Forcepoint NGFW and Azure Active Directory secure hybrid access

Integration Guide

Forcepoint

Integration Guide

Dlo Bagari

19 June 2020

Public

Table of Contents

Version	Date	Author	Notes
0.1	26 March 2020	Dlo Bagari	First draft
0.2	08 April 2020	Mattia Maggioli	Review
0.3	15 April 2020	Dlo Bagari	Resolved Comments
0.4	24 April 2020	Neelima Rai	Added troubleshooting chapter
0.5	19 June 2020	Jonathan Knepher	Review

Summary

This guide provides step by step instructions to set up an integration between **Azure AD Secure Hybrid** and **Forcepoint Next Generation Firewall (NGFW)**

The automated integration enables Forcepoint NGFW Security Management Center (SMC) access and authentication through Azure AD users/policies, and exposes SMC as an Azure app for remote management: selected Azure AD users can be assigned with different level of access into the SMC, enabling remote management of the entire fleet of NGFW engines controlled by SMC with the extra security layer provided by Azure AD authentication policies.

The code and instructions provided enable system administrators to **automatically**

→ Create Azure AD Domain Services with external LDAPs enabled

→ Configure an Azure application for automatic provisioning

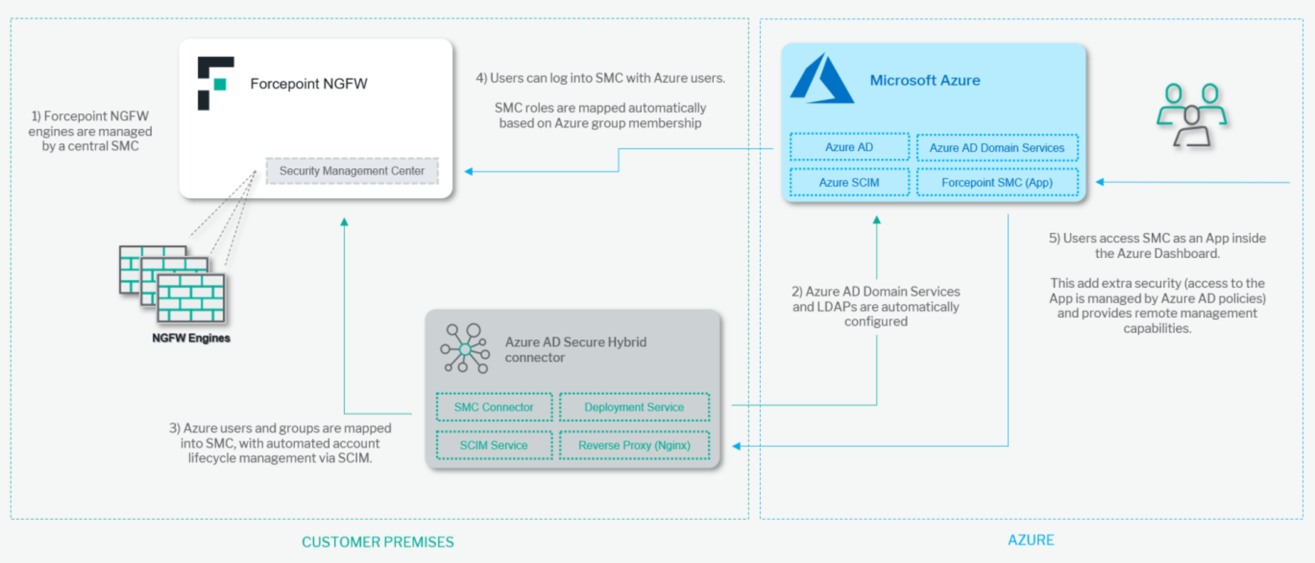
→ Create Azure Active Directory groups for Forcepoint SMC roles management

→ Create an external user's Active Directory and user authentication server in Forcepoint SMC

→ Expose Forcepoint SMC as an Azure app for remote management

→ Use the System for Cross-Domain Identity Management (SCIM) user management API to enable automatic provisioning of users between Forcepoint SMC and Azure AD

A description of the workflow between the components involved in this POC is depicted in this diagram:


1) Forcepoint NGFW engines are managed by a central SMC
Forcepoint NGFW
Security Management Center
NGFW Engines
4) Users can log into SMC with Azure users.
SMC roles are mapped automatically based on Azure group membership
Microsoft Azure
Azure AD
Azure AD Domain Services
Azure SCIM
Forcepoint SMC (App)
5) Users access SMC as an App inside the Azure Dashboard.
This add extra security (access to the App is managed by Azure AD policies) and provides remote management capabilities.
2) Azure AD Domain Services and LDAPs are automatically configured
Azure AD Secure Hybrid connector
SMC Connector
Deployment Service
SCIM Service
Reverse Proxy (Nginx)
3) Azure users and groups are mapped into SMC, with automated account lifecycle management via SCIM.
CUSTOMER PREMISES
AZURE

Caveats

The integration described in this document was developed and tested with the following products:

→ Forcepoint SMC 6.7.3 and Forcepoint NGFW 6.7.2

→ Azure Active Directory

This interoperability uses:

→ **Deployment Service**: a service that deploys Azure AD Domain Services template and app provisioning template, creates an external Active Directory authentication server in Forcepoint SMC, creates external users domain in

Forcepoint SMC and links Forcepoint SMC to an Azure app.

- **Reverse Proxy**: a server that handles requests from external clients (i.e. web browsers) to Forcepoint SCIM service and Forcepoint SMC web interface
- **Forcepoint SCIM Service**: a server that implements the SCIM v2 standard and listens to incoming SCIM requests from Azure SCIM for user provisioning.
- **SMC Connector**: a server that connects Forcepoint SCIM service with Forcepoint SMC.

Implementation options
Two implementation options are provided in this document

1. Docker – leverages docker images where the integration component is already installed with all necessary dependencies: the user only has to edit one docker-compose environment variable file and run containers on an existing docker setup.
2. Traditional – requires the manual deployment of the integration component inside a clean Centos 7 host-machine.

The docker images for this integration have been tested working with:

- Docker 19.03.6
- Docker-compose 1.25.4
- The docker host machine meets the minimum hardware requirements of 2GB of RAM and 20GB of storage

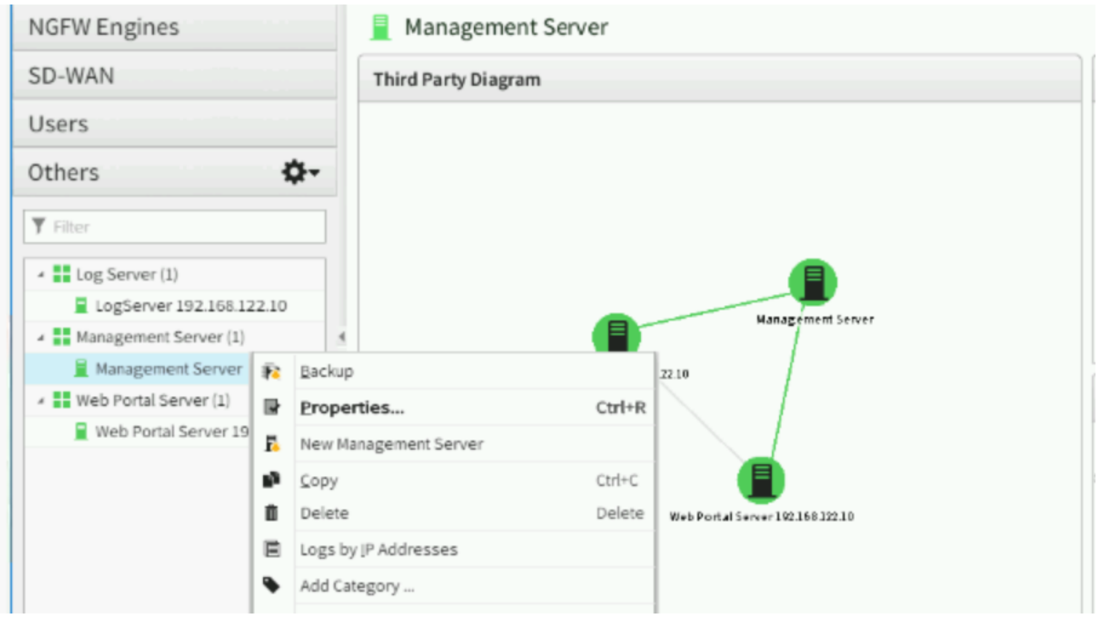
while the traditional version of this integration has been tested working with the following requirements

- Centos 7.3 with at least 2 GB RAM and 20 GB of storage

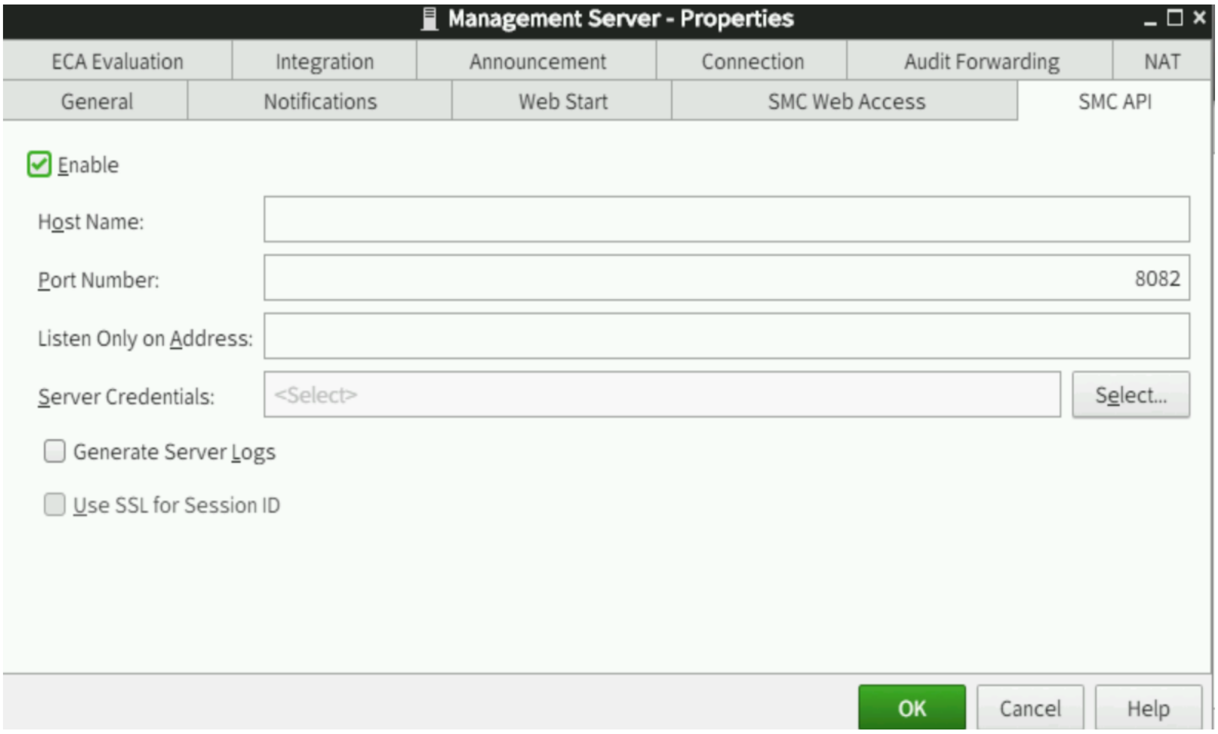
In this document we assume **Azure AD** is already in use but without **Azure AD Domain Services** and LDAPs connectivity. If either of those components are already in use, then the relative steps in the following chapters can be skipped.

Enable Forcepoint SMC Client API

1. Login to Forcepoint SMC with a superuser administrator account
2. Select **Home > Others** > right-click on **Management Server** and select **Properties**



3. Select **SMC API** and click on **Enable** then OK



☑ Enable

Host Name:

Port Number: 8082

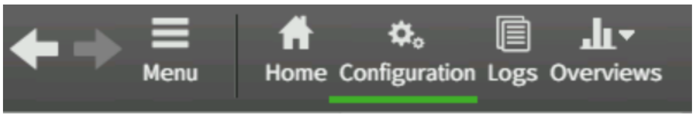
Listen Only on Address:

Server Credentials: <Select> Select...

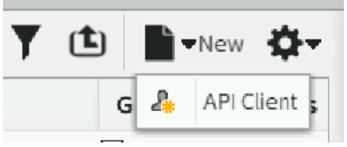
☐ Generate Server Logs

☐ Use SSL for Session ID

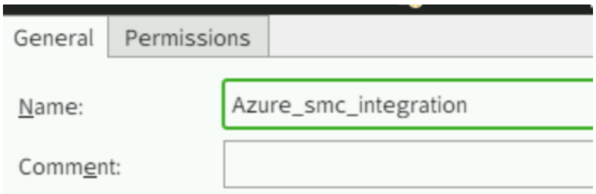
Now on the top menu of the SMC user interface



1. Click **Configuration** > **Administration** > **Access Rights** > **API Clients** > **New** > **API Client**



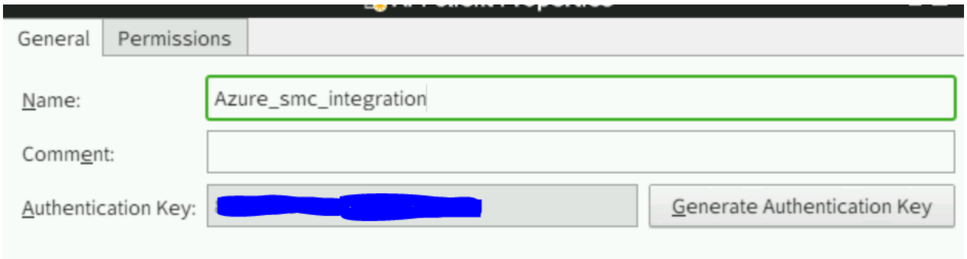
2. Insert a name for this API client



Name: Azure_smc_integration

Comment:

3. Save the **Authentication Key** in a safe location: this key will be used in the rest of this document and it will be referenced as **SMC_API_KEY**.

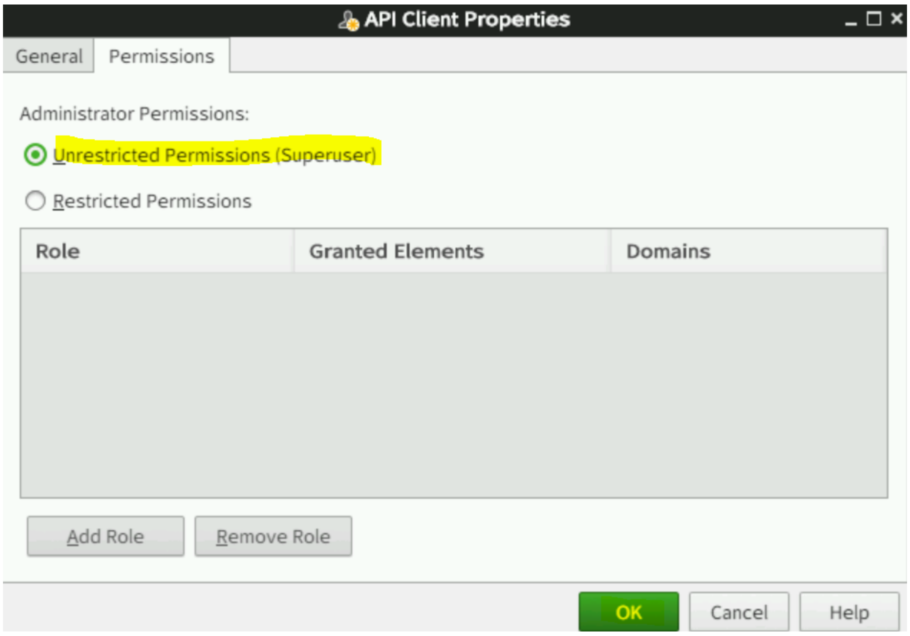


Name: Azure_smc_integration

Comment:

Authentication Key: [redacted]

4. Click **Permissions** > **Unrestricted Permissions (Superuser)** and click **OK**



Administrator Permissions:

● Unrestricted Permissions (Superuser)

○ Restricted Permissions

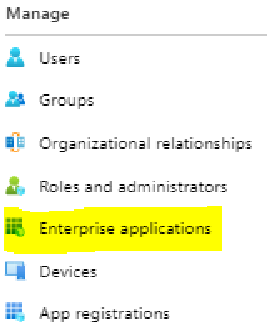
Create Azure Enterprise applications

This step shows how to create a new non-gallery application that will be used to link your on-premise Forcepoint SMC with this application.

