



Securing your Web App with SSL Certificates

Cybersecurity
Project Week





Day 1 Recap

In the first day of your project work, you:

01

Used Microsoft Azure to create a web application.

02

Chose a unique domain from one of two cost options.

03

Deployed a Docker container on your web app.

04

Designed your custom web application.

05

Answered review questions.

Today's Class

Today's class will include:



An introduction to Azure Key Vaults.



An overview of tasks that you will work on.



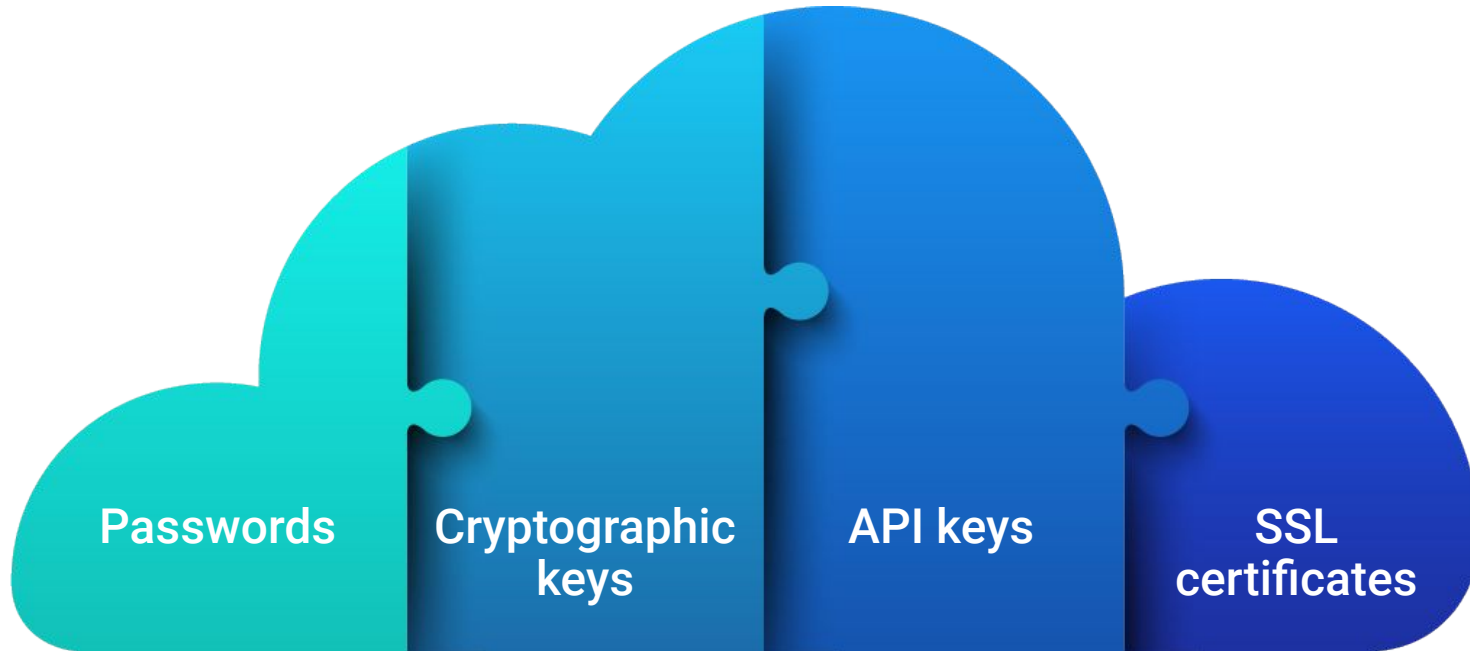
Project work.



Azure Key Vault

Azure Key Vault

When administrators manage cloud services such as web applications, they need to maintain lots of information securely, including:



Azure Key Vault

It is imperative that this information is stored securely, accessible only by specific individuals, and managed in a central location.



Microsoft can accomplish these tasks with **Azure Key Vault**.

Per Microsoft:

“Azure Key Vault is a cloud service for securely storing and accessing secrets.

A secret is anything that you want to tightly control access to, such as API keys, passwords, certificates, or cryptographic keys.”



