

Getting Ready For Production and Deploying Your Identity Provider



Kevin Dockx

Architect

@KevinDockx <https://www.kevindockx.com>



Coming Up



Deploying IdentityServer to Azure

- Configuration and operational data
- Data protection APIs
- Storing key material on Azure KeyVault
- Forwarded headers
- Applying a license



Deploying IdentityServer to Azure

It's just a web app, deploy it to an Azure App Service as any other web app

- Any way you see fit



Deploying IdentityServer to Azure



**Configuration
and operational
data**



Configuration Data

Configuration related to resources, clients, CORS, identity providers

- Hard-coded
- Settings file
- Persistent database-based store



Operational Data

Data required to correctly operate: grant results (tokens, codes), key management data, server-side sessions

- In-memory store is an issue in a multi-server environment: different requests may end up at different servers



Operational Data

SQLite databases are deployed together with the code to each host

- Same issues as in-memory data store



Operational
Data

**Deploy to a data store accessible from a
central location**



Deploying IdentityServer to Azure



**Configuration
and operational
data**



Data protection

Data Protection

ASP.NET Core's data protection feature is required for:

- Protecting keys at rest
- Protecting persisted grants at rest
- Protecting server-side session data at rest
- Session management

The data protection keys must be stored in a central location



Deploying IdentityServer to Azure



**Configuration
and operational
data**



Data protection



**Key material for
token signing**

Key Material

On the fly generation is handy during development

- Problematic in a multi-server environment

Azure KeyVault is the preferred location



Deploying IdentityServer to Azure



**Configuration
and operational
data**



Data protection



**Key material for
token signing**



**Forwarded
headers**



Forwarded Headers

Proxy servers, load balancers, ... often obscure information about the request

- Original scheme (HTTPS -> HTTP)
- Originating client IP address

These values must be forwarded in a header so they don't get lost



License

- A license must be applied before deploying IdentityServer to production**
 - Even if the license is free!



Deploying IdentityServer to Azure

Full-on Azure deployment

- Some basic Azure knowledge will come in handy



Demo



Persisting configuration data



Demo



Persisting operational data



Demo



Moving from SQLite to SQL Azure



Demo



Configuring data protection



Demo



Storing key material in Azure KeyVault



Demo



Configuring and using the forwarded headers middleware



Demo



Applying a license



Demo



The final deployment



Summary



Deployments are often to multi-server environments behind a load balancer

- In-memory data stores or host-specific stores for data that must be accessible regardless of the server must be replaced

Summary



Configuration data should go in a persistent store

- Resources, clients, ...

Operational data must go in a persistent store

- Grant results (tokens, codes), key management data, ...

Likewise story for data protection keys & other key material (e.g.: for signing)

Summary



Forwarded headers middleware ensures the original scheme & IP are forwarded in a header



The End is Nigh...



Questions?

@KevinDockx or the discussions
tab on the course page



Consider rating this course :-)





@KevinDockx

