
NGINX App Protect Documentation

Matthieu Dierick

Apr 23, 2021

CONTENTS:

- 1 Publish and protect on-prems apps with Azure AD as identity provider 1**
- 1.1 Class 1 - Check the Lab Architecture 2
- 1.2 Class 2 - Deploy APM to protect on-prems apps 7
- 1.3 Class 3 - Leverage Azure AD to protect Cloud Apps 27
- 1.4 Class 4 - Enable MFA 29
- 1.5 Class 5 - Clean up the lab 35

PUBLISH AND PROTECT ON-PREMS APPS WITH AZURE AD AS IDENTITY PROVIDER

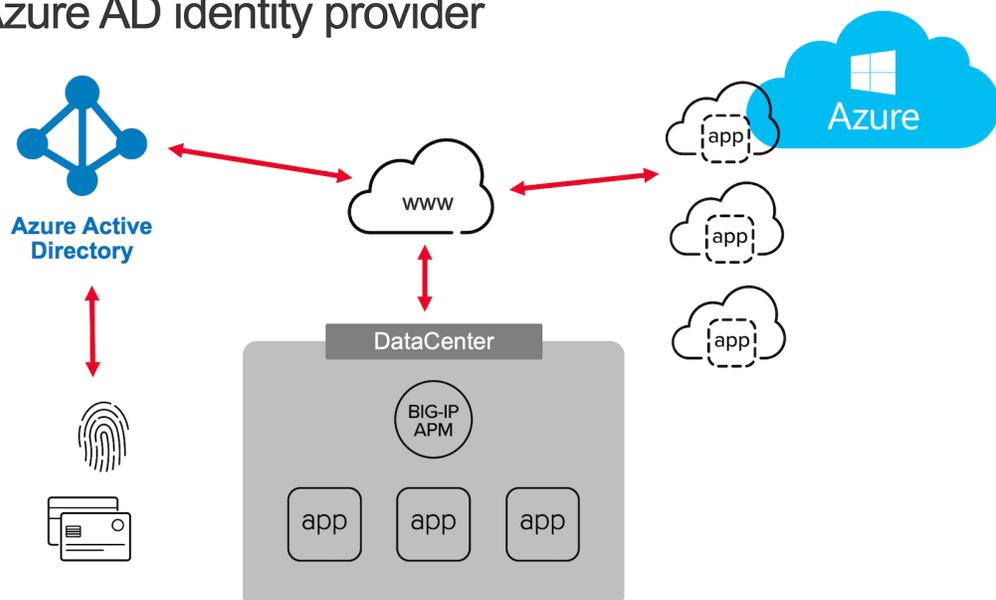
Warning: For any remark or mistake in this lab, please send a Teams chat to Matthieu DIERICK.

In this lab, you will learn how to connect APM to Azure AD as IDaaS. Since v15.1, you can enable APM as SAML SP and Azure AD as SAML IDP. In this lab, we will use the new **Easy Button** Guided Configuration template. This template:

1. Publish on-prems apps
2. Enable Single Sign on
3. Interconnect (SAML binding) APM with Azure AD tenant

Note: You will notice we will never connect to Azure AD interface. APM will use Microsoft Graph API to configure AAD tenant accordingly.

Leverage Azure AD identity provider



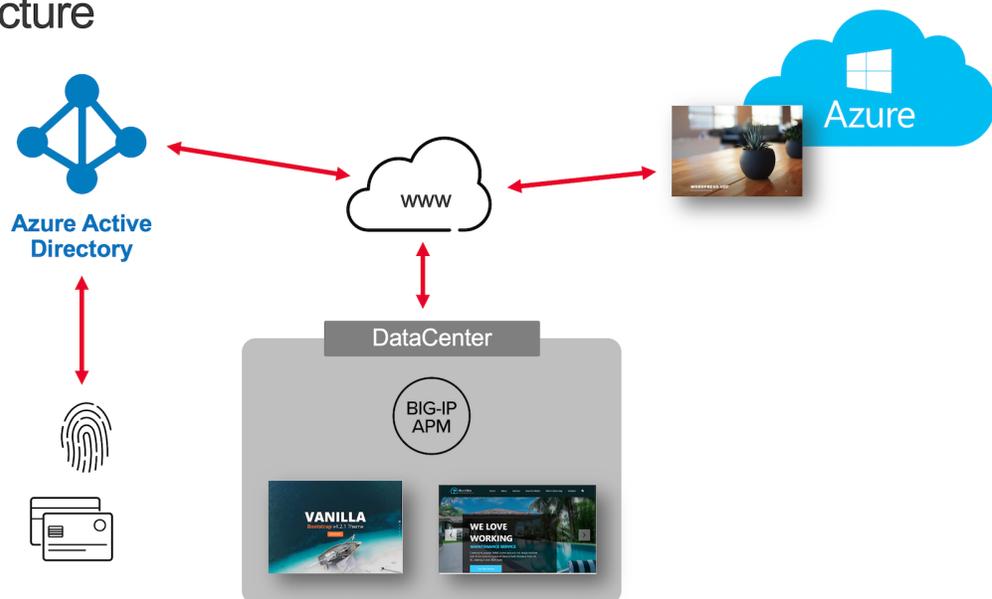
In the video below, you can see the use case. This is **not** the **lab video**, it is the public facing use case demo.

1.1 Class 1 - Check the Lab Architecture

In this class, we will protect 3 apps:

1. 2 internal apps
 1. Vanilla Application hosted in IIS
 2. Skyblue Application hosted in IIS
2. 1 cloud app hosted in Azure cloud
 1. Wordpress-UDF hosted in Azure cloud

Lab architecture



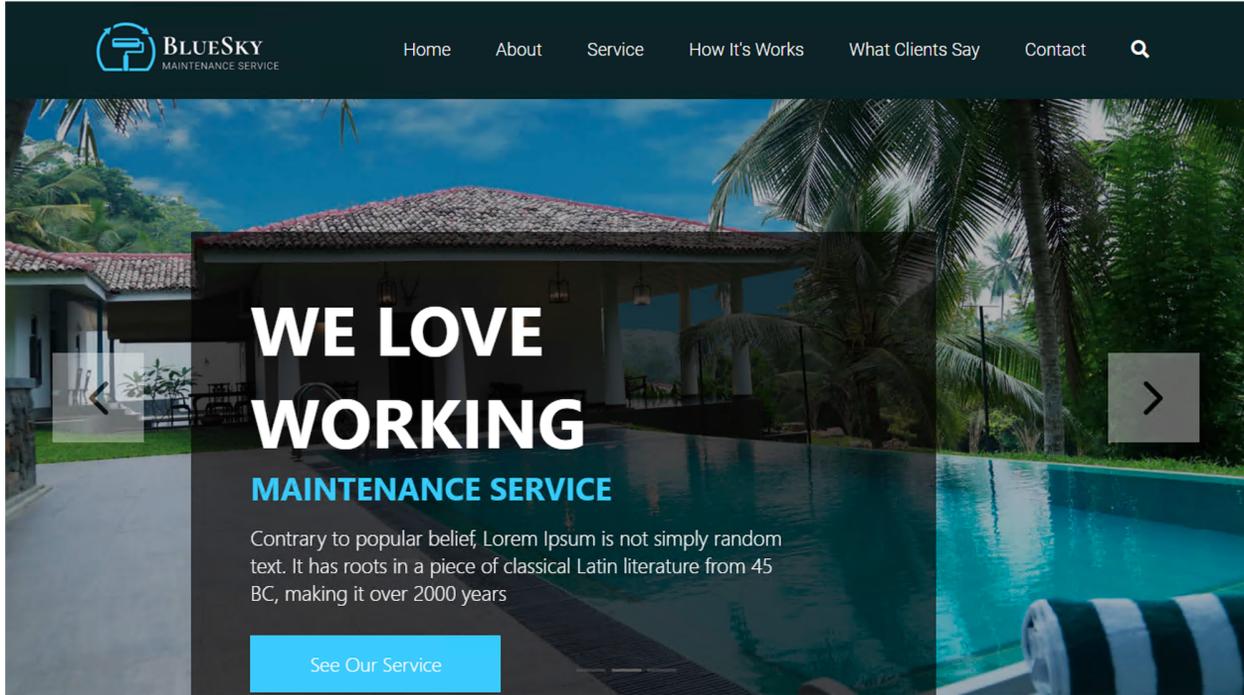
Class 1 - All sections

1.1.1 Architecture of Internal Apps

Bluesky application

This application resides on-prems in IIS server. Its FQDN is `https://bluesky.f5access.onmicrosoft.com`

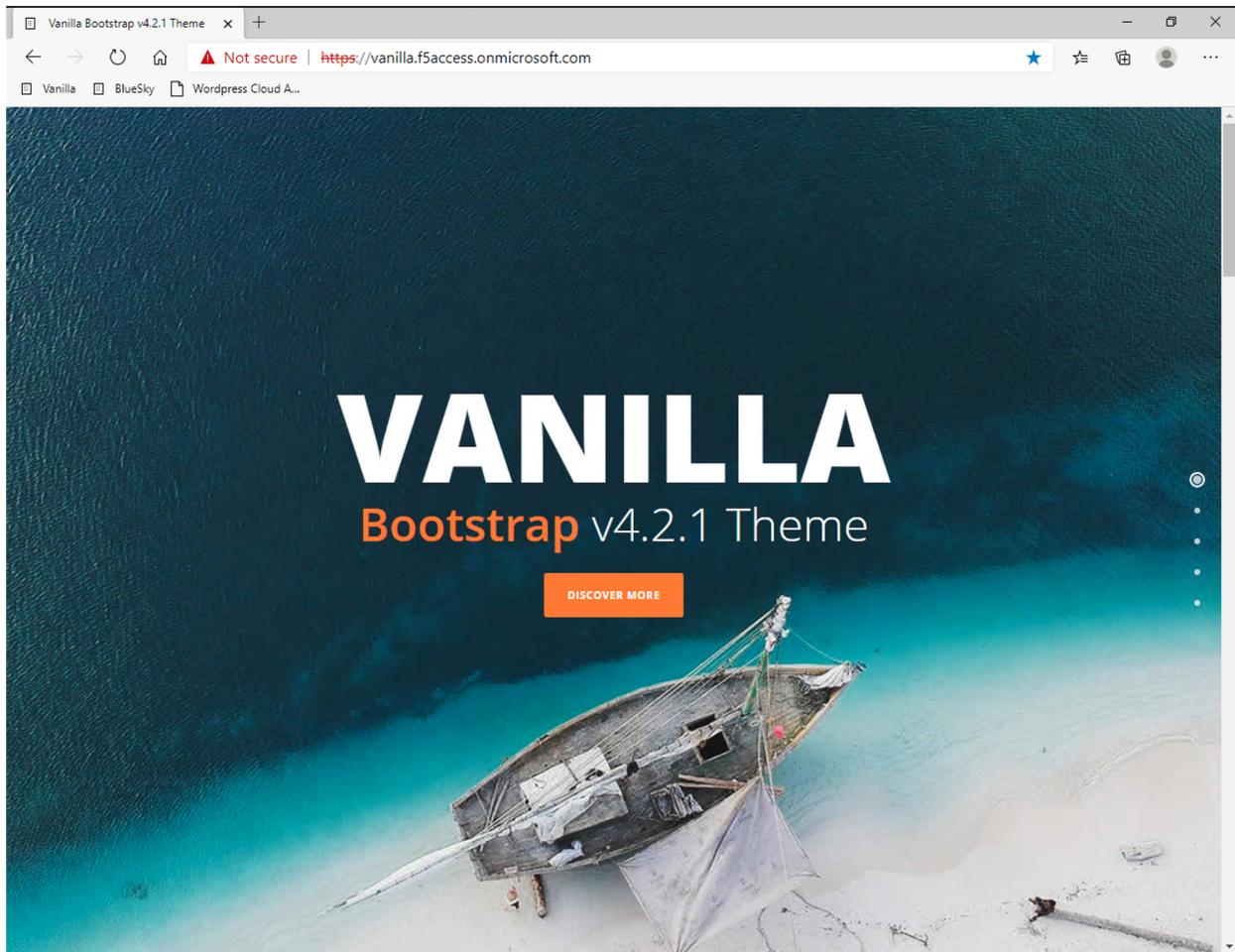
This application is not **authenticated**, meaning there is no **Single Sign on** required in front of this app.