Azure Key Vault

Benefits of Azure Key Vault include:



Azure Key Vault can store secret information in a single central location.



Key Vault secrets can be given access policies either by user or by application.



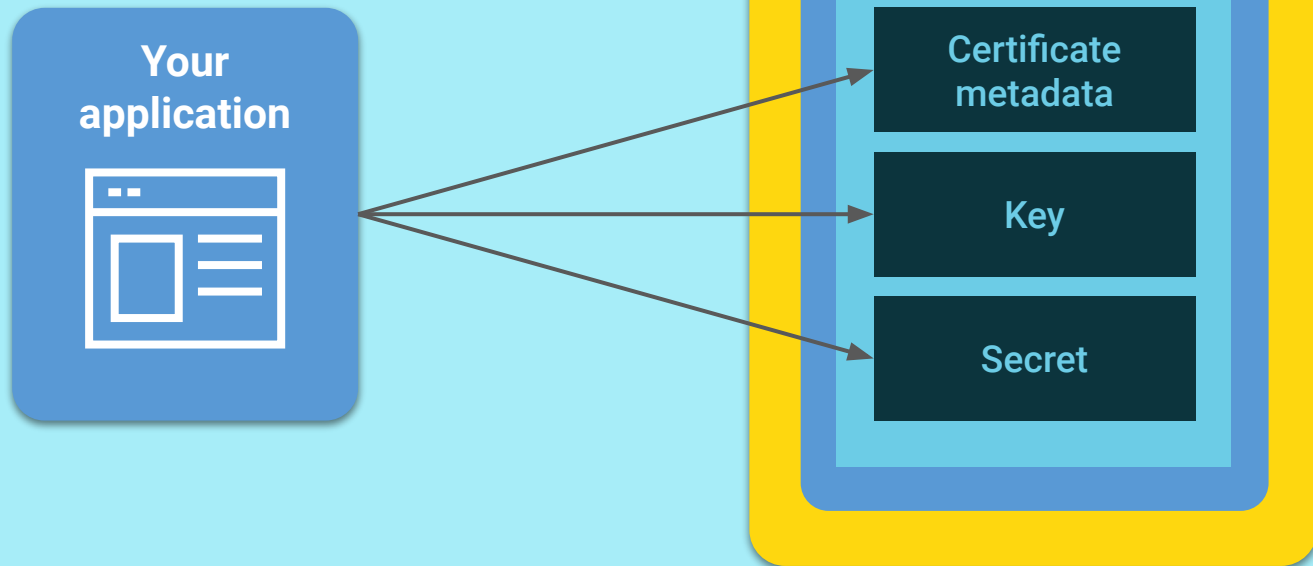
Secret information from the key vault can be accessed from other cloud resources.



Certificates and keys can be imported or directly generated from within the key vault.

Azure Key Vault

Today, we will create an Azure key vault and use it to store an SSL certificate.



Azure Key Vault

Today, you will create two types of SSL certificates and bind them to your web application. You will explore the advantages and disadvantages of each certificate.

