# Module 9: Managing Security & Identity for Azure Solutions

## Contents:



## Module overview

This module discusses both security and identity within the context of Azure. For security, this module reviews the various options for monitoring security, the options available for securing data and the options for securing application secrets. For identity, this module focuses specifically on Azure Active Directory (Azure AD) and the various features available such as Multi-Factor Authentication (MFA), Managed Service Identity, Azure AD Connect, ADFS and Azure AD B2B/B2C.

### Objectives

After completing this module, students will be able to:

- Integrate their existing solutions with external identity providers using Azure AD B2B or B2C.
- Design a hybrid identity solution.
- Determine when to use advanced features of Azure AD such as Managed Service Identity, MFA and Privileged Identity Management.
- Secure application secrets using Key Vault.
- Secure application data using SQL Database and Azure Storage features.

## Lesson 1: Security

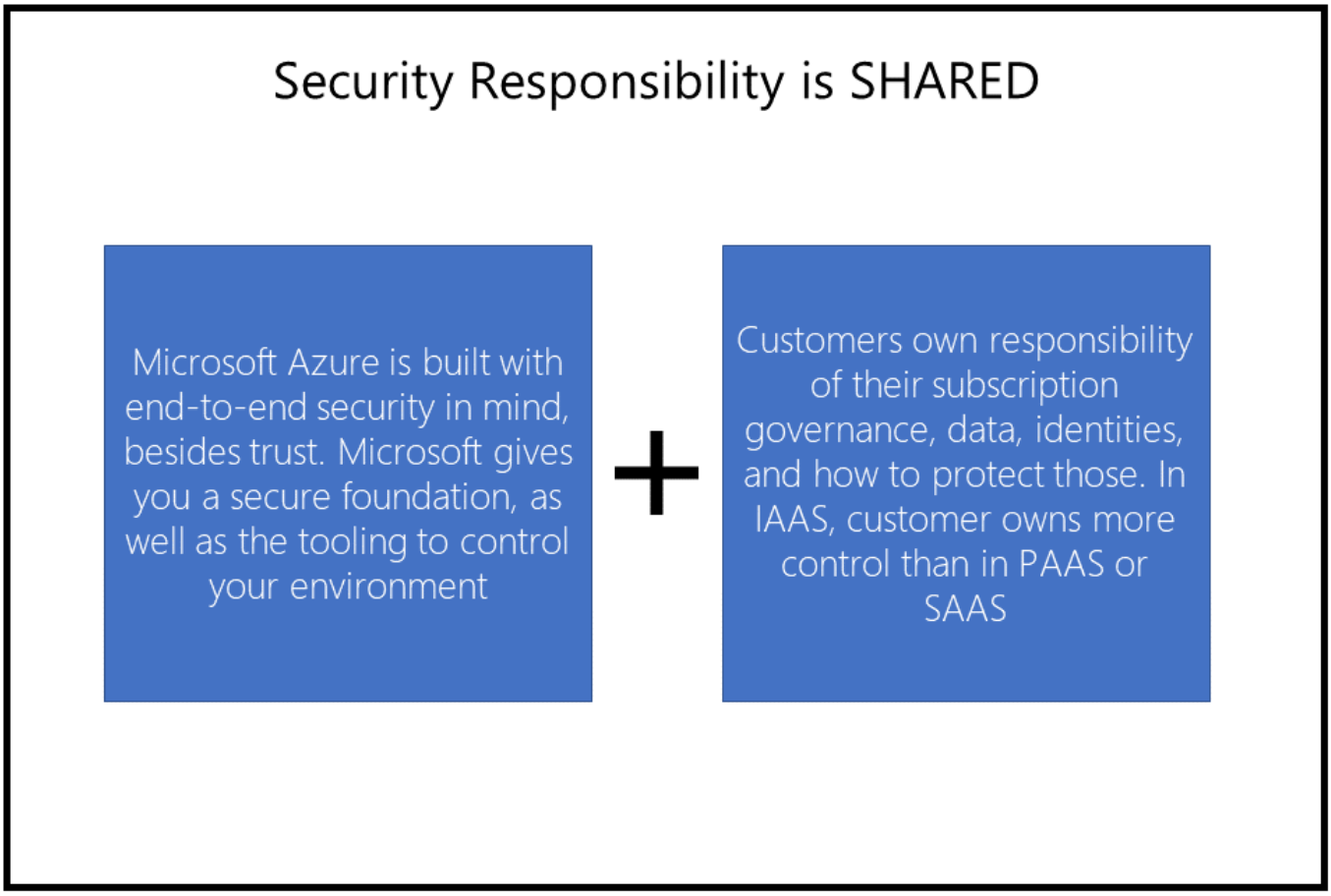
In this lesson, you learn how the Azure Platform is built with security in mind, followed by the sharing of responsibility with both Microsoft and you as a customer for security workloads hosted in Azure.

### Lesson objectives

After completing this lesson, you will be able to:

- Understand “shared responsibility” in a cloud model
- Explain the overall Azure Platform end-to-end security aspects
- Explain how Azure is offering and handling encryption on different levels
- Identify how different Azure services and resources are dealing with security, like Azure Networking, Azure Key Vault, Azure SQL, Azure Storage accounts and more

## Platform Security




Azure Security is a “difficult” topic, since it’s the core existence of the platform design and architecture. However, at the same time, a lot of organizations are hesitant from using Public Cloud services like Microsoft Azure, because they don’t think it is providing decent security.

