Forcepoint DLP and Azure Active Directory secure hybrid access

Integration Guide

Forcepoint

Integration Guide

Michael Nevin
14 September 2020
Public

Table of Contents

Summary	**2**
Implementation – Docker	**3**
Implementation - Traditional	**9**
Forcepoint DLP – Configure Azure AD as external identity source	**15**
Forcepoint DLP – Configure Azure Application proxy	**19**
Troubleshooting	**27**
Appendix – Mapping an IP address to an FQDN	**30**

Version	Date	Author	Notes
0.1	15 April 2020	Michael Nevin	First draft
0.2	05 May 2020	Neelima Rai	Updated and added Troubleshooting chapter
0.3	05 May 2020	Mattia Maggioli	Review
0.4	19 June 2020	Jonathan Knepher	Review
0.5	14 September 2020	Mattia Maggioli	Minor updates

Summary

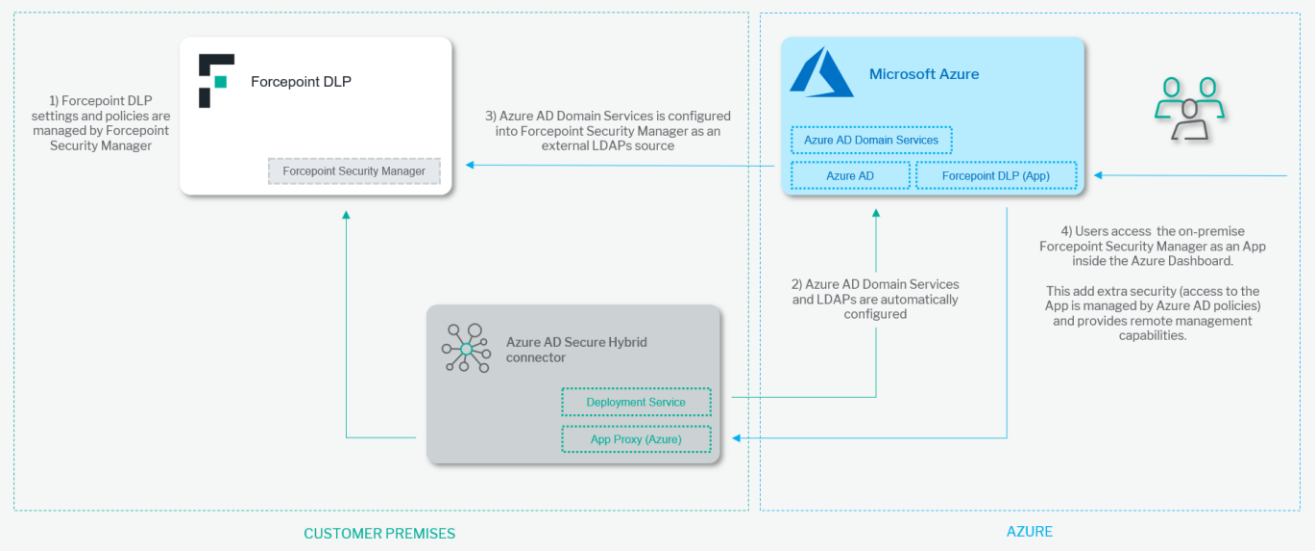
This guide provides step by step instructions to set up an integration between **Azure Active Directory secure hybrid access** and **Forcepoint DLP**.

The integration enables access and authentication to Forcepoint Security Manager with selected Azure AD users and to expose the Forcepoint Security Manager as an Azure app for remote management.

The code and instructions provided, enable system administrators to

- → Automatically deploy Azure AD Domain Services with LDAPs
- → Configure Azure AD as an external LDAPs source into Forcepoint Security Manager
- → Deploy and configure the App Proxy component of Azure to expose Forcepoint Security Manager as an Azure App

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



Caveats

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

- → Forcepoint DLP version 8.7.0.360
- → Azure Active Directory

This interoperability uses:

- → **Deployment Service**: a service that deploys Azure AD Domain Services template
- → **Azure AD Domain Services**: an extension to Azure Active Directory to enable LDAPs connectivity to Azure AD
- → **Azure App Proxy**: a component provided by Microsoft Azure to expose on-premises web applications

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

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

- → Centos 7.3

In this document we assume only Azure Active Directory is already in use. If **Azure AD Domain Services** with LDAPs is in use as well, then skip the **Implementation** chapter (either **Traditional** or **Docker**) and start directly from the **Forcepoint DLP – Configure Azure AD as external identity source** chapter.

Implementation – Docker

The solution described in this chapter requires a Linux machine (Centos 7.3 recommended) within the same network of Forcepoint Security Manager host machine. 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**: 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-dlp-azure-ad-ds-deployment-docker.tar.gz** available at the link https://frcpnt.com/dlp-deploy-azure-ad-docker-latest

The archive **fp-dlp-azure-ad-ds-deployment-docker.tar.gz** contains the following files:

- → **docker-compose-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 Security Manager.
- → **.env**: the environment variables files for docker-compose.

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-dlp-azure-ad-ds-deployment-docker.tar.gz** and change your directory to **fp-dlp-azure-ad-ds-deployment-docker**

```
tar -zxvf  fp-dlp-azure-ad-ds-deployment-docker.tar.gz
cd  fp-dlp-azure-ad-ds-deployment-docker
```

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

```
vi .env
```

Update the following variables:

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

2. Generate the PFX base64 certificate:

```
docker-compose 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 down
```

5. Insert the copied Base64 string as a value for PFX_CERTIFICATE_BASE64 variable in .env files variable. For example,
PFX_CERTIFICATE_BASE64=MIIQRQIBAzCCD9cGDSqGSIb3DUEHSAAaCCD8gEgg/EMIIPwDCCBf……

Step 4: Deploy Azure AD DS template

1. Run the deployment container:

```
docker-compose up -d
```

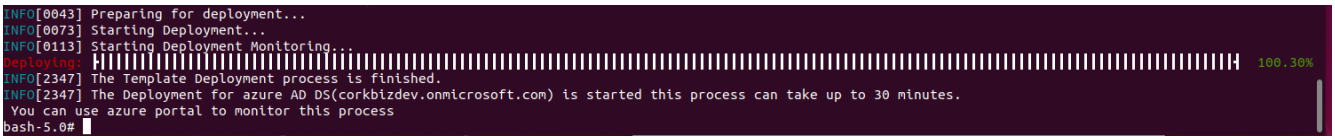
1. Interact with deployment container:

```
docker-compose exec deployment /bin/bash
```

2. Execute the following command to deploy the Azure AD DS, the application provisioning template and to create Azure groups for Forcepoint Security Manager roles:

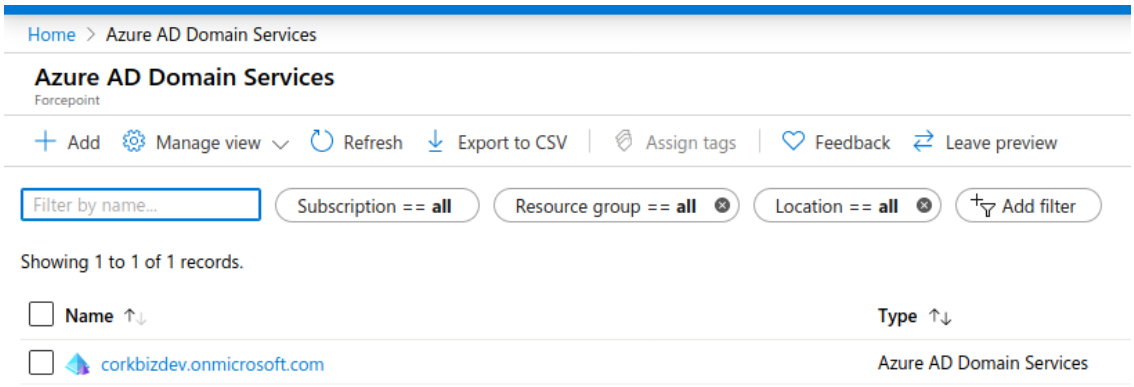
```
./deployment deploy-azure
```

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

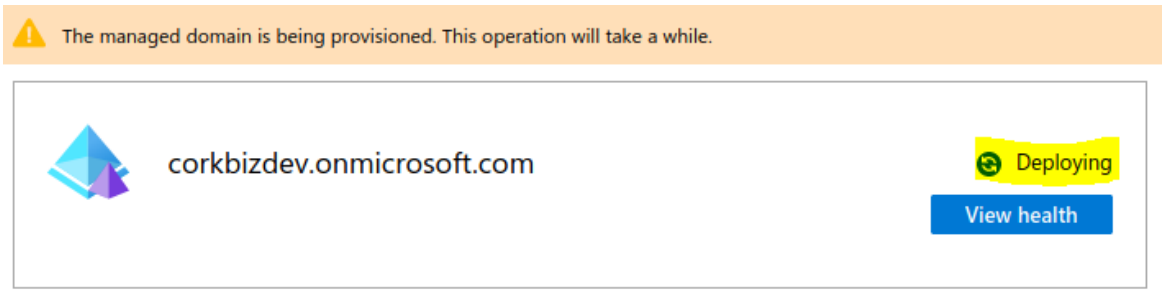


Once finished, Azure will start configuring Azure AD DS and this deployment will take up to 30 minutes and can only be monitored through Azure Portal.

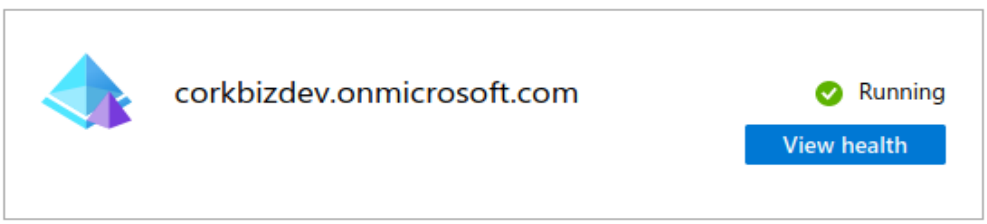
4. To monitor the ongoing deployment login to the Azure portal, search for **Azure AD Domain Services**, click on your **Azure AD Domain Services**



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



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 5**.