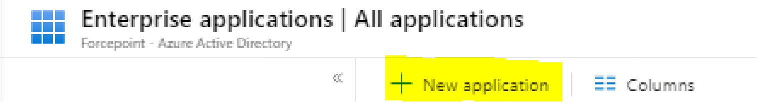
1. Sign in to your Azure account through the Azure portal with an administrator account that has **Global Administrator**

permissions

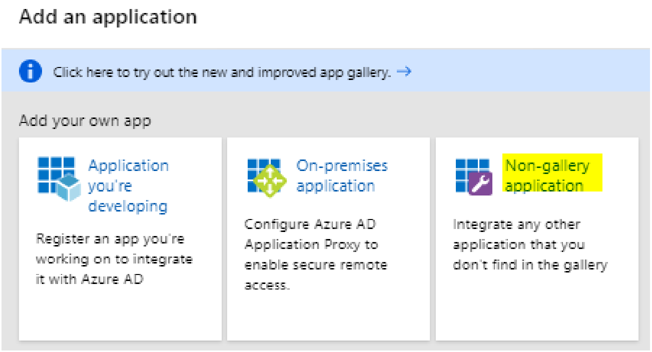
2. Go to **Azure Active Directory** > **Enterprise applications**



3. Click **New application**


Enterprise applications | All applications
Forcepoint - Azure Active Directory
«
+ New application
Columns

4. Click on the **Non-gallery application**.



Add an application

Click here to try out the new and improved app gallery. →

Add your own app

Application you're developing

Register an app you're working on to integrate it with Azure AD

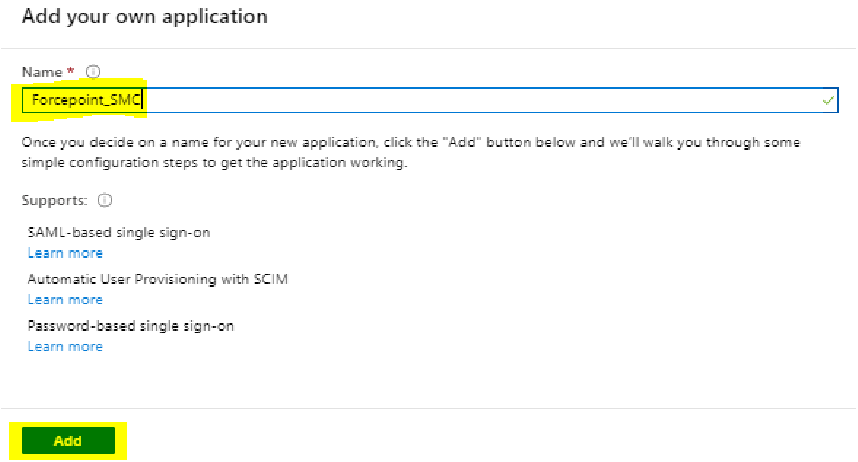
On-premises application

Configure Azure AD Application Proxy to enable secure remote access.

Non-gallery application

Integrate any other application that you don't find in the gallery

5. Enter a name for your new application and click **Add**



Add your own application

Name * ⓘ

Forcepoint_SMC ✓

Once you decide on a name for your new application, click the "Add" button below and we'll walk you through some simple configuration steps to get the application working.

Supports: ⓘ

SAML-based single sign-on
Learn more

Automatic User Provisioning with SCIM
Learn more

Password-based single sign-on
Learn more

Add

Implementation – Docker

The solution described in this chapter requires

→ A Linux machine (Centos 7.3 recommended with a minimum of 2GB of RAM and 20GB of storage) within the same network of Forcepoint SMC host machine. This machine requires a public IP address (or a public FQDN resolving into a public IP address) to expose its services to Azure. This machine will be referenced in the rest of this document as the **docker-host** machine**.**

The following components must be installed on the **docker-host** machine**:**

- Docker Engine installed on the Docker-host: if Docker Engine is not installed visit docker-installation-docs to install Docker Engine on Docker-host
- Docker Compose: if Docker Compose is not installed on the Docker-host machine, visit docker-compose-installation to install Docker, Compose on Docker-host
- The file **fp-ngfw-connect-Azure-ad-docker.tar.gz** available at the link https://frcpnt.com/ngfw-connect-Azure-ad-docker-latest

The archive **fp-ngfw-connect-Azure-ad-docker.tar.gz** contains the following files:

1. **docker-compse-deployment.yml**: docker-compose deployment file which will be used for deploying Azure templates into Azure, create an external Active Directory authentication server and external user domain in Forcepoint SMC.
2. **docker-compose-servers.yml**: docker-compose servers files which will be used to run all server containers (Nginx Reverse Proxy, Forcepoint SMC service, SMC connector)
3. **.env**: the environment variables files for docker-compose.
4. **certs**: is a directory for storing SSL certificates used by Nginx

Step 1: Login to Docker Registry

Use the following command and credentials to login into the Docker registry hosting the containers needed for this integration

```
root@linux:~# docker login docker.frcpnt.com
Username: fp-integrations
Password: t1knmAkn19s
```

Step 2: Modify .env file

Decompress **fp-ngfw-connect-Azure-ad-docker.tar.gz** and change your directory to **fp-ngfw-connect-Azure-ad**

```
tar -zxvf fp-ngfw-connect-Azure-ad-docker.tar.gz
cd fp-ngfw-connect-Azure-ad
```

Open **.env** file with a text editor such as vi

```
vi .env
```

Update the following variables:

1. **SMC_API_KEY:** is the SMC API Key which is generated in the chapter **Enable Forcepoint SMC Client API** of this document.
2. **SMC_IP_ADDRESS:** is the internal IP address of Forcepoint SMC.
3. **SMC_PORTAL:** is the Forcepoint SMC Web access portal, for example: SMC_PORTAL=192.168.122.10:8085
4. **AZURE_APP_NAME:** is the name of the Azure app which is created in the chapter **Create Azure Enterprise**

Application of this document

5. **AZURE_ADMIN_LOGIN_NAME:** is your Azure administrator login name. This administrator must have a **Global Administrator** role within Azure AD
6. **AZURE_DOMAIN_NAME:** is your Azure domain name
7. **AZURE_LOCATION:** is the Azure location where all resource will be created in Azure
8. **AZURE_RESOURCE_GROUP_NAME:** a name for Azure resource groups, if this resource group is not existing, the deployment process will create it.
9. **DOCKER_HOST_PUBLIC_IP_ADDRESS:** is the public IP address for the Docker-host machine
10. **PFX_CERTIFICATE_EXPIRY_DAYS:** the duration in days of the PFX certificate, after this the certificate will expire
11. **PFX_CERTIFICATE_PASSWORD:** a password that will be used for the PFX certificate

Once all variables are edited, save the **.env** file and move to the next step based on your existing Active Directory setup:

→ If you already have Azure AD Domain Services with LDAPs configured, move to **Step 8**

→ If you already have Azure AD Domain Services without LDAPs, move to **Step 5**

→ If you don’t have Azure AD Domain Services, continue to **Step 3**

Step 3: Create PFX certificate Base64 for secure LDAP

1. Run the deployment container:

```
docker-compose -f docker-compose-deployment.yml up -d
```

2. Generate the PFX base64 certificate:

```
docker-compose -f docker-compose-deployment.yml exec deployment /app/deployment generate-ssl-cert
```

3. The output of the above command is the Base65 string of the generated PFX certificate. Copy this output.
4. Stop and remove the deployment container:

```
docker-compose -f docker-compose-deployment.yml down
```

5. Insert the copied Base64 string as a value for PFX_CERTIFICATE_BASE64 variable in .env files variable. For example
   PFX_CERTIFICATE_BASE64=MIIQRQIBAzCCD9cGDSqGSId3DUEHSAAaCCD8gEgg/EMIIPwDCCBf……

Step 4: Deploy Azure AD DS template

1. Run the deployment container:

```
docker-compose -f docker-compose-deployment.yml up -d
```

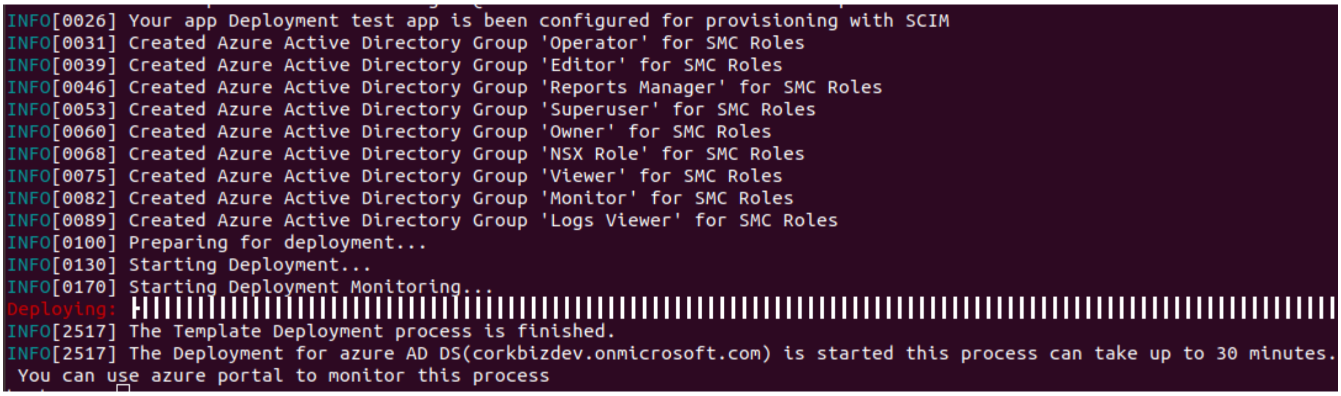
2. Interact with deployment container:

```
docker-compose -f docker-compose-deployment.yml exec deployment /bin/bash
```

3. Execute the following command to deploy the Azure AD DS, the application provisioning template and to create Azure groups for SMC roles:

```
./deployment deploy-azure -g
```

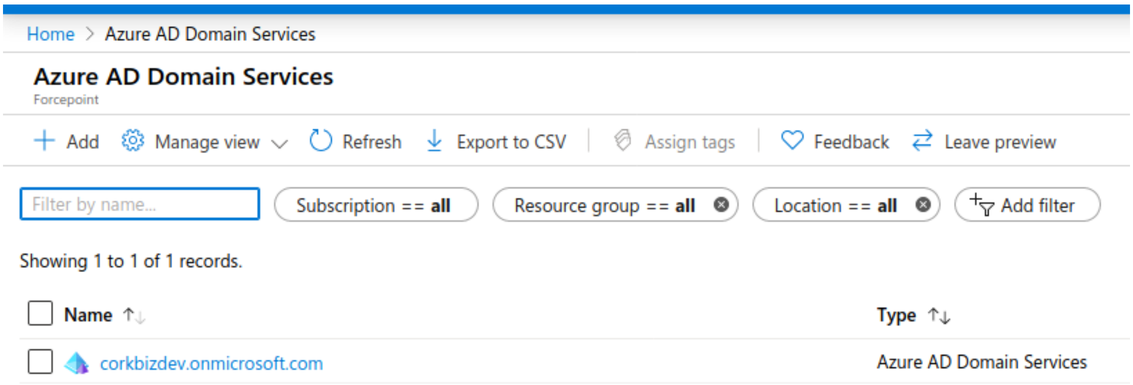
4. Enter your password for the administrator login name, then he deployment monitoring progress will start. Wait until the progress bar is completed. Provisioning of all resources inside Azure can take up to 55 minutes.



```
INFO[0026] Your app Deployment test app is been configured for provisioning with SCIM
INFO[0031] Created Azure Active Directory Group 'Operator' for SMC Roles
INFO[0039] Created Azure Active Directory Group 'Editor' for SMC Roles
INFO[0046] Created Azure Active Directory Group 'Reports Manager' for SMC Roles
INFO[0053] Created Azure Active Directory Group 'Superuser' for SMC Roles
INFO[0060] Created Azure Active Directory Group 'Owner' for SMC Roles
INFO[0068] Created Azure Active Directory Group 'NSX Role' for SMC Roles
INFO[0075] Created Azure Active Directory Group 'Viewer' for SMC Roles
INFO[0082] Created Azure Active Directory Group 'Monitor' for SMC Roles
INFO[0089] Created Azure Active Directory Group 'Logs Viewer' for SMC Roles
INFO[0100] Preparing for deployment...
INFO[0130] Starting Deployment...
INFO[0170] Starting Deployment Monitoring...
Deploying: ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
INFO[2517] The Template Deployment process is finished.
INFO[2517] The Deployment for azure AD DS(corkbizdev.onmicrosoft.com) is started this process can take up to 30 minutes.
 You can use azure portal to monitor this process
```

Once finished, Azure will start configuring Azure AD DS and this deployment will take up to 30 minutes and can only be monitored through Azure Portal.

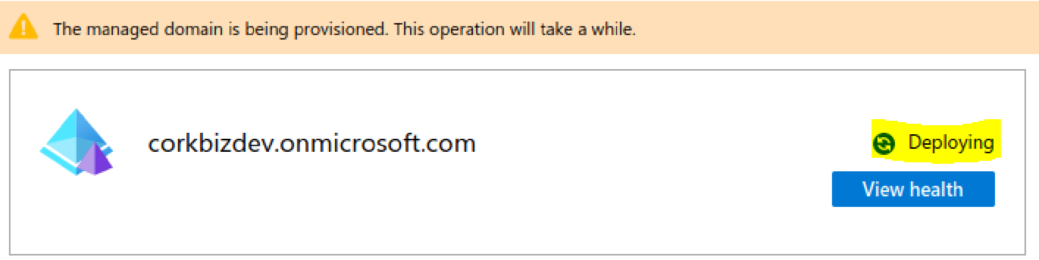
5. To monitor the ongoing deployment login to the Azure portal, search for **Azure AD Domain Services**, click on your **Azure AD Domain Services**



Showing 1 to 1 of 1 records.

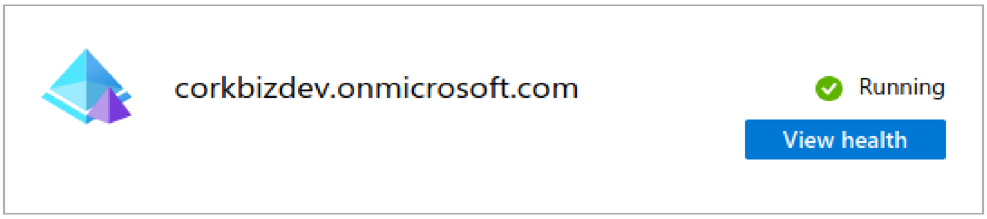
Name	Type
corkbizdev.onmicrosoft.com	Azure AD Domain Services

The status of the Domain Services will be **Deploying**



The managed domain is being provisioned. This operation will take a while.

Wait until the status of the Domain Services changes to **Running,** this can take up to 30 minutes



Once the new service is **Running** move to **step 6**.

Step 5: Enable LDAPs On Exist Azure AD DS.

In this section, we assume you already have an existing Azure AD Domain Service in your Azure Active Directory: the following steps show how to enable LDAPs.

Create a certificate for secure LDAP

1. Open a terminal
2. Create a private key with this command:

```
openssl genrsa 4096 > private.pem
```

3. Create a public key. Execute this command after replacing YOUR_AZURE_DOMAIN_NAME with your Azure domain name:

```
openssl req -x509 -days 365 -new -key private.pem -out public.pem -addext extendedKeyUsage=serverAuth,clientAuth -subj "/CN=*YOUR_AZURE_DOMAIN_NAME"
```

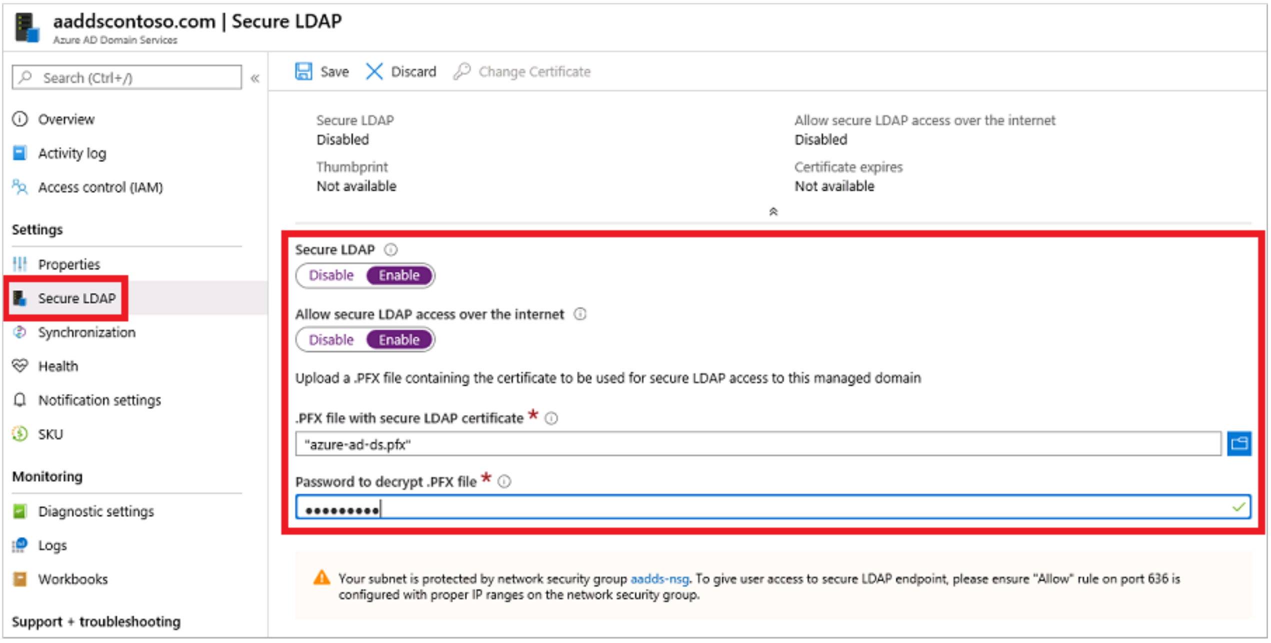
4. Create a PFX certificate. Execute this command after replacing PASSWORD with a password for FPX certificate, and store the password in a secure location as it will be used again in the next steps:

```
openssl pkcs12 -export -in public.pem -inkey private.pem -out Azure_cert.pfx  -password pass: PASSWORD
```

This will generate a PFX certificate named **Azure_cert.pfx** in your current directory. This certificate will be deployed to Azure AD DS in the next steps.