Vanilla application

This application resides on-prems in IIS server. Its FQDN is `https://vanilla.f5access.onmicrosoft.com`

This application is **authenticated** by Kerberos. So a **Signle Sign On** will be required to connect to this app.

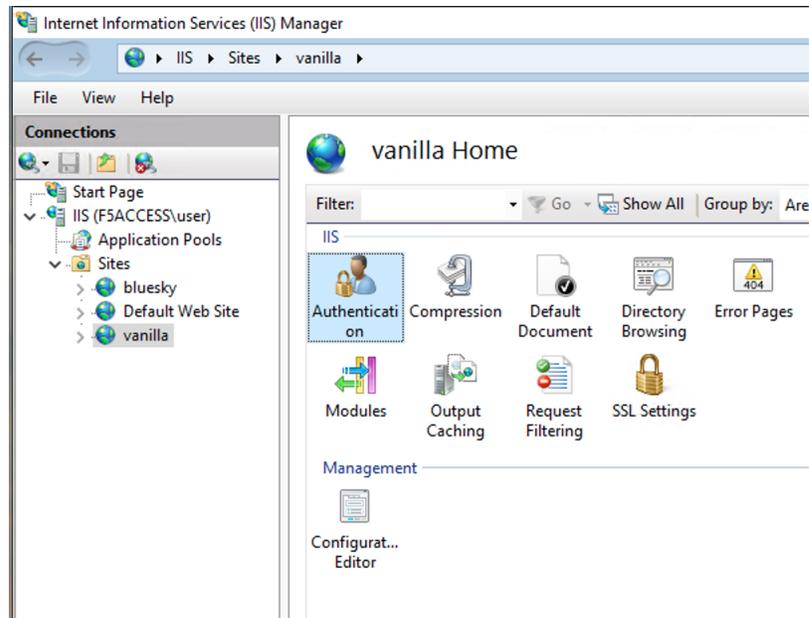


Check IIS configuration

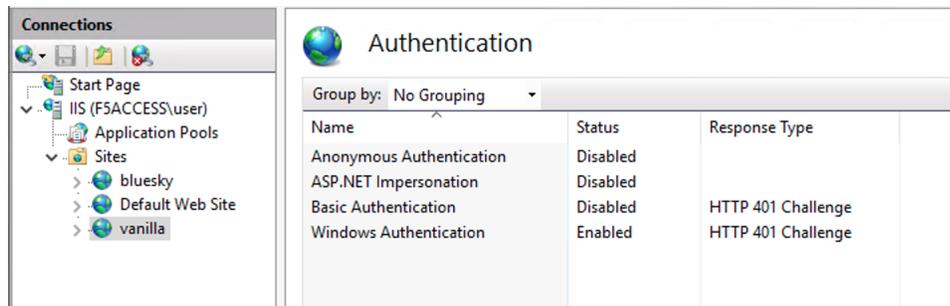
1. RDP to IIS with f5access\user as user, and user as password
2. Click IIS manager icon in the taskbar



3. In the Connections tree, click on vanilla and Authentication



4. You can notice Anonymous Auth is **Disabled** and Windows Authentication is **Enabled**

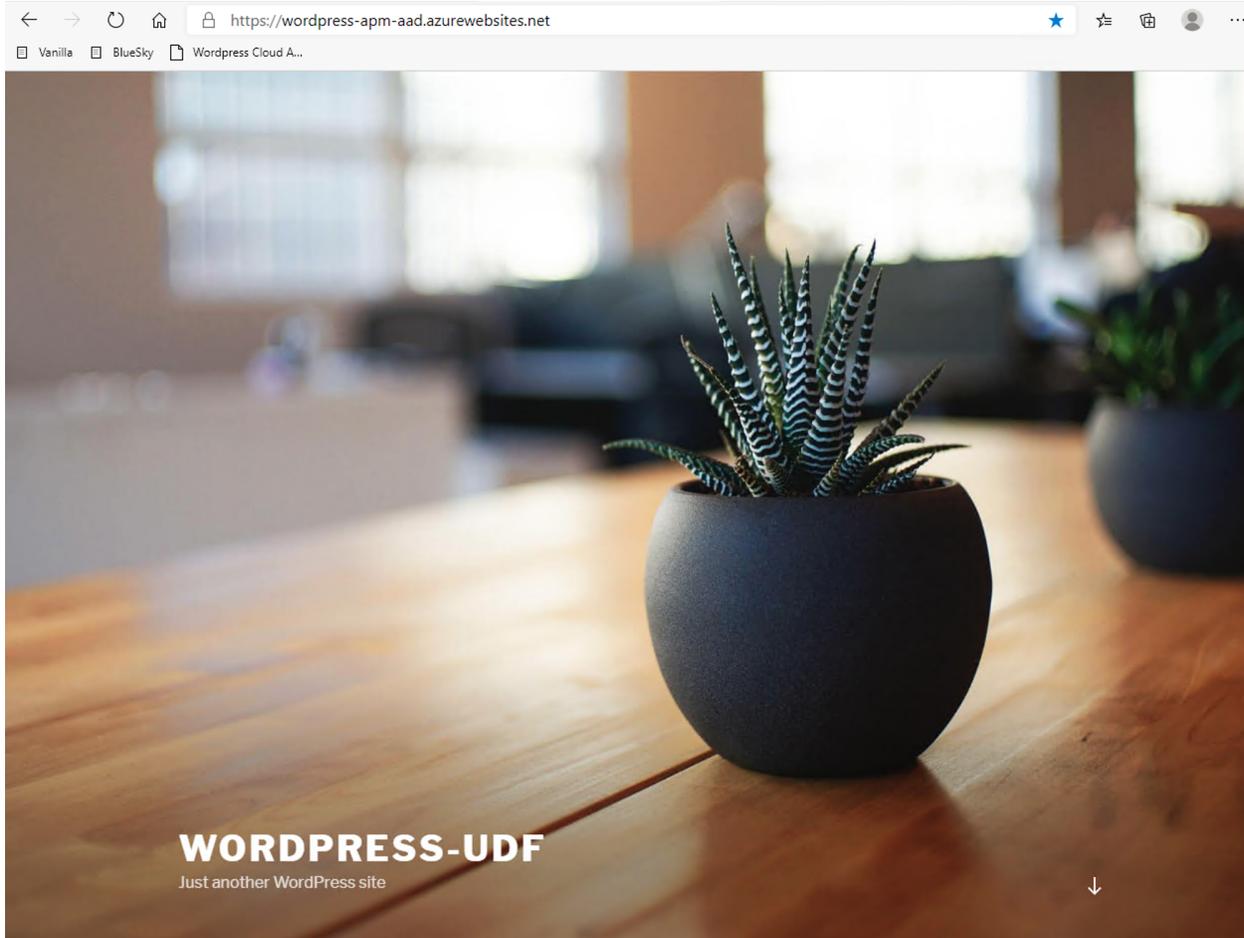


Note: In the next class we will configure APM to publish, protect and SSO to internal apps.

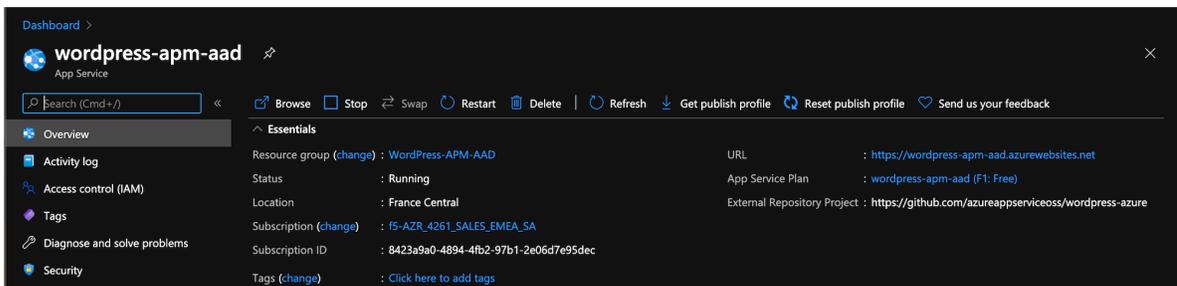
1.1.2 Architecture of Cloud App

Note: In this use case, we don't cover only internal, sensitive or legacy applications. In a real world, customers have applications on-prems and in the public cloud.

