600s
9      default_backend ldaps-back
10 backend ldaps-back
11     server slave1 obas-ldap-slave1.astro.unistra.fr:636 ssl verify none fall 3 rise 5 inter 5000
12     server slave2 obas-ldap-slave2.astro.unistra.fr:636 ssl verify none fall 3 rise 5 inter 5000
13     mode      tcp
14     balance leastconn
15     option tcpka
16     option ssl-hello-chk
17     timeout server 1800s
18     timeout connect 10s
```

Extrait de la configuration du frontend et backend HAProxy

La configuration HAProxy comprend un frontend écoutant pour toutes les adresses de la machine sur le port 636 en communication SSL. Il y a aussi un backend comprenant les deux serveurs LDAP consumers en mode d'équilibrage leastconn (selon le nombre de connexion actives) sur le port 636.

Une fois que les serveurs HAProxy fonctionnaient, j'ai installé Corosync et Pacemaker sur chaque serveur à partir des paquets Ubuntu. J'ai tout d'abord configuré Corosync pour créer la communication entre les deux nœuds et leur permettre de communiquer leur état. Ensuite j'ai utilisé l'outil pcs pour configurer le cluster pacemaker en rajoutant deux ressources : une ip virtuelle et un service HAProxy. Pour éviter que les ressources soient réparties sur les deux nœuds, j'ai forcé la colocation des deux ressources.

```
root@obas-hap1:~# pcs status
Cluster name: HAProxy
Stack: corosync
Current DC: obas-hap2 (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Tue Jul  2 17:19:25 2019
Last change: Tue Jul  2 17:19:15 2019 by root via cibadmin on obas-hap1

2 nodes configured
2 resources configured

Online: [ obas-hap1 obas-hap2 ]

Full list of resources:

VirtualIP      (ocf::heartbeat:IPaddr2):      Started obas-hap1
HAProxy        (lsb:haproxy):                 Started obas-hap1

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Illustration 8: Capture d'écran du statut du cluster HAProxy

4.4.7 Supervision avec Zabbix

Après la mise en place de tout les services de l'infrastructure d'authentification, j'ai mis en place la supervision des serveurs et des services s'y trouvant.

Dans un premier temps, j'ai mis en place la supervision des serveurs en y installant et configurant l'agent Zabbix. J'ai ensuite rajouté les serveurs à l'inventaire de Zabbix en utilisant l'interface web.

Par la suite, j'ai rajouté la supervision des services OpenLDAP, Haproxy et Pacemaker. Pour cela, j'ai dû installer sur le serveur Zabbix via l'interface web des templates créés par la communauté et spécifiques aux services. Avec ces templates, il y avait aussi des scripts à installer sur les serveurs où se trouvaient les services.

<input type="checkbox"/> Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface
<input type="checkbox"/> obas-hap1	Applications 14	Items 208	Triggers 19	Graphs 129	Discovery 5	Web	obas-hap1.astro.unistra.fr:10050
<input type="checkbox"/> obas-hap2	Applications 14	Items 208	Triggers 19	Graphs 129	Discovery 5	Web	obas-hap2.astro.unistra.fr:10050
<input type="checkbox"/> obas-ldap-master	Applications 11	Items 77	Triggers 19	Graphs 13	Discovery 2	Web	obas-ldap-master.astro.unistra.fr:10050
<input type="checkbox"/> obas-ldap-slave1	Applications 11	Items 77	Triggers 19	Graphs 13	Discovery 2	Web	obas-ldap-slave1.astro.unistra.fr:10050

Illustration 9: Capture d'écran de Zabbix montrant les machines supervisées

4.5 Authentification des services et des outils

Une fois le service d'annuaire en place, j'ai dû configurer les outils et services de l'observatoire pour que les utilisateurs puissent s'authentifier avec les identifiants de l'annuaire LDAP. La configuration de l'authentification entre les différents outils et services est très similaire mais comporte parfois quelques inconvénients. Cela est dû aux choix dans la manière d'implémenter l'authentification LDAP par les développeurs de l'outil.

4.5.1 Création d'un utilisateur pour l'authentification

Pour pouvoir connecter les différents services, j'ai d'abord dû créer un identifiant que les services peuvent utiliser pour récupérer la liste des utilisateurs et ainsi vérifier l'existence d'un utilisateur qui tente de s'authentifier ou récupérer des informations sur l'utilisateur pour compléter des informations du profil de l'utilisateur sur le service.

```

1 dn: cn=auth,o=annuaire,dc=obas-ldap
2 cn: auth
3 sn: Authentification Account for services and applications
4 userPassword: {SSHA}uRgRtNX6FPY65dTOM33dOWbSraG2KNLf
5 objectclass: top

