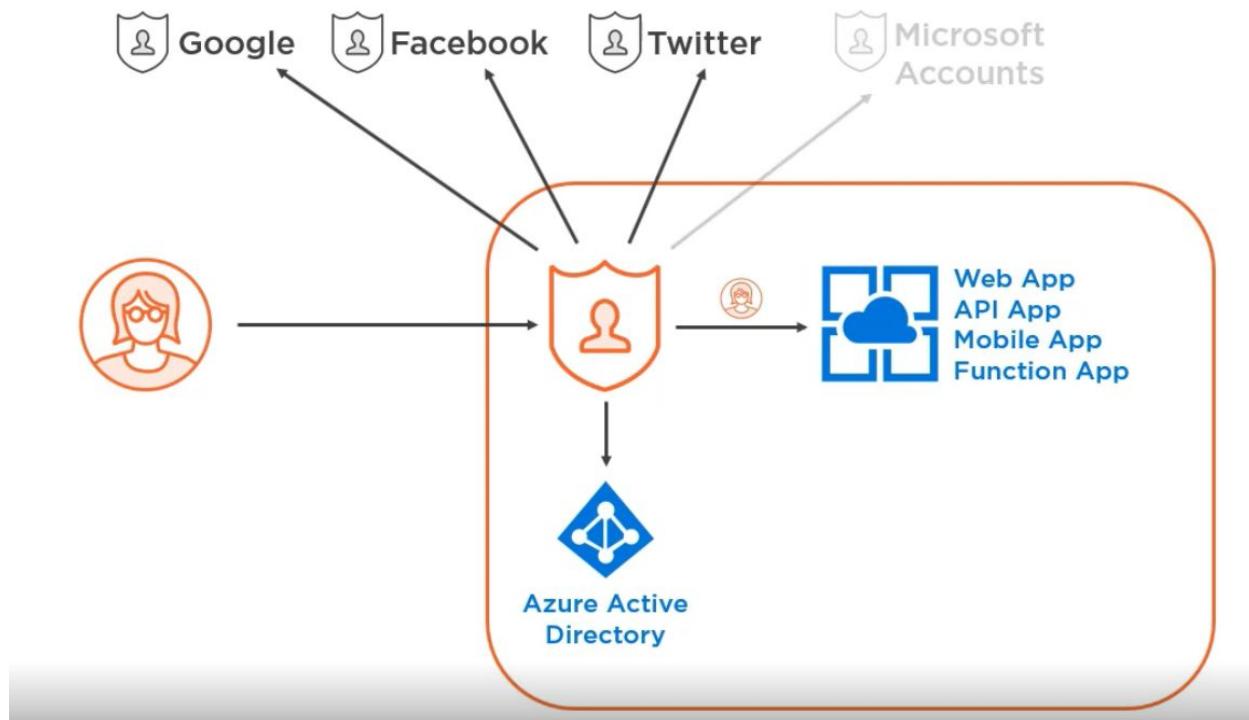


Managing Microsoft Azure App Service

Microsoft's Platform as a Service offering for hosting web applications, APIs, mobile back ends, and small pieces of event-driven code called function apps.



