Session 13:
Configuring Web Applications for Deployment

Objectives

- Identify Azure configuration settings
- Describe Azure Key Vault
- Explain how to configure deployment credentials for Azure App Service
- Explain how to work with and manage NuGet Packages

Introduction to Azure Configuration [1-2]

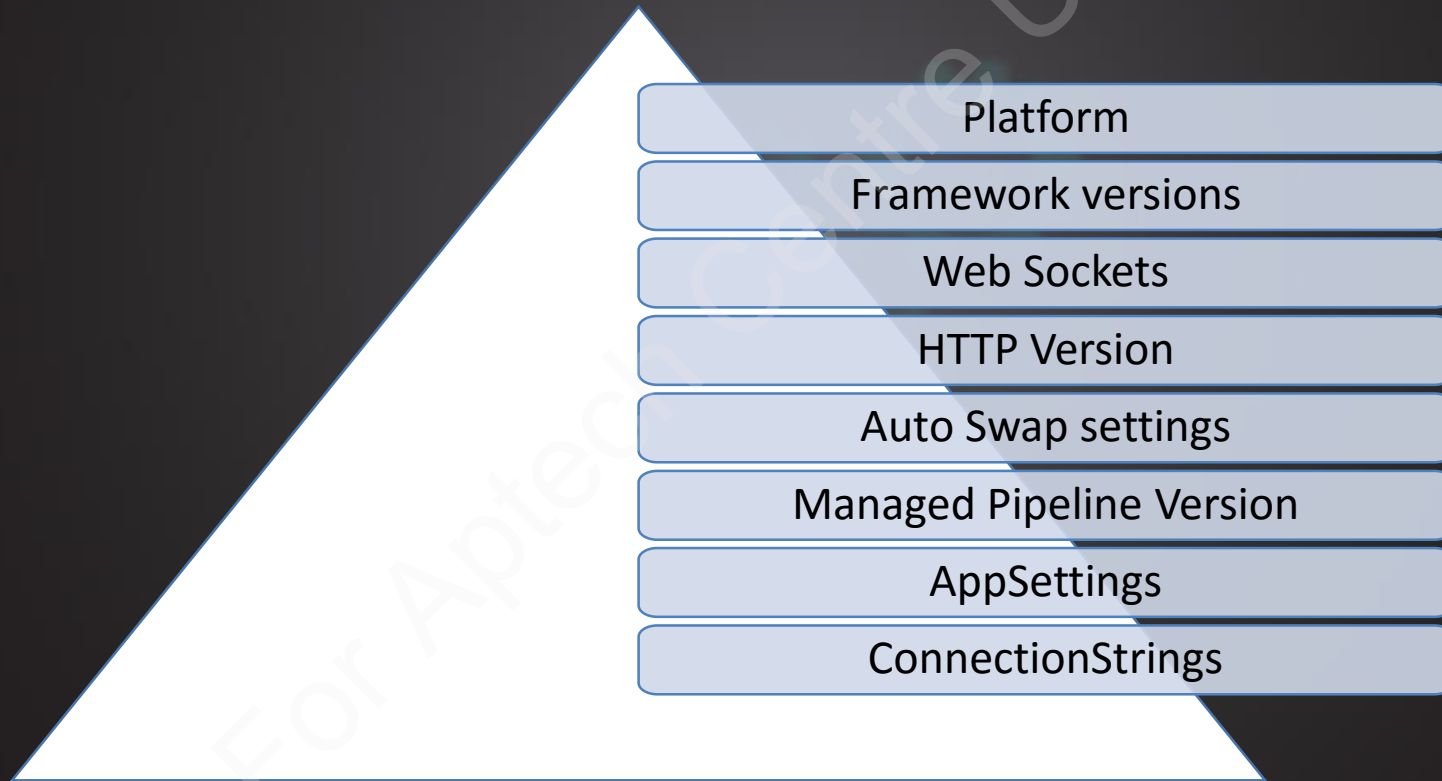Azure Web apps offer a convenient configuration feature for developers.

Allows storing string pairs in the form of keys and values in Azure.

In Azure portal, the **App Settings** tab allows entering string pairs for a Website.

Developers can specify the string pairs either as `ConnectionStrings` or `AppSettings`.

Introduction to Azure Configuration [2-2]

Options available in the left panel for viewing settings in the portal on clicking **App Settings**:

Platform
Framework versions
Web Sockets
HTTP Version
Auto Swap settings
Managed Pipeline Version
AppSettings
ConnectionStrings

AppSettings Configuration

AppSettings section under Application Settings has following name or value pairs, loaded during the Web app initialization:

Settings for .NET apps are added to the .NET configuration `AppSettings` at the time of execution.