When an organization starts using Azure, responsibility for securing workloads is shared.

- **Microsoft Azure** is built with end-to-end security in mind, besides trust. Microsoft gives you a secure foundation, as well as the tooling to control your environment
- **Customers own responsibility** of their subscription governance, data, identities, and how to protect those. In IAAS, customer owns more control than in PAAS or SAAS

Security controls are designed to ensure technology solutions are built and maintained in ways that ensure function and security successfully coexist. This ideal holds strong in Azure where we are constantly vetting and monitoring the implementation of our security controls, as well as watching our service teams continue to innovate new functionality in the cloud environment. With that said, the cloud presents a spectrum of responsibilities based on what types of services and/or features a customer may be consuming. This is unlike more traditional on-premises information systems where most, if not all, security is implemented by the same owner.

Azure is architected for secure multi-tenancy. It's designed to abstract much of the infrastructure that typically underlies applications (servers, operating systems, Web and database software, and so on) so that customers can focus on building applications—and not on managing resources. The goal is to provide a secure, consistent, scalable set of resources for each customer that they can manage through an Azure subscription. The subscription is associated with a Microsoft account or organizational account.

Technical separation in the Azure datacenter is based on the following components:

- The Azure Fabric Controller (FC) functions as the kernel of the Azure platform, managing resources as needed. The FC provisions, stores, delivers, monitors and commands the VMs and physical servers that make up the Azure customer environment and infrastructure.
- The Host OS is a configuration-hardened version of Windows Server
- The Hypervisor is Hyper-V from Windows Server 2012 R2, which has been battle-tested and proven in enterprise environments worldwide
- The Guest VM OS can be either Windows Server, several distributions of Linux, or an OS image supplied by the customer (much be supported Operating Systems, or starting from the Azure Marketplace images

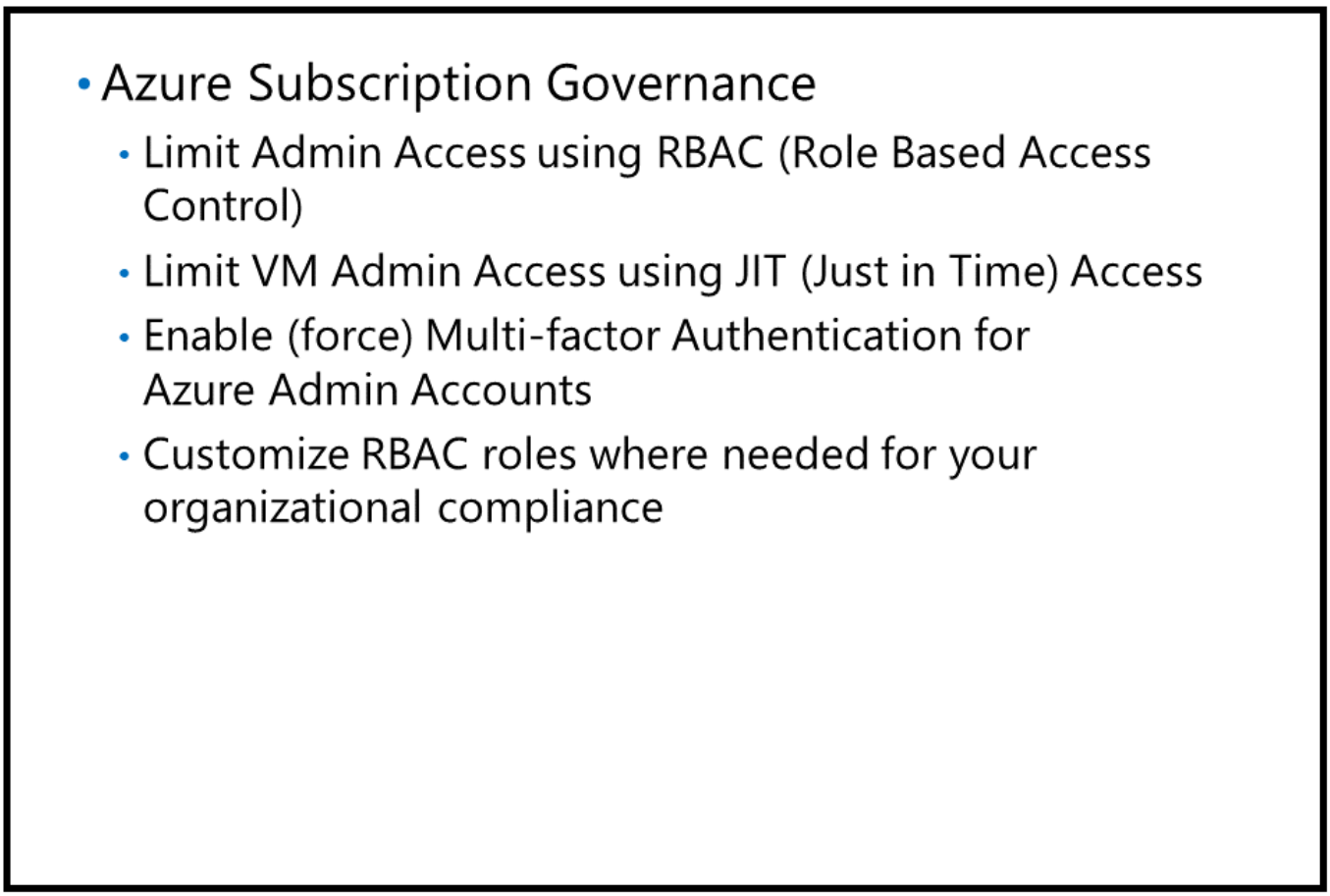
From an application & data perspective, the situation looks like this:

Microsoft Azure uses logical isolation to segregate each customer's data from that of others. This provides the scale and economic benefits of multitenant services while rigorously preventing customers from accessing one another's data.

- **Storage isolation**
- Data is accessible only through claims-based Identity Management & access control with a Storage Access Key (SAK). Shared Access Signature (SAS) tokens can be generated using storage access keys to provide more granular, restricted access. Storage access keys can be reset via the Microsoft Azure Portal or the Storage Management API.
- Storage blocks are hashed by the hypervisor to separate accounts
- **SQL isolation**
- SQL Azure isolates separate account databases

- **Network isolation**
- VM switch at the host level blocks inter-tenant communication

## Securing the Azure Platform




**Azure Key Vault**

Azure Key Vault helps safeguard cryptographic keys and secrets used by cloud applications and services. By using Key Vault, you can encrypt keys and secrets (such as authentication keys, storage account keys, data encryption keys, .PFX files, and passwords) by using keys that are protected by hardware security modules (HSMs). For added assurance, you can import or generate keys in HSMs. If you choose to do this, Microsoft processes your keys in FIPS 140-2 Level 2 validated HSMs (hardware and firmware).

Key Vault streamlines the key management process and enables you to maintain control of keys that access and encrypt your data. Developers can create keys for development and testing in minutes, and then seamlessly migrate them to production keys. Security administrators can grant (and revoke) permission to keys, as needed.

Anybody with an Azure subscription can create and use key vaults. Although Key Vault benefits developers and security administrators, it could be implemented and managed by an organization's administrator who manages other Azure services for an organization. For example, this administrator would sign in with an Azure subscription, create a vault for the organization in which to store keys, and then be responsible for operational tasks, such as:

- Create or import a key or secret
- Revoke or delete a key or secret

- Authorize users or applications to access the key vault, so they can then manage or use its keys and secrets
- Configure key usage (for example, sign or encrypt)
- Monitor key usage

This administrator would then provide developers with URIs to call from their applications, and provide their security administrator with key usage logging information.

# Lesson 2: Identity

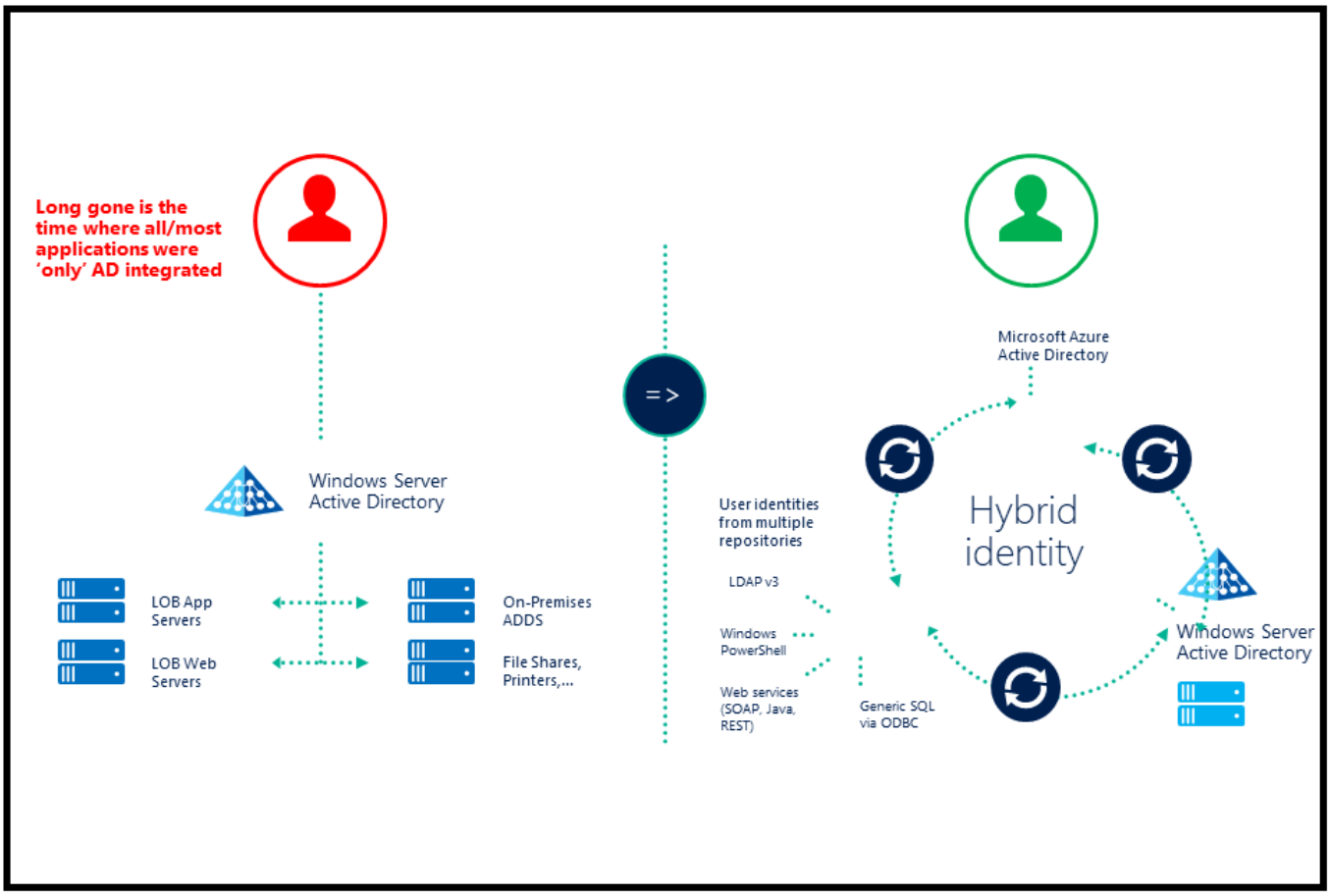
This lesson explores Azure Active Directory in the context of advanced identity architectures and solutions.