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"
```

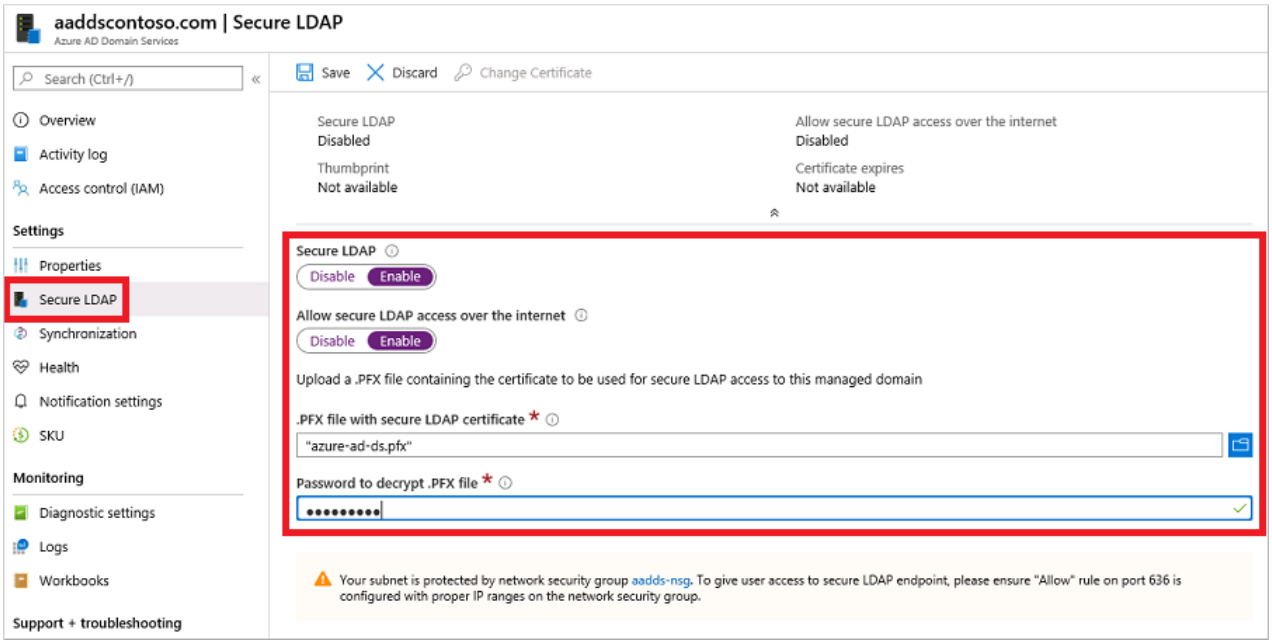
4. Create a PFX certificate. Execute this command after replacing PASSWORD with a password for FPX certificate, and store the password in a secure location as it will be used again in the next steps:

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

This will generate a PFX certificate named a**zure_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 a**zure_cert.pfx** file, then select the certificate a**zure_cert.pfx** .
7. Enter the password to decrypt .PFX file: this is the password that is used when a**zure_cert.pfx** is created.
8. Select **Save** to enable secure LDAP.

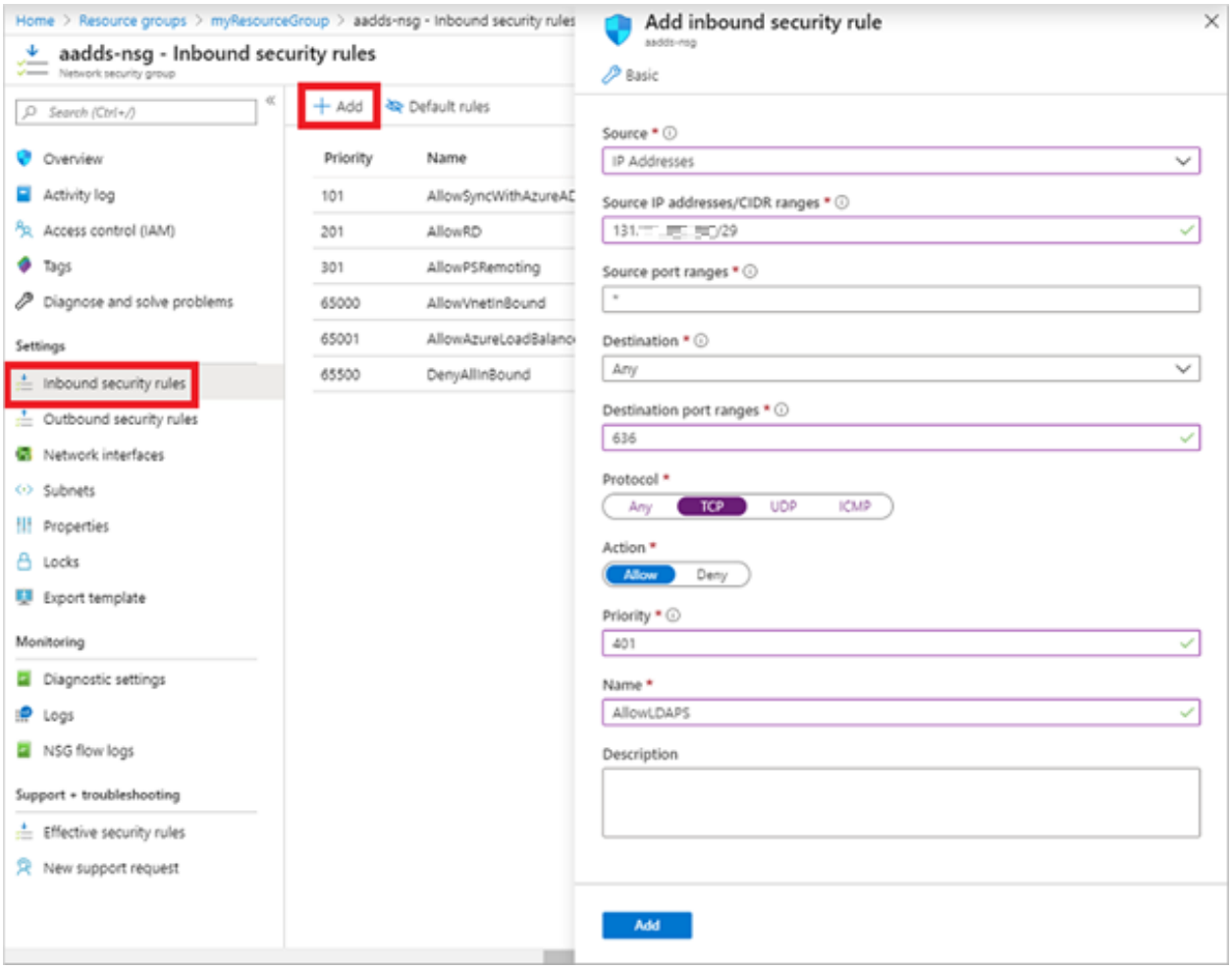


A notification is displayed that secure LDAP is being configured for the managed domain. You can't modify other settings for the managed domain until this operation is complete.

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 premises.



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

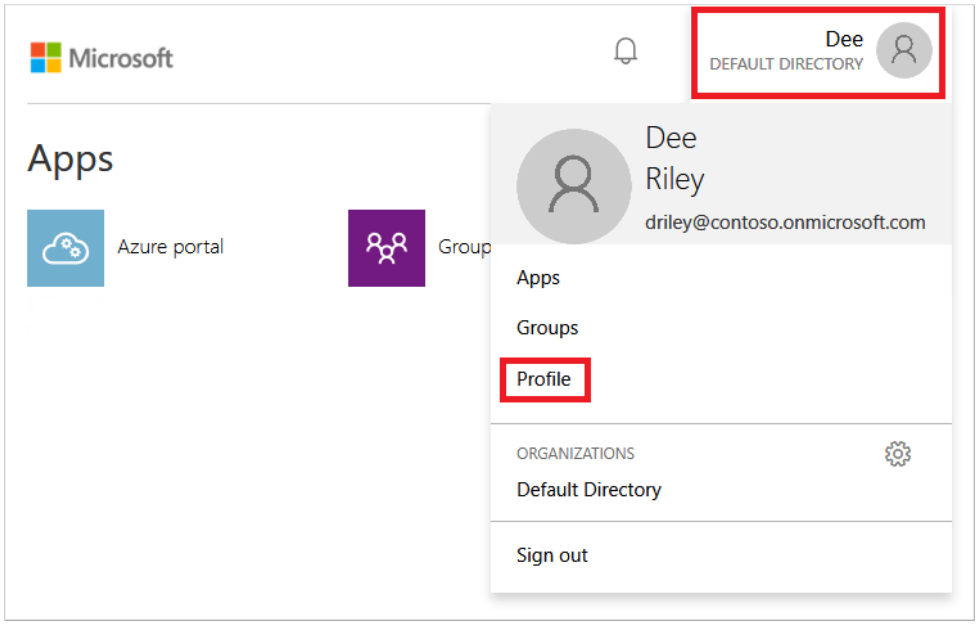
Step 6: Enable Azure AD Domain Services password hash synchronization

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 Forcepoint Security Manager authentication must have their password changed before authentication in Forcepoint Security Manager will work.

The password change process will store password hashes inside Azure AD Domain Services so that users authenticating through LDAPs from Forcepoint Security Manager 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 Forcepoint Security Manager 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 click on 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 (**Forcepoint DLP – Configure Azure AD as external identity source**).

Implementation - Traditional

The solution described in this chapter requires

→ A Centos 7.3 machine able to reach the Forcepoint Security Manager host machine over the network. This 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-dlp-azure-ad-ds-deployment.tar.gz** available at the link http://frcpnt.com/dlp-deploy-azure-ad-latest

The archive **fp-dlp-azure-ad-ds-deployment.tar.gz** contains the following files and folders:

→ **deployment**: the deployment application for deploying templates to Azure, creating external Active Directory authentication server and external users' domain in Forcepoint Security Manager.

→ **deployment.yml**: the configuration file for the deployment application.

→ **installation_script.sh**: a bash script to install all required dependencies.

Step 1: Modify configuration files
Inside the **host-machine** unpack the **fp-dlp-azure-ad-ds-deployment.tar.gz** archive and change your directory to **fp-dlp-azure-ad-ds-deployment**

```
tar -zxvf  fp-dlp-azure-ad-ds-deployment.tar.gz
```

```
cd fp-dlp-azure-ad-ds-deployment
```

Modify deployment.yml file

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

```
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_HOST_MACHINE_PUBLIC_IP_ADDRESS
LOGGER_JSON_FORMAT: false
RESOURCE_GROUP: INSERT_AZURE_RESOURCE_GROUP_NAME_HERE
DEPLOYMENT_TEMPLATE: /root/fp-dlp-azure-ad-ds-deployment/azure_smc_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_YOUR_AZURE_ADMINISTRATOR_LOGIN_NAME_HERE** with your Azure Administrator login name, this administrator most have **Global administrator** role.
2. Replace **INSERT_YOUR_AZURE_DOMAIN_NAME_HERE** with your Azure Domain Name
3. Replace **INSERT_AZURE_LOCATION_HERE** with an Azure region. All Azure resources will be created in this location
4. Replace **INSERT_YOUR_HOST_MACHINE_PUBLIC_IP_ADDRESS** with the public address of the **host-machine**.
5. 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.
6. Replace **INSERT_NUMBER_OF_DAYS_FOR_PFX_CERTIFICATE_EXPIRATION_HERE** with the number of days for PFX certificate to be expired
7. 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 install the following packages:

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

Execute **installation_script.sh**

```
sudo ./installation_script.sh
```

The installation takes about 30 minutes.