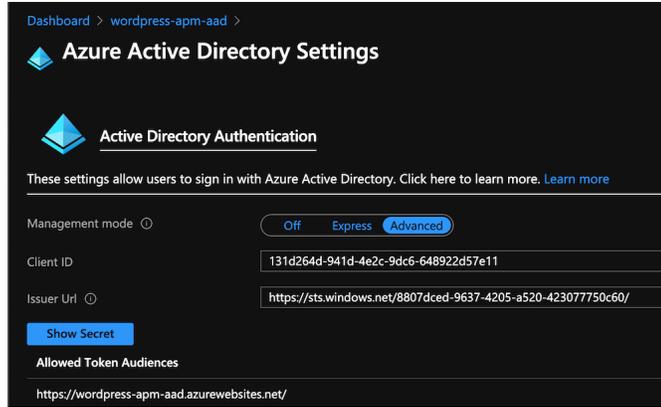
Note: A Wordpress application is already up and running in Azure Cloud at this address <https://wordpress-apm-aad.azurewebsites.net/>



1. This Wordpress application is an Azure App Service.



2. This App Service is already bound with our demo Azure AD tenant.

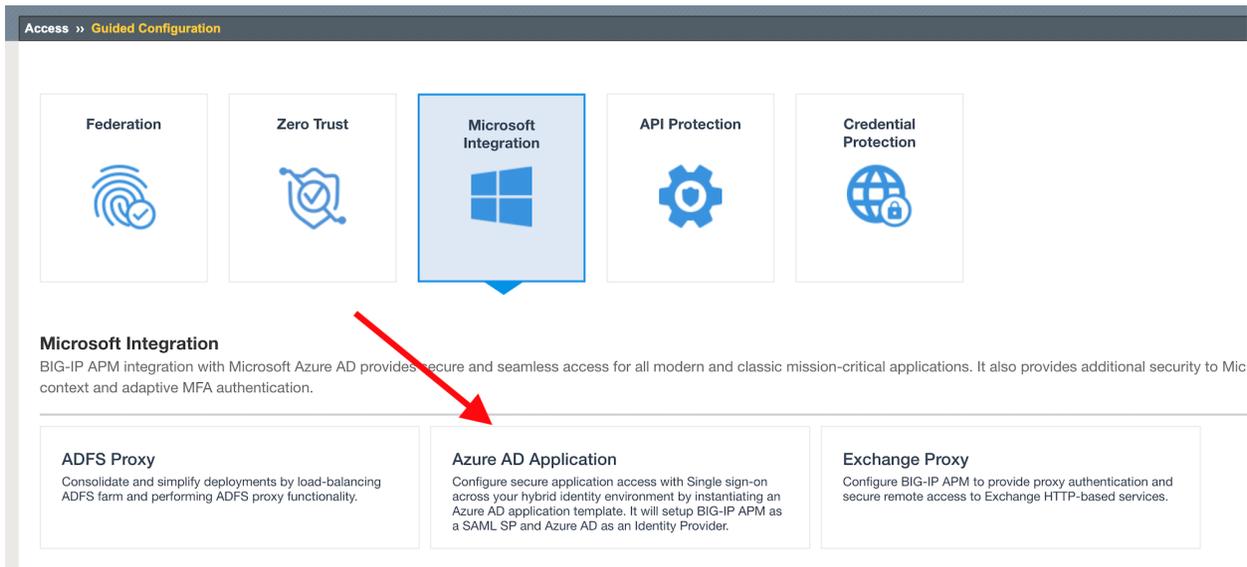


Warning: It is important to note this application is **not tied** to APM. APM only publishes and protects on-prems apps. All other cloud and SaaS apps are directly connected to Azure AD.

1.2 Class 2 - Deploy APM to protect on-prems apps

In this class, we will publish Vanilla and Bluesky applications hosted on-prems.

To do so, we will use Guided Configuration template Azure AD Application

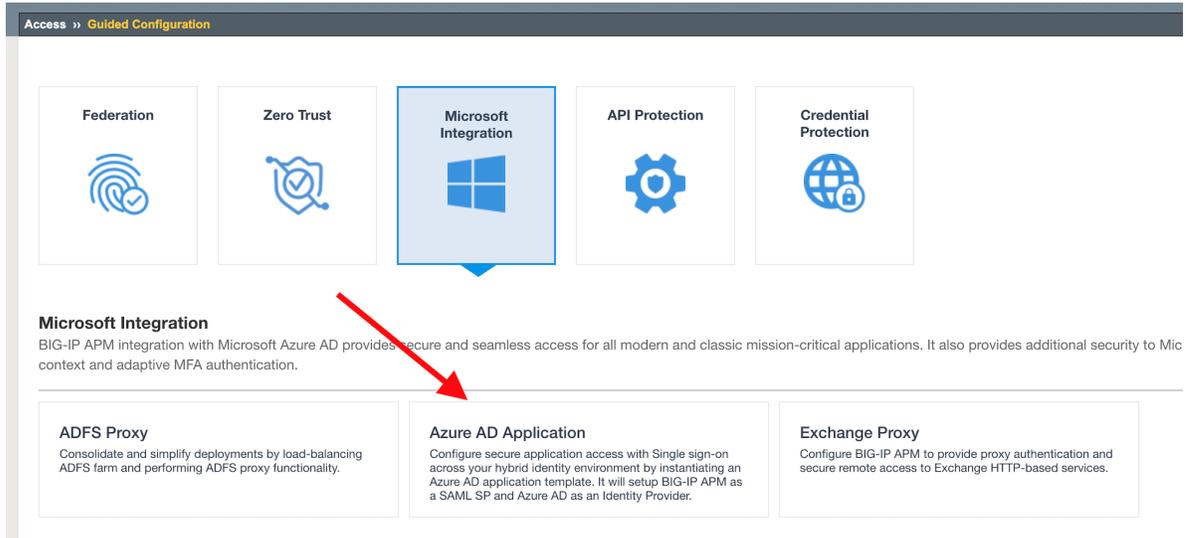


Class 2 - All sections

1.2.1 Publish and protect Bluesky app

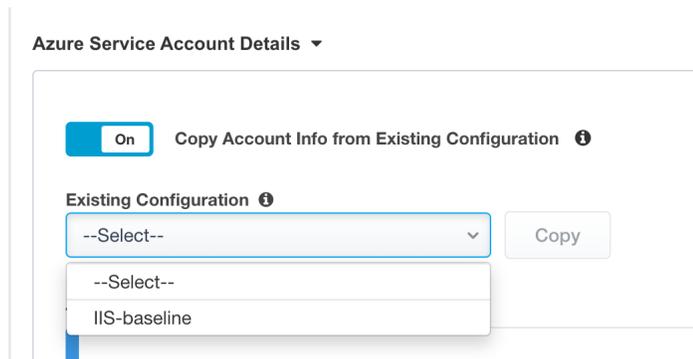
Let's start with Bluesky application. Reminder, Bluesky does not have any Authentication enabled.

1. Connect to BIG-IP HTTPS user interface from UDF as admin and password admin
2. In `Access > Guided Configuration`, select `Microsoft Integration > Azure AD application`



Configuration Properties

1. Click `Next` and start the configuration
2. Configure the page as below
 1. Configuration Name : `IIS-Bluesky-<My Name>` Why my name ? Because this app will be created in Azure AD tenant. And we need to differentiate all apps. Example : `IIS-Bluesky-Matt`
 2. In `Azure Service Account Details`, Select `Copy Account Info` form Existing Configuration, and select `IIS-baseline`, then click `Copy`



Note: In a real world, you will set here the values from the Azure Service Application created for APM. You have to create an Azure Application so that APM gets access to Microsoft Graph API. But for **security concerns**, I can't show in this lab the application secret.

Note: The steps to create this Azure applications are below

1. In Azure AD, create a service application under your organization's tenant directory using App Registration.
2. Register the App as Azure AD only single-tenant.
3. Request permissions for Microsoft Graph APIs and assign the following permissions to the application:
 1. Application.ReadWrite.All
 2. Application.ReadWrite.OwnedBy
 3. Directory.Read.All
 4. Group.Read.All
 5. Policy.Read.All
 6. Policy.ReadWrite.ApplicationConfiguration
 7. User.Read.All
4. Grant admin consent for your organization's directory.
5. Copy the Client ID, Client Secret, and Tenant ID and add them to the Azure AD Application configuration.

3. Click `Test Connection` button → Connection is valid

Azure Service Account Details ▾

On Copy Account Info from Existing Configuration ⓘ

Existing Configuration ⓘ

IIS-baseline ▾ Copy

Tenant ID ⓘ

8807dced-9637-4205-a520-423077750c60

Client ID ⓘ

1ef1f16f-0472-4d3e-9062-138030a5c41d

Client Secret ⓘ

....

Test Connection ✔ Connection is valid

4. Click Next

Service Provider

1. Configure the page as below

1. Host bluesky.f5access.onmicrosoft.com
2. Entity ID is auto-filled https://bluesky.f5access.onmicrosoft.com/IIS-Bluesky-myname>

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ	bluesky.f5access.onmicrosoft.com
Entity ID ⓘ	https://bluesky.f5access.onmicrosoft.com/IIS-Bluesky-Matt
Description ⓘ	
Relay State ⓘ	

Security Settings ▾

<input type="checkbox"/> Enable Encrypted Assertion ⓘ		
Assertion Decryption Private Key ⓘ	--Select--	↻
Assertion Decryption Certificate ⓘ	--Select--	↻

Cancel

Save Draft

Back

Save & Next

3. Click Save & Next

Azure Active Directory

1. Select Azure BIG-IP APM Azure AD... template

Note: As you can notice, there are several templates available for different applications. Here, in this lab, we will publish a generic app. So we select the first template.

2. Click Add
3. In the new screen, configure as below
 1. Signing Key : default.key
 2. Signing Certificate : default.crt
 3. Signing Key Passphrase : F5twister\$

SAML Signing Certificate ▾

Signing Key ⓘ

default.key ▾

Signing Certificate ⓘ

default.crt ▾

Signing Key Passphrase ⓘ

.....|

Signing Option ⓘ

Sign SAML assertion ▾

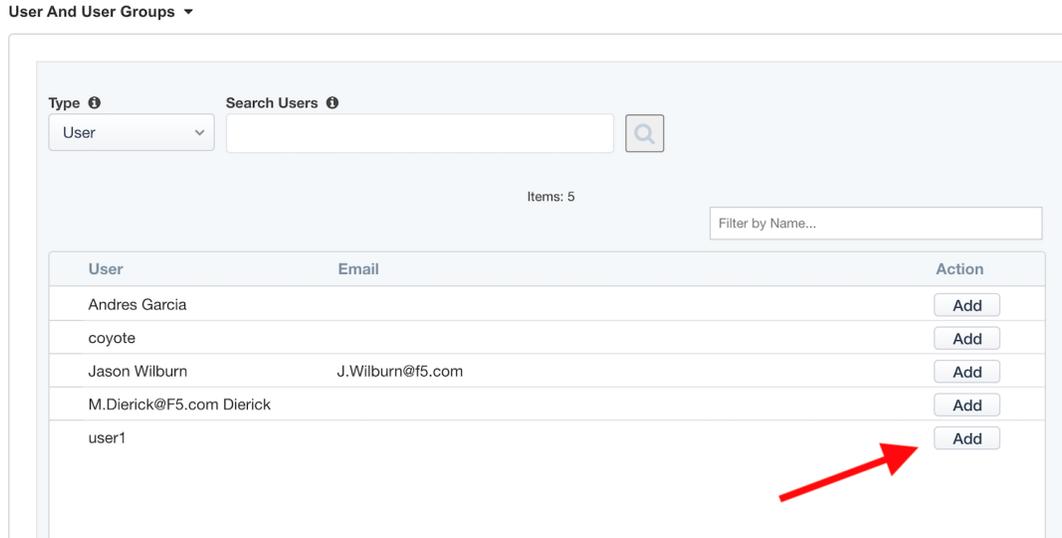
Signing Algorithm ⓘ

RSA-SHA256 ▾

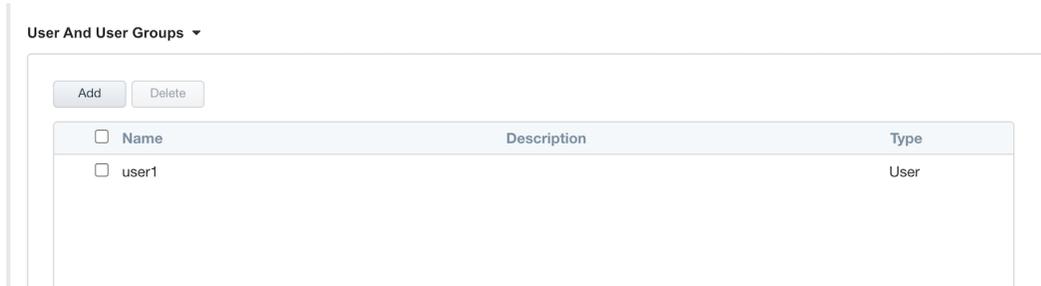
4. In User And User Groups, click Add

Note: We have to assign Azure AD users/group to this app, so that they can be allowed to connect to it.