01

Self-signed SSL



02

Azure-managed certificate





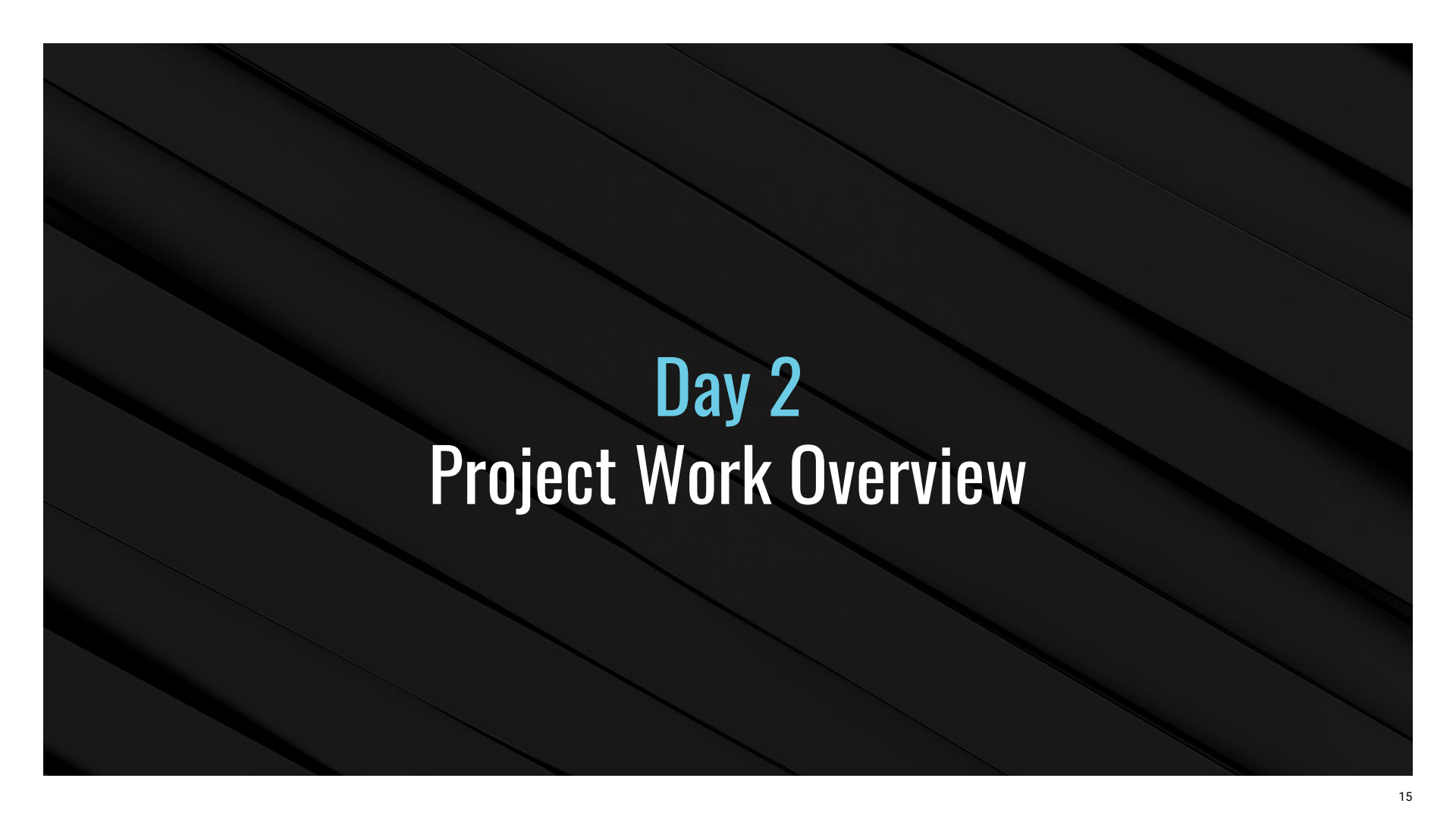
**Can anyone recall from
Cryptography week how
SSL certificates are used?**



SSL certificates are used to validate the authenticity of a web application and to assist with encrypting the user's traffic between the client and the server.

Questions?



The background of the slide is dark gray with a pattern of diagonal lines that create a sense of depth and movement. The lines are slightly lighter in some areas, giving the impression of a 3D effect.

Day 2

Project Work Overview

Day 2 Project Work

Today, you will complete the following:

01

Create a key vault.

02

Create a self-signed certificate.

03

Import and bind your self-signed certificate to your web application.

04

Create and bind an app service managed certificate.

05

Answer review questions.

Step 1

Create a Key Vault



REMEMBER

Use the same subscription and resource group that you did in the previous class.

Today's first step is to create an Azure key vault. Steps are included in the activity guide.