CONFIGURING APPLICATION SETTINGS

Module Overview



Understand App Service features

App Service Plan

Individual App Services

Web Apps, API Apps, Mobile Apps,
Function Apps, WebJobs

Create an App Service in the Azure Portal

Configuring application settings

App Service deployment options

Deploy a Web App

Azure App Service authentication

Azure appservices made of

Azure App Services

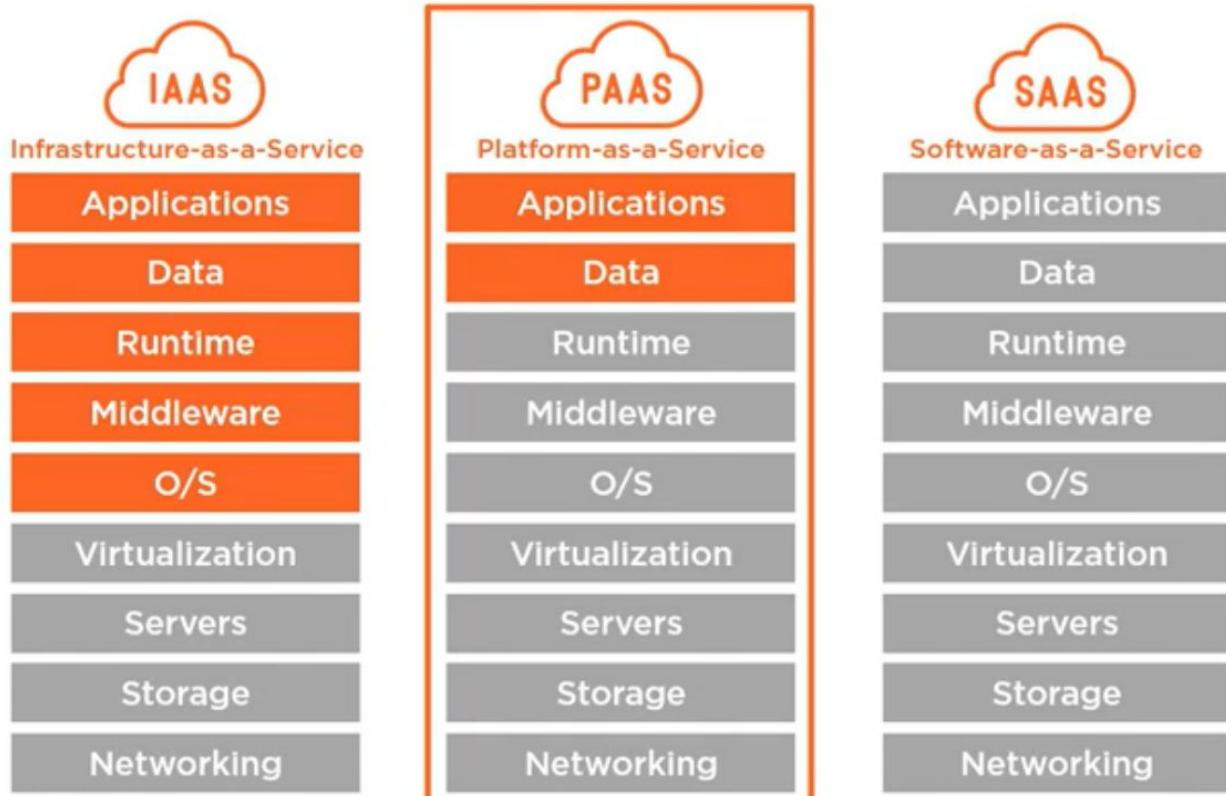
Web Apps

API Apps

Mobile Apps

Function Apps

There are a variety of Platform as a Service offerings in Azure, like App Services for hosting web-based applications, and there's also Azure SQL Database and Azure Storage for blob and file storage, as well as NoSQL tables and queues.



*grey->taken care by azure

*orange->user have flexibility to change

Azure App Service Features



Development frameworks>.net apps,.net core,java,ruby,node js,php,python apps

Web jobs> background services that run on the server like windows service applications(any language)

CI> continues integrating like integrating with VSTS,BIGBUCKET,GITHUB,DOCKER HUB

Deployment slots> to deploy your app to development or staging slot, and then swap the code with a production slot in order to promote the app to production

Backup and restore> back up and restore your app configuration, file content, and database right from within the App Service

Azure App Service Features

Hybrid Data Connections

Virtual Network Endpoints

Restrict Access to IP Addresses

Custom Domains

SSL Certificates

Auto-scaling

You can connect your app to on-premises data using hybrid data connections and to VMs running in Azure Infrastructure as a Service using virtual network endpoints. You can also restrict access to IP addresses or ranges of IP addresses so you can set it up. So the App Service can only be accessed by computers running on your corporate network, for example. You can set up custom domains for your App Service, either through an outside DNS provider or have Azure manage the domain registration and DNS for you. We'll be looking at that later in the course. And you can enable SSL for those domains too, which we'll also look at later in this course. You can set up auto-scaling for the servers that your app is running on to quickly scale-out to handle customer load. You can also scale-up your plan to run your App Services on bigger VMs if you find that you need more horsepower.