1. In the list, click Add for the user user1. If you can't find it, search for it in the search field.



2. Click Close
3. You can see user1 in the list.



4. Click Save & Next

Virtual Server Properties

1. Configure the VS as below
 1. IP address : 10.1.10.104
 2. ClientSSL profile. We will get a TLS warning in the browser, but it does not matter for this lab.

Virtual Server Properties

Advanced Settings

Virtual Server
 Create New Use Existing

Destination Address ⓘ

Service Port ⓘ

Enable Redirect Port ⓘ

Client SSL Profile ⓘ
 Create new Use Existing

Available

Filter

Common

clientssl-insecure-compatible

clientssl-quick

Create Profile in BIG-IP UI

Selected

Common

clientssl

2. Click Save & Next

Pool Properties

1. Select Create New
2. In Pool Servers, select /Common/10.1.20.9 This is the IIS server.

Pool Properties

Advanced Settings

Select a Pool
Create New

Select an existing pool or select Create New.

Resources Properties

Load Balancing Method
Round Robin

Specifies the load balancing method. The default is Round Robin.

Pool Servers
Select servers for the pool.

IP Address/Node name	Port	Priority Group	Action
<input type="text" value="Select..."/>	80	HTTP	0
--Select--			
/Common/10.1.20.9			

Session Management Properties

1. Nothing to change, click Save & Next

Deploy your app template

1. Click Deploy

Access >> Guided Configuration

Azure AD Application Configuration :IS-Bluesky-Matt NOT DEPLOYED

Configuration Properties Service Provider Azure Active Directory Virtual Server Pool Session Management Summary

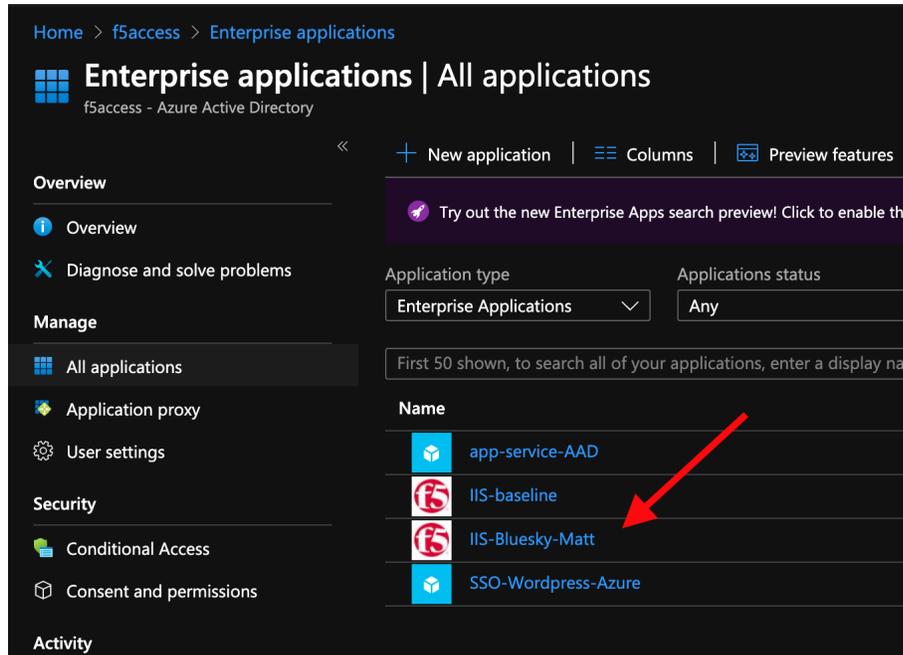
Your application is ready to be deployed.
The application is correctly configured, and ready to be deployed. Review the summary. You can click on any step to make changes.

Summary

- Configuration Properties
- Service Provider
- Azure Active Directory
- Virtual Server
- Pool
- Session Management

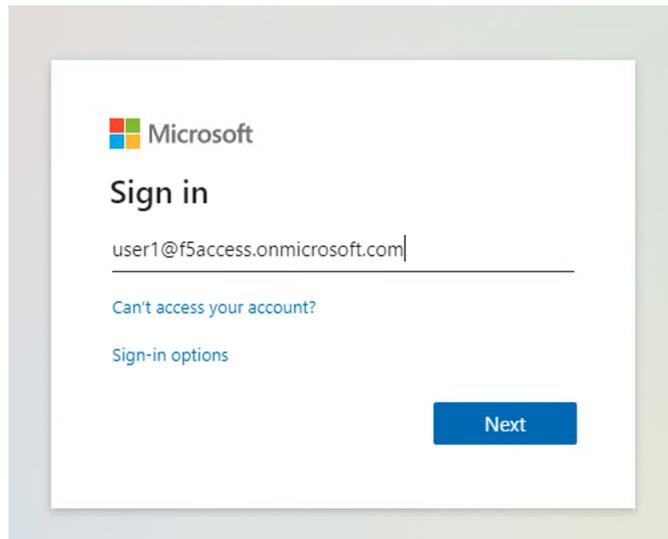
Cancel Save Draft Back Deploy

2. Behind the scene, the deployment creates an Azure Enterprise Application for Bluesky. We can see it in Azure portal (you don't have access in this lab). With this Enterprise Application, Azure knows where to redirect the user when authenticated. And this app has the certificate and key used to sign the SAML assertion.

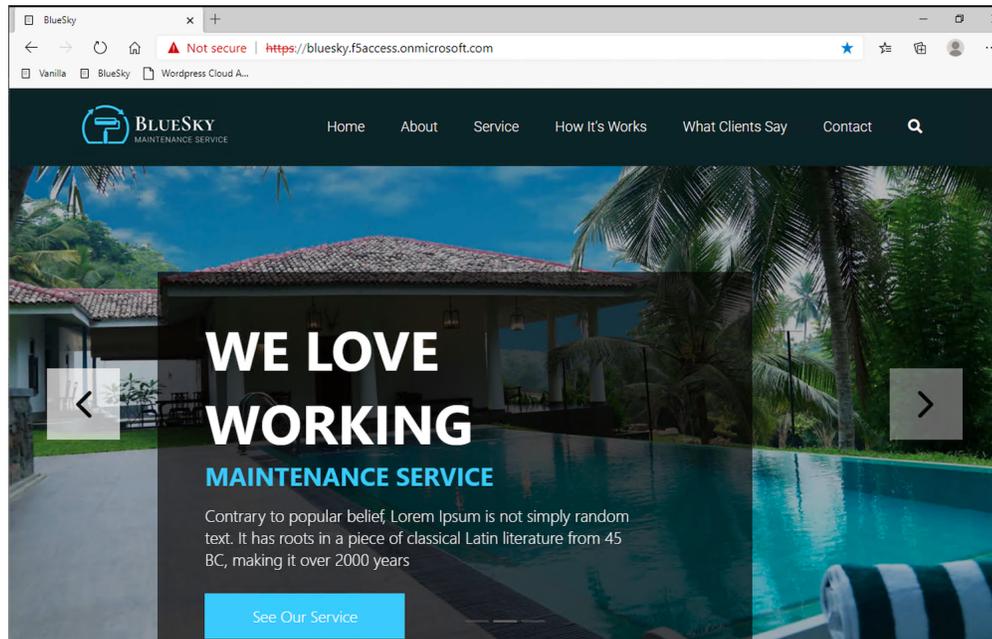


Test your deployment

1. RDP to Win10 machine as user and password user
2. Open Microsoft Edge browser - icon is on the Desktop
3. Click on the bookmark Bluesky
4. You will be redirected to Azure AD login page. Login as `user1@f5access.onmicrosoft.com`, and for the password please ask to your instructor.



5. You are redirected to APM with a SAML assertion, and can access to Bluesky application

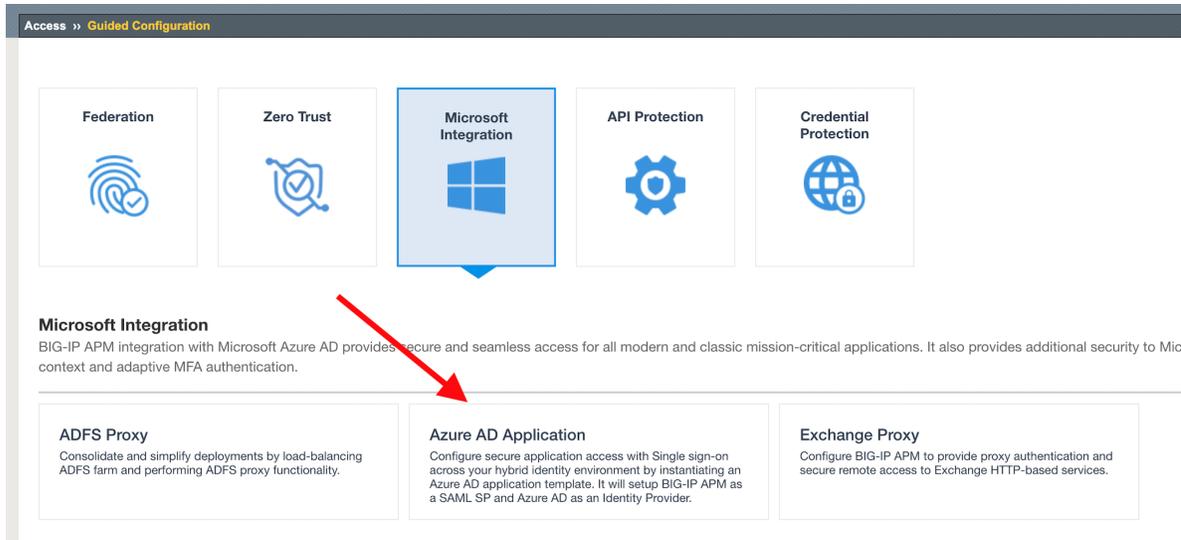


1.2.2 Publish and protect Vanilla app

Let's continue with Vanilla application. Reminder, Vanilla application as Authentication enabled with Kerberos auth. So, we will need to enable Kerberos Constrained Delegation.