Enable secure LDAP

1. Login to Azure portal, search for **Azure AD Domain Services**.
2. Click on your Azure AD Domain Service.
3. Select **Secure LDAP**
4. By default, secure LDAP access to your managed domain is disabled: toggle **Secure LDAP** to **Enable**.
5. Secure LDAP access to your managed domain over the internet is disabled by default. Toggle **Allow secure LDAP access over the internet** to **Enable**
6. Select the folder icon next to **.PFX file with secure LDAP certificate**. Browse to the path of the **Azure_cert.pfx** file, then select the certificate **Azure_cert.pfx** .
7. Enter the password to decrypt .PFX file: this is the password that is used when **Azure_cert.pfx** is created.
8. Select **Save** to enable secure LDAP.



aaddscontoso.com | Secure LDAP
Azure AD Domain Services

Secure LDAP
Disabled

Allow secure LDAP access over the internet
Disabled

Thumbprint
Not available

Certificate expires
Not available

Secure LDAP: Disable [✓ Enable]

Allow secure LDAP access over the internet: Disable [✓ Enable]

Upload a .PFX file containing the certificate to be used for secure LDAP access to this managed domain

.PFX file with secure LDAP certificate *
"azure-ad-ds.pfx"

Password to decrypt .PFX file *
•••••••••

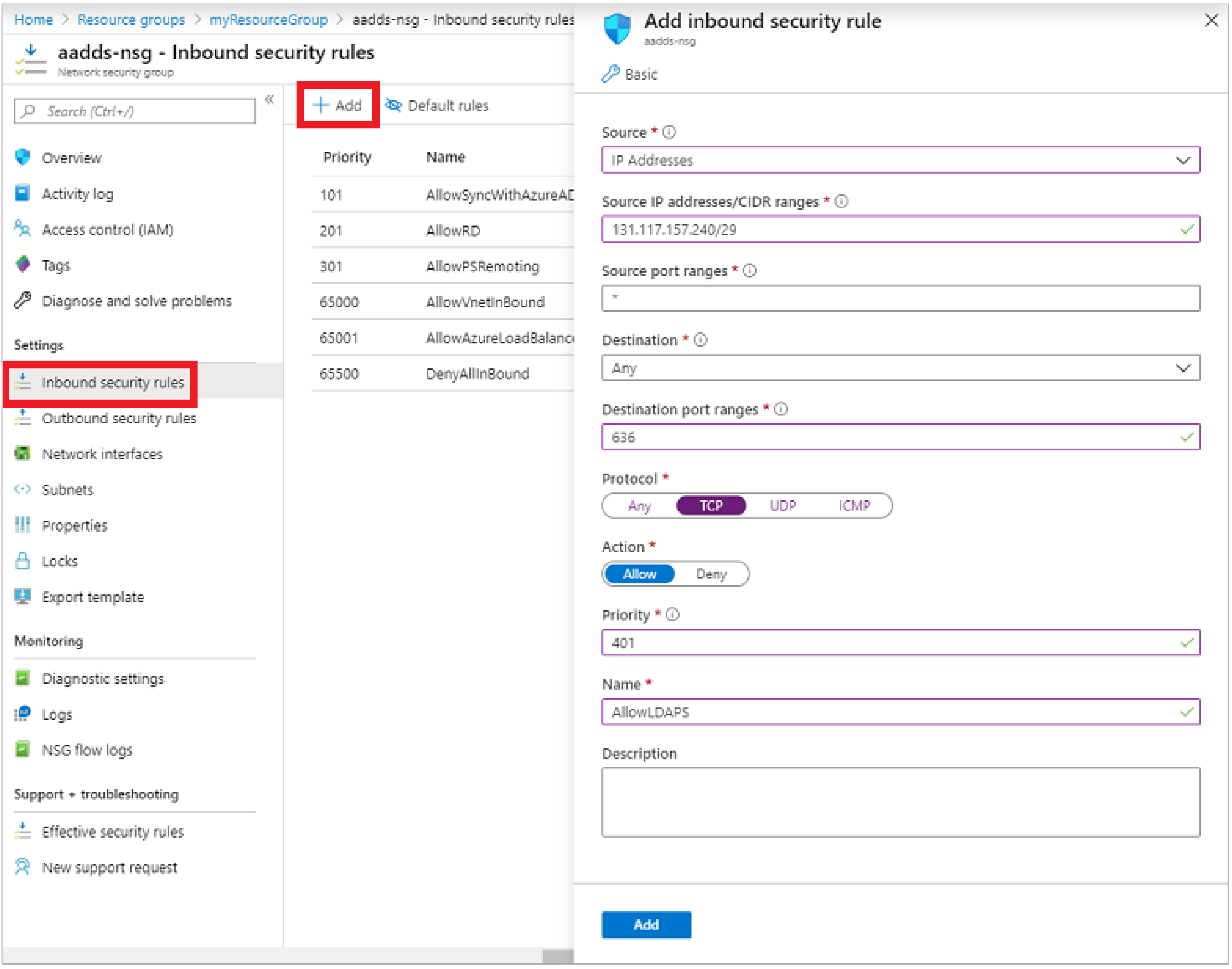
Your subnet is protected by network security group aadds-nsg. To give user access to secure LDAP endpoint, please ensure "Allow" rule on port 636 is configured with proper IP ranges on the network security group.

A notification is displayed that secure LDAP is being configured for the managed domain. You can't modify other settings for the managed domain until this operation is complete.

It takes a few minutes to enable secure LDAP for your managed domain.

Lockdown secure LDAP access over the internet

1. Click **Properties**, then select your network security group
2. On the left-hand side of the network security group pane, choose **Settings** > **Inbound security rules**
3. Click **Add**, then create a rule to allow **TCP** port **636:** For improved security, choose the source as **IP Addresses** and then specify your **Docker-host** machine public IP address. This is necessary to enable network connectivity to the Forcepoint SCIM service hosted on premise.



Priority	Name
101	AllowSyncWithAzureAD
201	AllowRD
301	AllowPSRemoting
65000	AllowVnetInBound
65001	AllowAzureLoadBalanc
65500	DenyAllInBound

Source *: IP Addresses

Source IP addresses/CIDR ranges *: 131.117.157.240/29

Source port ranges *: *

Destination *: Any

Destination port ranges *: 636

Protocol *: Any [✓ TCP] UDP ICMP

Action *: [✓ Allow] Deny

Priority *: 401

Name *: AllowLDAPS

Description:

4. Click **Add** to save and apply the rule.

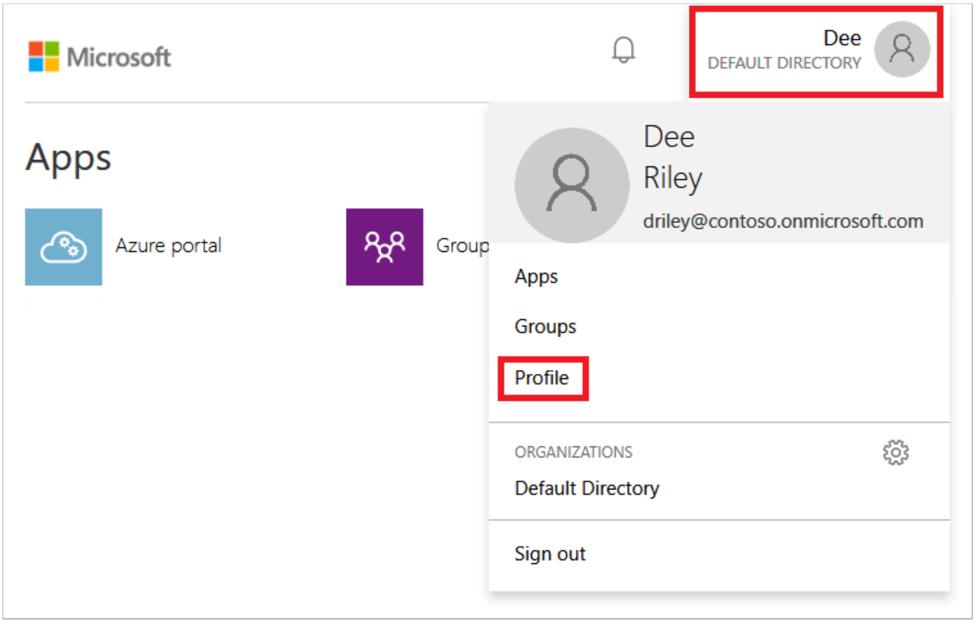
Step 6: Enable Azure AD Domain Services password hash synchronization

When Azure AD Domain Service is deployed for the first time, it does not contain any password hash for the existing users within Azure AD, therefore users intended to be used for SMC authentication must have their password changed before authentication in SMC will work.

The password change process will store password hashes inside Azure AD Domain Services so that users authenticating through LDAPs from SMC and other applications will be verified in a secure way. The preferred method to have password changes is left to the Azure AD administrator implementing this integration: for example manually expiring the passwords of all users who will use the SMC integration (this will force a password change upon a new sign-in attempt) or instructing users to manually change their password at their preferred schedule.

Manually password changing:

1. Go to the Azure AD Access Panel page at https://myapps.microsoft.com
2. In the top-right corner, select your name, then choose Profile from the drop-down menu.



3. On the Profile page, select Change password.
4. On the Change password page, enter your existing (old) password, then enter and confirm a new password.
5. Select Submit.

Wait 10 minutes after the password change has been completed (including the password of the user with Global Administrator role within Azure AD) then proceed to the next step.

Step 7: Create an external Active Directory authentication server and external user domain in SMC

1. Execute the following command:

```
./deployment deploy-smc
```

2. Entering the password for the Azure username with administrator role being used.
3. Exit from the deployment container with this command:

```
exit
```

4. Terminate docker-compose for the deployment with this command:

```
docker-compose -f docker-compose-deployment.yml down
```

Step 8: Run server containers

In the above steps we created all resources required on both Azure and Forcepoint SMC using the deployment docker-compose files. In this step we will configure the Nginx Reverse Proxy server, Forcepoint SCIM service and SMC Connector.

In the **Docker-host** machine do the following steps:

1. Open .env file.

```
vi /root/fp-ngfw-connect-Azure-ad/.env
```

2. Add this line to the end of .env file and replace the red Text with your Azure administrator password.

```
AZURE_ADMIN_LOGIN_PASSWORD=INSERT_YOUR_AZURE_ADMINISTRAOR_PASSWORD_HERE
```

3. Save the .env file

4. Change your directory to **/root/fp-ngfw-connect-Azure-ad/certs/.**

```
cd /root/fp-ngfw-connect-Azure-ad/certs
```

5. Create **cert.key** and **cert.crt** files to be used by Nginx for https connections.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout cert.key -out cert.crt -subj
"/CN=nginx-reverse"
```

6. Create **dhparam.pem** file to be used by Nginx for https connections.

```
sudo openssl dhparam -out dhparam.pem 2048
```

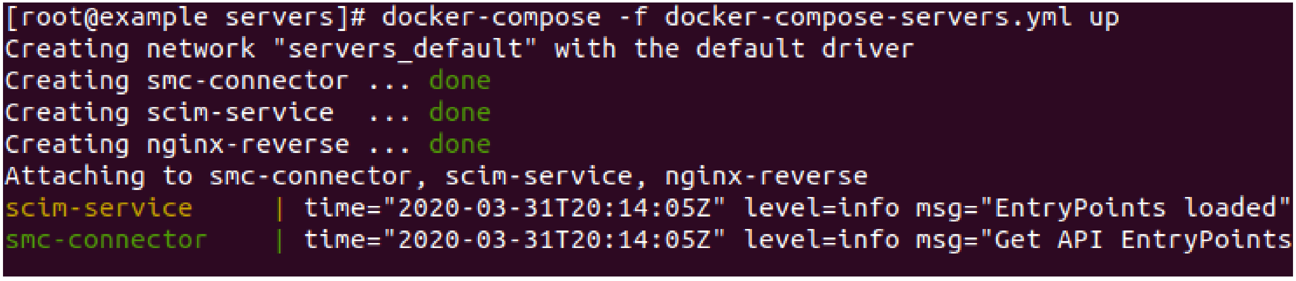
7. Return to **/root/fp-ngfw-connect-Azure-ad** directory.

```
cd /root/fp-ngfw-connect-Azure-ad
```

8. Run the following command to run Nginx Reverse Proxy server, Forcepoint SCIM service, and SMC Connector containers.

```
docker-compose -f docker-compose-servers.yml up
```

The result will be as in the following screenshot



```
[root@example servers]# docker-compose -f docker-compose-servers.yml up
Creating network "servers_default" with the default driver
Creating smc-connector ... done
Creating scim-service  ... done
Creating nginx-reverse ... done
Attaching to smc-connector, scim-service, nginx-reverse
scim-service      | time="2020-03-31T20:14:05Z" level=info msg="EntryPoints loaded"
smc-connector     | time="2020-03-31T20:14:05Z" level=info msg="Get API EntryPoints
```

Now all servers are running and ready to process any incoming request from Azure.

Step 9: Assign users to the Azure App

The Azure App is configured to sync only assigned users with Forcepoint SMC. To assign a user to your Azure app follow these steps:

1. Select your Azure app.
2. Select **Users and Groups**.
3. Click **Add user** > **Users and groups.**
4. Select the users to be assigned to the Azure app.
5. Click **Select** > **Assign**

Step 10: Add SCIM credentials

The last step is to add Forcepoint SCIM credentials to your Azure app for provisioning. Any Linux machine can be used for this step.

1. Open a terminal
2. Define these variables with your own information inside the terminal:

    user_name=YOUR_SMC_ADMIN_NAME

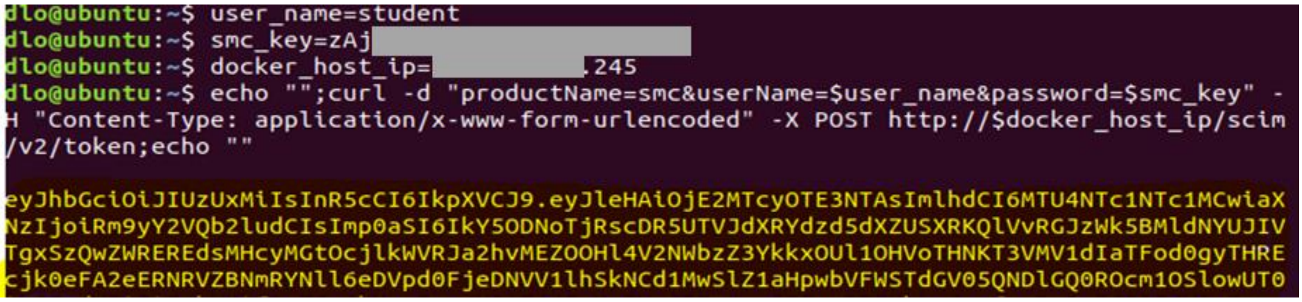
    smc_key=SMC_API_KEY

    docker_host_ip=YOUR_DOCKER_HOST_MACHINE_PUBLIC_IP

3. Execute the following command to obtain a valid access token for Forcepoint SCIM service.