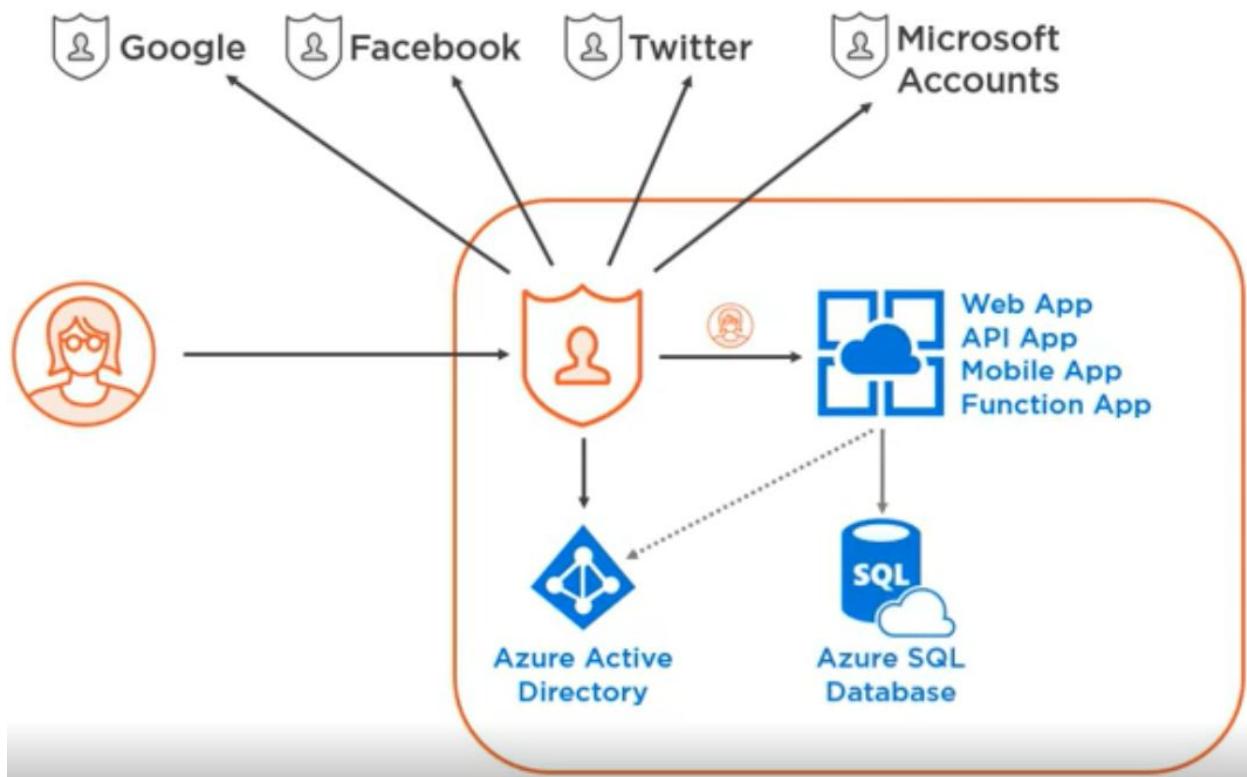
Azure App Service - Mobile App Features

Azure Notification Hubs Integration

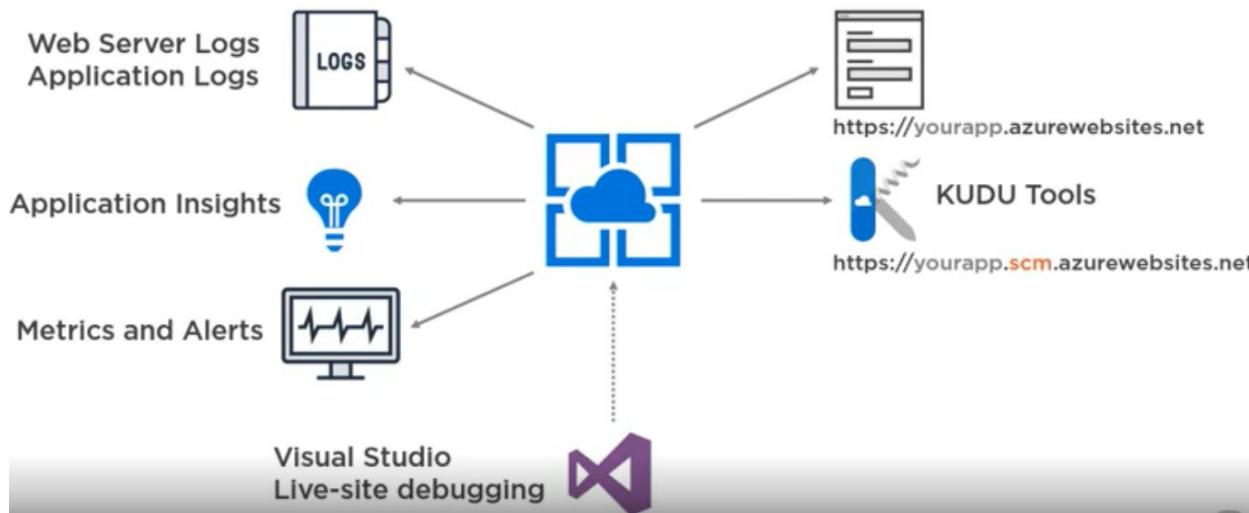
Easy Tables

Easy APIs

AUTHENTICATION

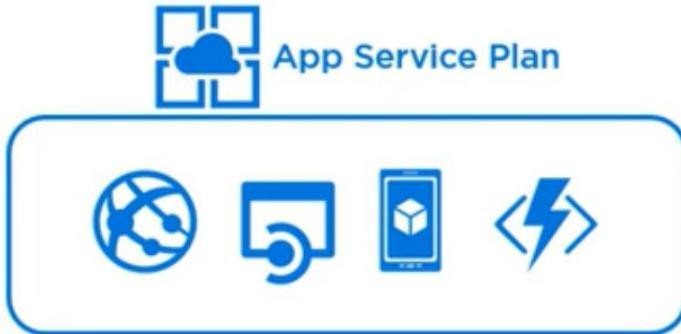


Monitoring and Troubleshooting Azure App Service



KUDU>KUDU provides a lot of functionality for managing and troubleshooting your App Service app. It's also the deployment engine for integrating with Git and other platforms. You can deploy the app from here, view deployment logs, interact with the files on the file system, view the processes running on your site, and even view the environment variables, so you can really see what's happening within the runtime environment that you're deploying to. You can run commands in PowerShell from the Debug console and view log files and download diagnostic stumps. When you're having issues with your site, KUDU can really help with diagnosis

App Service Plan:



Azure App Service Plan

Azure region

Number of VM instances (scale-out)

Size of VM instances (CPU/RAM/Storage)

Pricing Tier

<https://azure.microsoft.com/en-in/pricing/details/app-service/plans/>



App Service Environment (ASE)

Compute isolation (dedicated infrastructure)

- Windows and Linux Web/API Apps
- Docker Containers
- Mobile Apps and Function Apps

Network isolation

- Can be accessible only from VMs
- Can be accessible only from ExpressRoute