```

6 objectclass: person

Enregistrement au format LDIF d'un utilisateur auth

Cet utilisateur n'a pas la possibilité de lire les mots de passes des autres utilisateurs comme replicator et il ne peut pas changer les données présentes sur l'annuaire (y compris celles le concernant). Il doit aussi être répliqué sur les autres annuaires.

```
1 olcAccess: {0}to dn.subtree="cn=auth,o=annuaire,dc=obas-ldap" by
2 dn.exact="cn=replicator,o=annuaire,dc=obas-ldap" read by dn.exact="cn=auth,o=annuaire,dc=obas-ldap"
3 read by anonymous auth by * none
```

Extrait d'ACLs au format LDIF

4.5.2 Configuration de l'authentification GLPI

La configuration de l'authentification LDAP sur l'outil GLPI s'est fait via l'interface web. Pour configurer l'authentification, il faut renseigner l'URI du serveur (dans notre cas, on utilise le domaine vers l'adresse virtuel du cluster HAProxy) ainsi que son port, un filtre de connexion pour filtrer les entrées à prendre en compte (optionnel), le base DN pour définir à partir de quelle branche doit-on chercher les utilisateurs, les identifiants pour s'authentifier et pouvoir lire les entrées de l'annuaire (DN du compte et mot de passe) et finalement le champ à considérer comme identifiant.

Configuration > Authentification > Annuaires LDAP + 🔍

obas-ldap 1/1

Annuaire LDAP

Nom	<input type="text" value="obas-ldap"/>	Dernière modification	2019-06-27 09:50
Serveur par défaut	<input type="text" value="Oui"/>	Actif	<input type="text" value="Oui"/>
Serveur	<input type="text" value="ldaps://obas-ldap.astro.unis"/>	Port (par défaut 389)	<input type="text" value="636"/>
Filtre de connexion	<input type="text" value="(objectclass=inetorgperson)"/>		
BaseDN	<input type="text" value="o=annuaire,dc=obas-ldap"/>		
DN du compte (pour les connexions non anonymes)	<input type="text" value="cn=auth,o=annuaire,dc=obas-ldap"/>		
Mot de passe du compte (pour les connexions non anonymes)	<input type="text"/>	<input type="checkbox"/> Effacer	
Champ de	<input type="text"/>	Commentaires	<input type="text"/>

Illustration 10: Écran de configuration de l'authentification LDAP de GLPI

Après la configuration de l'authentification, il est possible de tester la connexion via l'onglet « Tester » (voir Illustration 11).



Illustration 11: Capture d'écran de test de la configuration LDAP de GLPI

Une fois l'authentification configurée et la connexion réussie, GLPI ajoutera automatiquement les utilisateurs dans sa propre base de donnée dès leur première connexion en utilisant l'annuaire LDAP (choix à l'écran de connexion) sur l'outil pour pouvoir renseigner des informations supplémentaires sur l'utilisateur. Toutefois, il est possible d'importer par avance les utilisateurs pour leur attribuer des rôles spécifiques avant leur première connexion.

4.5.3 Configuration de l'authentification GRR

La configuration de l'authentification LDAP de GRR nécessitait l'installation du module php-ldap pour pouvoir commencer la configuration.

Configuration de l'authentification LDAP

Informations de connexion à l'annuaire LDAP.

Adresse de l'annuaire

Laissez «localhost» si l'annuaire est installé sur la même machine que GRR. Sinon, indiquez l'adresse du serveur.

Numéro de port de l'annuaire

Dans le doute, laissez la valeur par défaut : 389
(3268 pour serveur de catalogues global AD)

Type d'accès

Si le serveur LDAP n'accepte pas d'accès anonyme, veuillez préciser un identifiant (par exemple « cn=jean, o=lycée, c=fr »).
Dans le doute, laissez les champs suivants vides pour un accès anonyme.

Identifiant :

Mot de passe :

Remarque : des problèmes liés à un mot de passe contenant un ou plusieurs caractères accentués ont déjà été constatés.

Utiliser TLS :

Oui Non

Remarque : pour le moment, aucune modification n'a été apportée au fichier de configuration "config_ldap.inc.php".
Les informations ne seront enregistrées qu'à la fin de la procédure de configuration.

Illustration 12: Capture d'écran de configuration LDAP de GRR

4.5.4 Configuration de l'authentification Zabbix

La configuration de l'authentification LDAP pour Zabbix permet d'authentifier un utilisateur avec un identifiant existant à la fois dans la base de données locale de Zabbix et l'annuaire LDAP. Le mot de passe utilisé est le mot de passe de l'annuaire LDAP et non celui de la base de données locale de Zabbix.

Authentication HTTP settings **LDAP settings**

Enable LDAP authentication

* LDAP host

* Port

* Base DN

* Search attribute

Bind DN

Case sensitive login

Bind password

Test authentication [must be a valid LDAP user]

* Login

* User password

Illustration 13: Capture d'écran de configuration LDAP de Zabbix

Un inconvénient de l'authentification LDAP avec Zabbix est l'impossibilité pour un utilisateur de s'authentifier en utilisant à la fois la base de données locale et celle de l'annuaire LDAP. Il a donc fallu désigner un utilisateur pour éviter la perte d'accès à Zabbix en cas de dysfonctionnement de l'authentification LDAP.

