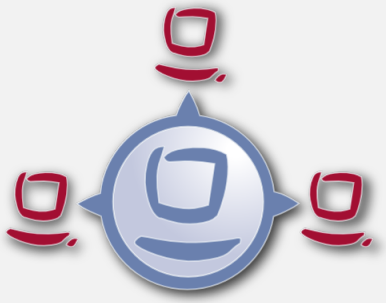

opsi Directory Connector

uib gmbh
Bonifaziusplatz 1b
55118 Mainz
Tel.:+49 6131 275610
www.uib.de
info@uib.de

Stand: 3. Januar 2018

Inhaltsverzeichnis

Kapitel 1

Copyright

Kapitel 2

opsi directory connector

Einführung

Der opsi Directory Connector ist ein Werkzeug um Daten aus einem Verzeichnisdienst in eine opsi-Installation zu überführen. Dadurch wird mehrfacher Pflegeaufwand in unterschiedlichen Systemen vermieden.

Vorbedingungen für die opsi Erweiterung *opsi directory connector*

Dieses Modul ist momentan eine kofinanzierte opsi Erweiterung.
Es sind eine Reihe von Vorbedingungen nötig, um dieses Modul einsetzen zu können. Das bedeutet, dass Sie zum Einsatz eine Freischaltdatei benötigen. Diese Freischaltung erhalten Sie, wenn Sie die Erweiterung kaufen. Zu Evaluierungszwecken stellen wir Ihnen auch eine zeitlich befristete Freischaltung kostenlos zur Verfügung (→ mail an info@uib.de).

Allgemeine Anforderungen

Der Quell-Verzeichnisdienst muss das LDAP-Protokoll implementieren.

Das Ziel-Opsi-System sollte mindestens opsi 4.0.7 verwenden. Ältere Versionen können funktionieren, wurden aber nicht getestet.

Die Maschine, auf welcher der Connector laufen soll, muss über das Netzwerk Zugriff auf den Directory- und opsi-Server haben. Es ist möglich alle Komponenten auf der gleichen Maschine zu betreiben, aber es wird davon ausgegangen, dass jeweils getrennte Maschinen verwendet werden.

Hardware-Anforderungen

Diese Anforderungen richten sich an eine einfache Verwendung in einer kleinen Umgebung mit bis zu 500 Clients. Diese Anforderungen fallen in großen Umgebungen gegebenenfalls größer aus, weshalb Anpassungen notwendig sein können.

- 256 MB freier Arbeitsspeicher
- Netzwerkverbindungen

Software-Anforderungen

Es wird nur die Installation und der Betrieb des Connectors unter Linux unterstützt. Eine Unterstützung für Windows ist nicht geplant.

Auf der Maschine muss folgendes installiert sein.

- Python 3.4 oder neuer

Die folgenden Module für Python 3:

- ldap3
- requests

Durch die Verwendung standardisierter Protokolle zur Kommunikation werden keine opsi- oder Verzeichnisdienst-spezifischen Komponenten benötigt.

Bei der Installation des Connectors über den Paket-Manager des Betriebssystems werden diese Abhängigkeiten automatisch installiert.

Installation

Bitte verwenden Sie den Paket-Manager des Betriebssystems um das Paket `opsi-directory-connector` zu installieren.

Auf einer Debian-basierten Maschine kann die Installation wie folgt durchgeführt werden:

```
apt-get install opsi-directory-connector
```

Konfiguration

Der Connector kann über eine Vielzahl an Einstellungsmöglichkeiten an verschiedenste Umgebungen angepasst werden.

Die Konfiguration geschieht über eine Konfigurationsdatei im JSON-Format, welche gültiges JSON enthalten muss. Zur Angabe von boolschen Werten verwenden Sie bitte `true` oder `false`. Text muss mit doppelten Anführungszeichen eingegeben werden, beispielsweise `"das ist Text"`.

Eine Beispiel-Konfiguration wird unter `/etc/opsi/opsidirectoryconnector.example.conf` bereitgestellt. Diese Datei kann als eine Vorlage für eigene Konfigurationen verwendet werden.

```
cp /etc/opsi/opsidirectoryconnector.example.conf /etc/opsi/opsidirectoryconnector-custom.conf
```

Directory-Einstellungen

Diese Einstellungen werden benötigt, um eine Verbindung zum Verzeichnisdienst herzustellen und den Suchbereich auf bestimmte Bereiche und Objekte einzugrenzen.

```
{
    "directory": {
        "address": "ldap://192.168.12.34",
        "user": "DOMAIN\\opsiconnector",
        "password": "insertpasswordhere",
        "search_base": "dc=testcompy,dc=local",
        "search_query_computers": "(objectClass=computer) ",
        "search_query_groups": "(objectClass=organizationalUnit)",
```

```
        "connection_options": {
            "start_tls": true
        }
    },
    ...
}
```

Unter `address` muss angegeben werden unter welcher Adresse der Server angesprochen wird. `user` und `password` werden für die Authentifikation an Selbigem verwendet.