## Lesson objectives

After completing this lesson, you will be able to:

- Understand Azure Active Directory as a directory service
- Understanding different topologies to enable Hybrid Identity
- What is Azure AD Connect, and how to implement and use it
- Enabling Azure Active Directory Seamless and Single Sign-On
- What is Azure Active Directory Application Proxy
- Concepts of Azure Active Directory B2B and B2C
- Azure Active Directory MFA (multi factor authentication)
- Using Azure Active Directory for advanced Identity Protection and Privileged Management
- Use case for Azure Active Directory Domain Services

## Azure Active Directory




Azure Active Directory (Azure AD) is Microsoft's multi-tenant, cloud based directory and identity management service. Azure AD combines core directory services, advanced identity governance, and application access management.

For IT Admins, Azure AD provides an affordable, easy to use solution to give employees and business partners single sign-on (SSO) access to thousands of cloud SaaS Applications like Office365, Salesforce.com, DropBox, and Concur.

For application developers, Azure AD lets you focus on building your application by making it fast and simple to integrate with a world class identity management solution used by millions of organizations around the world.

Azure AD also includes a full suite of identity management capabilities including multi-factor authentication, device registration, self-service password management, self-service group management, privileged account management, role based access control, application usage monitoring, rich auditing and security monitoring and alerting. These capabilities can help secure cloud based applications, streamline IT processes, cut costs and help ensure that corporate compliance goals are met.

Where Active Directory has always been our "trusted" source of identities, long gone is the time where all/most applications were 'only' AD integrated. The single user account that could log on to all business applications, as long as they were AD integrated, are not like that anymore in the present world.

Today, a typical business user logs on to 17 applications on average per day, requiring 12 different user accounts/passwords. This is what we call the hybrid identity.

To protect these user accounts from threats, from getting compromised, an end-to-end security model must be put in place.

The "cloud" way of authenticating, is possible using any of the 3 scenarios:

1. Azure ADConnect using **Password Hash Sync**
2. Azure ADConnect using **Federation (ADFS)**
3. Azure ADConnect using **Azure AD Passthrough Authentication Agent**

**Single Sign-On**

Single sign-on, also called identity federation, is a hybrid-based directory integration scenario of Azure Active Directory that you can implement when you want to simplify your user's ability to seamlessly access cloud services, such as Office 365 or Microsoft Intune, with their existing Active Directory corporate credentials. Without single sign-on, your users would need to maintain separate user names and passwords for your online and on-premises accounts.

An Secure Token Service (STS) enables identity federation, extending the notion of centralized authentication, authorization, and SSO to Web applications and services located virtually anywhere, including perimeter networks, partner networks, and the cloud. When you configure an STS to provide single sign-on access with a Microsoft cloud service, you will be creating a federated trust between your on-premises STS and the federated domain you've specified in your Azure AD tenant.

There is a clear benefit to users when you implement single sign-on: it lets them use their corporate credentials to access the cloud service that your company has subscribed to. Users don't have to sign in again and remember multiple passwords.