```
curl -d "productName=smc&userName=$user_name&password=$smc_key" -H "Content-Type: application/x-www-form-urlencoded" -X POST http://$docker_host_ip/scim/v2/token;echo ""
```

The output of the above command is the value of a valid access token for Forcepoint SCIM Service

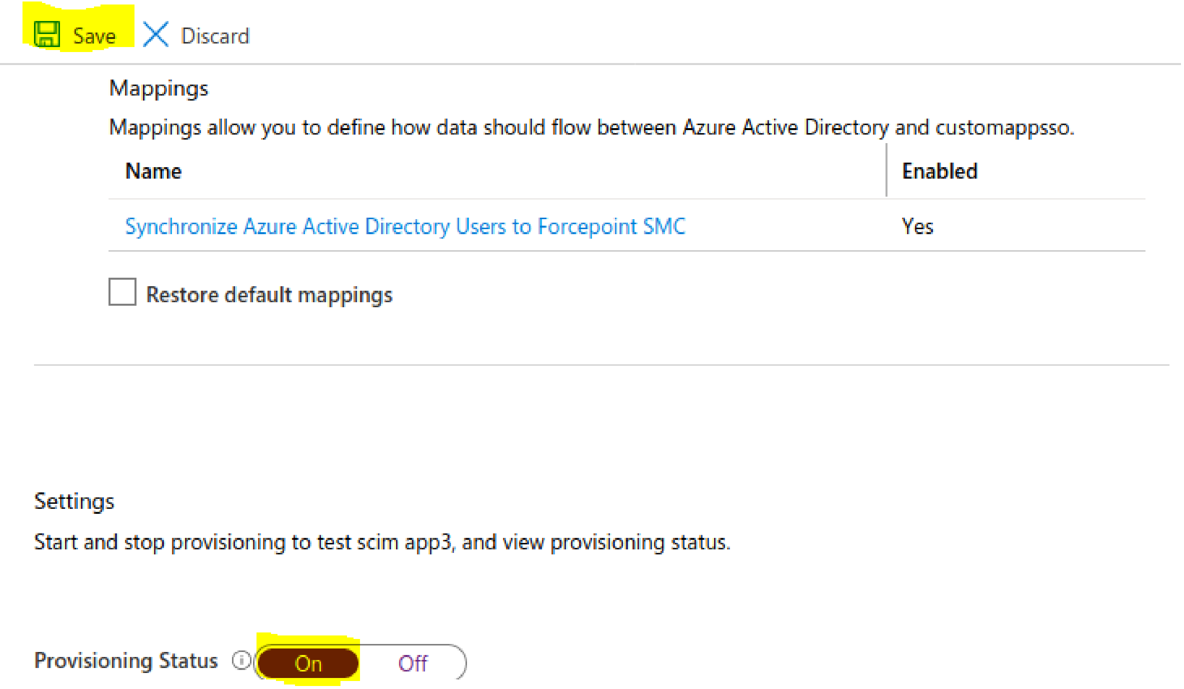


```
dlo@ubuntu:~$ user_name=student
dlo@ubuntu:~$ smc_key=zAj[redacted]
dlo@ubuntu:~$ docker_host_ip=[redacted].245
dlo@ubuntu:~$ echo "";curl -d "productName=smc&userName=$user_name&password=$smc_key" -
H "Content-Type: application/x-www-form-urlencoded" -X POST http://$docker_host_ip/scim
/v2/token;echo ""

eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2MTcyOTE3NTAsImlhdCI6MTU4NTc1NTc1MCwiaX
NzIjoiRm9yY2VQb2ludCIsImp0aSI6IkY5ODNoTjRscDR5UTVJdXRYdzd5dXZUSXRKQlVvRGJzWk5BMldNYUJIV
TgxSzQwZWREREdsMHcyMGtOcjlkWVRJa2hvMEZOOHl4V2NWbzZ3YkkxOUl1OHVoTHNKT3VMV1dIaTFod0gyTHRE
cjk0eFA2eERNRVZBNmRYNll6eDVpd0FjeDNVV1lhSkNCd1MwSlZ1aHpwbVFWSTdGV05QNDlGQ0ROcm1OSlowUT0
```

Copy the value of the access token (yellow text in the screenshot) and save it in a secure location: this access token will be used in the configuration of the Azure app for automated provisioning.

4. Login to the Azure portal.
5. Search **Azure Active Directory**.
6. Click **Enterprise applications.**
7. Find your Azure app and click on it.
8. Click **Provisioning**.
9. In the Tenant URL field enter **http://YOUR_DOCKER_HOST_PUBLIC_IP/scim/v2** replacing the red part with the actual public IP address of the **Docker-host** machine.
10. In the **Secret Token** field enter the access token for Forcepoint SCIM service obtained with the command at step 3 of this chapter.
11. Finally, change the **Provisioning status** to **On** and click **Save**.



Mappings

Mappings allow you to define how data should flow between Azure Active Directory and customappsso.

Name	Enabled
Synchronize Azure Active Directory Users to Forcepoint SMC	Yes

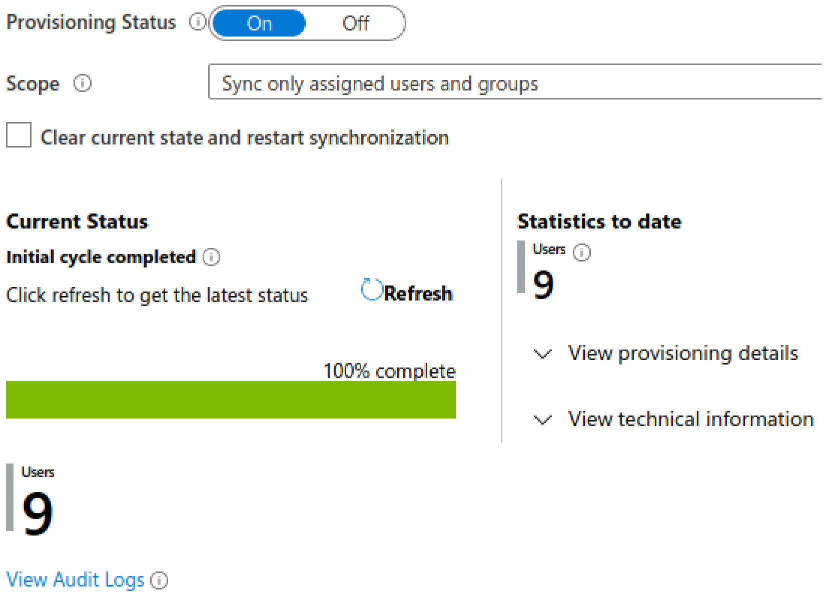
□ Restore default mappings

Settings

Start and stop provisioning to test scim app3, and view provisioning status.

Provisioning Status [✓ On] Off

Once you have saved the settings, the provisioning cycle will start. Provisioning cycle interval is 40 minutes. In each provisioning cycle, **Azure will only sync assigned users with your Forcepoint SMC**.



Provisioning Status ⓘ **On** Off

Scope ⓘ Sync only assigned users and groups

□ Clear current state and restart synchronization

Current Status

Initial cycle completed ⓘ

Click refresh to get the latest status **Refresh**

100% complete

Users
9

View Audit Logs ⓘ

Statistics to date

Users ⓘ
9

View provisioning details

View technical information

Once a provisioning cycle is completed, assigned users can login to **Forcepoint SMC** with their Azure credentials where login name will be in this format **<FirstName>.<LastName>.** For example, if the Azure login name is john.doe@Azuredomain.com, the login name for Forcepoint SMC will be **john.doe**

Step 11: Apply SMC roles to Azure users
For all newly synced users, their permissions assigned within SMC will be **Viewer**. Different SMC roles can be assigned to Azure users by simply changing the user's group membership.

In the Azure Active Directory, the following groups are automatically created to mirror the SMC administrator permissions:

→ **Editor**

→ **Logs Viewer**

→ **Monitor**

→ **Operator**

→ **Owner**

→ **Reports Manager**

→ **Viewer**

→ **Superuser**

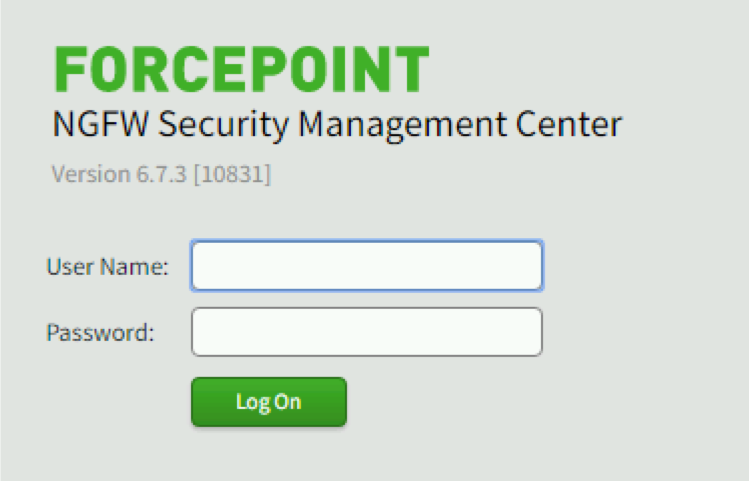
To add/remove permissions to a user, simply add/remove that user from the corresponding group.

Example: to give **Editor** and **Monitor** permission to user B, simply add user B as a member to the AD group **Editor** and to the AD group **Monitor**.

The members of superuser groups would have full administrator permissions into the SMC.

Step 12: Access on-promise Forcepoint SMC via Azure application
users how are assigned to your Azure app, can use your Azure app to access Forcepoint SMC on-promise with the following steps:

1. Login to https://myapplications.microsoft.com/
2. Find your Azure app and click on it. This will redirect your web browser to Forcepoint SMC on-promise.
3. Enter your Azure credentials: Username is your Azure nickName.


FORCEPOINT

NGFW Security Management Center

Version 6.7.3 [10831]

User Name:

Password:

Implementation - Traditional

The solution described in this chapter requires

- A Centos 7.3 machine (with at least 2GB of RAM and 20GB of storage) able to reach the Forcepoint SMC host machine over the network. This machine requires a public IP address or public DNS to expose its services and it will be referenced in the rest of this document with the name **host-machine.**
- The source files for this implementation, contained in the archive **fp-ngfw-connect-Azure-ad.tar.gz** available at the link http://frcpnt.com/ngfw-connect-Azure-ad-latest

The archive **fp-ngfw-connect-Azure-ad.tar.gz** contains the following files and folders:

- **scim-smc**: the Forcepoint SCIM service application.
- **smc-connector**: the SMC Connector application.
- **deployment**: the deployment application for deploying templates to Azure, creating external Active Directory authentication server and external users domain in Forcepoint SMC.
- **scim.yml**: the configuration for the Forcepoint SCIM service application.
- **connector.yml**: the configuration file for the SMC Connector application.
- **deployment.yml**: the configuration file for the deployment application.
- **Nginx** folder, where Nginx configurations are stored.
- **forcepoint_scim.service**: systemd file for the Forcepoint SCIM service.
- **smc_connector.service**: systemd file for SMC Connector service.
- **installation_script.sh**: a bash script to install all required dependencies.

Step 1: Modify configuration files

Inside the **host-machine** unpack the **fp-ngfw-connect-Azure-ad.tar.gz** archive and change your directory to **fp-ngfw-connect-Azure-ad**

```
tar -zxvf fp-ngfw-connect-Azure-ad.tar.gz
cd fp-ngfw-connect-Azure-ad
```

Create an SSL certificate that will be used inside the Nginx reverse proxy:

1. Change the directory to **/root/fp-ngfw-connect-Azure-ad/nginx/certs/.**

```
cd /root/fp-ngfw-connect-Azure-ad/nginx/certs
```

2. Create **cert.key** and **cert.crt** files.

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout cert.key -out cert.crt -
subj "/CN=nginx-reverse"
```

3. Create **dhparam.pem** file.

```
sudo openssl dhparam -out dhparam.pem 2048
```

4. Return to **root/fp-ngfw-connect-Azure-ad** directory.

```
cd /root/fp-ngfw-connect-Azure-ad
```

Modify deployment.yml file

The contents of **deployment.yml** file are as follows:

```
SMC:
  IP_ADDRESS: INSERT_USER_SMC_IP_ADDRESS_HERE
  PORT: "8082"
  API_VERSION: "6.7"
  KEY: INSERT_YOUR_SMC_API_KEY_HERE
APP_NAME: INSERT_YOUR_AZURE_APPLICATION_NAME
AZURE_ADMIN_LOGIN_NAME: INSERT_YOUR_AZURE_ADMINISTRATOR_LOGIN_NAME_HERE
DOMAIN_NAME: INSERT_YOUR_AZURE_DOMAIN_NAME_HERE
LOCATION: INSERT_AZURE_LOCATION_HERE
DOMAIN_SERVICES_VNET_NAME: domain-services-vnet
DOMAIN_SERVICES_VNET_ADDRESS_PREFIX: 10.0.0.0/16
DOMAIN_SERVICES_SUBNET_NAME: domain-services-subnet
DOMAIN_SERVICES_SUBNET_ADDRESS_PREFIX: 10.0.0.0/24
NGINX_PUBLIC_IP_ADDRESS: INSERT_YOUR_HOS_MACHINE_PUBLIC_IP_ADDRESS
LOGGER_JSON_FORMAT: false
RESOURCE_GROUP: INSERT_AZURE_RESOURCE_GROUP_NAME_HERE
PARAMETERS_PATH: /root/smc_connecotr/fp-ngfw-connect-Azure-ad
DEPLOYMENT_TEMPLATE: /root/fp-ngfw-connect-Azure-ad/Azure_smc_template.json
SCIM_TEMPLATE: /root/fp-ngfw-connect-Azure-ad/scim_template.json
PFX_CERTIFICATE_EXPIRY_DAYS: INSERT_NUMBER_OF_DAYS_FOR_PFX_CERTIFICATE_EXPIRATION_HERE
```

```
PFX_CERTIFICATE_PASSWORD: INSERT_A_PASSWORD_FOR_PFX_CERTIFICATE_HERE
PFX_CERTIFICATE_BASE64: PFX_BASE64_WILL_BE_INSERTED_HERE
```

Open **deployment.yml** file with a text editor such as vi and do the following steps.

1. Replace **INSERT_USER_SMC_IP_ADDRESS_HERE** with your Forcepoint SMC internal IP address.
2. Replace **INSERT_YOUR_SMC_API_KEY_HERE** with your Forcepoint SMC Client API key
3. Replace **INSERT_YOUR_AZURE_APPLICATION_NAME** with your Azure app name that was created in the initial chapter **Create Azure Enterprise Application** of this document
4. Replace **INSERT_YOUR_AZURE_ADMINISTRATOR_LOGIN_NAME_HERE** with your Azure Administrator login name, this administrator most have **Global administrator** role.
5. Replace **INSERT_YOUR_AZURE_DOMAIN_NAME_HERE** with your Azure Domain Name
6. Replace **INSERT_AZURE_LOCATION_HERE** with an Azure region. All Azure resources will be created in this location
7. Replace **INSERT_YOUR_HOS_MACHINE_PUBLIC_IP_ADDRESS** with the public address of the **host-machine**.
8. Replace **INSERT_AZURE_RESOURCE_GROUP_NAME_HERE** with your Azure resource group name, if the resource group name does not exist it will be created.
9. Replace **INSERT_NUMBER_OF_DAYS_FOR_PFX_CERTIFICATE_EXPIRATION_HERE** with the number of days for PFX certificate to be expired
10. Replace **INSERT_A_PASSWORD_FOR_PFX_CERTIFICATE_HERE** with a password that will be used as a password for the PFX certificate.

Save the deployment.yml file, and move to the next step

Step 2: Install dependencies

Execute the following command to make **installation_script.sh** executable

```
chmod +x installation_script.sh
```

installation_script.sh will create systemd services for **Forcepoint SCIM** service, **SMC connector** and will install the following packages:

- Python3
- Golang 1.14
- Azure CLI
- OpenSSL (upgrade to the latest version)
- Nginx

Execute **installation_script.sh** and pass your Forcepoint SMC internal IP address as a parameter to this script by replacing the part in red in the following example:

```
sudo ./installation_script.sh <YOUR_SMC_IP_ADDRESS>
```

Once the installation has finished, do NOT reboot the **host-machine** but move to the next step.

Step 3: Create PFX certificate Base64 for secure LDAP

Skip this step if you have Azure AD Domain Services already deployed in your Azure Active Directory.

1. Make sure you are inside **fp-ngfw-connect-Azure-ad** directory.
2. Run the following command which will generate a Base64 string of PFX certificate.

```
./deployment generate-ssl-cert --config ./deployment.yml
```

3. Copy the output of the above command and use it to replace the placeholder `PFX_BASE64_WILL_BE_INSERTED_HERE` in the **deployment.yml** file

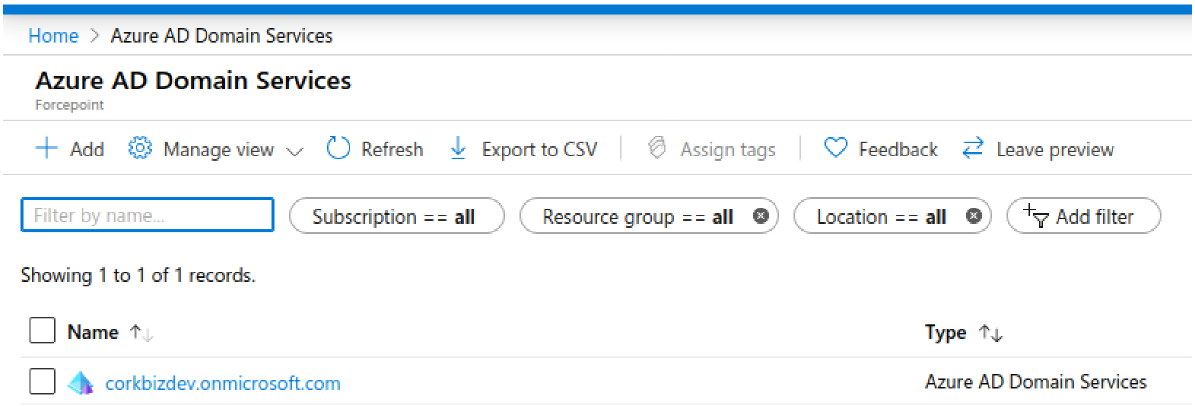
Step 4: Deploy Azure AD DS template

If **Azure AD Domain Services** is already deployed in Azure Active Directory, skip this step and **move to step 5**.