1. Connect to BIG-IP HTTPS user interface from UDF as admin and password admin
2. In Access > Guided Configuration, select Microsoft Integration > Azure AD application

Note: As you can notice, we deploy one template per application



Configuration Properties

1. Click **Next** and start the configuration
2. Configure the page as below
 1. Configuration Name : IIS-Vanilla-<My Name> Why my name ? Because this app will be created in Azure AD tenant. And we need to differentiate all apps.
 2. Enable Single Sign-on (SSO)

General Properties ▾

Configuration Name

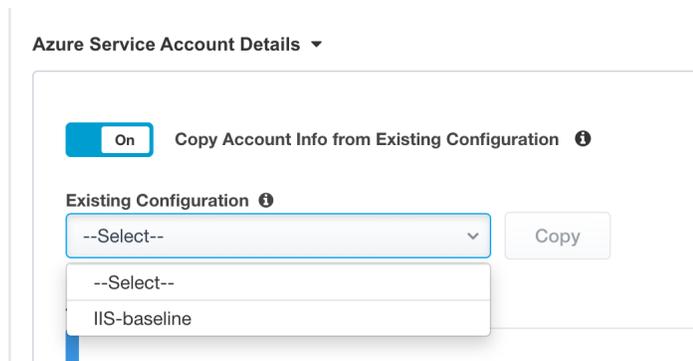
Type a name for this guided configuration.

Description ⓘ

On Single Sign-On (SSO) ⓘ

Endpoint Checks ⓘ

3. In Azure Service Account Details, Select Copy Account Info form Existing Configuration, and select IIS-baseline, then click Copy



Note: In a real world, you will set here the values from the Azure Service Application created for APM. You have to create an Azure Application so that APM get access to Microsoft Graph API. But for **security concerns**, I can't show in this lab the application secret.

Note: The steps to create this Azure applications are below

1. In Azure AD, create a service application under your organization's tenant directory using App Registration.
 2. Register the App as Azure AD only single-tenant.
 3. Request permissions for Microsoft Graph APIs and assign the following permissions to the application:
 1. Application.ReadWrite.All
 2. Application.ReadWrite.OwnedBy
 3. Directory.Read.All
 4. Group.Read.All
 5. Policy.Read.All
 6. Policy.ReadWrite.ApplicationConfiguration
 7. User.Read.All
 4. Grant admin consent for your organization's directory.
 5. Copy the Client ID, Client Secret, and Tenant ID and add them to the Azure AD Application configuration.
-
4. Click `Test Connection` button -> Connection is valid

Azure Service Account Details ▾

On Copy Account Info from Existing Configuration ⓘ

Existing Configuration ⓘ

IIS-baseline ▾ Copy

Tenant ID ⓘ

8807dced-9637-4205-a520-423077750c60

Client ID ⓘ

1ef1f16f-0472-4d3e-9062-138030a5c41d

Client Secret ⓘ

....

Test Connection ✔ Connection is valid

5. Click Next

Service Provider

1. Configure the page as below

1. Host `vanilla.f5access.onmicrosoft.com`
2. Entity ID is auto-filled `https://vanilla.f5access.onmicrosoft.com/IIS-Bluesky-myname>`

Service Provider

Advanced Settings

Service Provider Properties ▾

Host ⓘ

vanilla.f5access.onmicrosoft.com

Entity ID ⓘ

https://vanilla.f5access.onmicrosoft.com/IIS-Vanilla-Matt

Description ⓘ

Relay State ⓘ

3. Click Save & Next

Azure Active Directory

1. Select Azure BIG-IP APM Azure AD... template

Note: As you can notice, there are several templates available for different applications. Here, in this lab, we will publish a generic app. So we select the first template.

2. Click Add
3. In the new screen, configure as below.
 1. Signing Key : default.key
 2. Signing Certificate : default.crt
 3. Signing Key Passphrase : F5twister\$

SAML Signing Certificate ▾

Signing Key ⓘ
default.key ▾ ↻

Signing Certificate ⓘ
default.crt ▾ ↻

Signing Key Passphrase ⓘ
.....|

Signing Option ⓘ
Sign SAML assertion ▾

Signing Algorithm ⓘ
RSA-SHA256 ▾

4. In User And User Groups, click Add

Note: We have to assign Azure AD users/group to this app, so that they can be allowed to connect to it.

1. In the list, click Add for the user user1. If you can't find it, search for it in the search field.



2. Click Close
3. You can see user1 in the list.



4. Click Save & Next

Virtual Server Properties

1. Configure the VS as below
 1. IP address : 10.1.10.103
 2. ClientSSL profile. We will get a TLS warning in the browser, but it does not matter for this lab.

Virtual Server Properties

Advanced Settings

Virtual Server

Create New Use Existing

Destination Address

10.1.10.103

Service Port

443 HTTPS 

Enable Redirect Port 

Client SSL Profile

Create new Use Existing

Available

Filter 

Common

- clientssl-insecure-compatible
- clientssl-quick

[Create Profile in BIG-IP UI](#) 

Selected

- Common**
- clientssl

[Cancel](#)

[Save Draft](#)

[Back](#)

[Save & Next](#)

2. Click Save & Next

Pool Properties

1. Select Create New
2. In Pool Servers, select /Common/10.1.20.9 This is the IIS server.

Pool Properties

Advanced Settings

Select a Pool

Select an existing pool or select Create New.

Resources Properties

Load Balancing Method

Specifies the load balancing method. The default is Round Robin.

Pool Servers
 Select servers for the pool.

IP Address/Node name	Port	Priority Group	Action
<input type="text" value="Select..."/>	<input type="text" value="80"/>	<input type="text" value="HTTP"/>	<input type="button" value="+"/> <input type="button" value="x"/>
--Select--			
/Common/10.1.20.9			

Single Sign-On Settings

1. In Selected Single Sign-on Type, select Kerberos, and select Advanced Settings

Single Sign-On Settings

Advanced Settings On

Selected Single Sign-On Type

Select the authentication type from the list.

Credentials Source

Username Source ⓘ

User Realm Source ⓘ

2. In Credentials Source, fill as below
 1. Username Source : session.saml.last.identity
 2. Delete User Realm Source value - keep it empty. The domain is similar between Azure AD and on-prems AD.
3. In SSO Method Configuration, fill as below
 1. Kerberos Realm : f5access.onmicrosoft.com

2. Account name : `host/apm-deleg.f5access.onmicrosoft.com`
3. Account Password : `F5twister$`
4. KDC : `10.1.20.8`
5. UPN Support : Enabled
6. SPN Pattern : `HTTP/%s@f5access.onmicrosoft.com`

SSO Method Configuration ▾

Kerberos Realm ⓘ

Account Name ⓘ

Account Password

The password for the delegation account specified in the previous field.

Confirm Account Password

Re-type the password for the delegation account specified in the previous field.

KDC ⓘ

UPN Support

Enable this to allow the User Principal Name to be used for SSO.

SPN Pattern ⓘ

Ticket Lifetime ⓘ

Send Authorization ⓘ

[Cancel](#)

[Save Draft](#)

[Back](#)

[Save & Next](#)