Anmerkung
Wir empfehlen die Verwendung eines gesonderten Benutzerkontos.

Anmerkung
In vielen Systemen wird als User der komplette DN erwartet. Bspw. `uid=opsiconnect,cn=users,dc=test,dc=intranet`.

Über `search_base` wird angegeben ab welchem Punkt nach passenden Element gesucht wird. Über `search_query_computers` und `search_query_groups` können Bedingungen für die Suche nach Eintragen konfiguriert werden.

Entweder `search_query_computers` oder `search_query_groups` oder beides muss konfiguriert sein. Um eine Bedingung zu deaktivieren, kann der Wert auf `""` gesetzt werden. Die Suche nach Gruppen wird in einer zukünftigen Version implementiert werden. Bis dahin hat diese Einstellung keine Auswirkungen.

Der Parameter `connection_options` beinhaltet zusätzliche Optionen zur Konfiguration der Verbindung. Mit `start_tls` kann gesteuert werden, ob eine gesicherte Verbindung verwendet werden soll.

Anmerkung
Weitere Verbindungs-Optionen werden auf Nachfrage implementiert.

Verbindung zu UCS konfigurieren

Für eine Verbindung zu Univention Corporate Server muss für die Verbindung als Benutzername der komplette DN verwendet werden.

Soll das Verzeichnis komplett durchsucht werden, kann für `search_base` die Ausgabe des Befehls `ucr get ldap/base` verwendet werden.

Für die Suche nach Windows-Clients kann `(objectClass=univentionWindows)` als Wert für `search_query_computers` angegeben werden.

Seit Version 14 ist es möglich über den Aufrufparameter `--check-directory` die Verbindungseinstellungen zum Verzeichnis zu prüfen, ohne dass eine Verbindung zum opsi-Server hergestellt wird.

Verhaltens-Einstelllungen

Die Einstellungen steuern das Verhalten des Connectors.

```
{
    ...
    "behaviour": {
        "write_changes_to_opsi": true,
        "root_dir_in_opsi": "Directory",
        "update_existing_clients": true,
        "prefer_location_from_directory": true
```

```
    },
    ...
}
```

Wird `write_changes_to_opsi` auf `false` gesetzt werden keine Daten nach opsi geschrieben. Mit dieser Einstellung ist es möglich die Verbindungseinstellungen zu überprüfen, bevor sie angewendet werden.

Per `root_dir_in_opsi` wird angegeben welche Gruppe in opsi als Wurzelgruppe verwerndet werden soll. Es muss von Ihnen sichergestellt werden, dass diese Gruppe existiert.

Wird `update_existing_clients` auf `false` gesetzt, so werden bereits in opsi existierende Clients nicht verändert. Wird dieser Wert auf `true` gesetzt, so werden möglicherweise manuell gesetzte Daten mit den Werten aus dem Directory überschrieben.

Falls `prefer_location_from_directory` auf `true` gesetzt, werden Clients in opsi an die Position verschoben, welche sie im Directory haben. Für das Deaktivieren dieses Verhalten, muss dieser Wert auf `false` gesetzt werden.

Mappings

Mit einem derart flexiblen System wie ein Verzeichnisdienst benötigt der Connector Informationen darüber welche Attribute im Directory auf welche Attribute in opsi angewendet werden sollen.

```
{
    ...
    "mapping": {
        "client": {
            "id": "name",
            "description": "description",
            "notes": "",
            "hardwareAddress": "",
            "ipAddress": "",
            "inventoryNumber": "",
            "oneTimePassword": ""
        },
        "group": {
            "id": "name",
            "description": "description",
            "notes": ""
        }
    },
    ...
}
```

Es gibt jeweils ein Mapping für Clients und eines für Gruppen.

Der Schlüssel jedes Mappings ist das Attribut in opsi und der Wert ist das Attribut aus dem Verzeichnisdienst. Ist der Wert (in der Zuordnung) leer, so wird keine Zuordnung vorgenommen.

Anmerkung
Sollte der aus dem Verzeichnis ausgelesene Wert für die ID des Clients nicht als FQDN erkennbar sein, so wird ein enstprechender FQDN erstellt. Der Domain-Teil hierfür wird aus den DC-Werten des Elements gebildet.

opsi-Verbindungs-Einstellungen

Hierüber wird gesteuert wie der Connector sich zu opsi verbindet.

```
{
    ...
    "opsi": {
        "address": "https://localhost:4447",
        "username": "syncuser",
        "password": "secret",
        "connection_options": {
            "verify_certificate": true
        }
    }
}
```

Unter `address` ist die Adresse des opsi-Servers einzutragen. Vergessen Sie nicht die Angabe des Ports!