1. Execute the following command to deploy the Azure AD DS, the application provisioning template and to create Azure groups for SMC roles:

```
./deployment deploy-Azure --config ./deployment.yml -g
```

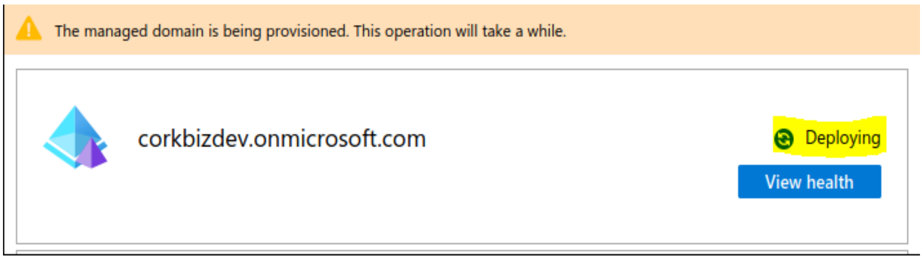
2. Enter your password for the administrator login name displayed on screen:
3. The deployment monitoring progress will start, wait until the progress bar is completed: this can take up to 55 minutes.
4. Once the above template deployment has finished, Azure will start deploying Azure AD DS and this deployment will take up to 30 minutes and can only be monitored through Azure Portal.
5. Login to the Azure portal, search for **Azure AD Domain Services**.
6. Click on your Azure AD Domain Services



Showing 1 to 1 of 1 records.

□ Name ↑↓	Type ↑↓
□ corkbizdev.onmicrosoft.com	Azure AD Domain Services

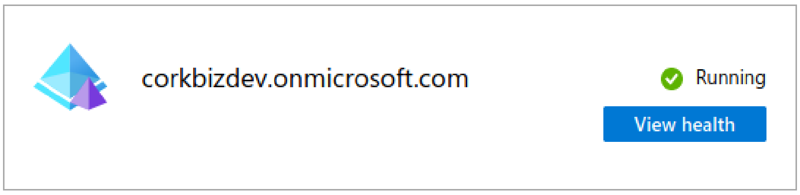
The initial status of the Domain Services will be **Deploying**



The managed domain is being provisioned. This operation will take a while.

corkbizdev.onmicrosoft.com

Wait until the status of the Domain Services changes to **Running**, then **move to step 6**.



corkbizdev.onmicrosoft.com

Step 5: Enable LDAPs On Existing Azure AD DS.
In this section we assume you already have an existing Azure AD Domain Service in your Azure Active Directory: the following steps show how to enable LDAPs.

Create a certificate for secure LDAP

1. Open a terminal
2. Create a private key with this command:

```
openssl genrsa 4096 > private.pem
```

3. Create a public key. Execute this command after replacing YOUR_AZURE_DOMAIN_NAME with your Azure domain name.

```
openssl req -x509 -days 365 -new -key private.pem -out public.pem -addext extendedKeyUsage=serverAuth,clientAuth -subj "/CN=*YOUR_AZURE_DOMAIN_NAME"
```

4. Create a PFX certificate. Execute this command after replacing PASSWORD with a password for FPX certificate, and store the password in a secure location as it will be used again in the next steps

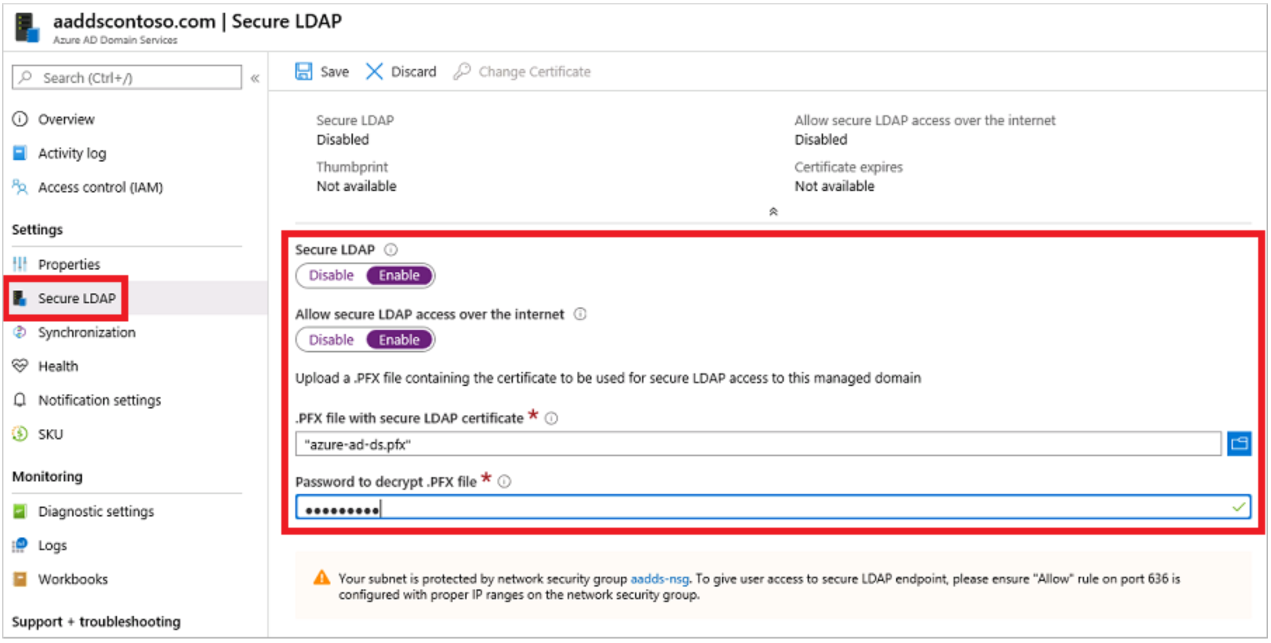
```
openssl pkcs12 -export -in public.pem -inkey private.pem -out Azure_cert.pfx  -password pass: PASSWORD
```

This will generate a PFX certificate named **Azure_cert.pfx** in your current directory. This certificate will be deployed to Azure AD DS in the next steps.

Enable secure LDAP

1. Login to Azure portal, search for **Azure AD Domain Services**.
2. Click on your Azure AD Domain Service.
3. Select **Secure LDAP**
4. By default, secure LDAP access to your managed domain is disabled: toggle **Secure LDAP** to **Enable**.
5. Secure LDAP access to your managed domain over the internet is disabled by default. Toggle **Allow secure LDAP access over the internet** to **Enable**
6. Select the folder icon next to **.PFX file with secure LDAP certificate**. Browse to the path of the **Azure_cert.pfx** file, then select the certificate **Azure_cert.pfx** .
7. Enter the password to decrypt .PFX file: this is the password that is used when **Azure_cert.pfx** is created.

8. Select **Save** to enable secure LDAP.



Secure LDAP
Disabled

Allow secure LDAP access over the internet
Disabled

Thumbprint
Not available

Certificate expires
Not available

Secure LDAP: Disable [✓ Enable]

Allow secure LDAP access over the internet: Disable [✓ Enable]

Upload a .PFX file containing the certificate to be used for secure LDAP access to this managed domain

.PFX file with secure LDAP certificate *: "azure-ad-ds.pfx"

Password to decrypt .PFX file *: ●●●●●●●●●

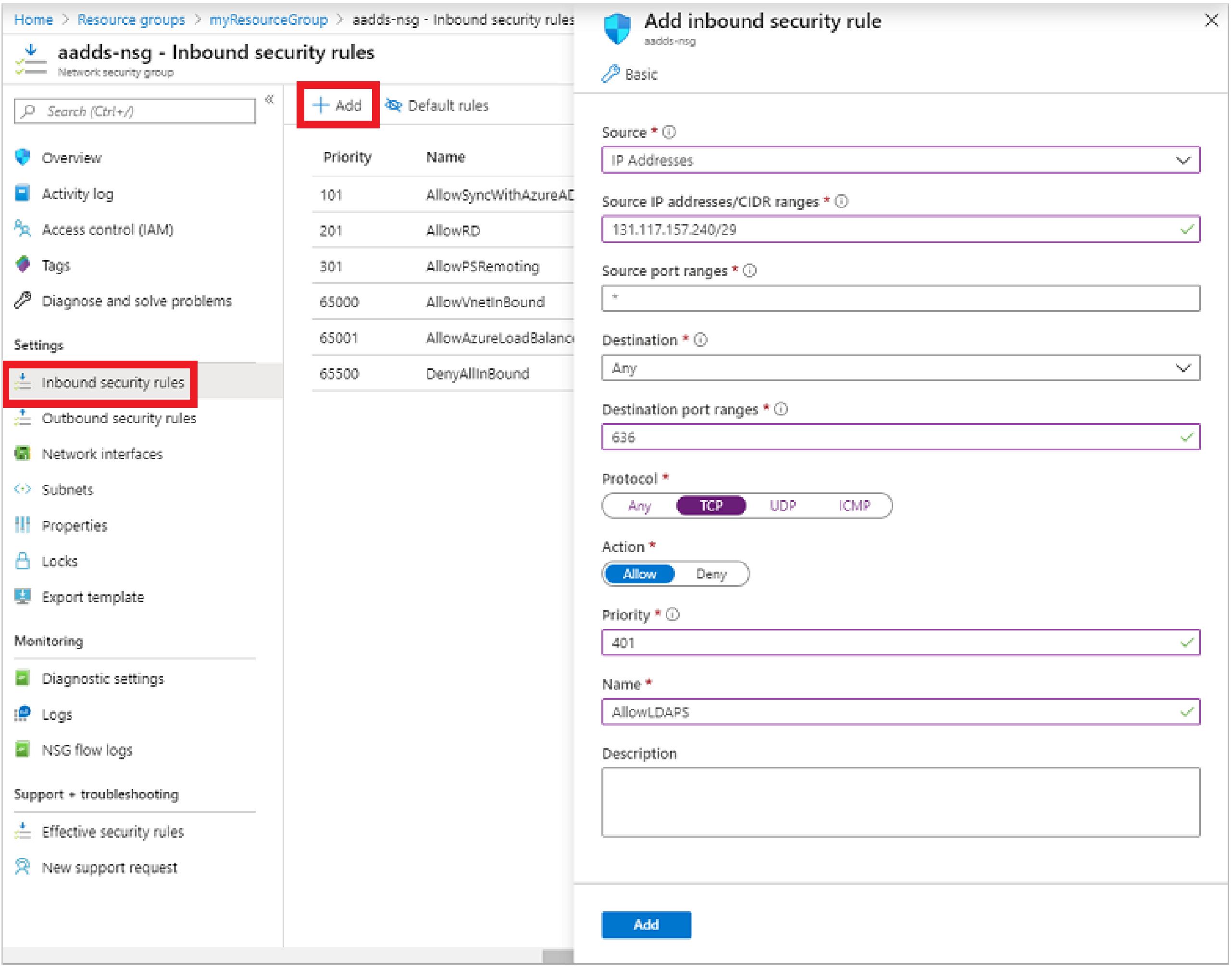
Your subnet is protected by network security group aadds-nsg. To give user access to secure LDAP endpoint, please ensure "Allow" rule on port 636 is configured with proper IP ranges on the network security group.

A notification is displayed that secure LDAP is being configured for the managed domain. You can't modify other settings for the managed domain until this operation is complete.

It takes a few minutes to enable secure LDAP for your managed domain.

Lockdown secure LDAP access over the internet

1. Click **Properties**, then select your network security group.
2. On the left-hand side of the network security group pane, choose **Settings** > **Inbound security rules.**
3. Click **Add**, then create a rule to allow **TCP** port **636:** for improved security, choose the source as **IP Addresses** and then specify your **Docker-host** machine public IP address. This is necessary to enable network connectivity to the Forcepoint SCIM service hosted on premise.



Priority	Name
101	AllowSyncWithAzureAD
201	AllowRD
301	AllowPSRemoting
65000	AllowVnetInBound
65001	AllowAzureLoadBalanc
65500	DenyAllInBound

Source * : IP Addresses

Source IP addresses/CIDR ranges * : 131.117.157.240/29

Source port ranges * : *

Destination * : Any

Destination port ranges * : 636

Protocol * : Any [✓ TCP] UDP ICMP

Action * : [✓ Allow] Deny

Priority * : 401

Name * : AllowLDAPS

Description

4. Click **Add** to save and apply the rule

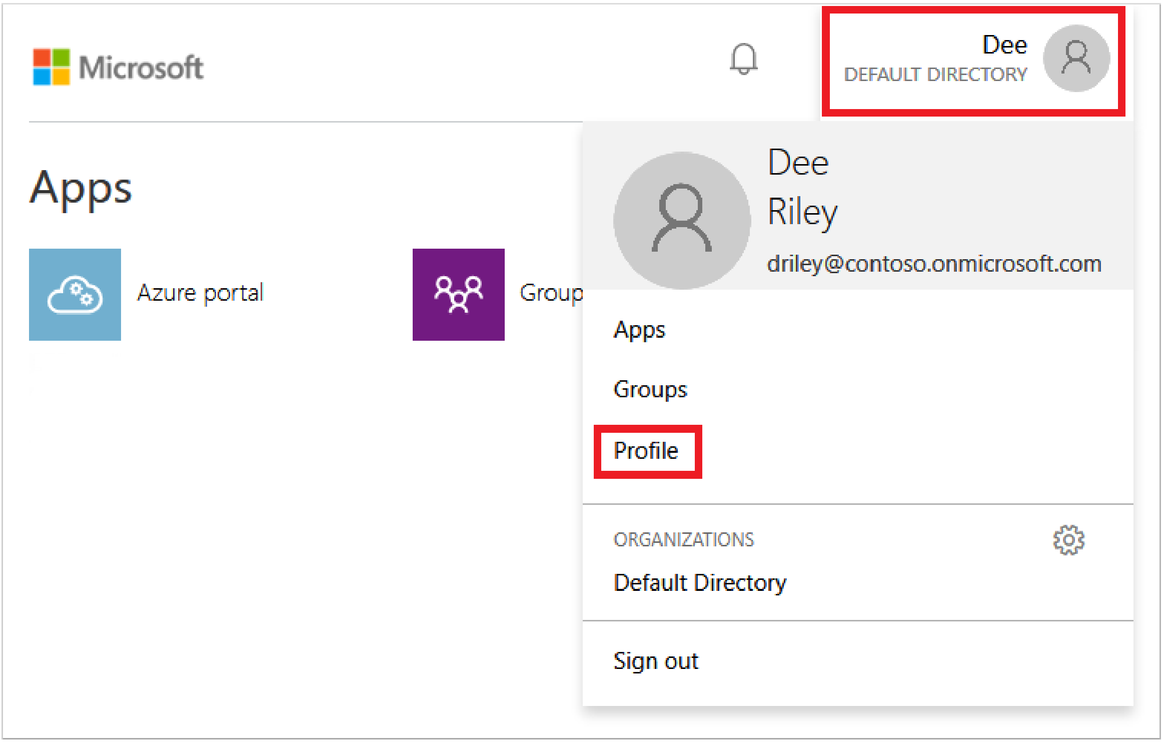
Step 6: Enable Azure AD Domain Services password hash synchronization

When Azure AD Domain Service is deployed for the first time, it does not contain any password hash for the existing users within Azure AD, therefore **users intended to be used for SMC authentication must have their password changed before authentication in SMC will work**.

The password change process will store password hashes inside Azure AD Domain Services so that users authenticating through LDAPs from SMC and other applications will be verified in a secure way. The preferred method to have password changes is left to the Azure AD administrator implementing this integration: for example manually expiring the passwords of all users who will use the SMC integration (this will force a password change upon a new sign-in attempt) or instructing users to manually change their password at their preferred schedule.

Manually password changing:

1. Go to the Azure AD Access Panel page at https://myapps.microsoft.com
2. In the top-right corner, select your name, then choose Profile from the drop-down menu.



3. On the Profile page, select Change password.
4. On the Change password page, enter your existing (old) password, then enter and confirm a new password.
5. Select Submit.

Wait 10 minutes after the password change has been completed (including the password of the user with Global Administrator role within Azure AD) then proceed to the next step.

Step 7: Create an external Active Directory authentication server and external user domain in SMC

1. Execute the following command.

```
./deployment deploy-smc --config ./deployment.yml
```

2. Enter your Azure administrator password for the displayed login name

Step 8: Modify /var/Azure_smc/connector.yml file

The content of /var/Azure_smc/connector.yml file is:

```
SMC:
  IP_ADDRESS: INSERT_USER_SMC_IP_ADDRESS_HERE
  PORT: 8082
  API_VERSION: 6.7
  NAME: smc
  KEY: INSERT_YOUR_SMC_API_KEY_HERE
CONNECTOR:
  HOSTNAME: localhost
  PORT: 8085
LOG_FORMAT_JSON: false
LDAP_DOMAIN: INSERT_YOUR_AZURE_DOMAIN_NAME_HERE
```

```
APP_NAME: INSERT_YOUR_AZURE_APP_NAME_HERE
AZURE_ADMIN_LOGIN_NAME: INSERT_YOUR_AZURE_ADMINSTRATOR_LOGIN_NAME_HERE
AZURE_ADMIN_LOGIN_PASSWORD: INSERT_YOUR_AZURE_ADMINSTRATOR_PASSWORD_HERE
ROLES_UPDATE_TIME_IN_MINUTES: 10
# one or multiple Permissions are required to be assigned to a newly created user.
# the permissions are: Logs Viewer, Reports Manager,  Owner, Viewer, Operator, Monitor, Editor, NSX Role, Superuser
# if you want to set restricted permissions select one or more permissions from:  Logs_Viewer, Reports_Manager,  Owner, Viewer, Operator, Monitor, Editor, NSX_Role
# for Unrestricted Permissions select Superuser only
# the roles with true value will be selected as default roles.
ROLES:
  PERMISSIONS:
    VIEWER: true
    LOGS_VIEWER: false
    REPORTS_MANAGER: false
    OWNER: false
    OPERATOR: false
    MONITOR: false
    EDITOR: false
    NSX_ROLE: false
    SUPPERUSER: false
  # can log in to SMC API
  CAN_USE_API: true
  # allow sudo on engines
  ALLOW_SUDO: false
  # user can sudo via SSH/console. this only can be true if the selected permission is Superuser
  CONSOLE_SUPPER_USER: false
  # user can log in to the shared domain
  ALLOW_TO_LOGS_IN_SHARED: true
```

Open **/var/Azure_smc/connector.yml** with a text editor such as vi and do the following steps.