4.6 Authentification des postes et serveurs

Pour l'authentification des postes et serveurs de l'Observatoire avec l'annuaire LDAP, j'ai dû choisir entre utiliser un module PAM LDAP ou une solution appelée SSSD. Après étude des deux procédés, j'ai choisi d'utiliser SSSD qui permettait beaucoup plus que le module PAM LDAP en termes de fonctionnalités et de performances. La fonctionnalité qui était principalement demandée était l'authentification hors-ligne si le serveur LDAP devient injoignable, SSSD gère cela via un système de cache.

4.6.1 SSSD

SSSD (System Security Services Daemon) est un set de daemons qui permet de gérer les mécanismes d'authentification sur une machine Linux en communiquant avec un ou plusieurs fournisseurs d'identités et d'authentications. Il vient à l'origine d'un projet open source nommé FreeIPA (Identity, Policy et Audit). L'objectif principal de ce projet est de fournir une solution de gestion d'identité, d'authentification et d'autorisation pour les ordinateurs Linux, UNIX, Windows et Apple. FreeIPA est en réalité une collection de projets réalisés dans l'objectif de fournir un framework incluant un client. L'un des principaux sponsors et contributeur au développement du projet FreeIPA est la compagnie RedHat.

4.6.2 Authentification sur les postes et serveurs avec SSSD

Pour l'authentification des postes avec SSSD, j'ai dû installer le paquet sssd en utilisant les commandes apt-get ou yum selon la machine. Une fois ce paquet installé, je devais modifier le fichier de configuration de SSSD (annexe page 48) pour que le service s'authentifie en utilisant le nom de domaine du cluster HAProxy et les identifiants créés précédemment pour l'authentification des outils et services.

```
1 ldap_schema = rfc2307bis
2 ldap_uri = ldaps://obas-ldap.astro.unistra.fr
3 ldap_search_base = o=annuaire,dc=obas-ldap
4 ldap_chpass_uri = ldaps://obas-ldap-master.astro.unistra.fr
5 # Authentification LDAP
6 ldap_default_bind_dn = cn=auth,o=annuaire,dc=obas-ldap
7 ldap_default_authtok =
8 AAAgAF4K3KNxR1tdDeQlwc2h981F3KEhPvWJRYi+cFi29uXs4lxiccIJoY0p6WI+olgpQjt8P4S5PCPA
9 AECAw==
10 ldap_default_authtok_type = obfuscated_password
```

Extrait de la configuration SSSD pour l'authentification au service d'annuaire

Une fois l'authentification configurée, j'ai rajouté pour certaines machines des droits d'accès pour limiter les utilisateurs pouvant se connecter à l'appareil.

```
1 access_provider = ldap
2 ldap_access_order = filter
3 ldap_access_filter = (memberof=cn=obas-admins,ou=groups,o=annuaire,dc=obas-ldap)
```

SSSD – Configuration des droits d'accès à la machine

Puis j'ai activé la mise en cache des hash des mots de passes des utilisateurs pour permettre l'authentification des utilisateurs même lors de l'indisponibilité du service d'annuaire.

```
1 # Allow offline logins by locally storing password hashes (default: false).
2 cache_credentials = True
```

SSSD – Mise en cache des identifiants pour authentification hors-ligne

Après la configuration de SSSD, il fallait que je modifie le fichier de configuration NSS pour utiliser SSSD pour l'authentification sur le poste. Pour finir, il suffisait de redémarrer le service SSSD avec la commande `systemctl` ou `service` pour pouvoir utiliser l'authentification SSSD sur le poste.

4.7 Déploiement de la configuration des machines

Une fois le procédé d'authentification LDAP choisi pour les machines, il a fallu que je recherche un moyen pour déployer rapidement cette configuration sur l'ensemble des serveurs et postes. Ce déploiement aurait été une tâche fastidieuse à la main.

4.7.1 Ansible

Ansible est un logiciel libre écrit principalement en Python par l'entreprise Ansible Inc. qui a été rachetée par RedHat en 2015. Ansible permet de configurer et de gérer de ordinateurs à distance en utilisant une connexion SSH. Il ne nécessite l'installation d'aucun logiciel supplémentaire (agent) sur les ordinateurs distants pour fonctionner. Ansible utilise des fichiers au format YAML appelés `playbook` qui servent à contenir le descriptif des instructions à réaliser sur les ordinateurs.

4.7.2 Création de `playbook` Ansible

Pour déployer la configuration de SSSD sur les machines distantes, j'ai d'abord créé un `playbook` contenant toutes les instructions à réaliser sur la machine distante (installation de paquets, modification de fichier de configuration...).