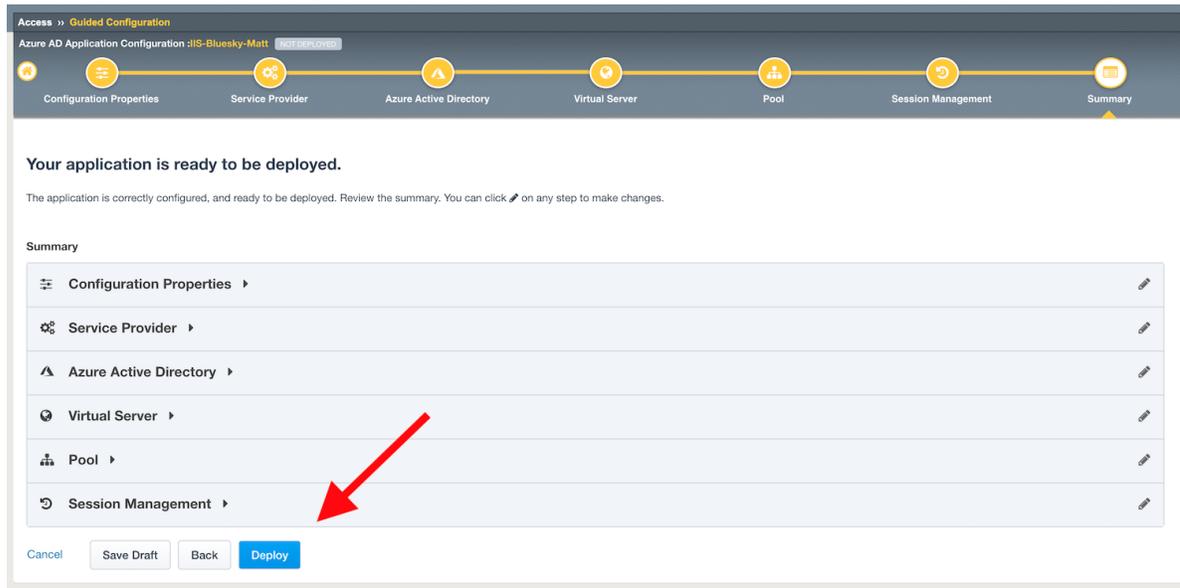
4. Click Save & Next

Session Management Properties

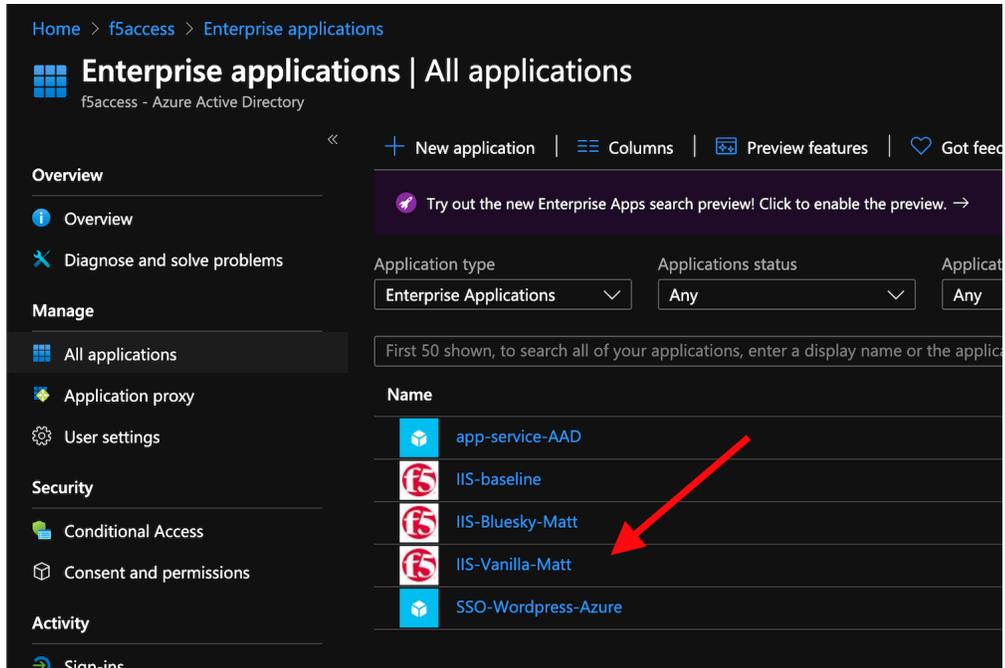
1. Nothing to change, click Save & Next

Deploy your app template

1. Click Deploy

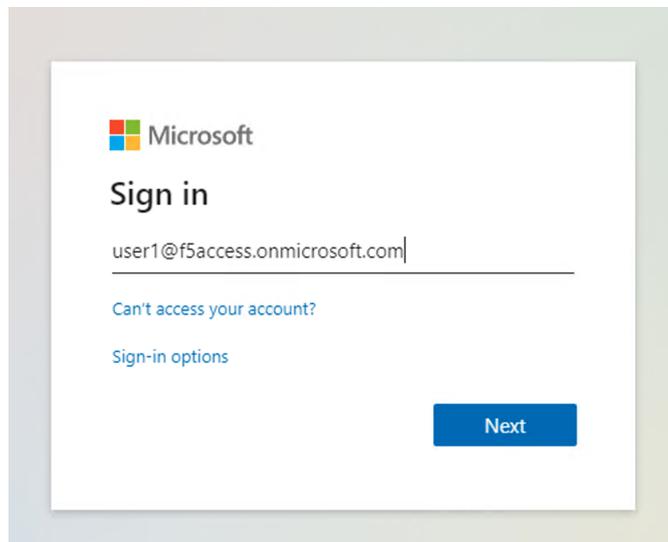


2. Behind the scene, the deployment creates an Azure Enterprise Application for Bluesky. We can see it in Azure portal (you don't have access in this lab). With this Enterprise Application, Azure knows where to redirect you when authenticated. And this app has the certificate and key used to sign the SAML assertion.

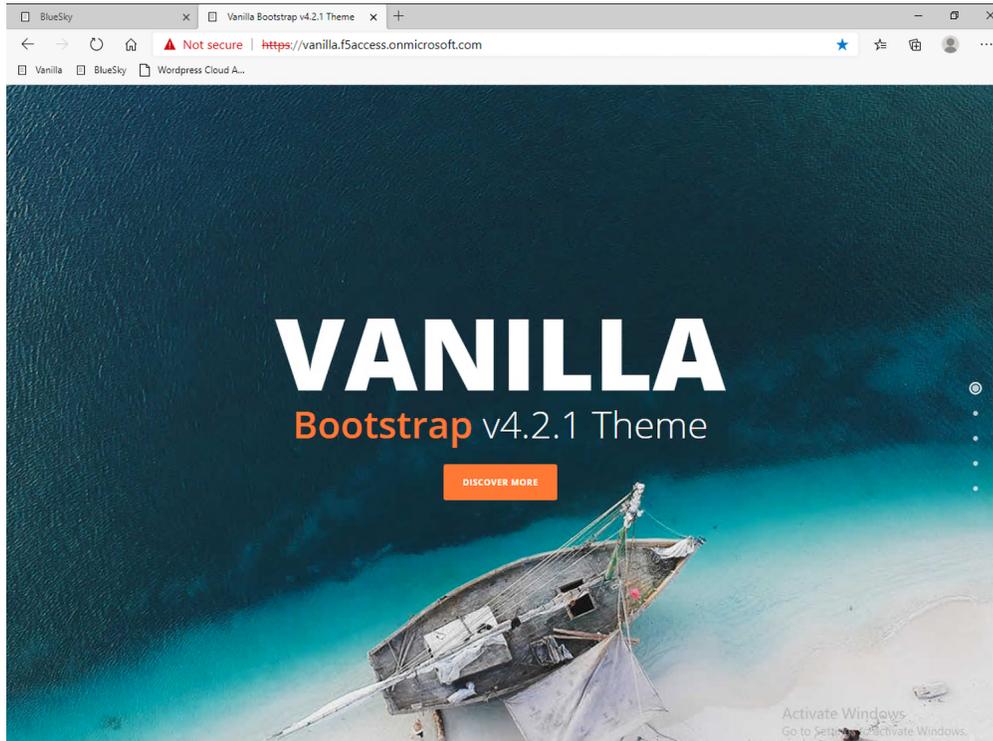


Test your deployment

1. RDP to Win10 machine as user and password user
2. Open Microsoft Edge browser - icon is on the Desktop
3. Click on the bookmark Vanilla
4. You will be redirected to Azure AD login page - only if your previous session with Bluesky expired in APM. Login as user1@f5access.onmicrosoft.com, and for the password please ask to your instructor (if you are prompted). But as you already authenticated against Azure AD, you still have a session in Azure AD.



5. You are redirected to APM with a SAML assertion, and can access to Vanilla application.
6. APM did Single Sign-on with Vanilla application (Kerberos Constrained Delegation)



7. Click Bluesky bookmark, you can access Bluesky application as well.
8. Extra lab, enable Inspect mode in Edge, and follow the SAML redirections to understand the workflow.

1.3 Class 3 - Leverage Azure AD to protect Cloud Apps

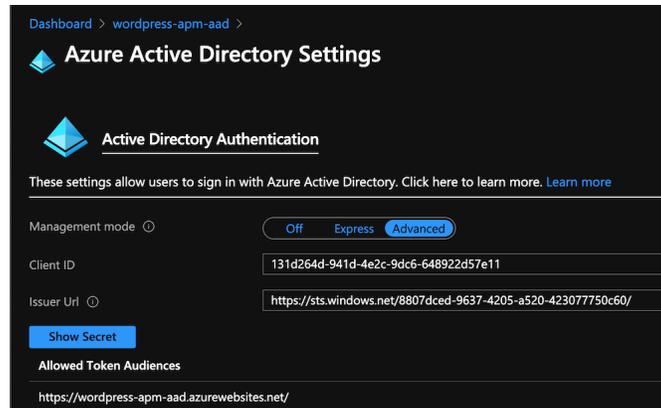
In this class, we will check that `user1` can access any cloud app federated with Azure AD.

1.3.1 The current config

In a real world, companies deploy applications on-prems and in public clouds. If the company uses **Azure AD as IDaaS**, it will federate all cloud apps with this Azure AD tenant.

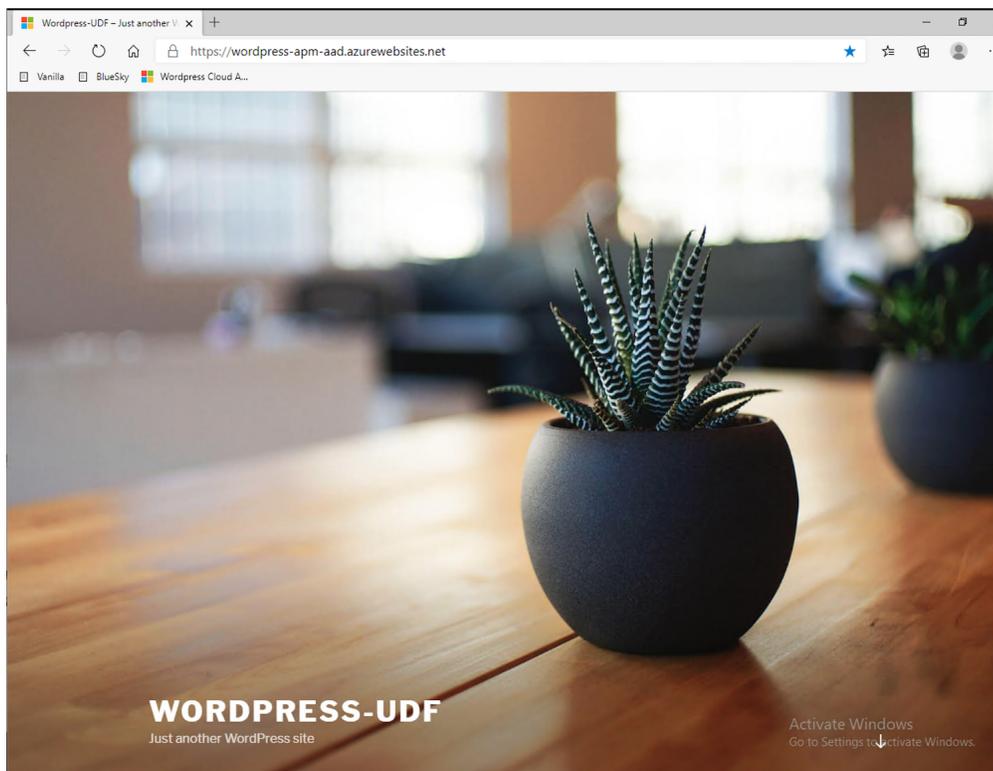
This is what we prepared for you in this lab. This application is **federated** with our Azure AD tenant.

You have **nothing** to configure on APM side, as everything is dealt between the cloud app and Azure AD. In Azure portal, we configured Oauth for the cloud app, so that every user reaching this app will be redirected to Azure login page.



1.3.2 Test your deployment

1. RDP to Win10 machine as user and password user
2. Open Microsoft Edge browser - icon is on the Desktop
3. Click on the bookmark Wordpress Cloud App
4. You will be redirected to Azure AD login page (it can take a while - look at the address bar). Login as `user1@f5access.onmicrosoft.com`, and for the password please ask to your instructor (if prompted). You already have a session up and running in Azure AD, from previous class.
5. You are redirected to the `cloud app` in Azure cloud, and can access to Wordpress-UDF application.



1.4 Class 4 - Enable MFA

Warning: You can **not** run this class without a F5 SA or F5 SME-UA lead. Please reach out to your local SA/SME-UA lead in order to activate an temporary account for you.

1. EMEA : Matthieu
2. USA : Jason or Shannon
3. APCJ : Shain

In this class, we will use another user account (created by SA/SME-UA), with MFA enabled for this account.