## Azure AD Authentication Strategies

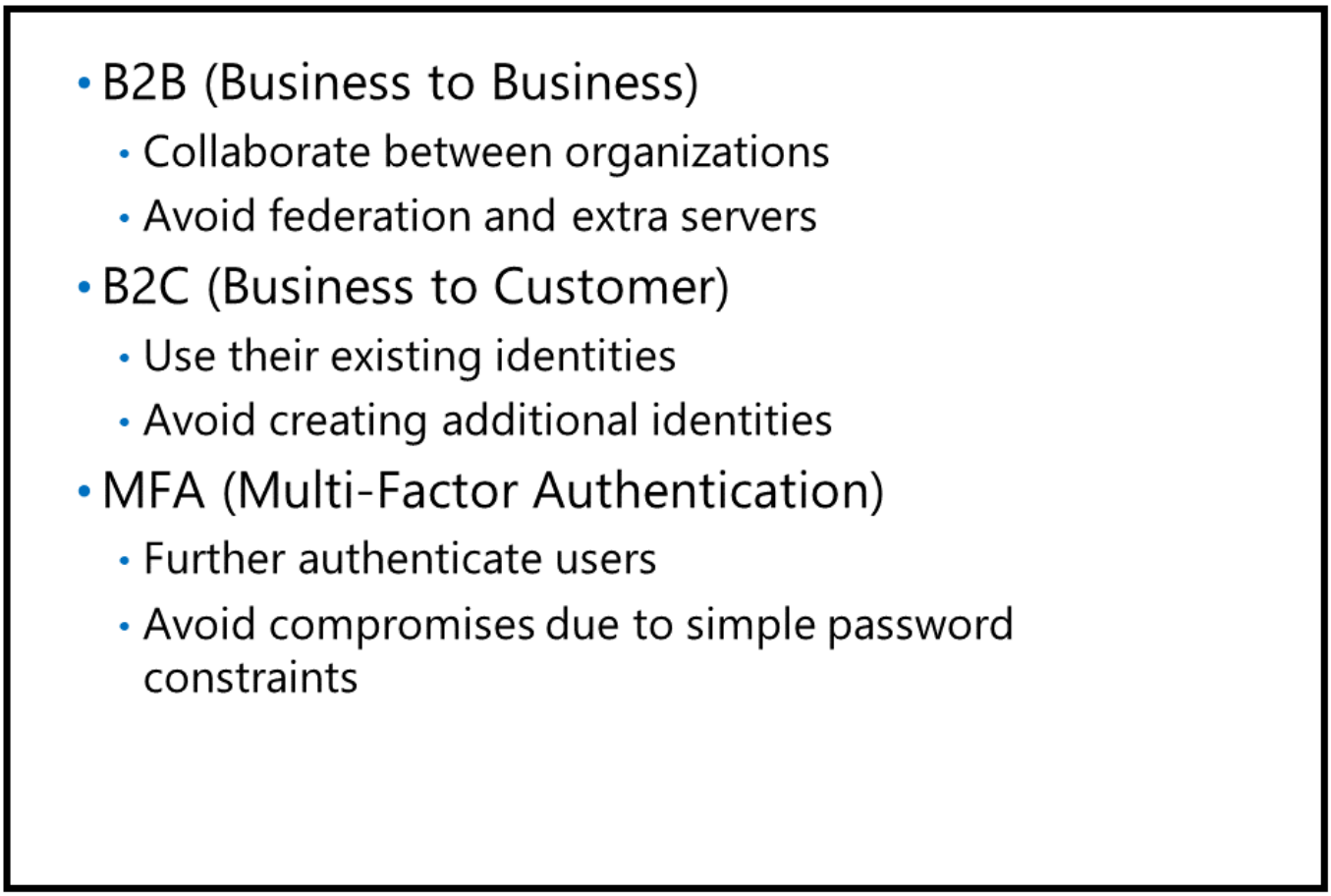
- In any of the "cloud" scenarios, AD Connect User/Group object sync is required
- Replaces legacy tools
  - DirSync, ADSync, FIM with AD Connector
- Benefits
  - Allows for write-back (passwords, devices, groups) to on-premises AD
  - Built-in deployment wizard for on-premises ADFS infrastructure
  - Azure AD Connect Synchronization Services dashboard
  - Managed user sign-in options

Regardless from what authentication mechanism your corporate organization is using, Azure AD Connect is always a required sync tool. Azure AD Connect supports synchronization from multiple Azure AD Forests/Domains, into a single Azure Active Directory environment.

Azure AD Connect can be installed on dedicated VMs, or directly on ADDS Domain Controllers (not recommended, but workable in an SMB environment). The underlying database that is used by AD Connect can be a SQL Server Express, or a full SQL Server 2008 R2 or newer database instance. Azure AD Connect requires an AD Connect Service Account. This account reads/write information from the Azure AD Tenant, as well as requiring an on-premises account in Active Directory, Enterprise Admin level rights, to read/write information back in the on-premises Active Directory.

Azure AD Connect allows for a two-way sync, e.g. password resets (optional – requires P1), account deletions and others strategies for connecting.

## Azure AD B2B & B2C




Both Azure Active Directory (Azure AD) B2B collaboration and Azure AD B2C allow you to work with external users in Azure AD. But how do they compare?

**Azure AD B2B**

- **Intended for**
  - Organizations that want to be able to authenticate users from a partner organization, regardless of identity provider.
- **Identities supported**
  -

Employees with work or school accounts, partners with work or school accounts, or any email address. Soon to support direct federation.