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-dlp-azure-ad-ds-deployment** directory.
2. Run the following command which will generate a Base64 string of PFX certificate.

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

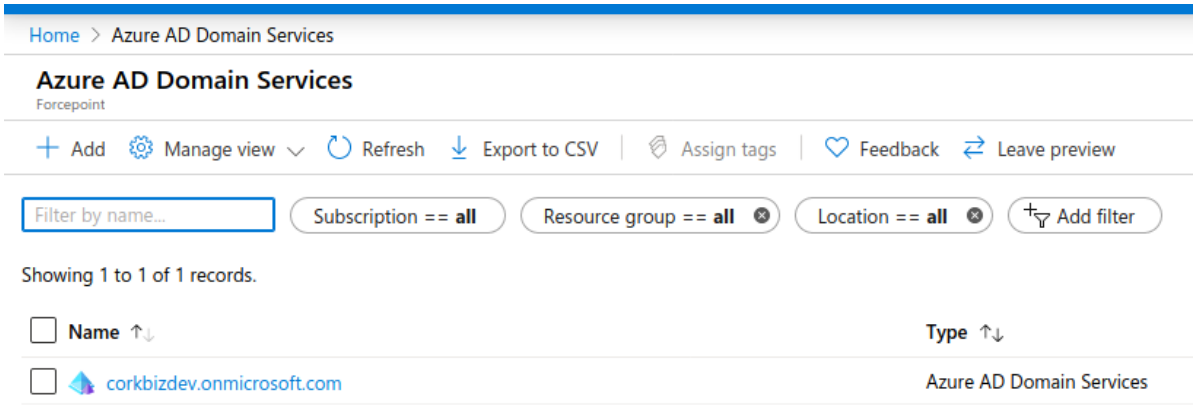
3. Copy the output of the above command and use it to replace the placeholder `PFX_BASE64_WILL_BE_INSERTED_HERE` in the **deployment.yml** file

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**.

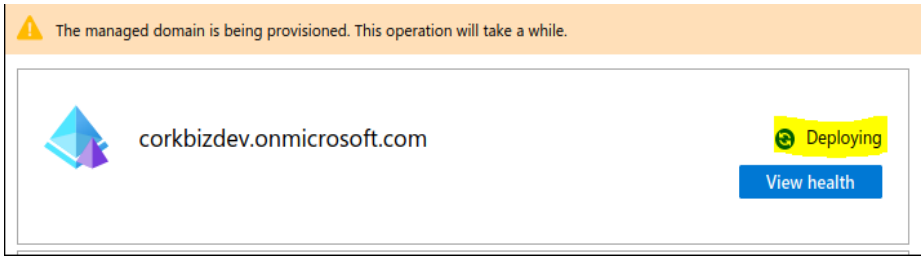
1. Execute the following command to deploy the Azure AD DS:

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

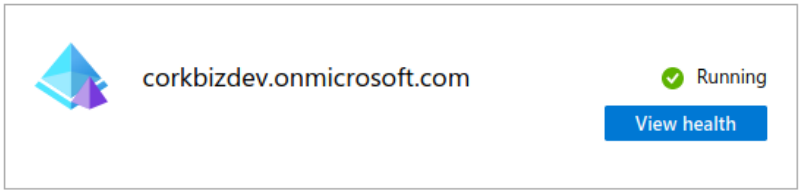
2. Enter your password for the administrator login name displayed on screen:
3. The deployment monitoring progress will start, wait until the progress bar is completed: this can take up to 55 minutes.
4. Once the above template deployment has finished, Azure will start deploying Azure AD DS and this 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



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



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



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"
```

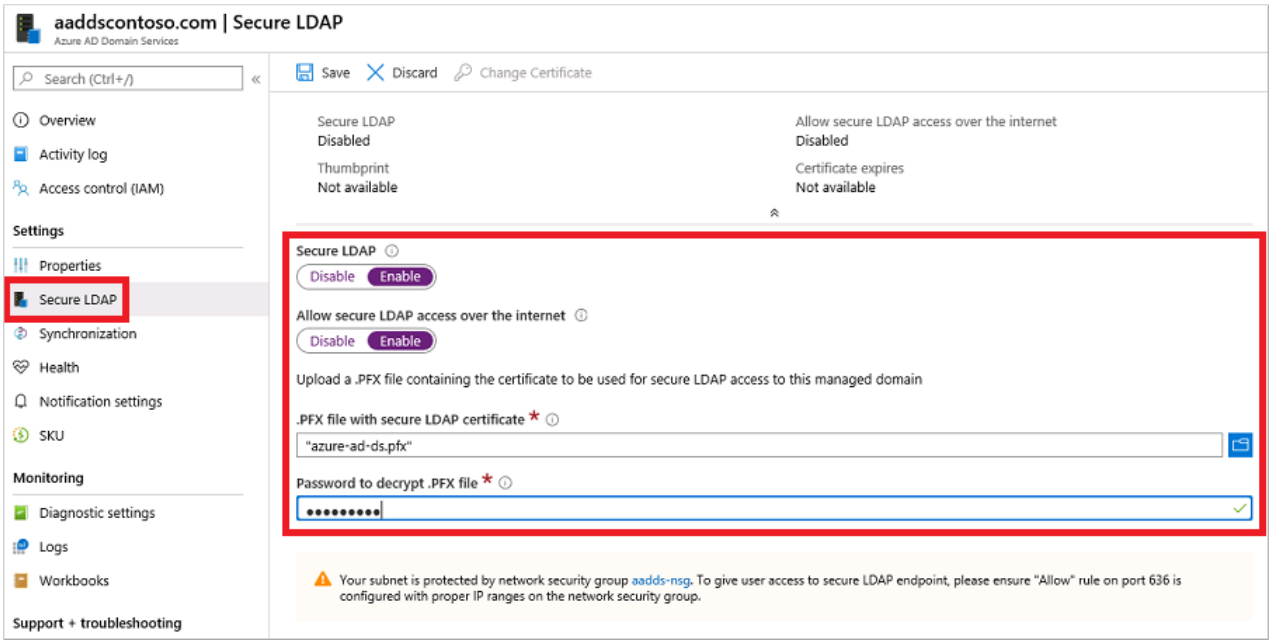
4. Create a PFX certificate. Execute this command after replacing PASSWORD with a password for FPX certificate, and store the password in a secure location as it will be used again in the next steps

```
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 a**zure_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.

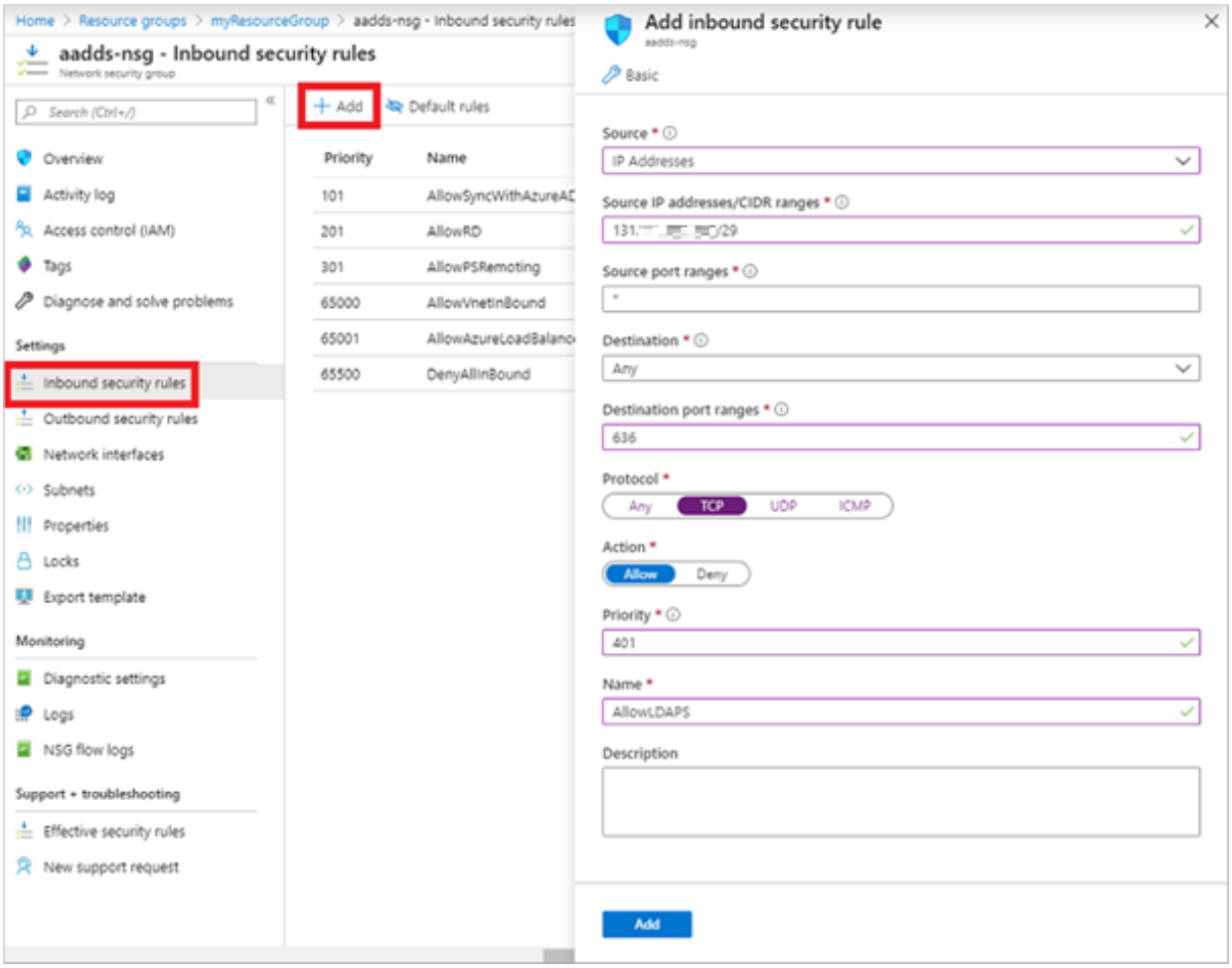


A notification is displayed that secure LDAP is being configured for the managed domain. You can't modify other settings for the managed domain until this operation is complete.