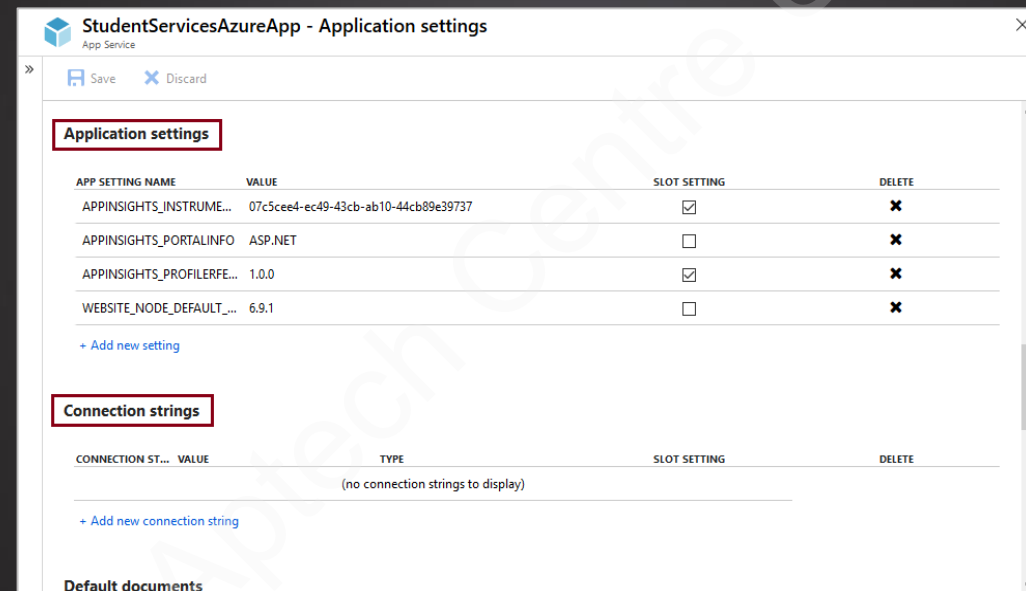The settings are accessible as environment variables for Python, Node, and Java applications.

If there is a nested JSON key structure for a Web app on a container or an app on Linux, such as `ApplicationInsights:InstrumentationKey`, its key name would be `ApplicationInsights__InstrumentationKey`.

ConnectionStrings Configuration

Connection strings are added to the `ConnectionStrings` settings at runtime for .NET apps. They replace the prevalent entries where the key represents the name of associated database.
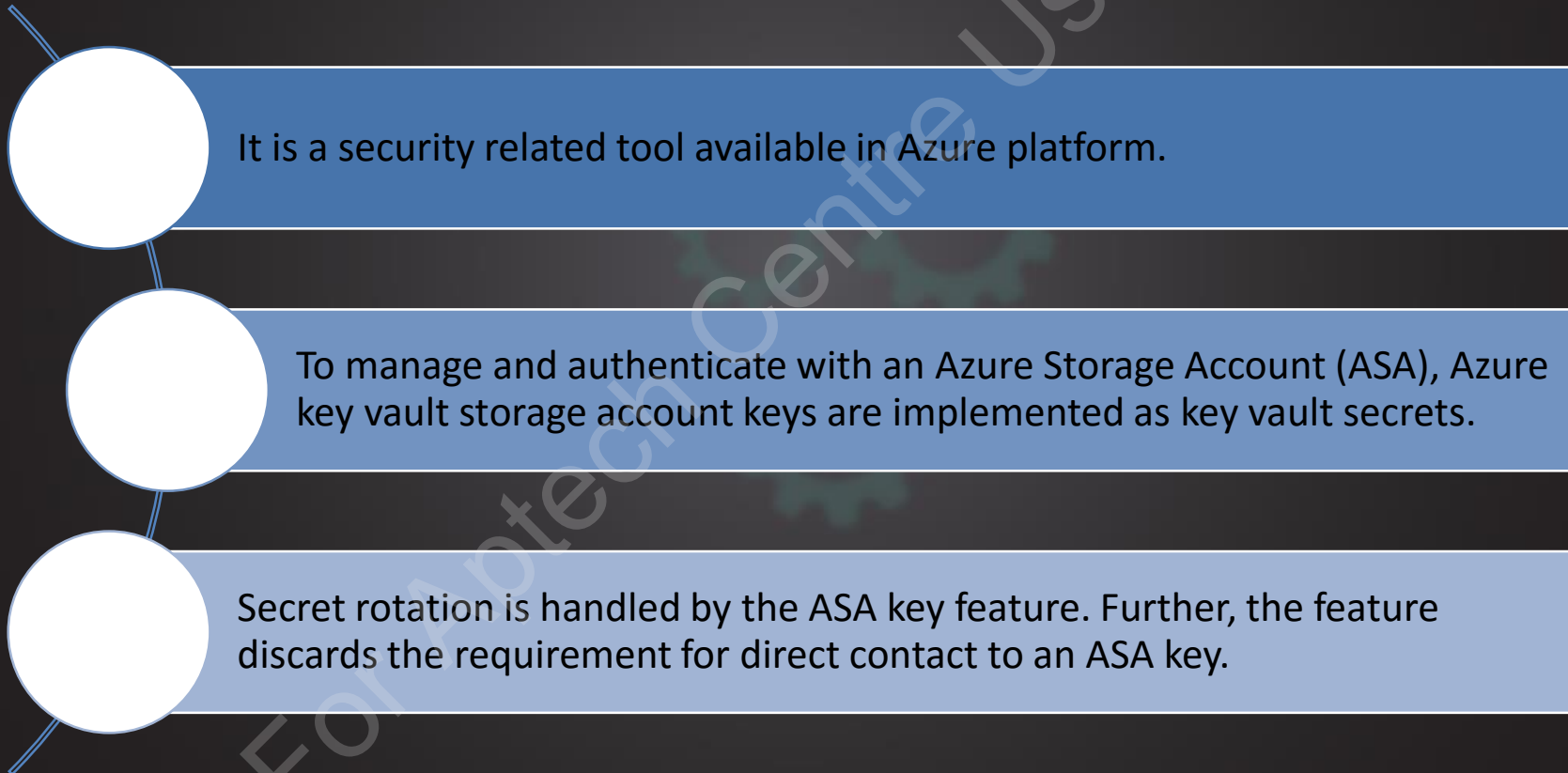
Application Settings and Connection Strings