- **Which directory the partner users are in**
  - Partner users from the external organization are managed in the same directory as employees, but annotated specially. They can be managed the same way as employees, can be added to the same groups, and so on
  - **Single sign-on (SSO)**
  - Single sign-on to all Azure AD-connected apps is supported. For example, you can provide access to Office 365 or on-premises apps, and to other SaaS apps such as Salesforce or Workday.
- **Partner lifecycle**
  - Managed by the host/inviting organization.
- **Security policy and compliance**
  - Managed by the host/inviting organization.
- **Branding**
  - Host/inviting organization’s brand is used.

**Azure AD B2C**

  - **Intended for**
  - Inviting customers of your mobile and web apps, whether individuals, institutional or organizational customers into your Azure AD.
  - **Identities supported**
  - Consumer users with local application accounts (any email address or user name) or any supported social identity with direct federation.
  - **Which directory the customer user entities are in**
  - In the application directory. Managed separately from the organization’s employee and partner directory (if any.
  - **Single sign-on (SSO)**
  - Single sign-on to customer owned apps within the Azure AD B2C tenants is supported. SSO to Office 365 or to other Microsoft and non-Microsoft SaaS apps is

not supported.

- **Customer lifecycle**
- Self-serve or managed by the application.
- **Security policy and compliance**
- Managed by the application.

- **Branding**
  - Managed by application. Typically tends to be product branded, with the organization fading into the background.



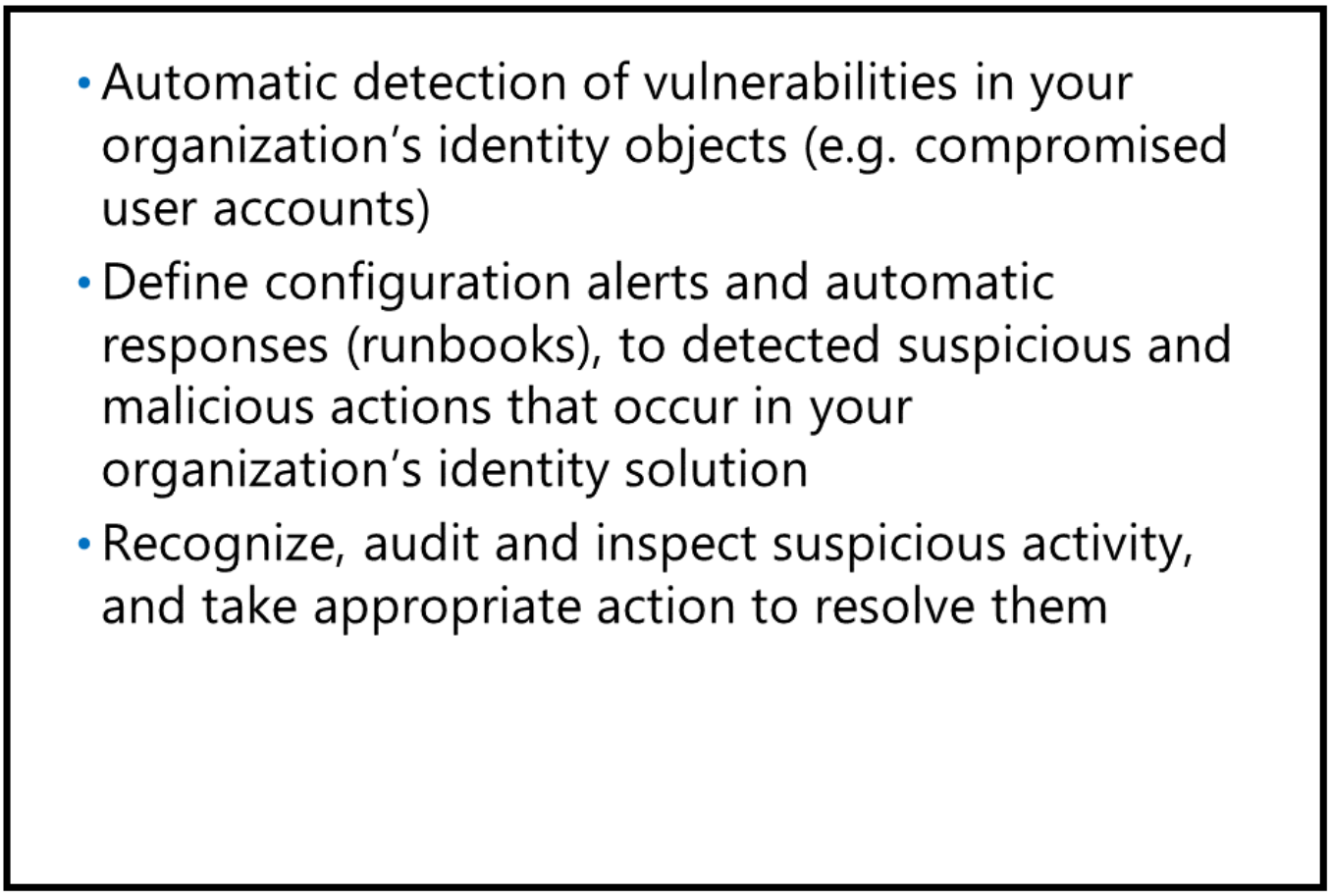
**Multi-Factor Authentication**

Two-step verification is a method of authentication that requires more than one verification method and adds a critical second layer of security to user sign-ins and transactions. It works by requiring any two or more of the following verification methods:

- Something you know (typically a password)
- Something you have (a trusted device that is not easily duplicated, like a phone)
- Something you are (biometrics)

Azure Multi-Factor Authentication (MFA) is Microsoft's two-step verification solution. Azure MFA helps safeguard access to data and applications while meeting user demand for a simple sign-in process. It delivers strong authentication via a range of verification methods, including phone call, text message, or mobile app verification.