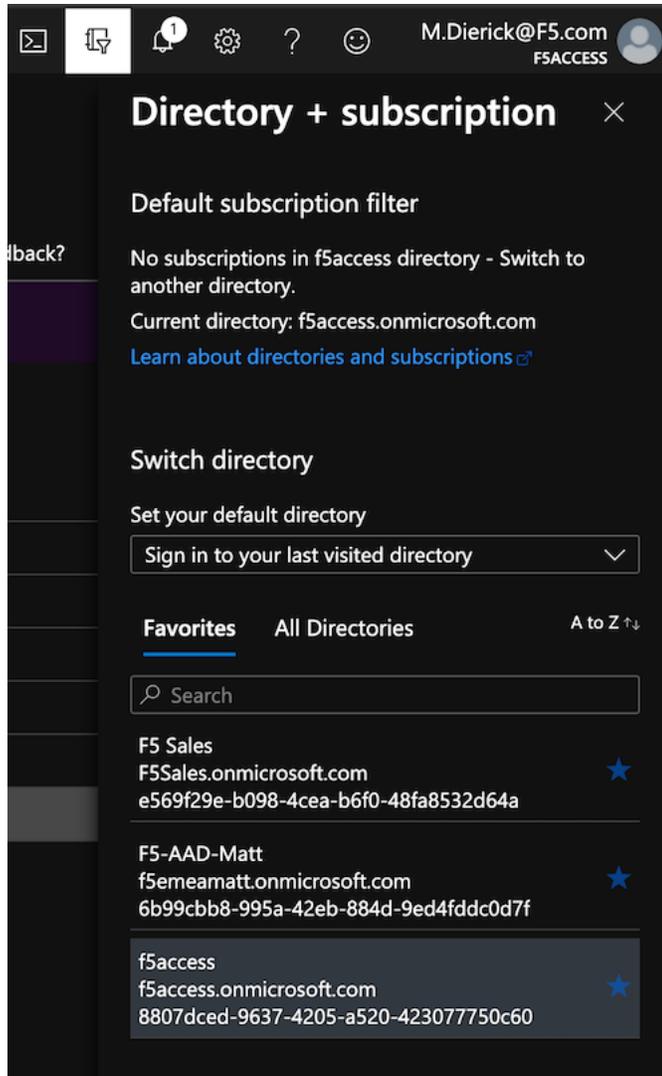
Let's say, a SA/SME created the account `matt@f5access.onmicrosoft.com` for me. Then he enabled the MFA for this account.

Class 4 - All sections

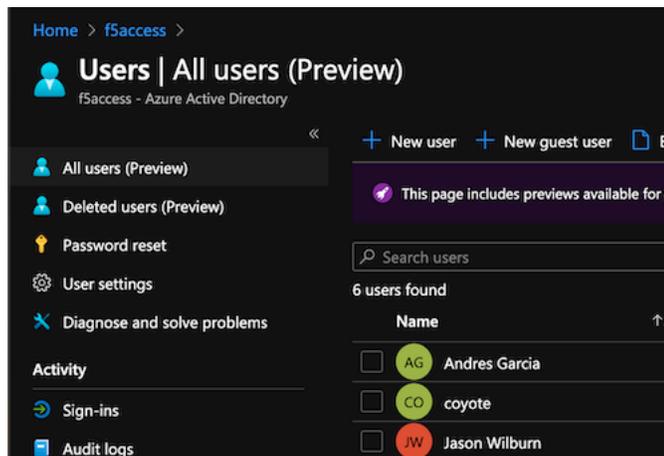
1.4.1 Procedure for SA/SME

Warning: Only SA and SME-UA with administrator role on this demo tenant, can create users. If you are not a SA or SME lead, move to the next section.

1. Connect to Azure Portal and select F5access tenant



2. Go to Azure Active Directory > Users
3. Click Create new



4. Enter the information, and click Create

Identity

User name * ⓘ ✓ @ ✓
The domain name I need isn't shown here

Name * ⓘ ✓

First name

Last name

Password

Auto-generate password

Let me create the password

Initial password * ⓘ ✓

Groups and roles

Groups 0 groups selected

Roles User

5. Click on Multi-Factor Authentication

Microsoft Azure Search resources, services, and docs (G+)

Home > f5access > **Users | All users (Preview)**
f5access - Azure Active Directory

+ New user + New guest user Bulk operations Refresh Reset password **Multi-Factor Authentication** Delete user

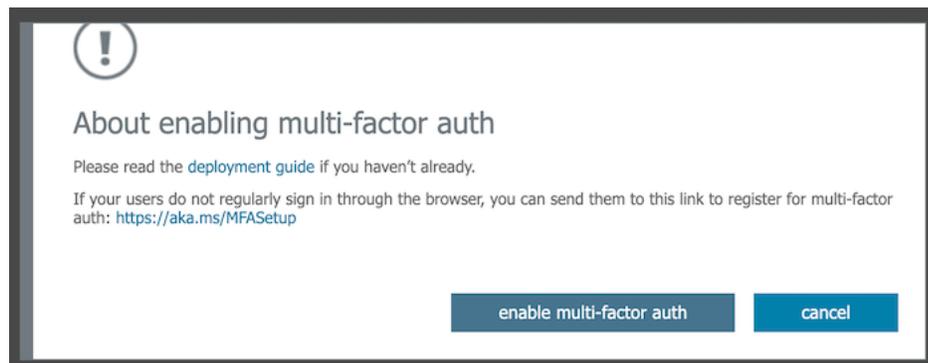
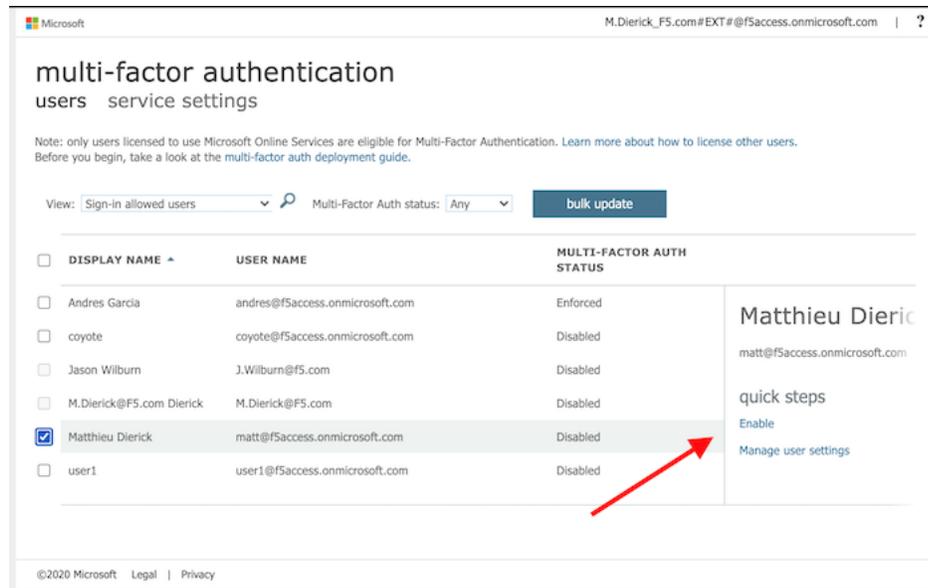
This page includes previews available for your evaluation. View previews →

Search users Add filters

7 users found

	Name	User principal name	User type	Directory synced
<input type="checkbox"/>	AG Andres Garcia	andres@f5access.onmicrosoft.com	Member	No
<input type="checkbox"/>	CO coyote	coyote@f5access.onmicrosoft.com	Member	No

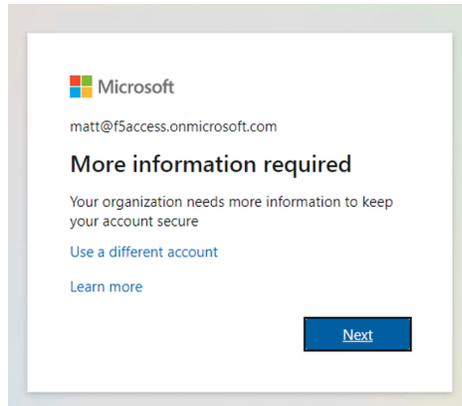
6. Enable MFA for the created user



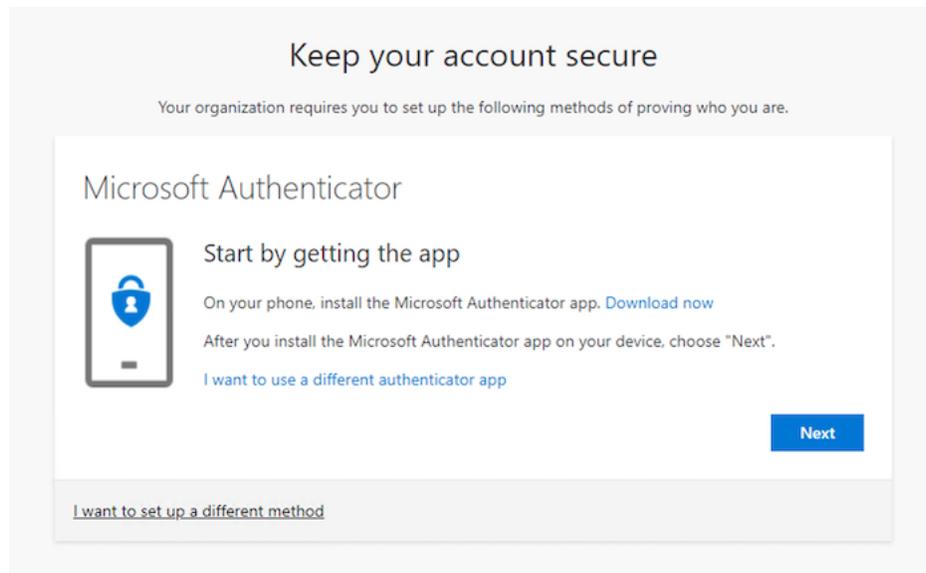
1.4.2 Test your deployment with MFA enabled

Warning: You should have received an email or teams chat from your SA/SME to continue.