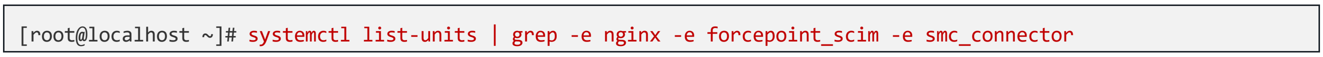
1. Replace **INSERT_USER_SMC_IP_ADDRESS_HERE** with your Forcepoint SMC internal IP address.
2. Replace **INSERT_YOUR_SMC_API_KEY_HERE** with your Forcepoint SMC Client API key
3. Replace **INSERT_YOUR_AZURE_DOMAIN_NAME_HERE** with your Azure Domain Name.
4. **Replace INSERT_YOUR_AZURE_APP_NAME_HERE** with your Azure application name.
5. **Replace INSERT_YOUR_AZURE_ADMINSTRATOR_LOGIN_NAME_HERE** with your Azure Administrator login-name.
6. **Replace INSERT_YOUR_AZURE_ADMINSTRATOR_PASSWORD_HERE** with your Azure Administrator password.
7. Save the **connector.yml** file

Step 9: Reboot Host-Machine

1. Reboot the host-machine
2. Check the status of **Nginx.service**, **forcepoint_scim.service** and **smc_connector.service**:

```
systemctl list-units | grep -e nginx -e forcepoint_scim -e smc_connector
```

The expected output is similar to the below screenshot: all services are running and ready to process any income request from Azure.



```
[root@localhost ~]# systemctl list-units | grep -e nginx -e forcepoint_scim -e smc_connector
```

```
forcepoint_scim.service             loaded active running   Forcepoint SCIM service
nginx.service                       loaded active running   The nginx HTTP and reverse proxy server
smc_connector.service               loaded active running   Forcepoint SMC connector service
```

.

Step 10: Add SCIM credentials

The last step is to add Forcepoint SCIM credentials to your Azure app for provisioning. Any Linux machine can be used for this step.

1. Open a terminal.
2. Define these variables with your own information inside the terminal:

   user_name=YOUR_SMC_ADMIN_NAME

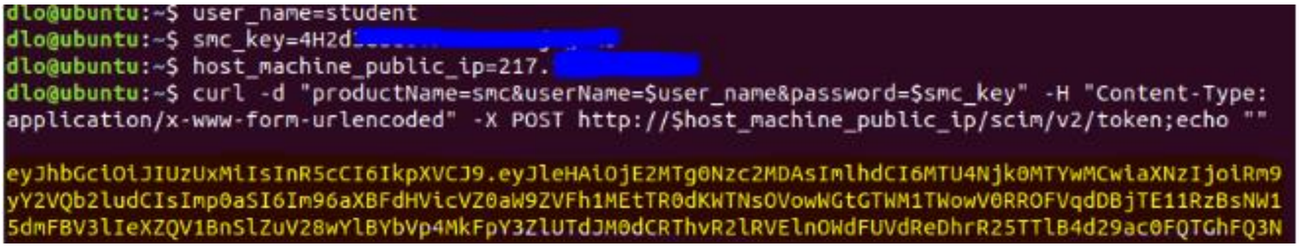
   smc_key=SMC_API_KEY

   host_machine_public_ip=YOUR_HOST_MACHINE_PUBLIC_IP

3. Execute the following command in any Linux machine in order to obtain a valid access token for Forcepoint SCIM service.

```
curl -d "productName=smc&userName=$user_name&password=$smc_key" -H "Content-Type: application/x-www-form-urlencoded" -X POST http://$host_machine_public_ip/scim/v2/token;echo ""
```

Output: the output of the above command is the value of a valid access token for Forcepoint SCIM Service

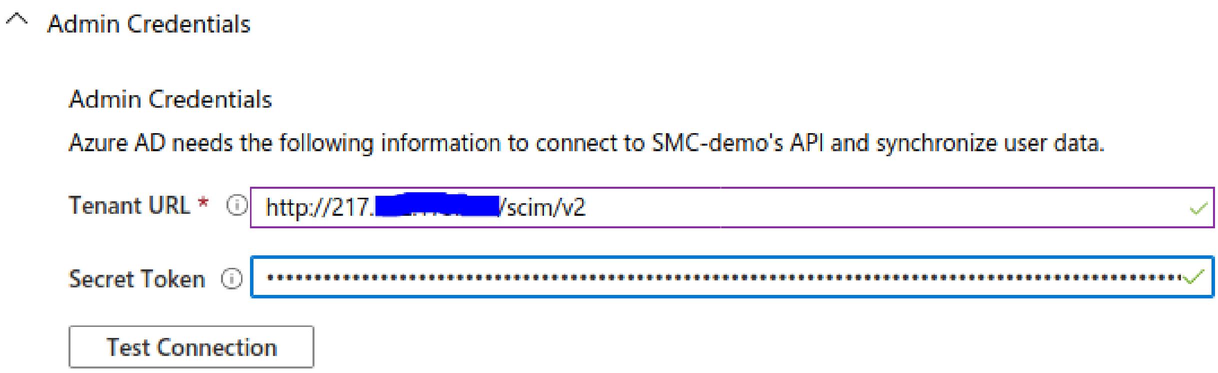


```
dlo@ubuntu:~$ user_name=student
dlo@ubuntu:~$ smc_key=4H2d[redacted]
dlo@ubuntu:~$ host_machine_public_ip=217.[redacted]
dlo@ubuntu:~$ curl -d "productName=smc&userName=$user_name&password=$smc_key" -H "Content-Type:
application/x-www-form-urlencoded" -X POST http://$host_machine_public_ip/scim/v2/token;echo ""

eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2MTg0Nzc2MDAsImlhdCI6MTU4NjkOMTYwMCwiaXNzIjoiRm9
yY2VQb2ludCIsImp0aSI6Im96aXBFdHVicVZ0aW9ZVFh1MEtTR0dKWTNsOVowWGtGTWM1TWowV0RROFVqdDBjTE11RzBsNW1
5dmFBV3lIeXZQV1BnSlZuV28wYlBYbVp4MkFpY3ZlUTdJM0dCRThvR2lRVElnOWdFUVdReDhrR25TTlB4d29ac0FQTGhFQ3N
```

Copy the value of the access token (yellow text in the screenshot) and save it in a secure location: this access token will be used in the configuration of the Azure app for automated provisioning.

4. Login to the Azure portal
5. Search **Azure Active Directory.**
6. Select **Enterprise applications.**
7. Find your Azure app and click on it.
8. Select **Provisioning.**
9. In **Tenant URL** field enter: http:// HOST_MACHINE_PUBLIC_IP/scim/v2.
10. In the **Secret Token** field enter the access token for Forcepoint SCIM service



^ Admin Credentials

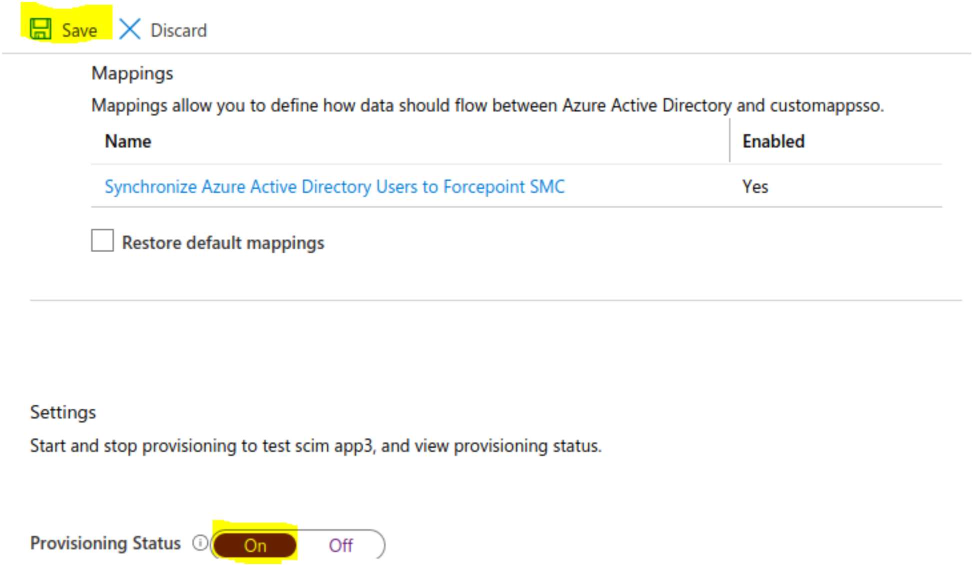
Admin Credentials

Azure AD needs the following information to connect to SMC-demo's API and synchronize user data.

Tenant URL * ⓘ http://217.[redacted]/scim/v2 ✓

Secret Token ⓘ •• ✓

11. In the **Secret Token** field enter the access token for Forcepoint SCIM service.
12. Finally, change the **Provisioning Status** to **On** and click **Save**.



Mappings

Mappings allow you to define how data should flow between Azure Active Directory and customappsso.

Name	Enabled
Synchronize Azure Active Directory Users to Forcepoint SMC	Yes

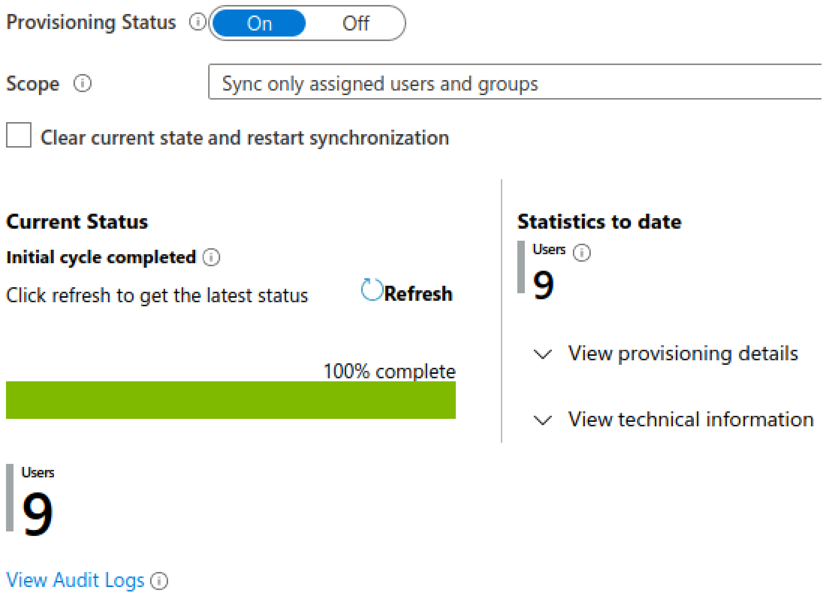
□ Restore default mappings

Settings

Start and stop provisioning to test scim app3, and view provisioning status.

Provisioning Status ⓘ [✓ On] Off

Once you have saved the settings, the provisioning cycle will start. Provisioning cycle interval is 40 minutes. In each provisioning cycle, **Azure will only sync assigned users with your Forcepoint SMC**.



Provisioning Status: On

Scope: Sync only assigned users and groups

□ Clear current state and restart synchronization

Current Status

Initial cycle completed

Click refresh to get the latest status

100% complete

Users 9

Statistics to date

Users 9

Once a provisioning cycle is completed, assigned users can login to Forcepoint SMC with their Azure credentials where login name will be in this format **userFirstName.userLastName**

Step 11: Assign users to Azure App

Your Azure App is configured to sync only assigned users with Forcepoint SMC. To Assign a user to your Azure app, follow these steps:

1. Select your Azure app.
2. Select **Users and Groups**.
3. Click on **Add user**.
4. Click on **Users and groups**.
5. Select the user/users you would like to assign to your Azure app.
6. Click on **Select** > **Assign**

Step 12: Apply SMC Roles to Azure Users

For all newly synced users, their permissions assigned within SMC will be **Viewer**. Different SMC roles can be assigned to Azure users by simply changing the user's group membership.

In the Azure Active Directory, the following groups are automatically created to mirror the SMC administrator permissions:

- **Editor**
- **Logs Viewer**
- **Monitor**
- **Operator**
- **Owner**
- **Reports Manager**
- **Viewer**
- **Superuser**

To add/remove permissions to a user, simply add/remove that user from the corresponding group.

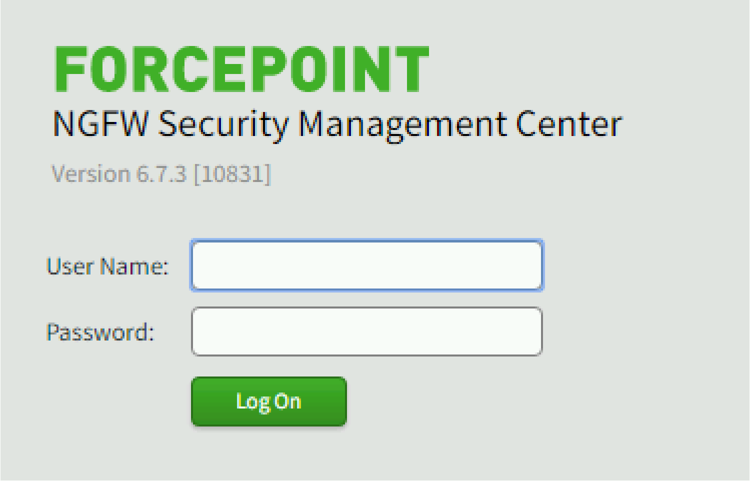
Example: to give **Editor** and **Monitor** permission to user B, simply add user B as a member to the AD group **Editor** and to the AD group **Monitor**.

The members of superuser groups would have full administrator permissions into the SMC.

Step 13: Access on-promise Forcepoint SMC via Azure application

Users assigned to your SMC Azure app can use your Azure to access Forcepoint SMC on-promise with the following steps:

1. Login to https://myapplications.microsoft.com/
2. Find your Azure app and click on it. This will redirect your web browser to Forcepoint SMC on-premise.
3. Enter your Azure credentials in the format **userFirstName.userLastName**.



Troubleshooting

Follow these steps to identify issues impacting the normal operation of the integration described in this document.