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 premises.



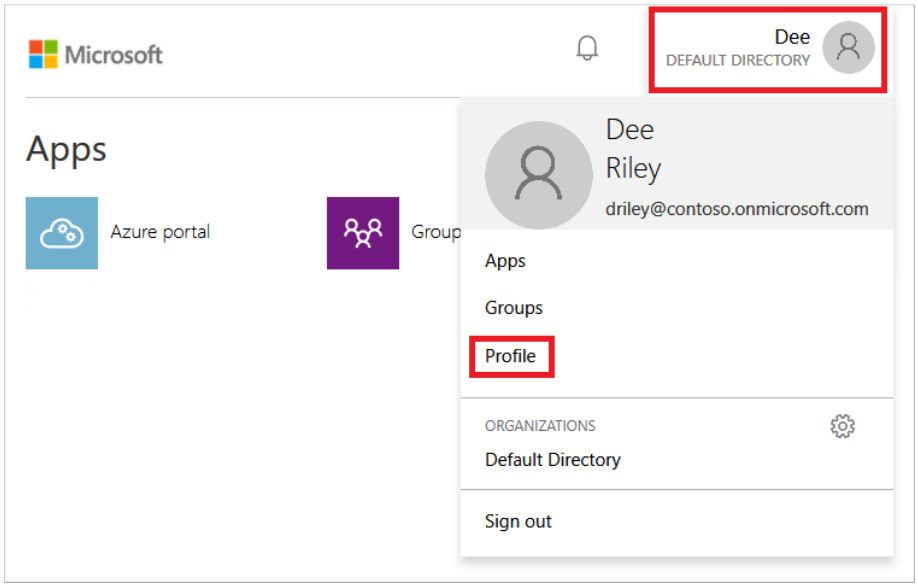
4. Click **Add** to save and apply the rule

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 Forcepoint Security Manager authentication must have their password changed before authentication in Forcepoint Security Manager will work**.

The password change process will store password hashes inside Azure AD Domain Services so that users authenticating through LDAPs from Forcepoint Security Manager 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 Forcepoint Security Manager 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.



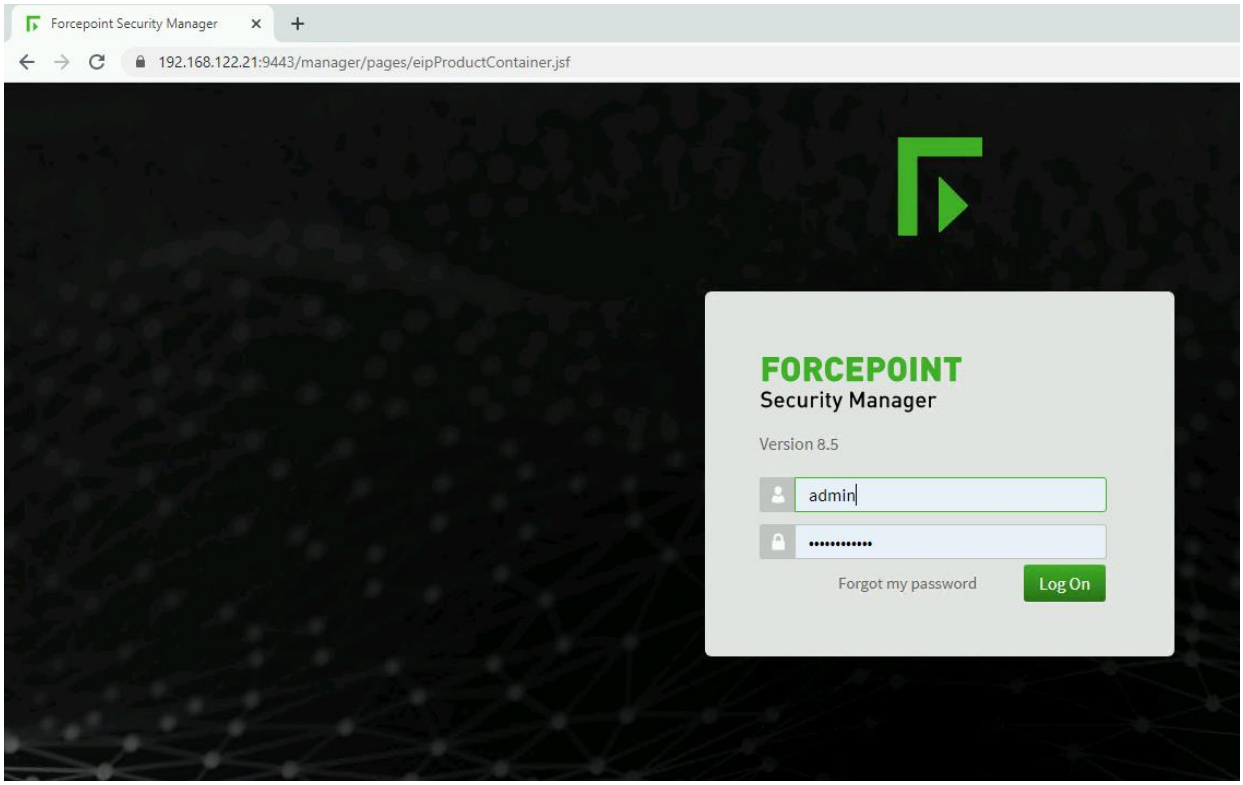
3. On the Profile page, select Change password.
4. On the Change password page, enter your existing (old) password, then enter and confirm a new password.
5. Select Submit.

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.

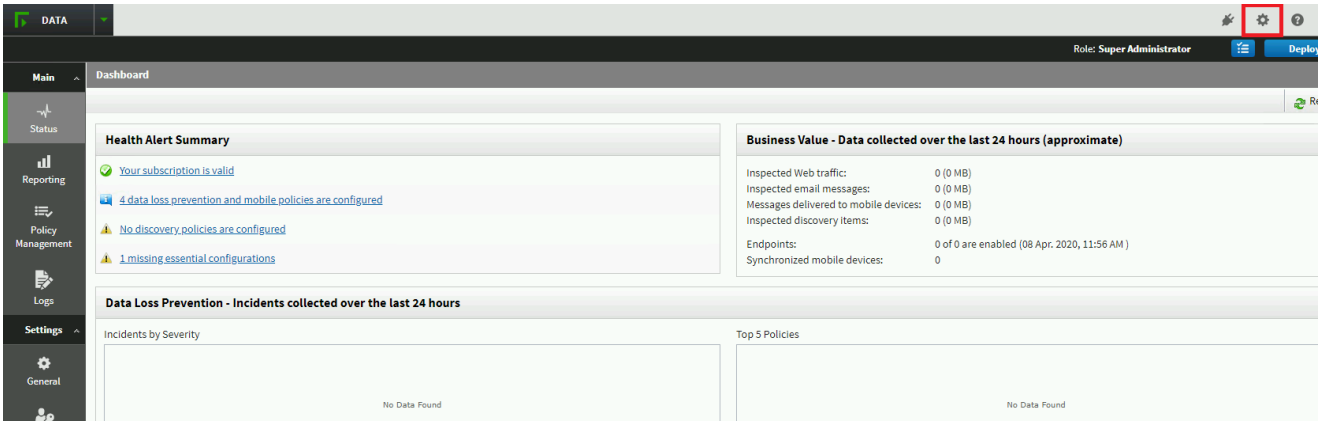
Forcepoint DLP – Configure Azure AD as external identity source

In order to configure Azure AD as an external source do as follows:

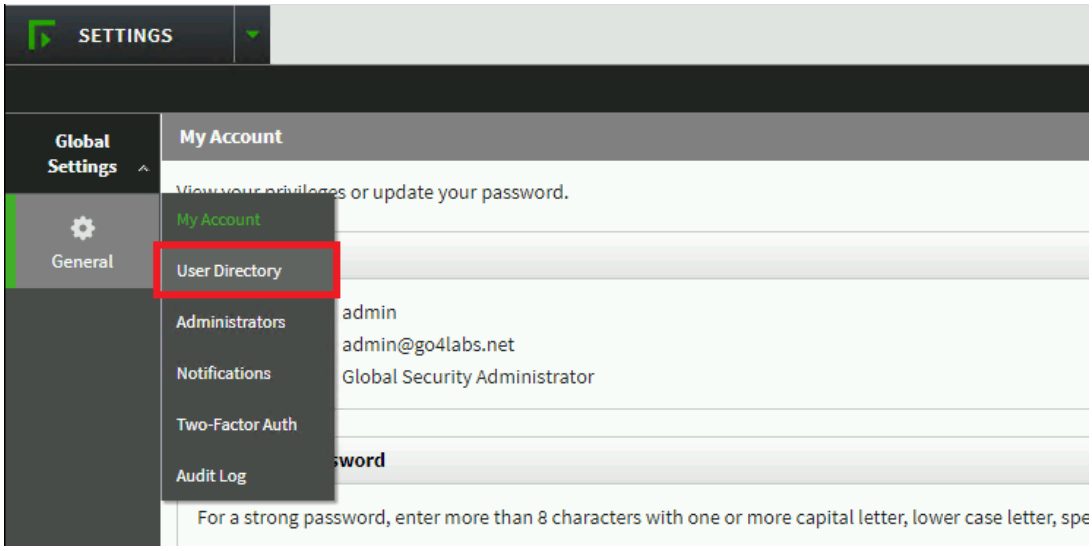
1. Log into your local Forcepoint Security Manager using your local admin credentials.