Par la suite, j'ai créé un `playbook` plus sophistiqué qui faisait appel à plusieurs fichiers différents selon le système détecté sur la machine distante (basé sur les distributions RedHat ou Debian présent sur les machines de l'observatoire) pour permettre d'adapter les instructions au système cible. Dans ce `playbook`, j'ai aussi fait l'usage du templating d'Ansible avec le format `jinja2` pour adapter le fichier de configuration de SSSD de façon plus précise. (annexe page 49)

```
1 - hosts: astromas
2   tasks:
3     - name: Install packages
4       apt:
5         name: "{{item}}"
6       with_items:
7         - sssd
```


5. Conclusion

Ce stage m'a permis de développer de nouvelles compétences dans le domaine de l'administration systèmes et de renforcer mon expérience professionnelle. Il m'a aussi permis d'affirmer mon projet professionnel qui est de travailler et acquérir davantage de connaissances dans l'administration systèmes et réseaux.

Premièrement, j'ai eu l'opportunité de découvrir de nouvelles méthodes de travail. J'ai appris à utiliser des technologies permettant d'automatiser le déploiement de configurations. Ces technologies ont énormément changé ma manière de travailler lorsque je suis amené à faire face à une infrastructure de taille trop importante pour y réaliser des actions manuelles. En mettant en place un annuaire LDAP, j'ai aussi appris à mieux planifier et organiser mon travail lors de la conception d'une base de données pour éviter de devoir recommencer à zéro.

Par ailleurs, ce stage m'a aussi permis de découvrir l'importance des solutions d'authentification centralisée que ça soit pour la facilité d'authentification des utilisateurs n'ayant pas à se souvenir de plusieurs identifiants différents ou la facilité de gestion de l'authentification pour l'administrateur de la solution.

Enfin à l'issue de la période de stage, mon tuteur m'a proposé de prolonger ma présence à l'observatoire par un CDD pendant le mois de juillet pour mettre en place une solution de déploiement automatique de systèmes sur les machines de l'observatoire. Je suis donc très content de pouvoir continuer à travailler à l'observatoire et d'être amené à me pencher sur une autre technologie d'automatisation.

6. Bibliographie

[observatoire astronomique, 2019] observatoire astronomique. Site officiel : <http://astro.unistra.fr>

[Wikipedia, 2019a] Wikipedia : Corosync Cluster Engine. https://fr.wikipedia.org/wiki/Corosync_Cluster_Engine

[Wikipedia, 2019b] Wikipedia : Pacemaker (logiciel). [https://fr.wikipedia.org/wiki/Pacemaker_\(logiciel\)](https://fr.wikipedia.org/wiki/Pacemaker_(logiciel))

[OpenLDAP, 2019] Documentation officielle OpenLDAP. <https://www.openldap.org/doc/admin24/>

[FusionDirectory, 2019a] Documentation officielle FusionDirectory 1.3. <https://fusiondirectory-user-manual.readthedocs.io/en/1.3/>

[FusionDirectory, 2019b] Gitlab officiel FusionDirectory. <https://gitlab.fusiondirectory.org>

[HAProxy, 2019] Documentations officielle HAProxy. <http://www.haproxy.org/#docs>

[Ansible, 2019] Documentation officielle Ansible. <https://docs.ansible.com/ansible/latest/index.html>