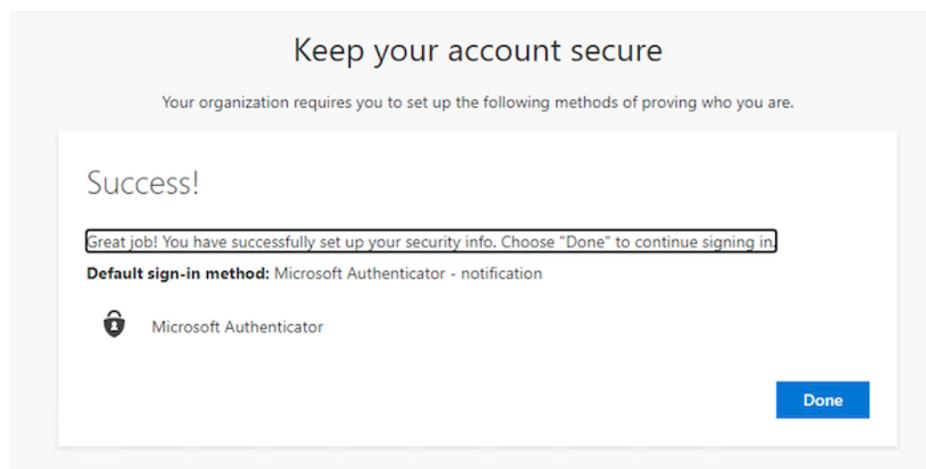
1. Close any opened browser and re-open Microsoft Edge
2. Connect to Bluesky. Don't try with Vanilla as your MFA test account does not exist in on-prem AAD. Thus, the SSo will not work. You can add this user in ADDS if you are confident with AD.
3. If you are not prompted at Azure AD login page, open an incognito window. It means you still have Azure AD cookies from previous session with ``user1`` account.
4. At prompt, login with your MFA account. In my case, `matt@f5access.onmicrosoft.com` and the password provided by your SA/SME
5. You will be asked to enroll and select an MFA method



6. Click Next
7. You have the choice to use the Microsoft Authenticator mobile app, or use SMS. Make your choice and follow the step to enroll your device (or phone number)



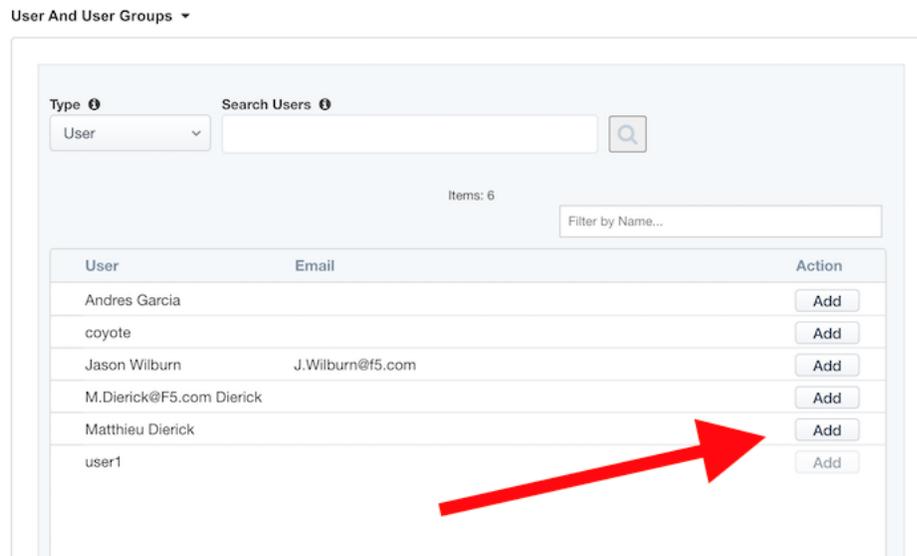
8. I select the mobile app, scan the QR code, and approve the push notification on my mobile phone.
9. I click Next and Done



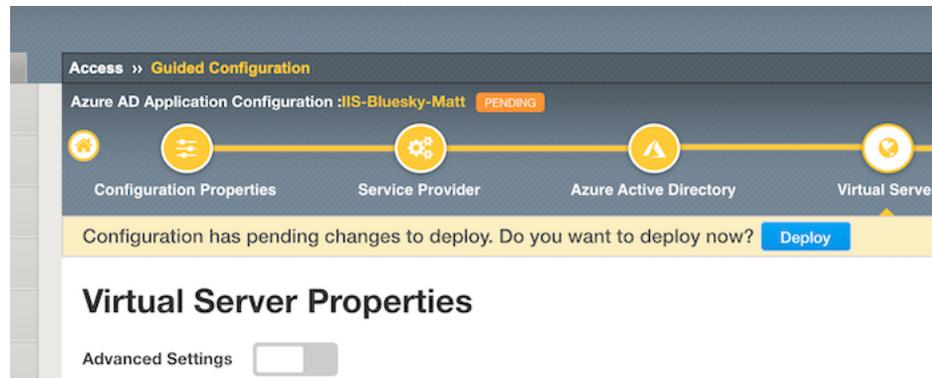
- Azure AD asks you to change your password set by your SA/SME.
- When done, and redirected to Bluesky, you can notice **it does not work**. The user has to be assigned with the Bluesky app.



- In the BIG-IP, edit the ISS-Bluesky-<my name> template, and in Azure Active Directory step, add your account.



- Click Save and Next and Deploy



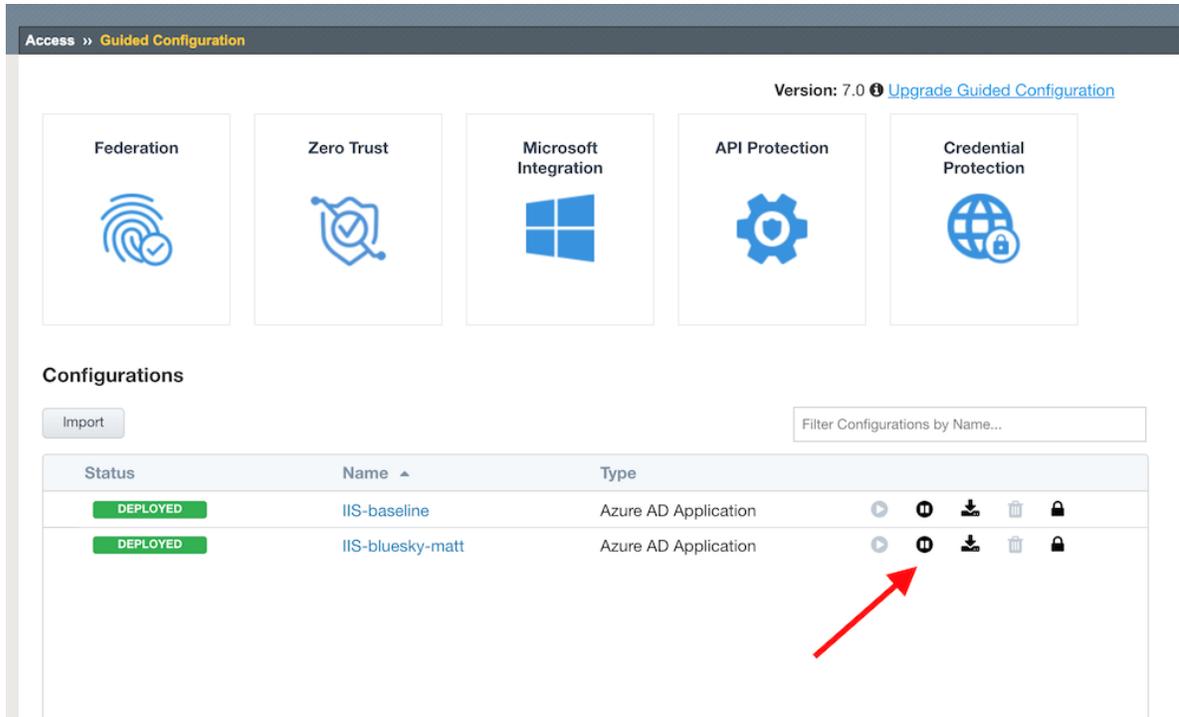
14. Make a new test, approve the push notification or enter the OTP received by SMS.

Note: This lab is not **Azure AD Conditional Access**. This is just **user MFA**. Conditional Access is similar but it is tied to a policy (group, location, app ...). In this lab, matt will be prompted for MFA whatever the apps he connects to.

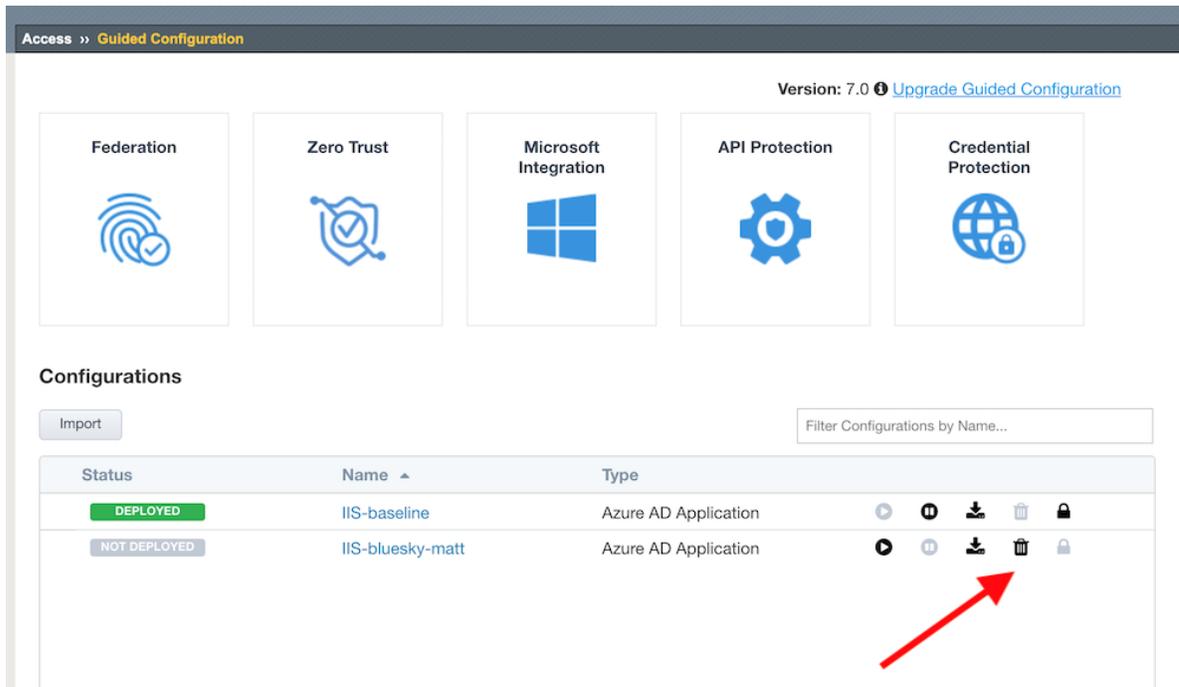
1.5 Class 5 - Clean up the lab

Warning: In order to keep the Azure AD tenant clean, it is important you delete your application in Guided Configuration, when your demo is finished.

1. In Guided Configuration menu, click on the Undeploy icon, then OK



2. When finished, click on Delete icon



Note: Thanks a lot, you cleaned up your config on both sides (APM and AAD). FYI, all old deployments will be

deleted automatically in Azure AD.