2. Click the gear in the top right corner to access the settings page.



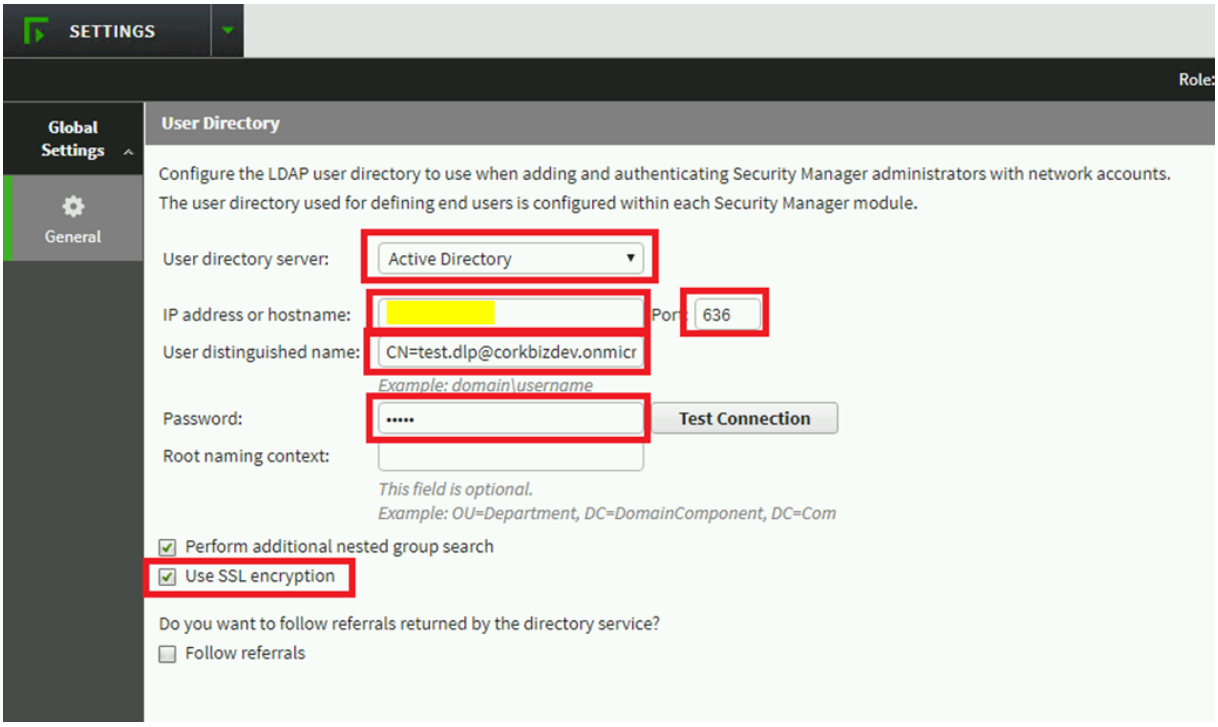
3. Move your mouse over the **General** tab and reveal the roll over menu, click **User Directory** tab



4. Fill in the details to connect to the Azure Active Directory as follows:
    - **User directory server**: Set this to **Active Directory**
    - **IP address or hostname**: The public IP address or FQDN of your Azure AD, this can be found in the **Properties** section of **Azure AD Domain Services**.
    - **User distinguished name**: This should be the distinguished name of an account with admin access to the Azure AD. Format for this field is similar to:

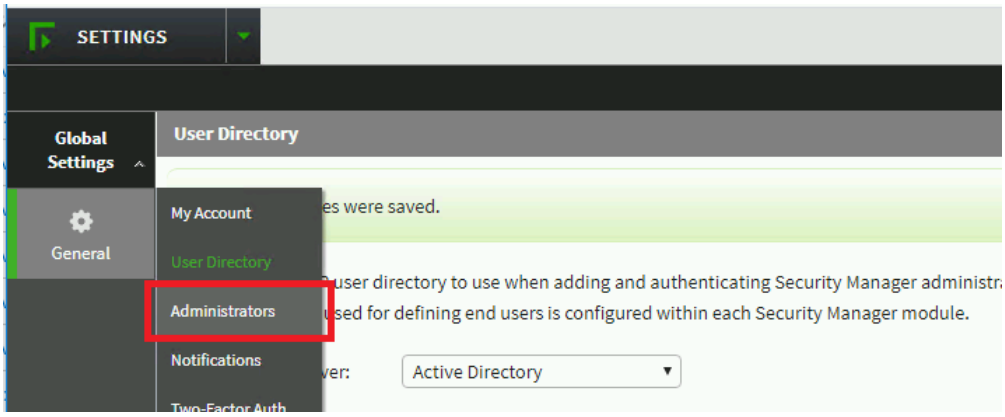
      CN=test.dlp@corkbizdev.onmicrosoft.com,OU=AADDC Users,DC=corkbizdev,DC=onmicrosoft,DC=com

      For more information on distinguished name refer to:
      https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ldap/distinguished-names
    - **Password**: the password of the account that has admin access to the Azure AD
    - **Use SSL encryption**: make sure this is ticked for Azure Active Directory

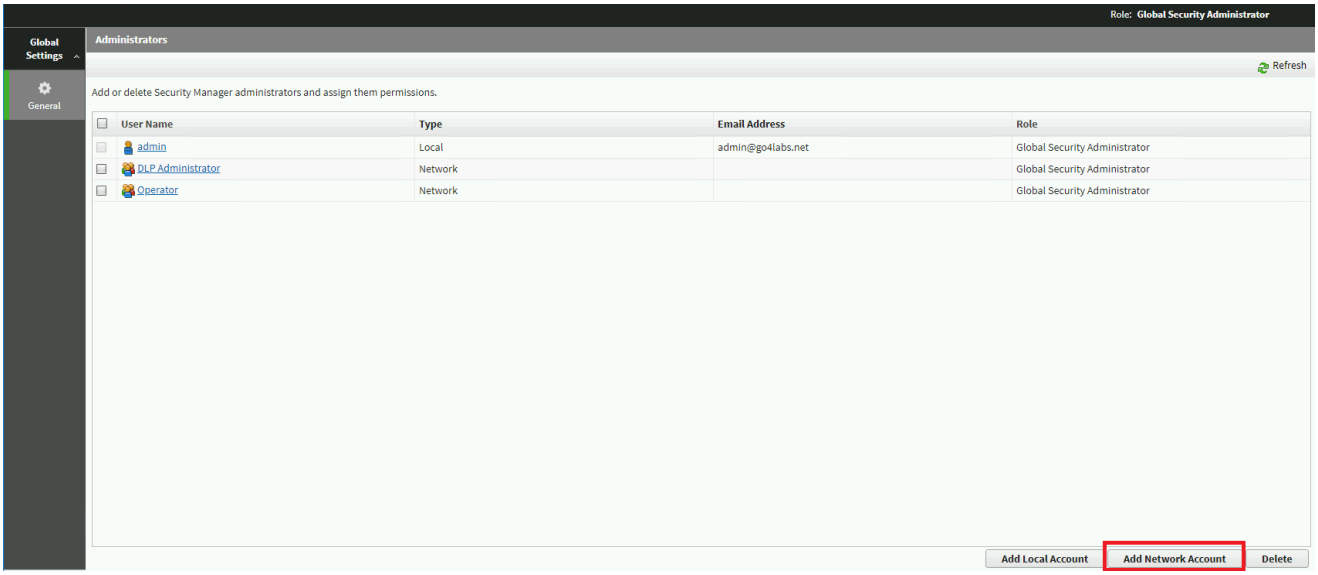


Click **OK** when you are done.

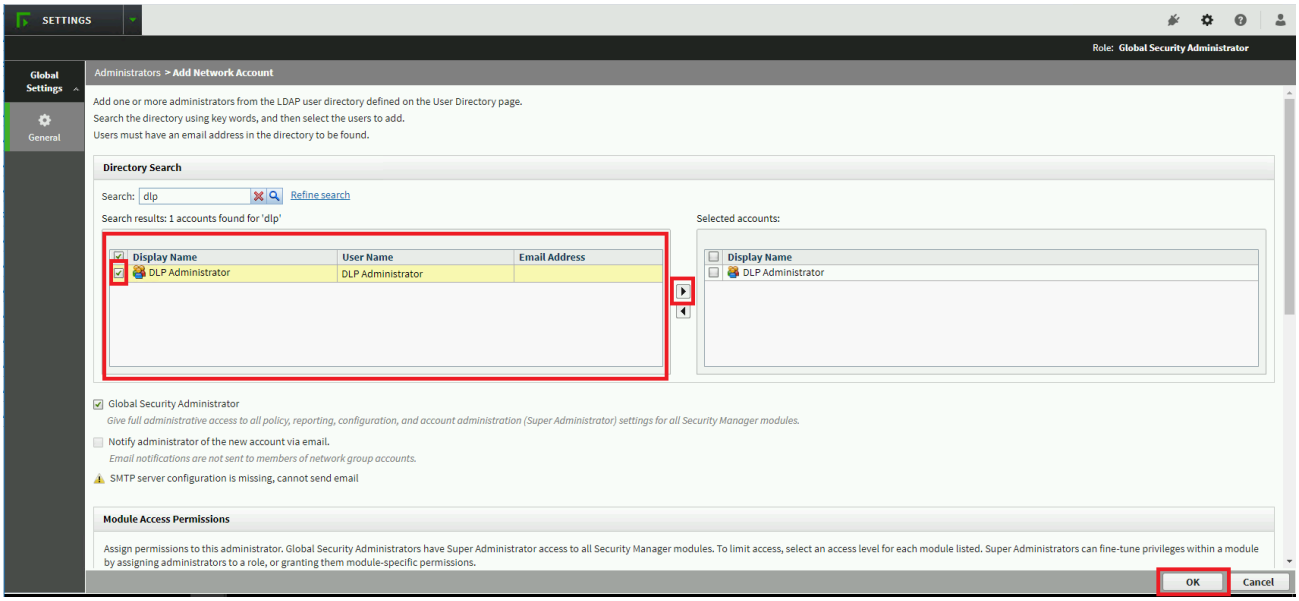
5. After you have successfully added the Active Directory mouse over **General** and select **Administrators** from the roll over menu



6. From here select A**dd Network Account** in the bottom right.



7. From here you can search for the Azure AD user or groups you want to assign permissions to on the Forcepoint Security Manager. Once you have selected the users or groups, use the arrow icon to add it to the selected accounts box.