7. Annexes

7.1 Fichier de configuration LSC

Ce fichier de configuration est celui que j'ai réalisé lors du stage pour établir la synchronisation entre les deux annuaires LDAP. Il permet aussi à LSC de créer des groupes POSIX par utilisateur par rapport au numéro attribué de façon incrémentale lors de la synchronisation.

```
1 <?xml version="1.0" ?>
2 <lsc xmlns="http://lsc-project.org/XSD/lsc-core-2.1.xsd" revision="0">
3 <connections>
4 <ldapConnection>
5 <name>ldap-unistra-conn</name>
6 <!-- ./url mandatory, the JNDI URL -->
7 <url>ldaps://ldapr.unistra.fr:636/o=annuaire</url>
8 <!-- ./username mandatory, the DN to bind with -->
9 <username>cn=observatoire,ou=accounts,ou=operateurs,o=annuaire</username>
10 <!-- ./password mandatory, credentials to bind with -->
11 <password>secretPassword</password>
12 <!-- ./authentication mandatory, must contain either ANONYMOUS, SIMPLE, SASL, GSSAPI or
13 DIGEST_MD5 -->
14 <authentication>SIMPLE</authentication>
15 <!-- ./referral mandatory, must contain either IGNORE, THROUGH, THROW or FOLLOW -->
16 <referral>IGNORE</referral>
17 <!-- ./derefAliases mandatory, must contain either NEVER, SEARCH, FIND, ALWAYS -->
18 <derefAliases>NEVER</derefAliases>
19 <!-- ./version mandatory, must contain either VERSION_2, VERSION_3 -->
20 <version>VERSION_3</version>
21 <!-- ./pageSize optional, specify the paged size when searching -->
22 <pageSize>-1</pageSize>
23 <factory>com.sun.jndi.ldap.LdapCtxFactory</factory>
24 <tlsActivated>>false</tlsActivated>
25 </ldapConnection>
26 <ldapConnection>
27 <name>obas-ldap-master-conn</name>
28 <!-- ./url mandatory, the JNDI URL -->
29 <url>ldaps://obas-ldap-master.astro.unistra.fr:636/dc=obas-ldap</url>
30 <!-- ./username mandatory, the DN to bind with -->
31 <username>cn=admin,dc=obas-ldap</username>
32 <!-- ./password mandatory, credentials to bind with -->
33 <password>secretPassword</password>
34 <!-- ./authentication mandatory, must contain either ANONYMOUS, SIMPLE, SASL, GSSAPI or
35 DIGEST_MD5 -->
```

```

36     <authentication>SIMPLE</authentication>
37 <!-- ./referral mandatory, must contain either IGNORE, THROUGH, THROW or FOLLOW -->
38     <referral>IGNORE</referral>
39 <!-- ./derefAliases mandatory, must contain either NEVER, SEARCH, FIND, ALWAYS -->
40     <derefAliases>NEVER</derefAliases>
41 <!-- ./version mandatory, must contain either VERSION_2, VERSION_3 -->
42     <version>VERSION_3</version>
43 <!-- ./pageSize optional, specify the paged size when searching -->
44     <pageSize>-1</pageSize>
45 <!-- ./factory mandatory, points to LDAP Context Factory, com.sun.jndi.ldap.LdapCtxFactory for a SUN JDK -
46 ->
47     <factory>com.sun.jndi.ldap.LdapCtxFactory</factory>
48 <!-- ./tlsActivated optional, specify if SSL/TLS is activated to connect to the LDAP server -->
49     <tlsActivated>>false</tlsActivated>
50 </ldapConnection>
51 </connections>
52 <!-- ./tasks Task list node, must contain at least one task -->
53 <tasks>
54
55 <!-- ./task Task node, this is the main node, in which synchronization is defined -->
56 <task>
57 <!-- ./name mandatory task node this is the main node, in which synchronization is defined -->
58     <name>T10-SyncUsers</name>
59 <!-- ./bean optional bean node, default to org.lsc.beans.SimpleBean, define the pivot object used to store datasets
60 and values -->
61     <bean>org.lsc.beans.SimpleBean</bean>
62 <!-- ./sourceService mandatory node containing definition of the source service settings
63         possible builtin types are :
64             databaseSourceService, ldapSourceService
65             Plugins also provides: syncreplSourceService, nisSourceService
66         -->
67 <ldapSourceService>
68     <name>ldap-unistra-service</name>
69     <connection reference="ldap-unistra-conn" />
70     <baseDn>ou=uds,ou=people,o=annuaire</baseDn>
71     <pivotAttributes>
72         <string>udsDirectoryId</string>
73     </pivotAttributes>
74     <fetchAttributes>
75         <string>uid</string>
76         <string>cn</string>

```



```

77     <string>sn</string>
78     <string>udsDirectoryId</string>
79     <string>userPassword</string>
80     <string>givenName</string>
81     <string>displayName</string>
82     <string>mail</string>
83     <string>displayName</string>
84     <string>telephoneNumber</string>
85     </fetchAllAttributes>
86     <getAllFilter>(udsNetManagerGroup=netmagis-obs)</getAllFilter>
87     <getOneFilter>(&(udsNetManagerGroup=netmagis-
88 obs)(udsDirectoryId={udsDirectoryId}))</getOneFilter>
89     <cleanFilter>(&(udsNetManagerGroup=netmagis-
90 obs)(udsDirectoryId={udsDirectoryId}))</cleanFilter>
91     </ldapSourceService>
92     <ldapDestinationService>
93     <name>obas-ldap-master-service</name>
94     <connection reference="obas-ldap-master-conn" />
95     <baseDn>ou=people,o=annuaire,dc=obas-ldap</baseDn>
96     <pivotAttributes>
97     <string>udsDirectoryId</string>
98     </pivotAttributes>
99     <fetchAllAttributes>
100    <string>udsDirectoryId</string>
101    <string>uid</string>
102    <string>cn</string>
103    <string>sn</string>
104    <string>userPassword</string>
105    <string>homeDirectory</string>
106    <string>uidNumber</string>
107    <string>gidNumber</string>
108    <string>loginShell</string>
109    <string>objectClass</string>
110    <string>givenName</string>
111    <string>displayName</string>
112    <string>mail</string>
113    <string>telephoneNumber</string>
114    <string>obs-datasource</string>
115    </fetchAllAttributes>
116    <getAllFilter>(&(objectClass=inetorgperson)(obs-datasource=Unistra))</getAllFilter>
117    <getOneFilter>(&(objectClass=inetorgperson)(udsDirectoryId={udsDirectoryId}))</getOneFilter>