Microsoft Azure

Search resources, services, and docs (G+)

Home >

Key vaults

Default Directory

+ Create | Manage deleted vaults | Manage view | Refresh | Export to CSV | Open query | Assign tags | Feedback

Filter for any field... | Subscription == Azure subscription 1 | Resource group == all | Location == all | Add filter

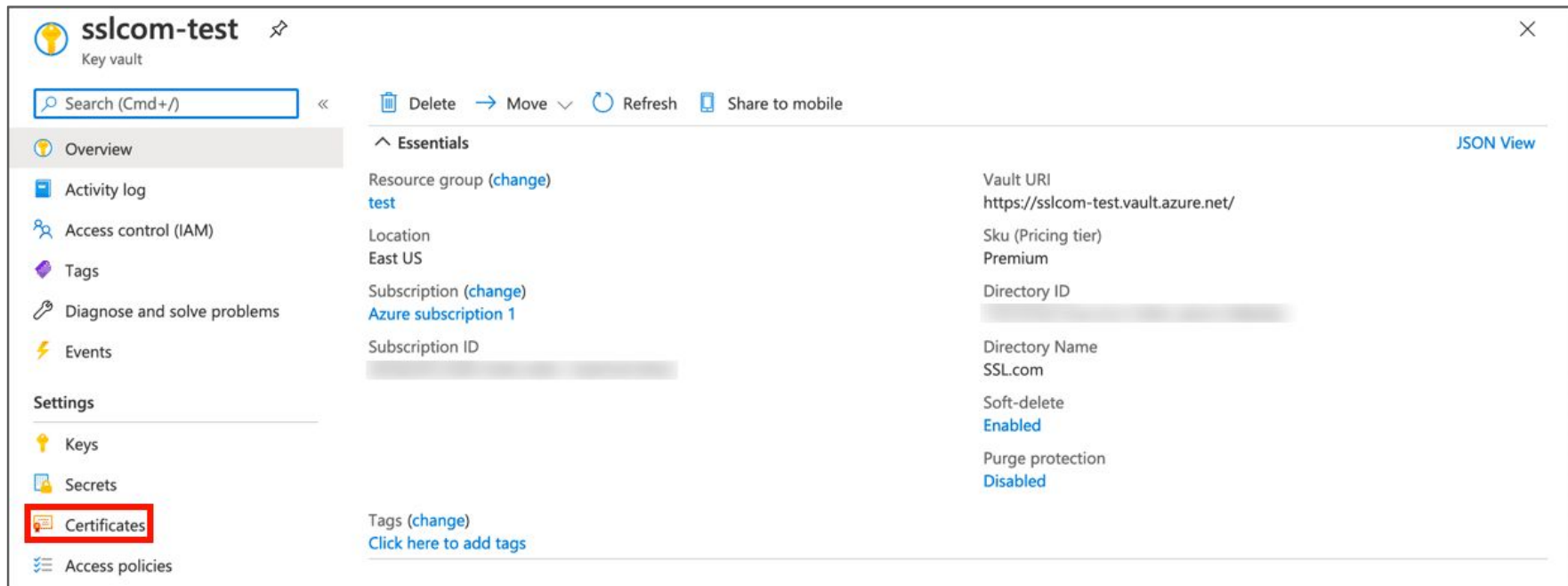
Showing 1 to 1 of 1 records.

<input type="checkbox"/> Name ↑↓	Type ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓
<input type="checkbox"/> Key vault1	Key vault	redteamRG	East US	Azure subscription 1

Step 2

Create a Self-Signed Certificate

In this step, you will return to the command line to use OpenSSL.



The screenshot shows the Azure portal interface for a Key Vault named 'sslcom-test'. The left sidebar contains navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Settings, Keys, Secrets, **Certificates** (highlighted with a red box), and Access policies. The main content area is divided into two sections. The 'Essentials' section on the left provides summary information: Resource group (test), Location (East US), Subscription (Azure subscription 1), and Subscription ID. The right section lists specific properties: Vault URI (https://sslcom-test.vault.azure.net/), Sku (Premium), Directory ID (redacted), Directory Name (SSL.com), Soft-delete (Enabled), and Purge protection (Disabled). A 'JSON View' link is visible in the top right corner of the Essentials section.