## Azure AD Identity Protection




The vast majority of security breaches take place when attackers gain access to an environment by stealing a user's identity. Over the years, attackers have become increasingly effective in leveraging third party breaches and using sophisticated phishing attacks. As soon as an attacker gains access to even low privileged user accounts, it is relatively easy for them to gain access to important company resources through lateral movement.

As a consequence of this, you need to:

- Protect all identities regardless of their privilege level
- Proactively prevent compromised identities from being abused

Discovering compromised identities is no easy task. Azure Active Directory uses adaptive machine learning algorithms and heuristics to detect anomalies and suspicious incidents that indicate potentially compromised identities. Using this data, Identity Protection generates reports and alerts that enable you to evaluate the detected issues and take appropriate mitigation or remediation actions. Azure Active Directory Identity Protection is a feature of the Azure AD that enables you to:

- Detect potential vulnerabilities affecting your organization's identities
- Configure automated responses to detected suspicious actions that are related to your organization's identities
- Investigate suspicious incidents and take appropriate action to resolve them

**Privileged Identity Management**

Securing privileged access is a critical first step to help protect business assets in a modern organization. Privileged accounts are accounts that administer and manage IT systems. Cyber-attackers target these accounts to gain access to an organization's data and systems. To secure privileged access, you should isolate the accounts and systems from the risk of being exposed to a malicious user.

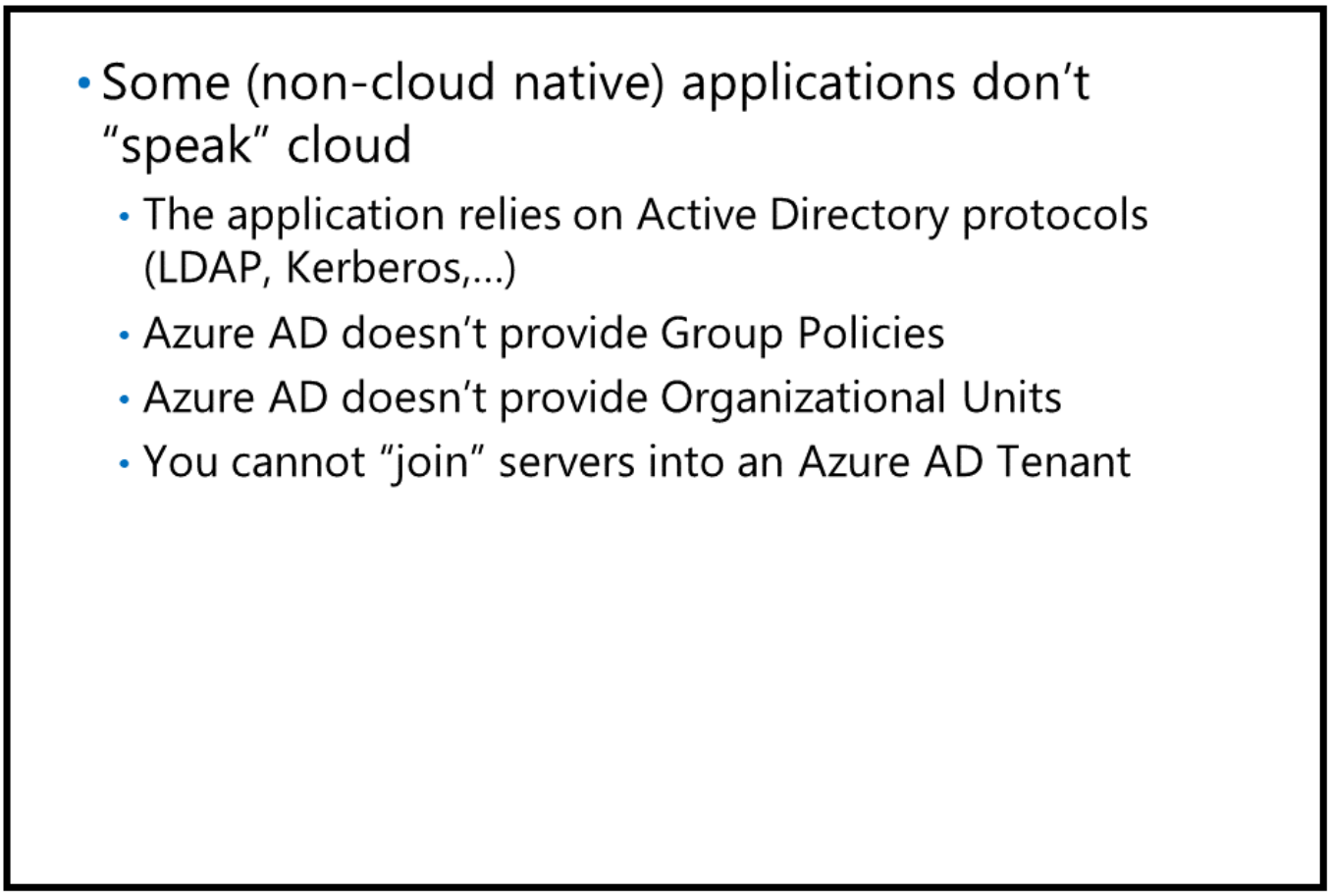
More users are starting to get privileged access through cloud services. This can include global administrators of Office365, Azure subscription administrators, and users who have administrative access in VMs or on SaaS apps.