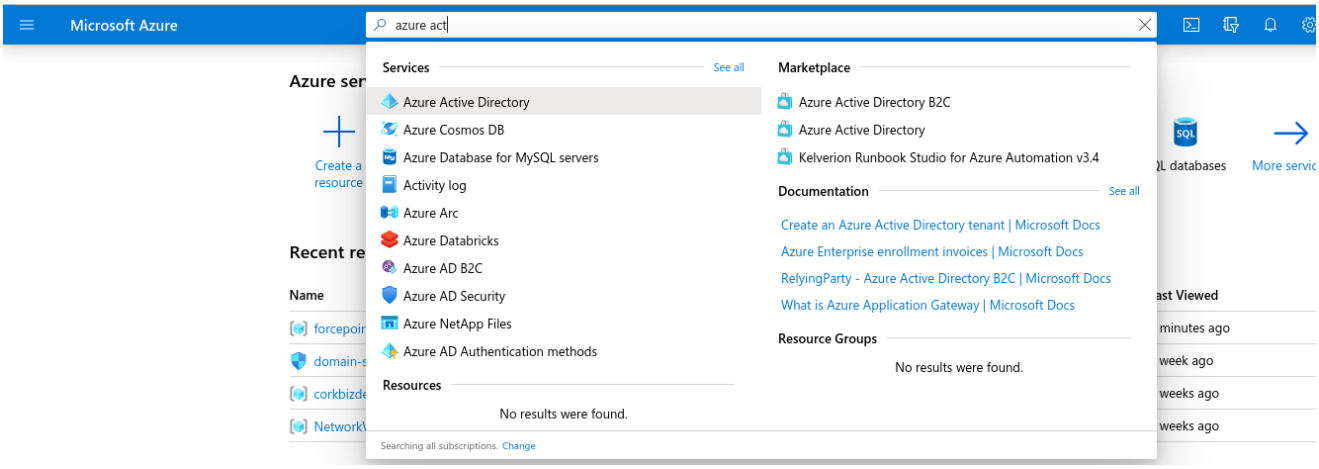


Click **OK** at the bottom right of the screen.

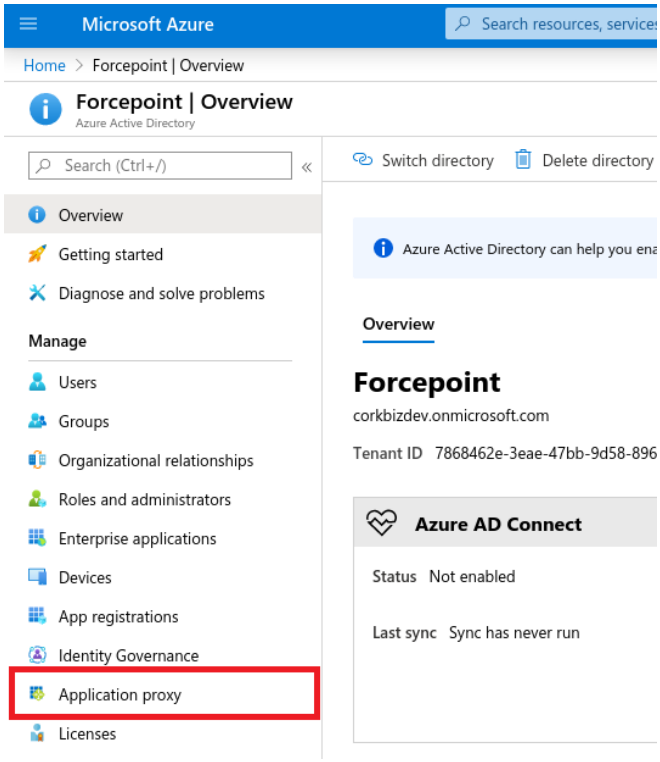
Forcepoint DLP – Configure Azure Application proxy

Azure Application proxy is the component provided by Azure to expose your local web application through Azure. To register Forcepoint DLP into Application proxy follow these steps:

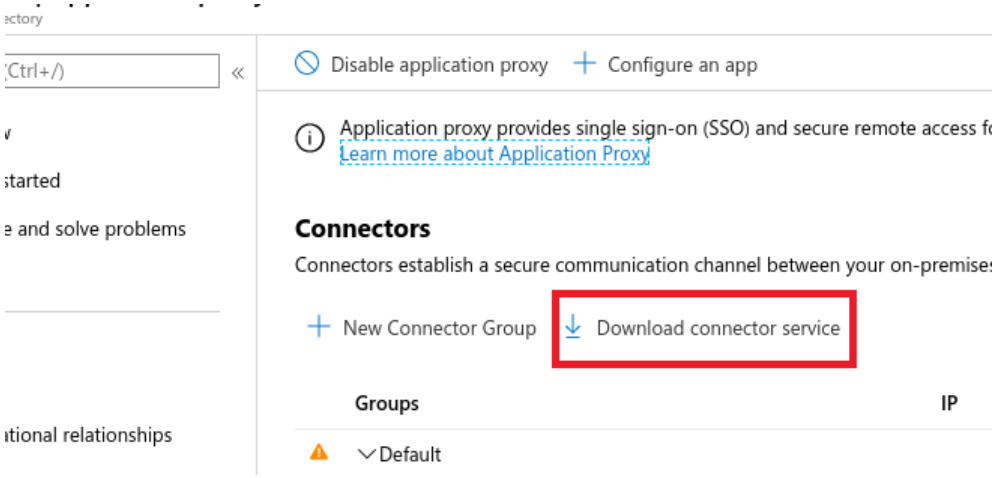
1. From the Azure portal navigate to the **Azure Active Directory**



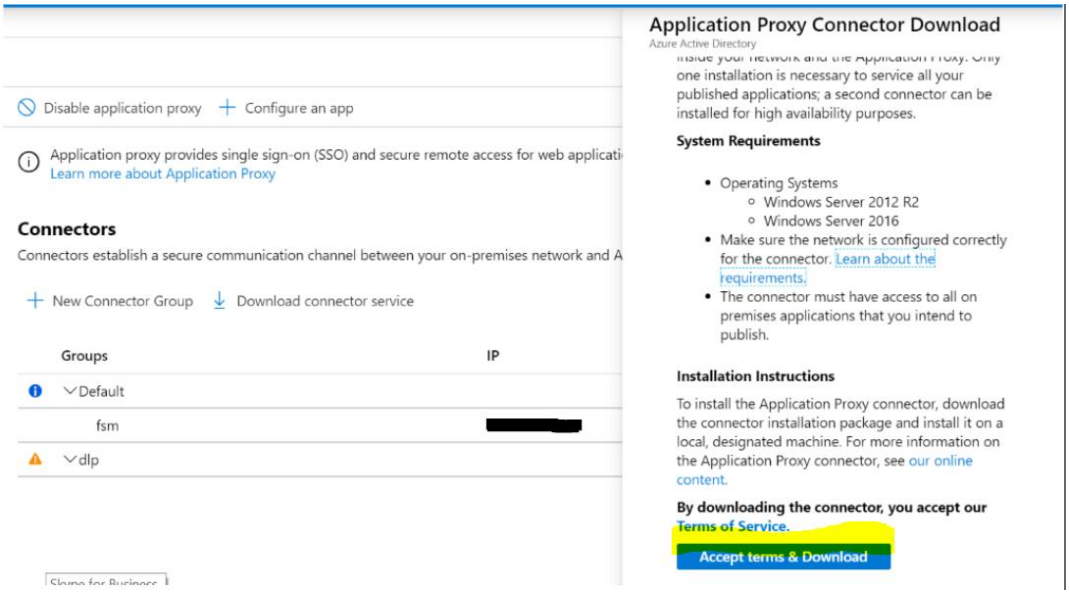
2. In the left pane select **Application proxy**



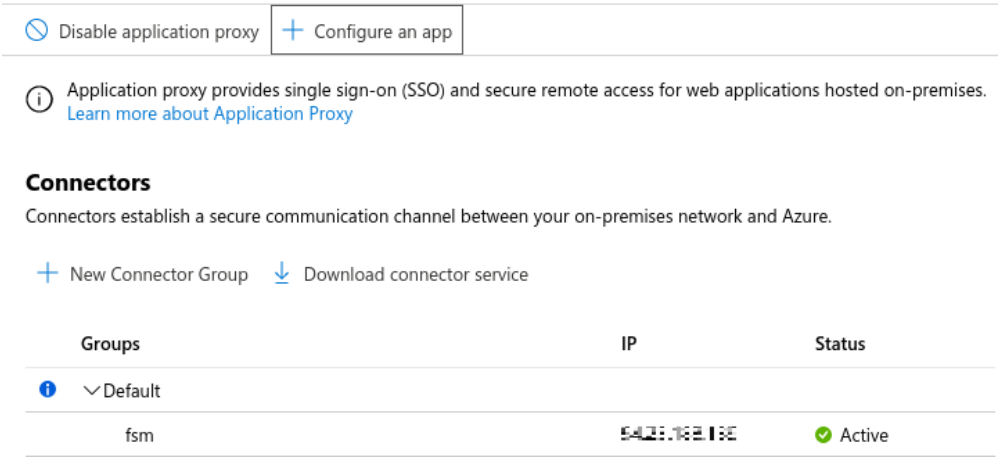
3. Click **Download connector service** and install it on the Forcepoint Security Manager machine.



4. Then click **Accept Terms & Download.**



5. After the installation is completed you should see the IP address of the registered machine in the list of **Connectors** of the **Application proxy**:

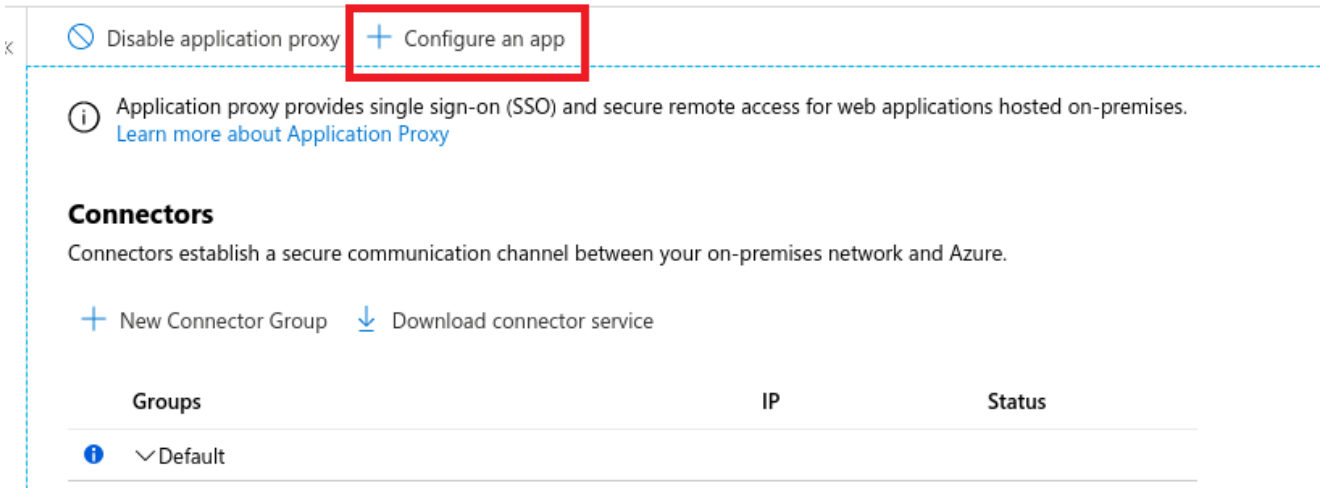


This shows that the **Application proxy** installed in the Forcepoint Security Manager machine has successfully connected to the Azure.