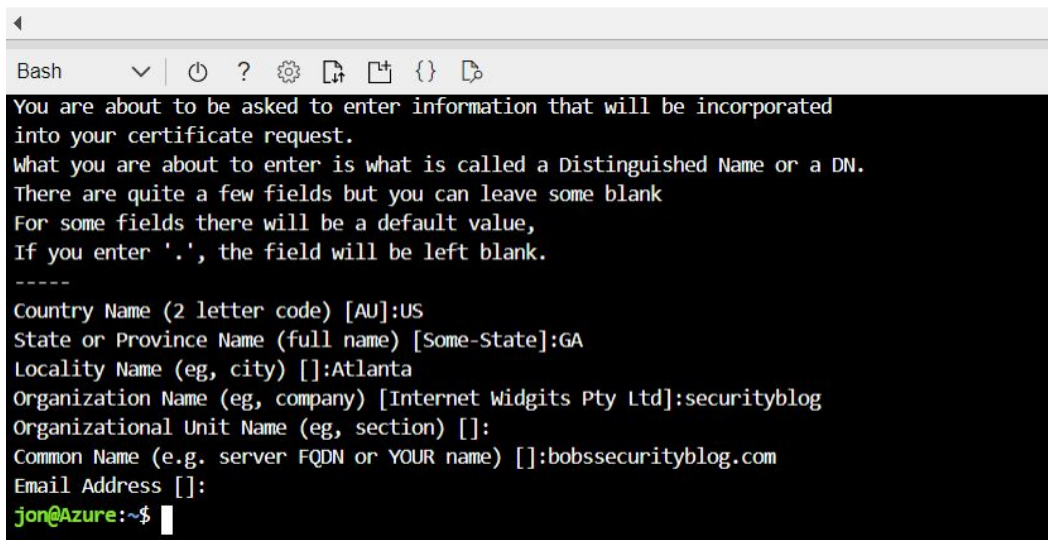


**Can anyone recall what we previously
used OpenSSL to do?**

Step 2

Create a Self-Signed Certificate

During Cryptography week, we used OpenSSL to generate a symmetric key and initialization vector.
In this project, we will use OpenSSL to generate a certificate called a **self-signed certificate**.

A terminal window with a dark background and light text. The window title is 'Bash'. The prompt is 'jon@Azure:~\$'. The text shows the user running 'openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem' and being prompted for information to be incorporated into the certificate request. The user has entered the following values: Country Name (2 letter code) [AU]:US, State or Province Name (full name) [Some-State]:GA, Locality Name (eg, city) []:Atlanta, Organization Name (eg, company) [Internet Widgits Pty Ltd]:securityblog, Organizational Unit Name (eg, section) [], Common Name (e.g. server FQDN or YOUR name) []:bobssecurityblog.com, and Email Address []. The prompt 'jon@Azure:~\$' is shown at the bottom.

```
Bash
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:GA
Locality Name (eg, city) []:Atlanta
Organization Name (eg, company) [Internet Widgits Pty Ltd]:securityblog
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:bobssecurityblog.com
Email Address []:
jon@Azure:~$
```

Step 3

Import and Bind the Self-Signed Certificate to the Web App

After you have created your self-signed certificate, you will import your certificate into your Azure key vault.

- You will use the TLS/SSL feature inside your web application to bind your self-signed certificate to your web application.
- Then, you'll view the certificate that you added directly on your browser, and analyze the security risks associated with your new certificate.

The screenshot shows the 'Create a certificate' page in the Azure portal. The breadcrumb navigation at the top reads 'Home > sslcom-test >'. The title is 'Create a certificate'. Under 'Method of Certificate Creation', there are two options: 'Generate' and 'Import'. The 'Import' option is highlighted with a red rectangular box. Below this, 'Type of Certificate Authority (CA)' is set to 'Self-signed certificate'. The 'Subject' field contains the placeholder text 'For example: "CN=mydomain.com".'. The 'DNS Names' section shows '0 DNS names'. The 'Validity Period (in months)' is set to '12'. The 'Content Type' section has two buttons: 'PKCS #12' (which is selected and highlighted in blue) and 'PEM'. The 'Lifetime Action Type' is set to 'Automatically renew at a given percentage lifetime'. The 'Percentage Lifetime' is shown as a slider set to '80'. The 'Advanced Policy Configuration' section shows 'Not configured'. At the bottom, there is a blue 'Create' button.