Docker Implementation

Validate the prerequisites

Make sure the prerequisites described in the **Summary** chapter are all satisfied:

→ Check the versions of Forcepoint NGFW SMC in use are listed as compatible

Forcepoint NGFW Security Management Center (SMC) version 6.7.3.2 or higher

→ Docker images for this integration have been tested with

Docker 19.03.6

Docker-compose 1.25.4

→ The docker host machine should meet the minimum hardware requirements of at least 2GB of RAM and 20GB of storage

→ User needs sudo permissions in the docker host machine

→ Check the user can download the file with the below command:

wget --content-disposition https://frcpnt.com/ngfw-connect-Azure-ad-docker-latest

Check network connectivity

Make sure firewalls or other security appliances are not impacting the network connectivity necessary for the operation of all components involved into this integration:

→ Check the docker host machine can be accessed via its public ip address or its public DNS name: execute the following command on any machine:

ping -c 2 YOUR_DOCKER_PUBLIC_IP_ADDRESS

Replacing the YOUR_DOCKER_PUBLIC_IP_ADDRESS with the docker public ip. Once done check the result is similar to below:

```
PING YOUR_DOCKER_PUBLIC_IP_ADDRESS.url (10.10.120.12) 56(84) bytes of data.
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=179 ms
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=181 ms
```

→ Check the docker host machine has connectivity to SMC: execute the following command on docker host machine:

ping -c 2 SMC_PRIVATE_IP_ADDRESS

Replacing the SMC_PRIVATE_IP_ADDRESS with your Forcepoint SMC private IP address or the hostname. Once done check the result is similar to below:

```
PING SMC_PRIVATE_IP_ADDRESS.url (10.10.120.12) 56(84) bytes of data.
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=179 ms
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=181 ms
```

Check dependencies are installed

Make sure the software dependencies needed by the components involved into this integration are installed:

→ Check all dependencies are installed: execute the following command on docker host machine to check docker-compose is installed:

```
docker-compose --version
```

Check the output presents a version of 1.25.4 or higher (example below):

```
docker-compose version 1.25.4, build 8d51620a
```

→ Check the host machine has docker installed: Execute the following command on the host machine:

```
docker info
```

Check the first few lines of the output are similar to below:

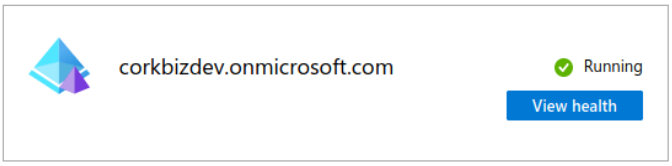
```
Client:
Debug Mode: false

Server:
Containers: 3
  Running: 2
  Paused: 0
  Stopped: 1
Images: 3
Server Version: 19.03.8
```

Check all components are configured and running properly

Make sure the products and services involved into this integration are configured as expected and they are running:

→ Check the domain service is successfully running in Azure

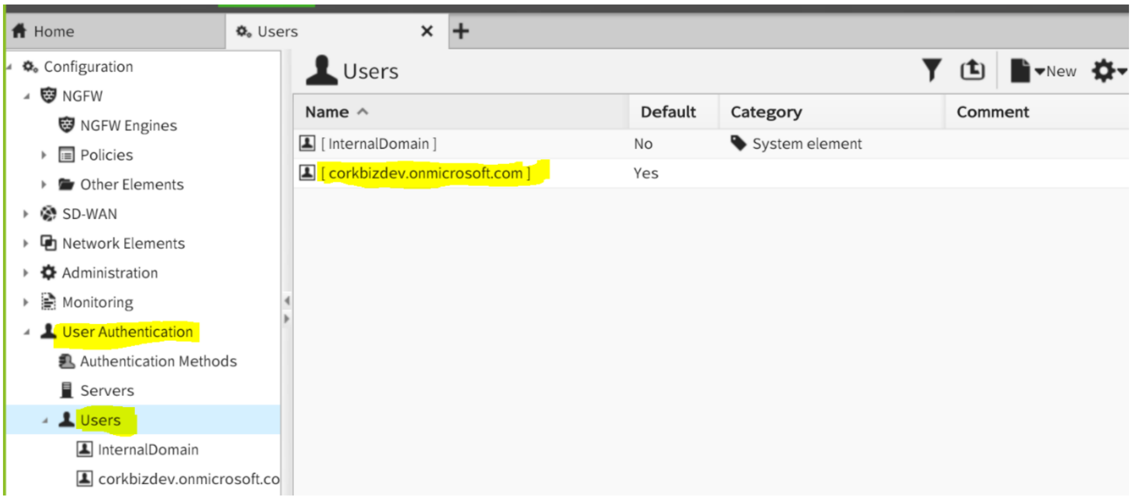


corkbizdev.onmicrosoft.com

✓ Running

→ After Azure user provisioning between SMC and Azure has finished and the user still cannot login to SMC with Azure credentials, please check the following cases:

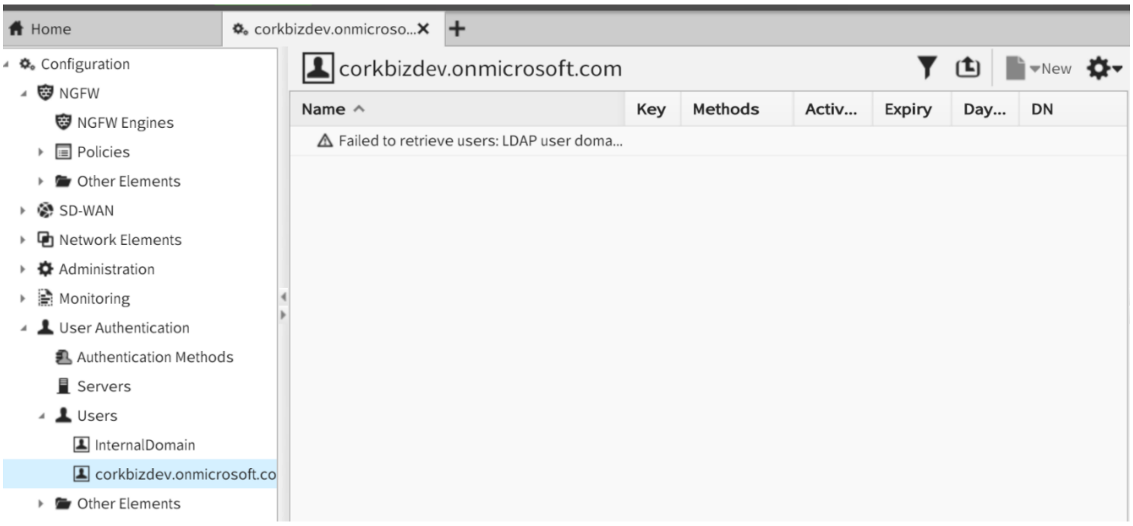
Ensure your Azure admin password is correct in SMC. Login with the local admin account into SMC and go to User Authentication and Click on Azure domain name.



Users

Name	Default	Category	Comment
[InternalDomain]	No	System element	
[corkbizdev.onmicrosoft.com]	Yes		

Double-click on the domain name and if you get the result as shown in the following picture



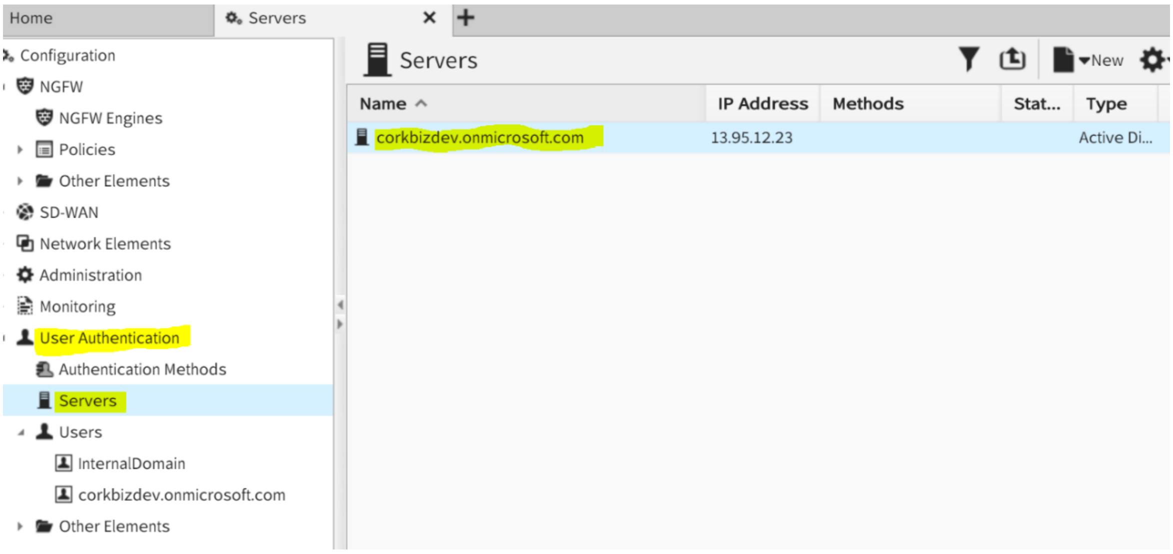
corkbizdev.onmicrosoft.com

Name	Key	Methods	Activ...	Expiry	Day...	DN
⚠ Failed to retrieve users: LDAP user doma...						

that could result from 2 cases:

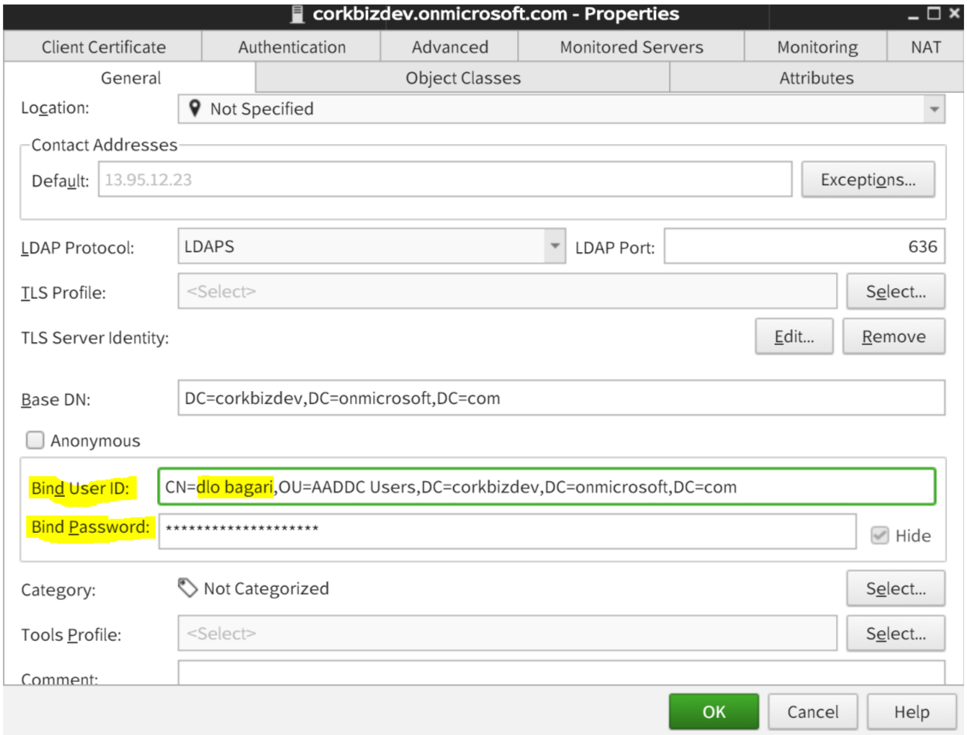
- → Azure admin password in SMC or the CN(common name) value in SMC external active directory can be wrong
- → Azure Active Directory domain service LDAP does not accept traffic from SMC

1. For the first case (Azure admin password in SMC can be wrong), go to User Authentication in SMC, click on Servers, double-click on corkbizdev.onmicrosoft.com



Name	IP Address	Methods	Stat...	Type
corkbizdev.onmicrosoft.com	13.95.12.23			Active Di...

Check that the CN value is correct in Bind User ID and enter your Azure admin password in Bind Password.



Location: Not Specified

Contact Addresses

Default: 13.95.12.23

LDAP Protocol: LDAPS

LDAP Port: 636

TLS Profile: <Select>

TLS Server Identity:

Base DN: DC=corkbizdev,DC=onmicrosoft,DC=com

□ Anonymous

Bind User ID: CN=dlo bagari,OU=AADDC Users,DC=corkbizdev,DC=onmicrosoft,DC=com

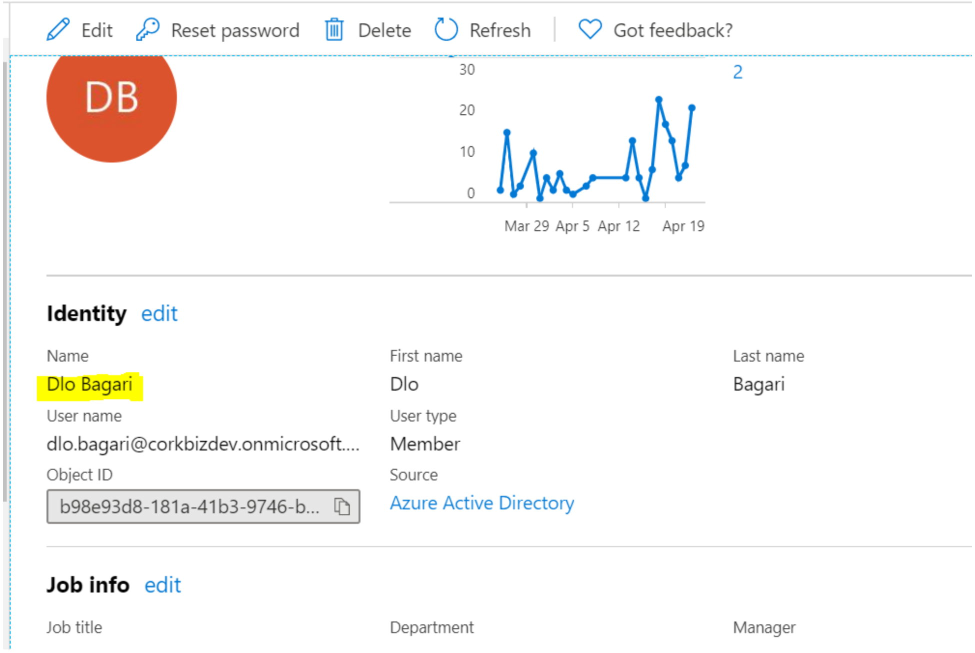
Bind Password: ******************** ☑ Hide

Category: Not Categorized

Tools Profile: <Select>

Comment:

The CN value is the Name of the admin user Azure Active Directory as shown below:



Identity edit

Name	First name	Last name
Dlo Bagari	Dlo	Bagari
User name	User type	
dlo.bagari@corkbizdev.onmicrosoft....	Member	
Object ID	Source	
b98e93d8-181a-41b3-9746-b...	Azure Active Directory	

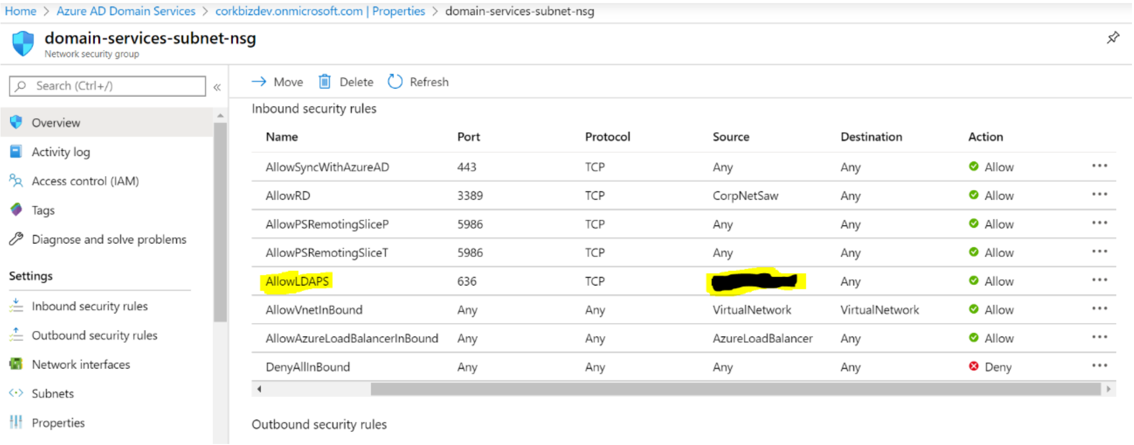
Job info edit

Job title　Department　Manager

Click Ok button after entering correct value in CN field of Bind User ID and Bind Password.

2. For the case 2, go to Azure -> Home ->Azure AD Domain Services -> corkbizdev.onmicrosoft.com -> Properties -> domain-services-subnet-nsg

   Make sure the source ip address for AllowLDAPs is your docker host machine’s ip address. If ip address is not correct, double-click on AllowLDAPs and enter the correct ip address.