```

```

118     </ldapDestinationService>
119     <propertiesBasedSyncOptions>
120 <!-- ./mainIdentifier This mandatory node must contain a string Javascript expression that will enforce the object
121 main identifier.-->
122     <mainIdentifier>"uid=" + srcBean.getDatasetFirstValueById("uid") + ",ou=people,o=annuaire,dc=obas-
123 ldap"</mainIdentifier>
124 <!-- ./defaultDelimiter This mandatory node must contain a string Javascript expression that will enforce the
125 object main identifier.-->
126     <defaultDelimiter>;</defaultDelimiter>
127 <!-- ./defaultPolicy This mandatory node must contain a string Javascript expression that will enforce the object
128 main identifier.-->
129     <defaultPolicy>FORCE</defaultPolicy>
130 <!-- ./conditions This optional node may contain one or more of the four node : create, update, delete and
131 changeId -->
132     <conditions>
133 <!-- ./create This optional node may contain a boolean Javascript expression that will indicate whenever a new
134 entry must be created or not -->
135         <create>>true</create>
136 <!-- ./update This optional node may contain a boolean Javascript expression that will indicate whenever a
137 existing entry must be updated or not -->
138         <update>>true</update>
139 <!-- ./delete This optional node may contain a boolean Javascript expression that will indicate whenever a
140 existing entry must be deleted or not -->
141         <delete>dstBean.getDatasetFirstValueById("obs-datasource") == "Unistra"</delete>
142 <!-- ./changeId This optional node may contain a boolean Javascript expression that will indicate whenever
143 an existing object main identifier must be changed or not -->
144         <changeId>>false</changeId>
145     </conditions>
146 <!-- ./dataset This multi-valued node may contain a structure that will describe how to synchronize the
147 corresponding dataset -->
148     <dataset>
149 <!-- ./name Mandatory node containing the dataset name -->
150         <name>objectClass</name>
151 <!-- ./policy Mandatory node containing the policy to apply to this dataset. Contains KEEP, FORCE or MERGE
152 value -->
153         <policy>KEEP</policy>
154 <!-- ./defaultValues Optional node containing a list of string values that will be used if noone is provided by
155 datasource -->
156         <defaultValues></defaultValues>
157 <!-- ./forceValues Optional node containing a list of string values that will be used to force destination service
158 dataset values -->

```

```

159     <forceValues></forceValues>
160 <!--   ./createValues Optional node containing a list of string values that will be used to force destination service
161 dataset values when creating object -->
162     <createValues>
163         <string>"inetOrgPerson"</string>
164         <string>"posixAccount"</string>
165         <string>"obs-custom"</string>
166     </createValues>
167     <delimiter>,</delimiter>
168 </dataset>
169 <dataset>
170     <name>uidNumber</name>
171     <policy>KEEP</policy>
172     <defaultValues></defaultValues>
173     <forceValues></forceValues>
174     <createValues>
175
176 <string>SequencesFactory.getInstance(Ildap.getJndiServices()).getNextValue("cn=uidnumberseq,ou=sequences,
177 o=annuaire,dc=obas-ldap","serialnumber")</string>
178     </createValues>
179 </dataset>
180 <dataset>
181     <name>gidNumber</name>
182     <policy>KEEP</policy>
183     <defaultValues></defaultValues>
184     <forceValues></forceValues>
185     <createValues>
186
187 <string>SequencesFactory.getInstance(Ildap.getJndiServices()).getNextValue("cn=gidnumberseq,ou=sequences,
188 o=annuaire,dc=obas-ldap","serialnumber")</string>
189     </createValues>
190 </dataset>
191 <dataset>
192     <name>homeDirectory</name>
193     <policy>KEEP</policy>
194     <defaultValues></defaultValues>
195     <forceValues></forceValues>
196     <createValues>
197         <string>"/home/" + srcBean.getDatasetFirstValueById("uid")</string>
198     </createValues>
199 </dataset>

```

```

200     <dataset>
201       <name>loginShell</name>
202       <policy>KEEP</policy>
203       <defaultValues></defaultValues>
204       <forceValues></forceValues>
205       <createValues>
206         <string>"/bin/bash"</string>
207       </createValues>
208     </dataset>
209     <dataset>
210       <name>obs-datasource</name>
211       <policy>KEEP</policy>
212       <defaultValues></defaultValues>
213       <forceValues></forceValues>
214       <createValues>
215         <string>"Unistra"</string>
216       </createValues>
217     </dataset>
218   </propertiesBasedSyncOptions>
219 </task>
220 <task>
221   <name>T20-CreateGroups</name>
222   <bean>org.lsc.beans.SimpleBean</bean>
223   <ldapSourceService>
224     <name>obas-ldap-master-users</name>
225     <connection reference="obas-ldap-master-conn" />
226     <baseDn>ou=people,o=annuaire,dc=obas-ldap</baseDn>
227     <pivotAttributes>
228       <string>gidNumber</string>
229     </pivotAttributes>
230     <fetchAttributes>
231       <string>uid</string>
232       <string>gidNumber</string>
233     </fetchAttributes>
234     <getAllFilter>(&(objectClass=posixAccount)(obs-datasource=Unistra))</getAllFilter>
235     <getOneFilter>(&(objectClass=posixAccount)(gidNumber={gidNumber}))</getOneFilter>
236     <cleanFilter>(&(objectClass=posixAccount)(gidNumber={gidNumber}))</cleanFilter>
237   </ldapSourceService>
238   <ldapDestinationService>
239     <name>obas-ldap-master-groups</name>
240     <connection reference="obas-ldap-master-conn" />