Important:

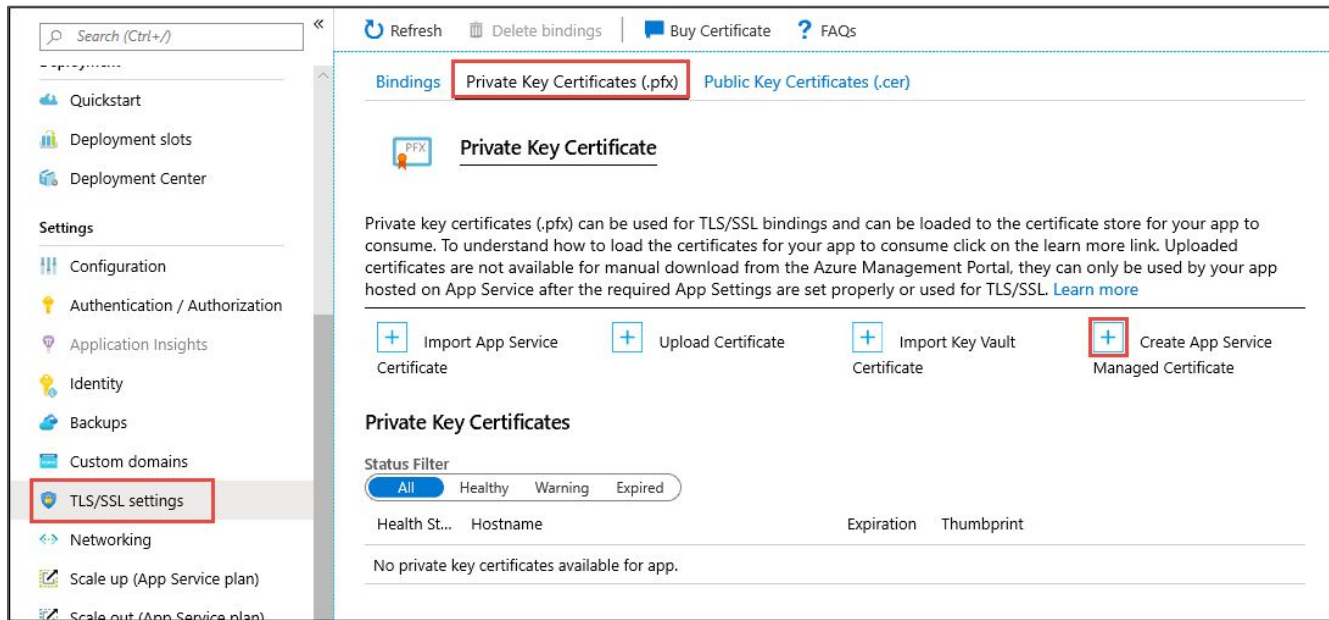
If you chose a free domain, you will not be able to bind any certificate to your web application, since one has already been provided by Azure.

Your daily guide will provide you with a theoretical exercise.

Step 4

Create and Bind an App Service Managed Certificate

Based on the security risks that you found, you will **replace** your first certificate by using Azure's services to add a more secure certificate, an app service managed certificate, directly to your web application.