Inbound security rules

Name	Port	Protocol	Source	Destination	Action
AllowSyncWithAzureAD	443	TCP	Any	Any	✓ Allow
AllowRD	3389	TCP	CorpNetSaw	Any	✓ Allow
AllowPSRemotingSliceP	5986	TCP	Any	Any	✓ Allow
AllowPSRemotingSliceT	5986	TCP	Any	Any	✓ Allow
AllowLDAPS	636	TCP	[redacted]	Any	✓ Allow
AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	✓ Allow
DenyAllInBound	Any	Any	Any	Any	✗ Deny

Outbound security rules

Traditional Implementation

Validate the prerequisites

Make sure the prerequisites described in the **Summary** chapter are all satisfied:

→ Check the versions of Forcepoint NGFW SMC in use are listed as compatible

Forcepoint NGFW Security Management Center (SMC) version 6.7.3.2 or higher

→ Verify the integration is operating correctly on a CentOS 7.3 machine with at least 2GB of RAM and 20GB of storage

→ User needs sudo permissions for installing necessary certificates and keys

→ Check the user can download the file with the below command:

wget --content-disposition http://frcpnt.com/ngfw-connect-Azure-ad-latest

Check network connectivity

Make sure firewalls or other security appliances are not impacting the network connectivity necessary for the operation of all components involved into this integration:

→ Check the host machine can be accessed via its public ip address or its public DNS name: execute the following command on any machine:

ping -c 2 YOUR_HOST_MACHINE_PUBLIC_IP_ADDRESS

Replacing the YOUR_DOCKER_PUBLIC_IP_ADDRESS with the docker public ip. Once done check the result is similar to below:

```
PING YOUR_ HOST_MACHINE_PUBLIC_IP_ADDRESS.url (10.10.120.12) 56(84) bytes of data.
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=179 ms
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=181 ms
```

→ Check the Centos 7.3 host machine can reach the Forcepoint SMC host machine over the network: execute the following command on host machine:

ping -c 2 SMC_PRIVATE_IP_ADDRESS

Replacing the SMC_PRIVATE_IP_ADDRESS with your Forcepoint SMC private Ip address or the host-name. Once done check the result is similar to below:

```
PING SMC_PRIVATE_IP_ADDRESS.url (10.10.120.12) 56(84) bytes of data.
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=179 ms
64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=181 ms
```

Check dependencies are installed

Make sure the software dependencies needed by the components involved into this integration are installed:

→ Check all dependencies are installed: execute the following command on host machine to check go is installed:

go version

Check the output is similar to below:

```
go version go1.14.1 linux/amd64
```

→ Check Azure CLI is installed: Execute following command on host machine:

az version

Check the output is similar to below:

```
{
  "Azure-cli": "2.3.1",
  "Azure-cli-command-modules-nspkg": "2.0.3",
  "Azure-cli-core": "2.3.1",
  "Azure-cli-nspkg": "3.0.4",
  "Azure-cli-telemetry": "1.0.4",
  "extensions": {}
```

→ Check openssl is installed: Execute following command on host machine:

```
openssl version
```

Check the output is similar to below:

```
OpenSSL 1.0.2k-fips  26 Jan 2017
```

→ Check openssl is installed: Execute following command on host machine:

```
openssl version
```

Check the output is similar to below:

```
OpenSSL 1.0.2k-fips  26 Jan 2017
```

→ Check nginx service is installed and running properly: Execute following command on host machine:

```
systemctl status nginx
```

Check the output is similar to below:

```
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2020-04-21 15:55:18 UTC; 8min ago
  Process: 998 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
  Process: 983 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
  Process: 976 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
 Main PID: 1005 (nginx)
   CGroup: /system.slice/nginx.service
           ├─1005 nginx: master process /usr/sbin/nginx
           └─1006 nginx: worker process
```

→ Check python3.6 is installed: Execute following command on host machine:

```
python3 --version
```

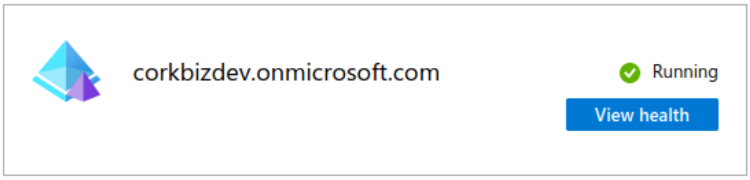
Check the output is similar to below:

```
Python 3.6.8
```

Check all components are configured and running properly

Make sure the products and services involved into this integration are configured as expected and they are running:

→ Check the domain service is successfully running in Azure

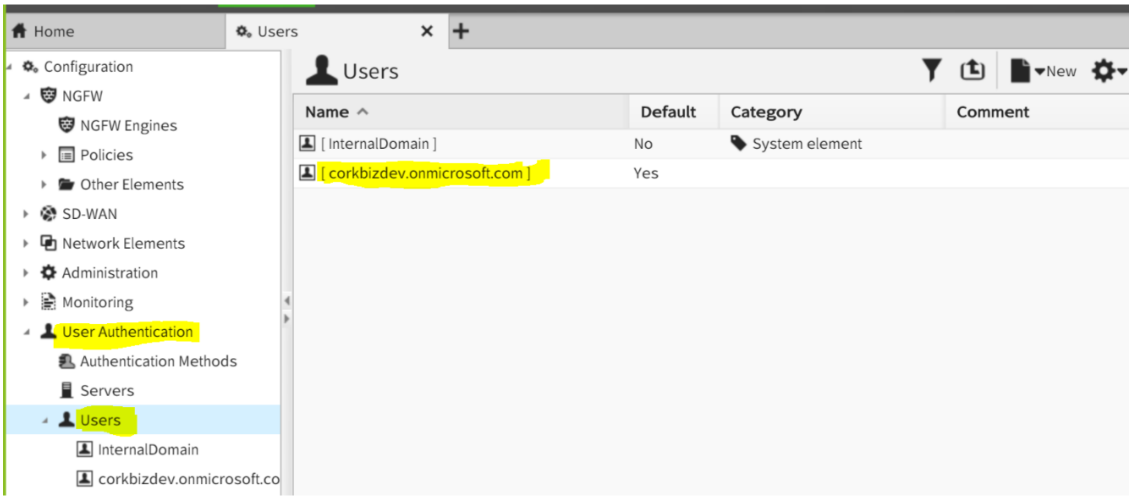


corkbizdev.onmicrosoft.com

Running

→ After Azure user provisioning between SMC and Azure has finished and the user still cannot login to SMC with Azure credentials, please check the following cases:

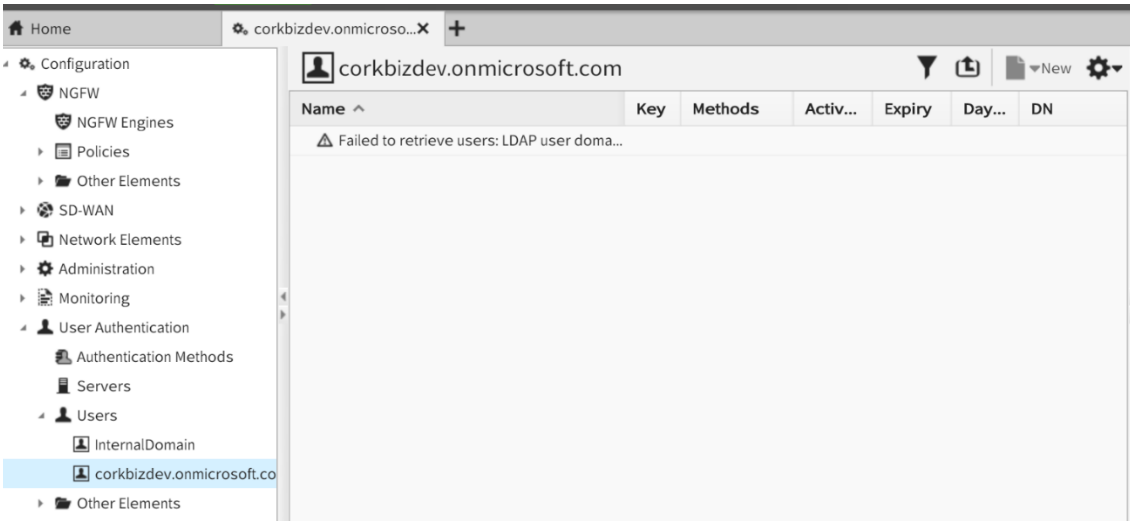
Ensure your Azure admin password is correct in SMC. Login with the local admin account into SMC and go to User Authentication and Click on Azure domain name.



Users

Name	Default	Category	Comment
[InternalDomain]	No	System element	
[corkbizdev.onmicrosoft.com]	Yes		

Double-click on the domain name and if you get the result as shown in the following picture



corkbizdev.onmicrosoft.com

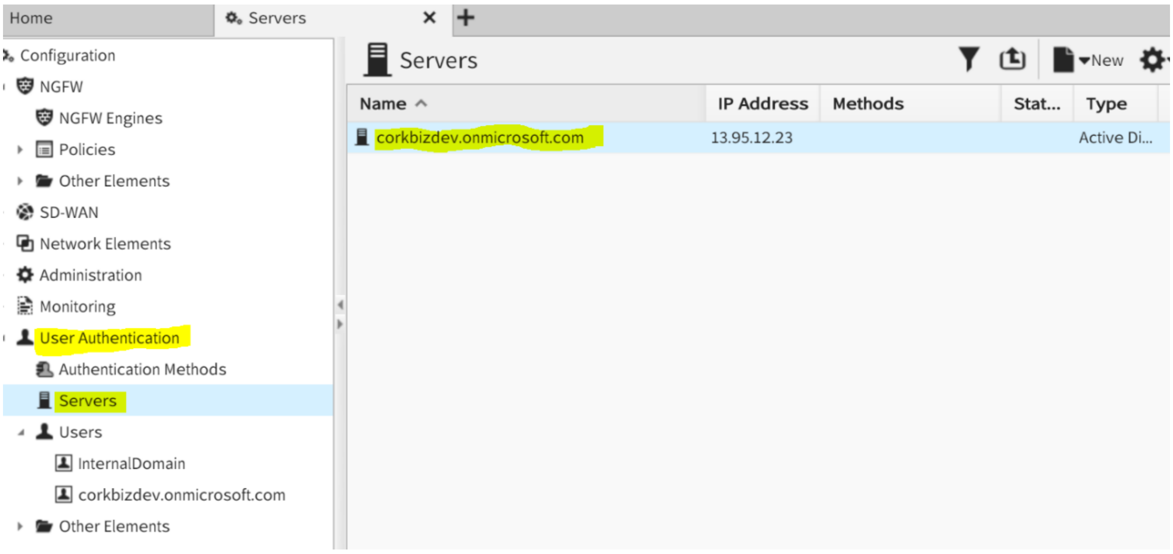
Name	Key	Methods	Activ...	Expiry	Day...	DN
⚠ Failed to retrieve users: LDAP user doma...						

that could result from 2 cases:

→ Azure admin password in SMC or the CN(common name) value in SMC external active directory can be wrong

→ Azure Active Directory domain service LDAP does not accept traffic from SMC

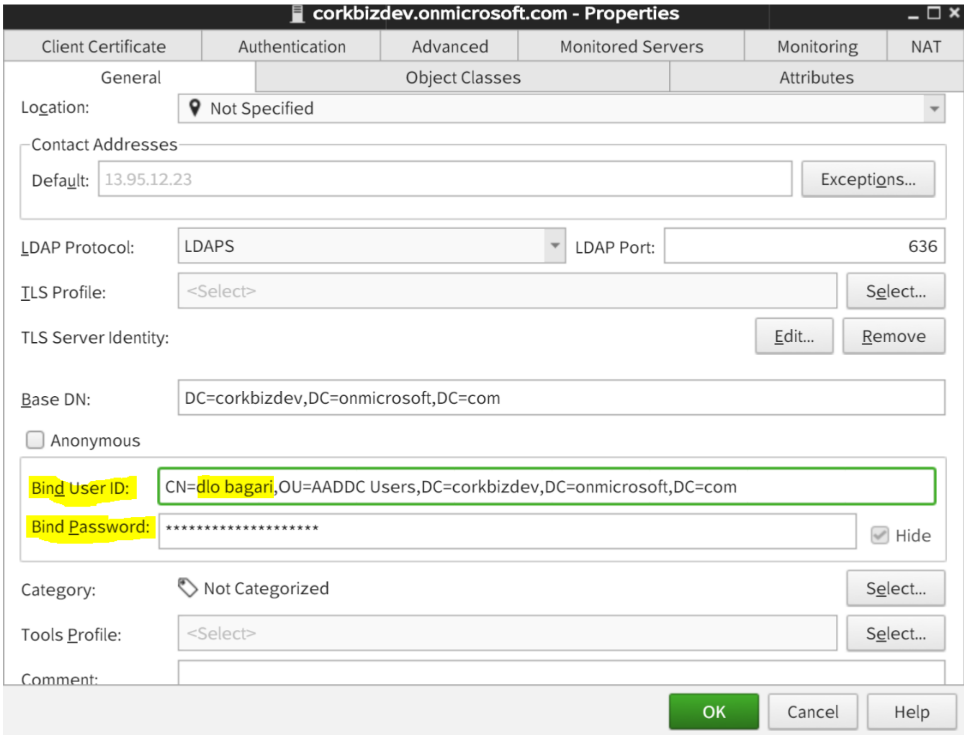
1. For the first case (Azure admin password in SMC can be wrong), go to User Authentication in SMC, click on Servers, double-click on corkbizdev.onmicrosoft.com



Servers

Name	IP Address	Methods	Stat...	Type
corkbizdev.onmicrosoft.com	13.95.12.23			Active Di...

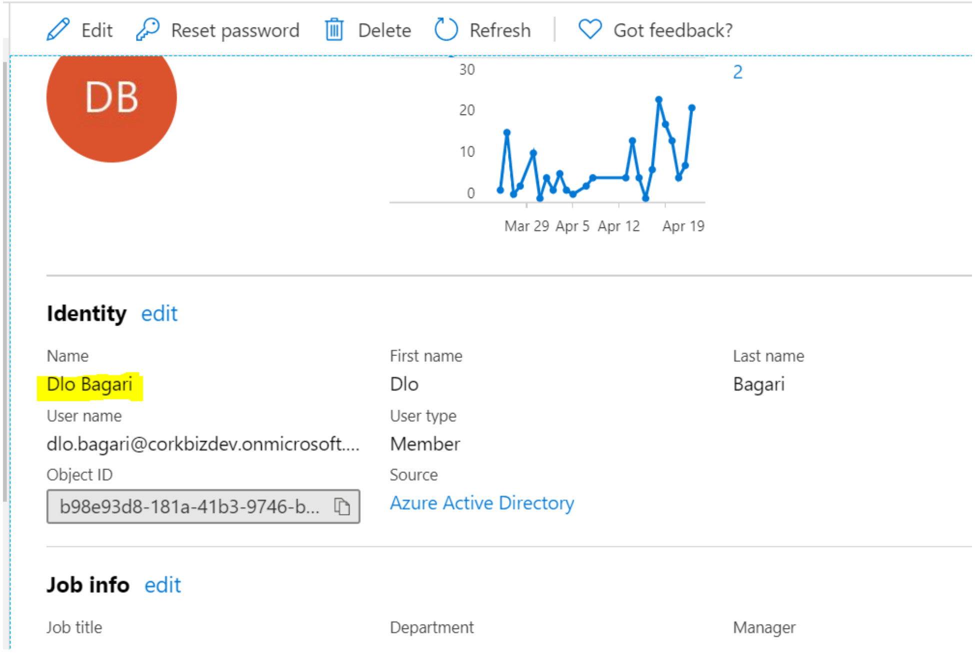
Check that the CN value is correct in Bind User ID and enter your Azure admin password in Bind Password.



corkbizdev.onmicrosoft.com - Properties

Location:	Not Specified
Contact Addresses — Default:	13.95.12.23
LDAP Protocol:	LDAPS
LDAP Port:	636
TLS Profile:	<Select>
TLS Server Identity:	
Base DN:	DC=corkbizdev,DC=onmicrosoft,DC=com
□ Anonymous	
Bind User ID:	CN=dlo bagari,OU=AADDC Users,DC=corkbizdev,DC=onmicrosoft,DC=com
Bind Password:	******************** ☑ Hide
Category:	Not Categorized
Tools Profile:	<Select>
Comment:	

The CN value is the Name of the admin user Azure Active Directory as shown below:



Identity edit

Name	First name	Last name
Dlo Bagari	Dlo	Bagari
User name	User type	
dlo.bagari@corkbizdev.onmicrosoft....	Member	
Object ID	Source	
b98e93d8-181a-41b3-9746-b...	Azure Active Directory	

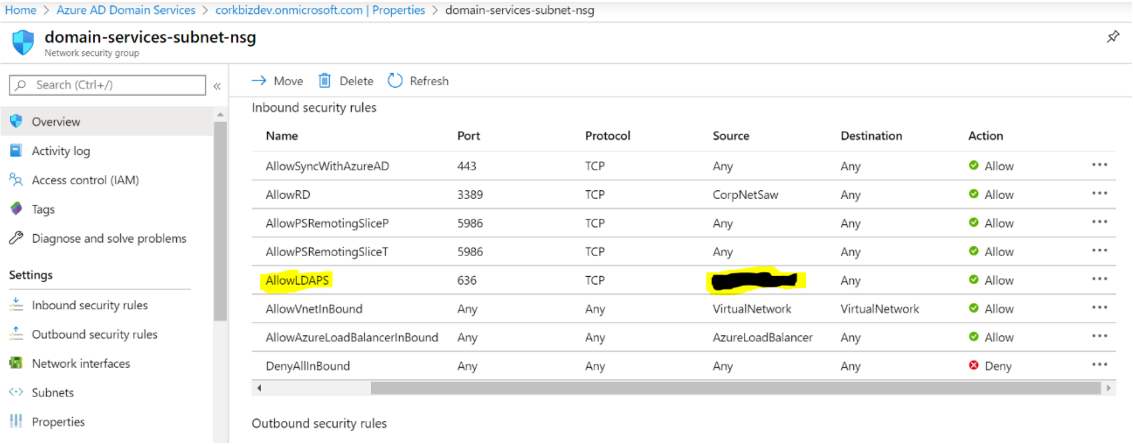
Job info edit

Job title	Department	Manager

Click Ok button after entering correct value in CN field of Bind User ID and Bind Password.

2. For the case 2, go to Azure -> Home ->Azure AD Domain Services -> corkbizdev.onmicrosoft.com -> Properties -> domain-services-subnet-nsg

   Make sure the source ip address for AllowLDAPs is your host machine's ip address. If ip address is not correct, double-click on AllowLDAPs and enter the correct ip address.



Inbound security rules

Name	Port	Protocol	Source	Destination	Action
AllowSyncWithAzureAD	443	TCP	Any	Any	Allow
AllowRD	3389	TCP	CorpNetSaw	Any	Allow
AllowPSRemotingSliceP	5986	TCP	Any	Any	Allow
AllowPSRemotingSliceT	5986	TCP	Any	Any	Allow
AllowLDAPS	636	TCP	[redacted]	Any	Allow
AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
DenyAllInBound	Any	Any	Any	Any	Deny

Outbound security rules

forcepoint.com/contact

About Forcepoint

Forcepoint is the global human-centric cybersecurity company transforming the digital enterprise by continuously adapting security response to the dynamic risk posed by individual users and machines. The Forcepoint human point system delivers risk-adaptive protection to continuously ensure trusted use of data and systems. Based in Austin, Texas, Forcepoint protects the human point for thousands of enterprise and government customers in more than 150 countries.