```

```

241     <baseDn>ou=groups,o=annuaire,dc=obas-ldap</baseDn>
242     <pivotAttributes>
243         <string>gidNumber</string>
244     </pivotAttributes>
245     <fetchedExceptions>
246         <string>cn</string>
247         <string>member</string>
248         <string>gidNumber</string>
249         <string>objectClass</string>
250         <string>obs-datasource</string>
251     </fetchedExceptions>
252     <getAllFilter>(&!(objectClass=posixGroup)(obs-datasource=Unistra))</getAllFilter>
253     <getOneFilter>(&!(objectClass=posixGroup)(gidNumber={gidNumber}))</getOneFilter>
254 </ldapDestinationService>
255 <propertiesBasedSyncOptions>
256 <!-- ./mainIdentifier This mandatory node must contain a string Javascript expression that will enforce the object
257 main identifier.-->
258     <mainIdentifier>"cn=" + srcBean.getDatasetFirstValueById("uid") + ",ou=groups,o=annuaire,dc=obas-
259 ldap"</mainIdentifier>
260 <!-- ./defaultDelimiter This mandatory node must contain a string Javascript expression that will enforce the
261 object main identifier.-->
262     <defaultDelimiter>;</defaultDelimiter>
263 <!-- ./defaultPolicy This mandatory node must contain a string Javascript expression that will enforce the object
264 main identifier.-->
265     <defaultPolicy>FORCE</defaultPolicy>
266 <!-- ./conditions This optional node may contain one or more of the four node : create, update, delete and
267 changeId →
268     <conditions>
269 <!-- ./create This optional node may contain a boolean Javascript expression that will indicate whenever a new
270 entry must be created or not -->
271         <create>>true</create>
272 <!-- ./update This optional node may contain a boolean Javascript expression that will indicate whenever a
273 existing entry must be updated or not -->
274         <update>>true</update>
275 <!-- ./delete This optional node may contain a boolean Javascript expression that will indicate whenever a
276 existing entry must be deleted or not -->
277         <delete>dstBean.getDatasetFirstValueById("obs-datasource") == "Unistra"</delete>
278 <!-- ./changeId This optional node may contain a boolean Javascript expression that will indicate whenever
279 an existing object main identifier must be changed or not -->
280         <changeId>>false</changeId>
281     </conditions>

```

```

282 <!-- ./dataset This multi-valued node may contain a structure that will describe how to synchronize the
283 corresponding dataset -->
284     <dataset>
285 <!-- ./name Mandatory node containing the dataset name -->
286     <name>objectClass</name>
287 <!-- ./policy Mandatory node containing the policy to apply to this dataset. Contains KEEP, FORCE or MERGE
288 value -->
289     <policy>KEEP</policy>
290 <!-- ./defaultValues Optional node containing a list of string values that will be used if noone is provided by
291 datasource -->
292     <defaultValues></defaultValues>
293 <!-- ./forceValues Optional node containing a list of string values that will be used to force destination service
294 dataset values -->
295     <forceValues></forceValues>
296 <!-- ./createValues Optional node containing a list of string values that will be used to force destination service
297 dataset values when creating object -->
298     <createValues>
299         <string>"top"</string>
300         <string>"posixGroup"</string>
301         <string>"groupOfNames"</string>
302         <string>"obs-custom"</string>
303     </createValues>
304 <!-- ./delimiter Used when multiples values are provided in a single joined value -->
305     <delimiter>,</delimiter>
306 </dataset>
307 <dataset>
308     <name>cn</name>
309     <policy>KEEP</policy>
310     <defaultValues></defaultValues>
311     <forceValues></forceValues>
312     <createValues>
313         <string>srcBean.getDatasetFirstValueById("uid")</string>
314     </createValues>
315 </dataset>
316 <dataset>
317     <name>obs-datasource</name>
318     <policy>KEEP</policy>
319     <defaultValues></defaultValues>
320     <forceValues></forceValues>
321     <createValues>
322         <string>"Unistra"</string>