Anmerkung
Ein Proxy für die Verbindung kann über die Umgebungsvariable *HTTPS_PROXY* gesetzt werden.

Mittels `username` und `password` wird geregelt welche Zugangsdaten zur Authentifizierung am opsi-Server verwendet werden.

Anmerkung
Wir empfehlen die Verwendung eines gesonderten Benutzers. Die Anlage zusätzlicher Benutzer ist im Dokument *Getting Started* beschrieben.

Unter `connection_options` werden Optionen für die Verbindung zum opsi-Server festgelegt. Mittels `verify_certificate` wird die Überprüfung des Server-Zertifikats gesteuert. Für selbstsignierte Zertifikate kann dieser Wert auf `false` gesetzt werden.

Seit Version 14 ist es möglich über den Aufrufparameter `--check-opsi` die Verbindung zum opsi-Server zu testen, ohne dass eine Verbindung zum Verzeichnisdienst hergestellt wird.

Den Connector ausführen

Nach der Installation existiert ein Binary `opsidirectoryconnector` auf dem System.

Dieses muss einen Parameter `--config` zusammen mit dem Pfad zur Konfigurationsdatei übergeben bekommen.

```
opsidirectoryconnector --config /etc/opsi/opsidirectoryconnector-custom.conf
```

Anmerkung
Der ausführende Benutzer benötigt keinen Zugriff auf das opsi-System, da der zugreifende Benutzer in der Konfigurationsdatei hinterlegt ist.

Beispiel: wiederkehrende Verarbeitung mit systemd

Der Connector macht aktuell bei der Ausführung eine Synchronisationslauf, aber die Chancen stehen gut, dass eine ständige Synchronisation erfolgt.

Es ist einfach, die Ausführung wiederkehrender Läufe zu automatisieren.

Wir werden hierbei systemd verwenden. Im Gegensatz zu cronjobs wird systemd verhindern, dass überlappende Läufe stattfinden, weshalb systemd eine gute Wahl ist.

Das folgende Beispiel wird den Connector so einrichten, dass er fünf Minuten nach dem Start der Maschine ausgeführt wird und danach jede Stunde.

Wir benötigen zwei Dateien, welche in dem entsprechenden Verzeichnis für benutzerdefinierte Units abgelegt werden müssen. Der Pfad kann je nach verwendetem Betriebssystem unterschiedlich ausfallen. Bitte verwenden Sie nachfolgend `pkg-config` um an den entsprechenden Pfad zu kommen:

```
pkg-config systemd --variable=systemduserunitdir
```

In diesem Verzeichnis müssen die zwei folgenden Dateien abgelegt werden. Eine Datei ist der Timer, welche unseren Job wiederkehrend aufruft und die Andere ist für den Job selbst.

Bitte füllen Sie die Datei `opsi-directory-connector.timer` mit dem folgenden Inhalt:

```
[Unit]
Description=Start the opsi-directory-connector in regular intervals

[Timer]
OnBootSec=5min
OnUnitActiveSec=1hour

[Install]
WantedBy=timers.target
```

Und dies muss nach `opsi-directory-connector.service`:

```
[Unit]
Description=Sync clients from AD to opsi.
Wants=network.target

[Service]
Type=oneshot
ExecStart=/usr/bin/opsidirectoryconnector --config /etc/opsi/opsidirectoryconnector-custom.conf
```

Um den Timer zu aktivieren und ihn sofort zu starten, können die folgenden Befehle verwendet werden:

```
systemctl enable opsi-directory-connector.timer
systemctl start opsi-directory-connector.timer
```

Falls der Timer nicht gestartet wird, wird er erst nach dem nächsten Neustart der Maschine ausgeführt werden.

Beispiel: wiederkehrende Verarbeitung als Cronjob

Es ist einfach, die Ausführung wiederkehrender Läufe über einen Crobjob zu automatisieren.

Bitte beachten Sie, dass überlappende Läufe stattfinden können, weshalb der Synchronisationsintervall am besten größer gewählt werden sollte. Zur Vermeidung dieses Problems wird die Verwendung von **systemd** anstatt **cron** empfohlen!

Zur Bearbeitung der Cronjob-Datei wird in der Regel `crontab -e` aufgerufen. Für eine zu jeder Stunde stattfindenden Synchronisation kann dort folgendes als Cronjob hinterlegt werden.

```
0 * * * * /usr/bin/opsidirectoryconnector --config /etc/opsi/opsidirectoryconnector-custom.conf
```