Azure AD Privileged Identity Management helps to mitigate the risk of excessive, unnecessary or misused access rights. Azure AD Privileged Identity Management helps your organization:

- See which users are assigned privileged roles to manage Azure resources (Preview), as well as which users are assigned administrative roles in Azure AD
- Enable on-demand, "just in time" administrative access to Microsoft Online Services like Office 365 and Intune, and to Azure resources (Preview) of subscriptions, resource groups, and individual resources such as Virtual Machines
- See a history of administrator activation, including what changes administrators made to Azure resources (Preview)
- Get alerts about changes in administrator assignments
- Require approval to activate Azure AD privileged admin roles (Preview)
- Review membership of administrative roles and require users to provide a justification for continued membership

In Azure AD, Azure AD Privileged Identity Management can manage the users assigned to the built-in Azure AD organizational roles, such as Global Administrator. In Azure, Azure AD Privileged Identity Management can manage the users and groups assigned via Azure RBAC roles, including Owner or Contributor.

## Azure AD Domain Services




Azure Infrastructure Services enable you to deploy a wide range of computing solutions in an agile manner. With Azure Virtual Machines, you can deploy nearly instantaneously and you pay only by the minute. Using support for Windows, Linux, SQL Server, Oracle, IBM, SAP, and BizTalk, you can deploy any workload, any language, on nearly any operating system. These benefits enable you to migrate legacy applications deployed on-premises to Azure, to save on operational expenses.

A key aspect of migrating on-premises applications to Azure is handling the identity needs of these applications. Directory-aware applications may rely on LDAP for read or write access to the corporate directory or rely on Windows Integrated Authentication (Kerberos or NTLM authentication) to authenticate end users. Line-of-business (LOB) applications running on Windows Server are typically deployed on domain joined machines, so they can be managed securely using Group Policy. To 'lift-and-shift' on-premises applications to the cloud, these dependencies on the corporate identity infrastructure need to be resolved.

Administrators often turn to one of the following solutions to satisfy the identity needs of their applications deployed in Azure:

Deploy a site-to-site VPN connection between workloads running in Azure Infrastructure Services and the corporate directory on-premises.

Extend the corporate AD domain/forest infrastructure by setting up replica domain controllers using Azure virtual machines.

Deploy a stand-alone domain in Azure using domain controllers deployed as Azure virtual machines.

All these approaches suffer from high cost and administrative overhead. Administrators are required to deploy domain controllers using virtual machines in Azure. Additionally, they need to manage, secure, patch, monitor, backup, and troubleshoot these virtual machines. The reliance on VPN connections to the on-premises directory causes workloads deployed in Azure to be vulnerable to transient network glitches or outages. These network outages in turn result in lower uptime and reduced reliability for these applications.

Azure AD Domain Services provides managed domain services such as domain join, group policy, LDAP, Kerberos/NTLM authentication that are fully compatible with Windows Server Active Directory. You can consume these domain services without the need for you to deploy, manage, and patch domain controllers in the cloud. Azure AD Domain Services integrates with your existing Azure AD tenant, thus making it possible for users to log in using their corporate credentials. Additionally, you can use existing groups and user accounts to secure access to resources, thus ensuring a smoother 'lift-and-shift' of on-premises resources to Azure Infrastructure Services.

Azure AD Domain Services functionality works seamlessly regardless of whether your Azure AD tenant is cloud-only or synced with your on-premises Active Directory.

# Lab: Deploying Services to Secure Secrets in Azure

### Scenario

A local credit union client has been excitedly using Azure Resource Manager to deploy virtual machines to the cloud with ARM templates. Unfortunately, they quickly discovered that sending virtual machine passwords as part of the JSON object in an HTTP request body violates some of their industry's best practices. While they already use SSL for in-flight request security, they would like to use the Azure Key Vault to store their virtual machine passwords moving forward. The client has hired you to put together a prototype showing them how to accomplish this task.

### Objectives

- Deploy a Key Vault using an ARM template.
- Use a Key Vault secret in an ARM template.

***Lab setup***

Estimated Time: 60 minutes

Virtual machine: **20535A-SEA-ARCH**

User name: **Admin**

Password: **Pa55w.rd**

The lab steps for this course change frequently due to updates to Microsoft Azure. Microsoft Learning updates the lab steps frequently, so they are not available in this manual. Your instructor will provide you with the lab documentation.

### Exercise 1: Deploy Key Vault using ARM Template

---

### Exercise 2: Deploy Virtual Machine using Key Vault Secret

---

### Exercise 3: Cleanup Subscription

---

### Review Question(s)

# Module review and takeaways



## Review Question(s)