```

```

323     </createValues>
324 </dataset>
325 <dataset>
326     <name>member</name>
327     <policy>KEEP</policy>
328     <defaultValues></defaultValues>
329     <forceValues></forceValues>
330     <createValues>
331         <string>"uid="+srcBean.getDatasetFirstValueById("uid")+","ou=people,o=annuaire,dc=obas-
332 ldap"</string>
333     </createValues>
334 </dataset>
335 </propertiesBasedSyncOptions>
336 </task>
337 </tasks>
338 <!-- ./security This mandatory node contains the security settings used by LSC -->
339 <security>
340 <!-- ./encryption This optional node contains the encryption settings -->
341 <encryption>
342 <!-- ./keyfile This optional node contains the keyfile location -->
343 <keyfile>etc/lsc.key</keyfile>
344 <!-- ./algorithm This optional node contains the encryption algorithm -->
345 <algorithm>AES</algorithm>
346 <!-- ./strength This optional node contains the algorithm key length -->
347 <strength>128</strength>
348 </encryption>
    </security>
</lsc>

```

7.2 Fichier de configuration SSSD d'un serveur

Ce fichier de configuration est un fichier que j'ai réalisé en utilisant le templating jinja2 d'ansible. Il permet de configurer l'authentification SSSD sur un serveur en n'autorisant que les administrateurs à s'authentifier.

```
1  [sssd]
2  config_file_version = 2
3  services = nss, pam, sudo
4  domains = LDAP
5
6  [nss]
7  filter_users = root,obs
8  filter_groups = root,obs
9
10 [pam]
11
12 [sudo]
13
14 [domain/LDAP]
15 id_provider = ldap
16 auth_provider = ldap
17 chpass_provider = ldap
18
19 access_provider = ldap
20 ldap_access_order = filter
21 ldap_access_filter = (memberof=cn=obas-admins,ou=groups,o=annuaire,dc=obas-ldap)
22
23 # ldap_schema can be set to "rfc2307", which stores group member names in the
24 # "memberuid" attribute, or to "rfc2307bis", which stores group member DNs in
25 # the "member" attribute. If you do not know this value, ask your LDAP
26 # administrator.
27 ldap_schema = rfc2307bis
28 ldap_uri = ldaps://obas-ldap.astro.unistra.fr
29 ldap_search_base = o=annuaire,dc=obas-ldap
30 ldap_chpass_uri = ldaps://obas-ldap-master.astro.unistra.fr
31
32 # Authentication LDAP
33 ldap_default_bind_dn = cn=auth,o=annuaire,dc=obas-ldap
34 ldap_default_authtok =
35 AAF4K3AgACKNxR1tdPvWJRYi+cFi29uXs4lxiGoX3yZZ/ccIJoY0p6WI+olqgS9fRt6PCPAAECAw==
36 ldap_default_authtok_type = obfuscated_password
37
38 # Configuration sudo
```



```

39 sudo_provider = ldap
40 ldap_sudo_search_base = ou=sudoers,o=annuaire,dc=obas-ldap
41
42 # Note that enabling enumeration will have a moderate performance impact.
43 # Consequently, the default value for enumeration is FALSE.
44 # Refer to the sssd.conf man page for full details.
45 # enumerate = false
46 # Allow offline logins by locally storing password hashes (default: false).
47 cache_credentials = True
48

```

7.3 Extrait du template Ansible pour la configuration SSSD

Ce fichier est un template au format jinja2 que j'ai réalisé lors du stage pour créer le fichier de configuration SSSD sur les machines de l'observatoire en fonction de plusieurs variables définies lors du lancement du playbook.

```

1 #jinja2: lstrip_blocks: "True"
2 [sssd]
3 config_file_version = 2
4 {% if autofs is defined and autofs == true %}
5 services = nss, pam, sudo, autofs
6 {% else %}
7 services = nss, pam, sudo
8 {% endif %}
9 domains = LDAP
10
11 [nss]
12 filter_users = {{filter_users}}
13 filter_groups = {{filter_groups}}
14
15 [pam]
16
17 [sudo]
18
19 {% if autofs is defined and autofs == true %}
20 [autofs]
21 {% endif %}
22
23 [domain/LDAP]
24 id_provider = ldap
25 auth_provider = ldap
26 chpass_provider = ldap

```

```
27
28  {# Authentication access rules #}
29  {% if sssd_access_order is defined and sssd_access_order is not none %}
30  access_provider = ldap
31  ldap_access_order = {{ sssd_access_order }}
32      {% if sssd_groups_filter is defined and sssd_groups_filter is not none %}
33          {% if sssd_groups_filter | length > 1 %}
34              {% set ldap_access_filter %}
35  (|{% for group in sssd_groups_filter %}(memberof=cn={{ group }},{{ ldap_group_search_base }}){% endfor %})
36          {% endset %}
37          {% else %}
38              {% set ldap_access_filter %}
39  (memberof=cn={{ sssd_groups_filter[0] }},{{ ldap_group_search_base }})
40          {% endset %}
41          {% endif %}
42  ldap_access_filter = {{ ldap_access_filter }}
43      {% endif %}
44  {% endif %}
45
```