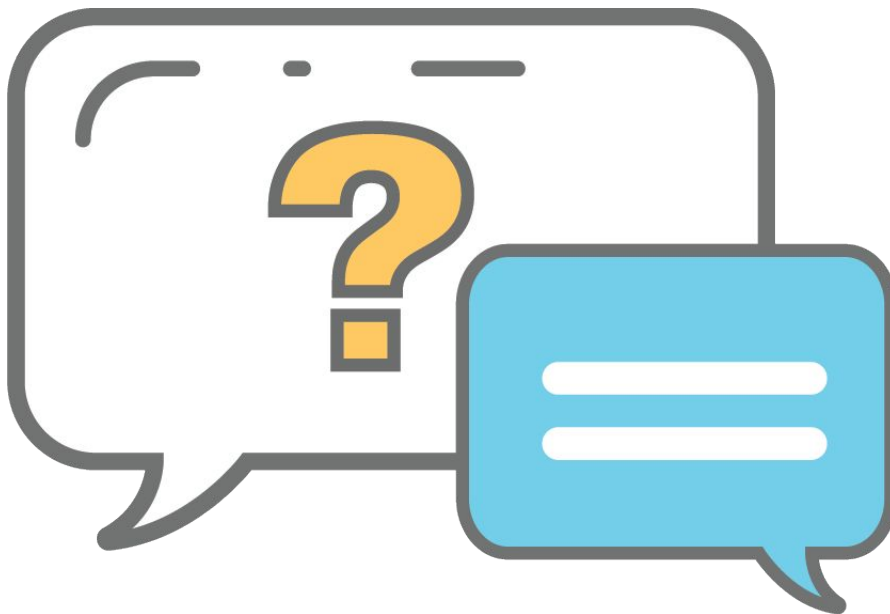


Step 5

Answer Review Questions

Once you complete today's activities, you will answer several questions about the project and how it relates to concepts that we've covered in class.

Feel free to use any resources available (class notes, slides, online resources) to answer these questions.



Securing Your Web Application with SSL certificates



Let's Get Started



For the remainder of today's class, you will work on the daily project tasks.

While each student is responsible for completing their own project, you can use classmates, TAs, or the instructor to assist if you have any questions.

The milestones that you need to complete in order to continue to the next day include: adding an SSL certificate to your web application.

Make sure to use the appropriate guide for the domain type that you selected: (free or paid).



Securing Your Web Application with SSL certificates

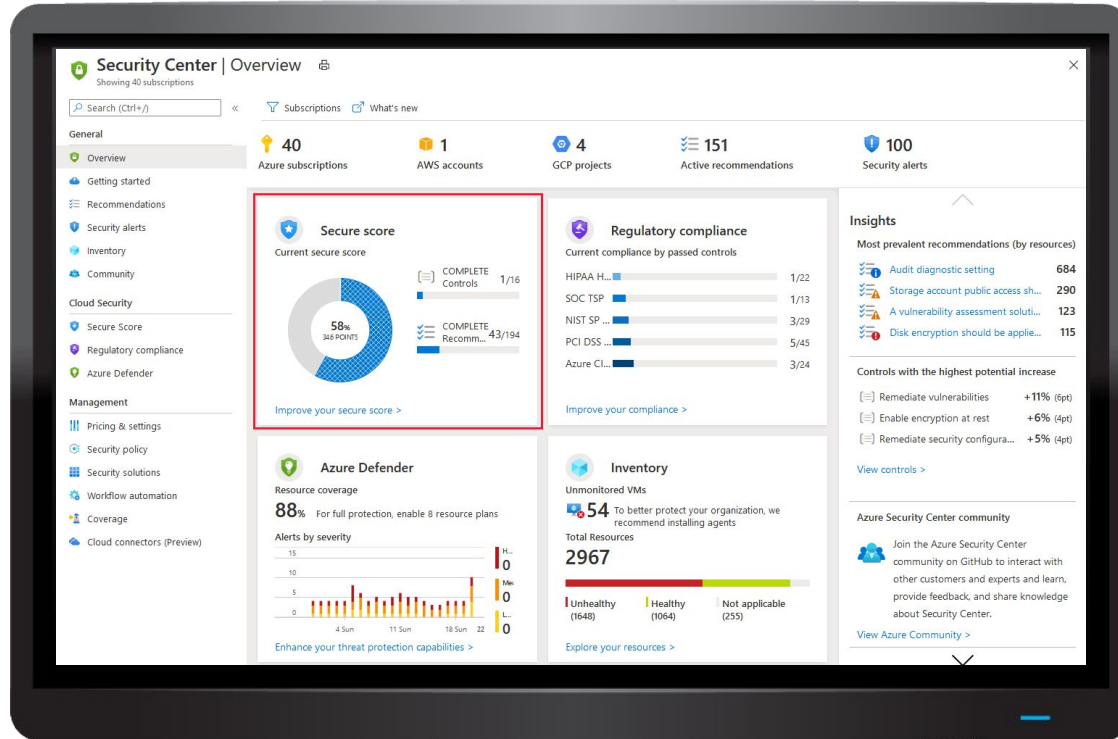
In this activity, you'll work on completing the
Day 2 tasks for your project.

Suggested Time:

To end of class

Project Work Time

Tomorrow, you will protect your web application with Azure security features.



Questions?



*The
End*