Following figure shows the application settings and connection strings:

Working with Azure Key Vault

Following are the features of an Azure key vault:

It is a security related tool available in Azure platform.

To manage and authenticate with an Azure Storage Account (ASA), Azure key vault storage account keys are implemented as key vault secrets.

Secret rotation is handled by the ASA key feature. Further, the feature discards the requirement for direct contact to an ASA key.

Where to Use Azure Key Vault?

Following are scenarios where one can use Azure Key Vault:

To centralize application secrets.	• By centralizing the storage of application secrets in Azure Key Vault, their distribution can be controlled.
To store secrets and keys securely.	• Azure protects secrets and keys by using key lengths, Industry-standard algorithms, and Hardware Security Modules (HSMs)
To monitor access and use.	• After creating a few key vaults, it might be required to verify the usage and time of access to the keys and secrets.
To simplify the administration of application secrets.	• Developers should follow certain steps when storing crucial data.
To integrate with other Azure services.	• Key vault acts as a secure store in Azure.

Configuring Deployment Credentials for Azure App Service

Credential types supported by Azure App Service for FTP/S deployment and local Git deployment:

User-level Credentials	• This presents a set of credentials that can be applied for the total Azure account. These credentials are used for deploying to App Service, in any subscription, for any app, for which the Azure account has access permission.
App-level Credentials	• This presents a set of credentials that can be applied for each app. These credentials are used for deploying only to the particular app.

Working and Managing NuGet [1-2]

Following points describe NuGet:

NuGet is the mechanism (Microsoft-supported) for sharing code for .NET (including .NET Core).

It defines the creation of a package, as well its hosting and consumption.

NuGet also offers the tools for all of these roles.

Working and Managing NuGet [2-2]

Methods for downloading and installing NuGet packages:

Method	Description
`Package Manager UI (Visual Studio)`	It supports transport protocols such as TCP, HTTP, UDP, and custom transports. It also allows switching between them.
`nuget.exe CLI` `nuget install <package_name>`	Binary, text, and Message Transmission Optimization Mechanism (MTOM) encoding is supported by WCF. MTOM is a method of efficiently sending binary data to and from Web services.
`dotnet.exe CLI` `dotnet add package <package_name>`	It supports building services with WS-* standards such as message security, transactions, and reliable messaging.
`Package Manager Console (Visual Studio)` `Install-Package <package_name>`	It allows describing WCF SOAP services in WSDL.

Summary

- Azure Web apps retrieve values from Azure configuration settings and run them in Web apps.

- Azure Key Vault is used to securely store tokens, passwords, certificates, API keys, and other secrets.

- Deployment credentials for Azure app services play an important role for local Git deployment and FTP/S deployment of Web app.