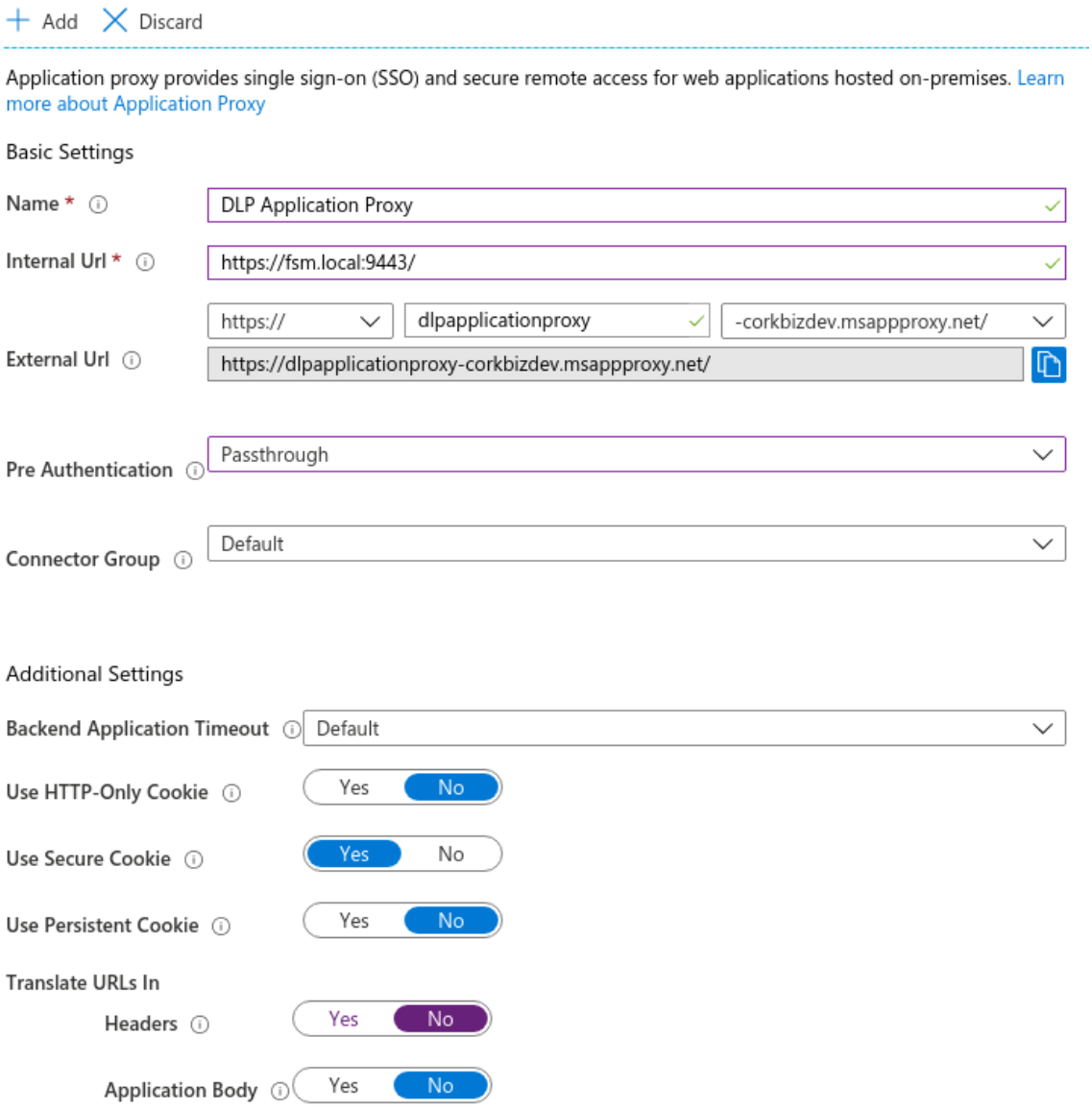
Next step is configuring the **Application proxy** hosted on the Forcepoint Security Manager machine to connect locally to Forcepoint Security Manager. The Application proxy **can only connect to local applications using an FQDN:** if your Forcepoint Security Manager is installed with the service binded to the IP address of the hosting machine, see the **Appendix** for two ways to address this.

Once the Forcepoint Security Manager is reachable using a FQDN, proceed with the following steps.

6. Back on the **Application proxy** page on Azure click **Configure an app**



7. Configure the **Basic Settings** and **Advanced Settings** as follows
    - **Name:** name the app being configured with the Application proxy
    - **Internal URL**: enter the FQDN of the Forcepoint Security Manager including the port
    - **External URL:** use **https** and define part of the FQDN, the last part of the URL will be the Azure AD name
    - Set **Pre-Authentication** as **Passthrough**
    - **Connector group**: this is the group your connector is in, by default is **Default**
    - **Backend Application Timeout**: leave this **Default**
    - **Translate URLs In**
        - **Headers**: No
    - Leave everything else default



8. Click **Add** in the top left corner.

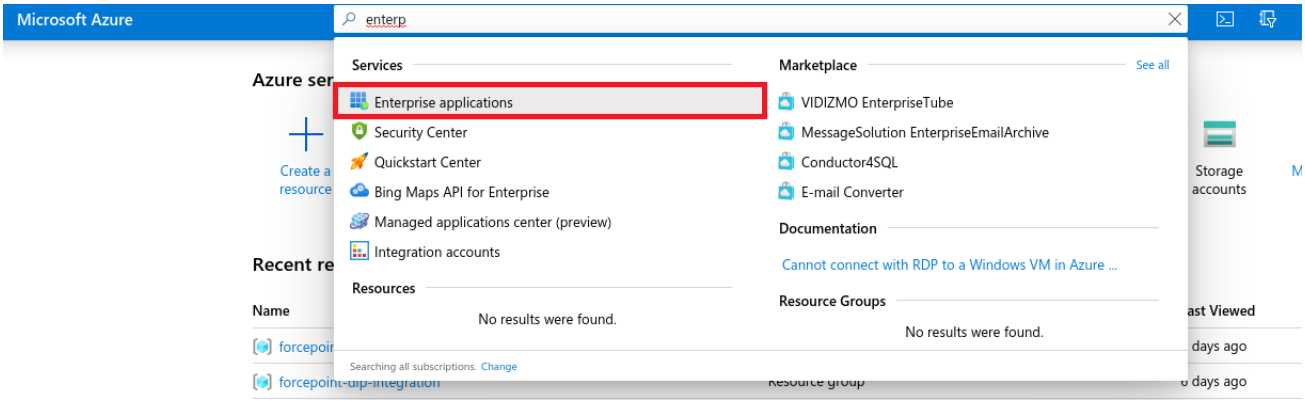
After the page has been created you will be directed to a page **App name | Overview** (in the example above, the app name is DLP Application Proxy).

9. Go to **Application Proxy** under **Manage** on the left. Click **Test Application** and **Open Application** to verify all settings work correctly: you will be able to reach the Forcepoint Security Manager application from the public FQDN provided by Azure.

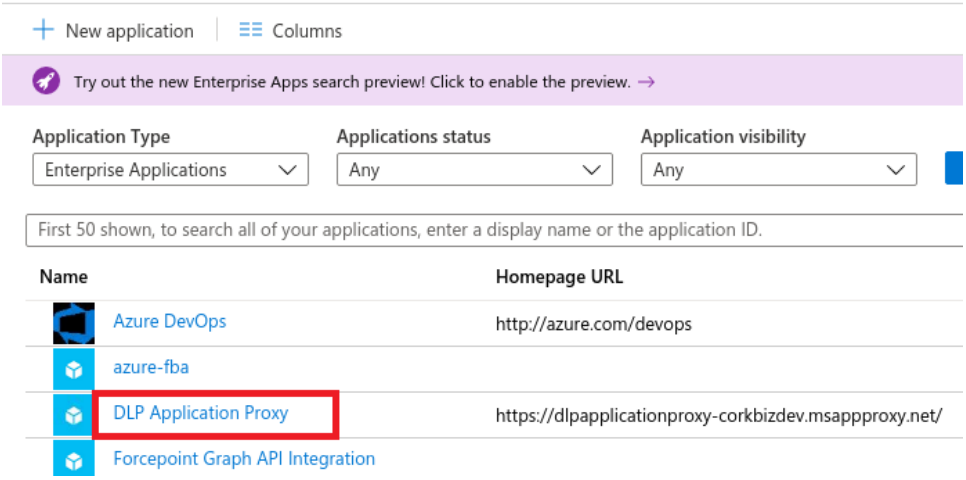
Provisioning access to on-premise Forcepoint Security Manager via Azure application

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

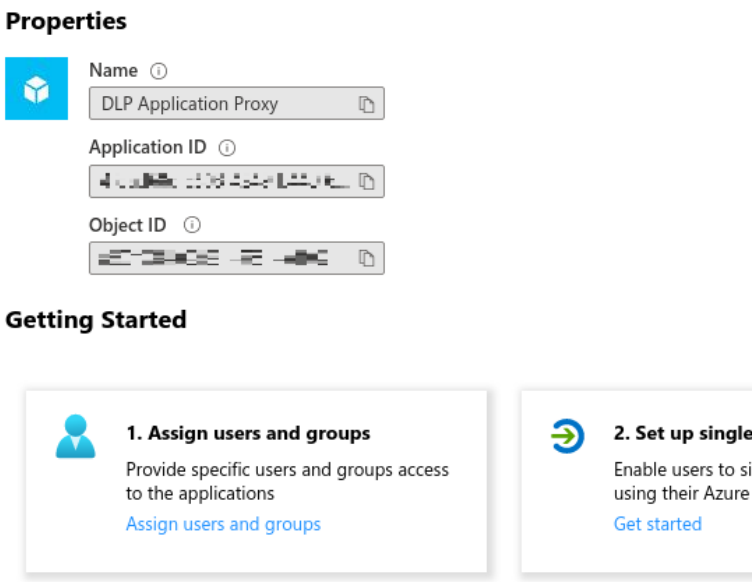
1. From the Microsoft Azure portal search for Enterprise applications