When running on the basic tier and up. You also have a service level agreement of 99.95% availability. There's no S.L.A. For the free and shared tiers. The isolated tier runs your V EMS on a dedicated azure virtual network, so you get network isolation. In addition to the compute isolation of the dedicated tiers, the isolated tier is also known as Asher App Service Environment or E.S.E. You can run Windows and Lennox, Web, APS, doctor containers, mobile APS and

function APS. In a sec. You get true network isolation in that you can configure an A S C to not be accessible from the Internet at all. You can configure it to only be accessible from the azure virtual network.



**App Service
on Linux**

Web Apps

Docker images

- Node.js
- Java
- PHP
- .NET Core
- Ruby
- Go
- Apache Tomcat

Custom Docker Images

Basic, Standard, Premium, Isolated pricing tiers

Web Apps:



**App Service
Web Apps**

Formerly “Azure Websites”

Deploy apps written in

- ASP.NET
- ASP.NET Core
- Java
- Ruby
- Node.js
- PHP
- Python

Web Apps and/or REST APIs

Web Apps - Benefits for Developers

Continuous Integration

Deployment Slots

Remote Debugging

Logging

Application Insights

Authentication

Implementing Windows Server 2016 Identity Federation and Access

by Neil Morrissey

Identity federation and access control allow you to authenticate users inside & outside your corporate domain, and provide secure access to applications and content. This course will teach you how to install and configure ADFS in Windows Server 2016.