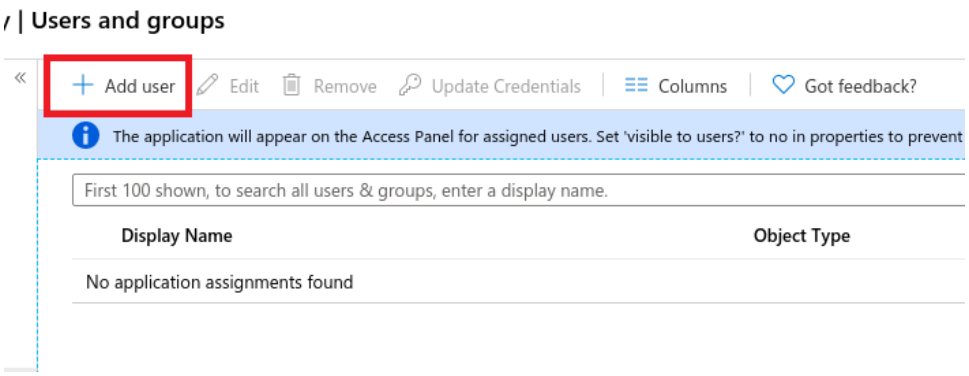
2. Select the name of the application you created in previous steps of this guide (in the example below, the app name is **DLP Application Proxy**)



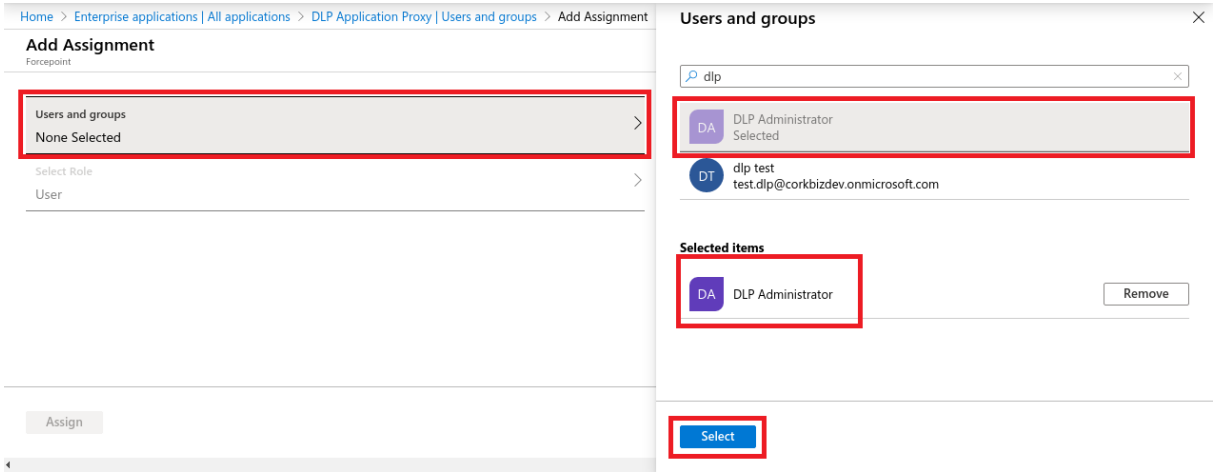
3. From this page select **1. Assign users and groups**.



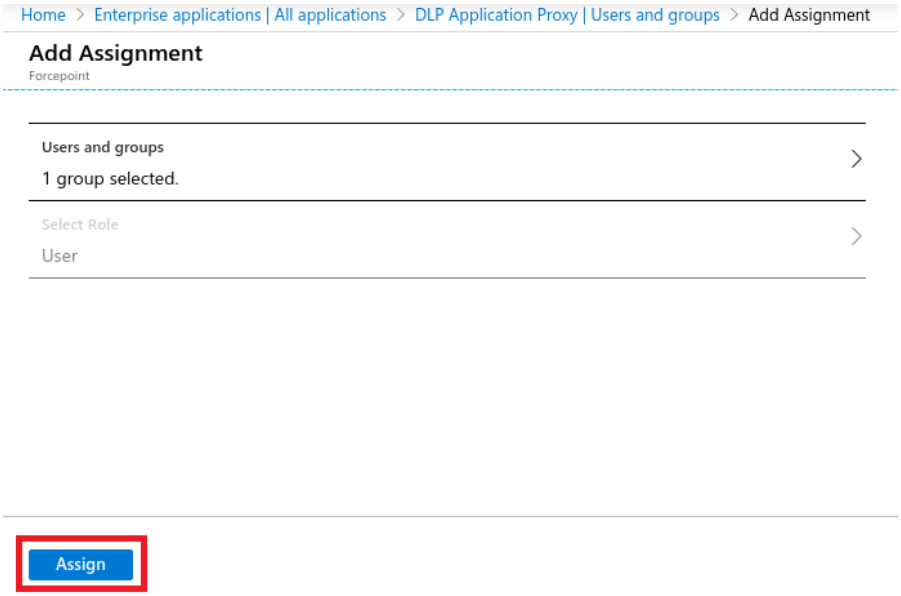
4. Select **Add user**



5. From this page you can select groups of users or individual users. Shown below is the same group we added to the Forcepoint Security Manager. All users belonging to this group will have access to this application.



6. Finally click **Assign**.



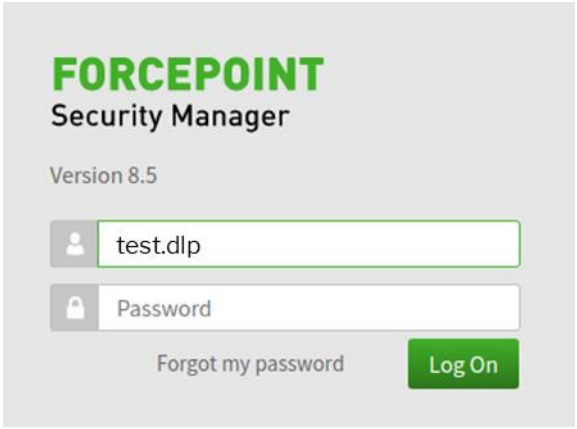
Access via My Azure Applications

1. Login to https://myapplications.microsoft.com/ with a user assigned or belonging to a group that was assigned to the Forcepoint Security Manager application.
2. Find your Azure app and click on it. This will redirect your web browser to Forcepoint Security Manager on-premise.

3. Enter your Azure credentials: the username is the part before the @ symbol in your Azure email address

Example

*If your username is **test.dlp@myazuredirectory.com** then the username is **test.dlp***



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 version of Forcepoint DLP in use is listed as compatible

Forcepoint DLP version 8.7.0.360

→ Docker images for this integration have been tested with

Docker 19.03.6

Docker-compose 1.25.4

→ The docker implementation has been tested on a Centos 7.3 machine

→ 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/dlp-deploy-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_HOST_PUBLIC_IP_ADDRESS

replacing YOUR_DOCKER_PUBLIC_IP_ADDRESS with the public IP of the docker host machine. Once done check the result is similar to below:

```
PING YOUR_DOCKER_HOST_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 Forcepoint Security Manager: execute the following command on docker host machine:

ping -c 2 FSM_PRIVATE_IP_ADDRESS

replacing the FSM_PRIVATE_IP_ADDRESS with your Forcepoint Security Manager private IP address or the hostname. Once done check the result is similar to below:

```
PING FSM_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:

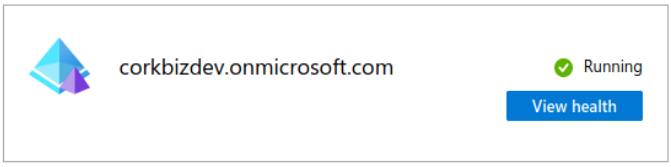
Client:
Debug Mode: false

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



Traditional Implementation

Validate the prerequisites

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

→ Check the version of Forcepoint DLP in use is listed as compatible

Forcepoint DLP version 8.7.0.360

→ This integration requires to be run on a CentOS 7.3 machine

→ 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/dlp-deploy-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_HOST_MACHINE_PUBLIC_IP_ADDRESS with the public IP of the host machine. 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 Security Manager host machine over the network: execute the following command on host machine:

ping -c 2 FSM_PRIVATE_IP_ADDRESS

replacing the FSM_PRIVATE_IP_ADDRESS with your Forcepoint Security Manager private IP address or the host-name. Once done check the result is similar to below:

```
PING FSM_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 python3.6 is installed: Execute following command on host machine:

python3 --version

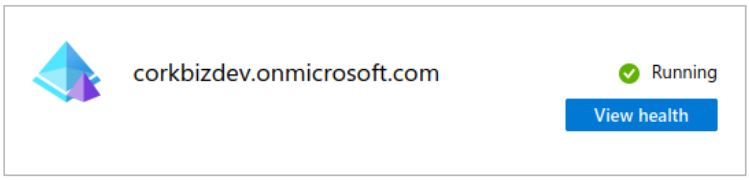
Check the output is similar to below:

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:

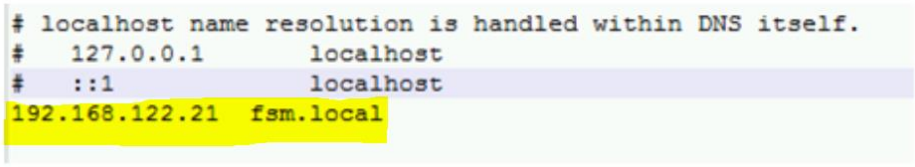
→ Check the domain service is successfully running in Azure



Appendix – Mapping an IP address to an FQDN

If the Forcepoint Security Manager was installed binding the service to the IP address of the machine rather to the hostname, one extra step is necessary in order to configure the **Application proxy** with the Forcepoint Security Manager. This can be accomplished by either

→ Adding a static entry to the end of the **hosts** file located in **C:\Windows\System32\drivers\etc** so the **Application proxy** can locally resolve the hostname of Forcepoint Security Manager to the IP (private IP address) of the Forcepoint Security Manager machine. As highlighted in the screenshot below:



→ If a local DNS server is available, adding a proper entry in the DNS server for the Forcepoint Security Manager machine which resolves to the private IP of the Forcepoint Security Manager machine itself

Forcepoint

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.