<https://app.pluralsight.com/library/courses/windows-server-2016-identity-federation-access-solutions>

Web Apps - Benefits for Administrators

O/S and
Middleware
Patching

Auto-scale

Auto-heal

Metrics and Logs

Custom Domains
and SSL

Availability SLA

API Apps:



App Service
API Apps

Same features as Web Apps

Deploy existing APIs without code changes

Features of App Services for APIs

App Service - API Features



CORS Settings
Browser security
restrictions
Server sends header
to indicate allowed
source domains



Open API (Swagger)
Interactive documentation
and help pages
Exposed as json endpoint
Helps with integration



Azure API Management
Monitor and throttle
usage
Manipulate input/output
Portal for developers

Mobile Apps:



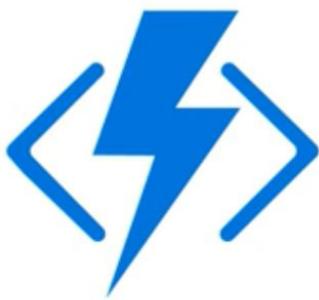
**App Service
Mobile Apps**

Formerly “Mobile Services”

Mobile Services are still supported

Existing apps can be migrated or upgraded

Function Apps:



App Service
Function Apps

Run code without host application

Event-driven model

- Function can be called directly
- Listen for events from Azure services and services outside Azure

Azure Function Apps - Triggers

HTTP Request

Timer events

Storage Account events

CosmosDb events

Azure Event Grid
Azure Event Hubs

WebHook events



Version 2

- C#, JavaScript, F# and Java

Consumption-based plan

- Execution time and resources used

Existing App Service plan

- No additional cost for Function Apps
- Leverage App Service Plan features
- “Always On” setting so functions are always ready to respond

Web Jobs:



WebJobs



WebJobs

WebJobs run within an App Service

Run program or script in the background

- .exe, .bat, .cmd (Windows cmd)
- .ps1 (PowerShell)
- .sh (Bash)
- .php (PHP)
- .py (Python)
- .js (Node.js)
- .jar (Java)

Execution models

- Run continuously
- Run on a schedule
- Run when triggered

View history and logs for WebJobs

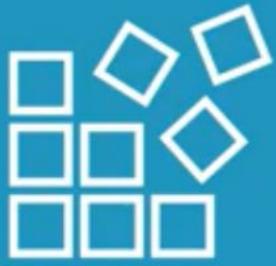
WebJobs SDK

- For custom development
- Azure Functions built on WebJobs SDK

No additional cost

App Services and WebJobs are subject to PaaS restrictions.

No access to:



Registry



Windows
Event Logs

GDI+

Graphics
subsystem

Creating an App Service in the Azure Portal:

Microsoft Azure

Search resources, services, and docs (G+/)

Dashboard > New > Web App

Web App

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Visual Studio Professional Subscription

Resource Group * ⓘ sample-website-name1 [Create new](#)

Instance Details

Name * sample-website-name2 .azurewebsites.net

Publish * [Code](#) Docker Container

Runtime stack * .NET Core 3.0

Operating System * [Linux](#) [Windows](#)

Region * Central US [Not finding your App Service Plan? Try a different region.](#)

App Service Plan

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. [Learn more](#)

Windows Plan (Central US) * ⓘ ASP-samplewebsitename1-8666 (S1) [Create new](#)

Sku and size * **Standard S1**
100 total ACU, 1.75 GB memory

[Review + create](#) [< Previous](#) [Next : Monitoring >](#)

Microsoft Azure

Search resources, services, and docs (G+/)

Dashboard > sample-website-name1

sample-website-name1

App Service

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Deployment

Quickstart

Deployment slots

Deployment Center

Settings

Configuration

Authentication / Authorization

Application Insights

Identity

Backups

Custom domains

TLS/SSL settings

Resource group (change) : sample-website-name1 URL : https://sample-website-name1.azurewebsites...

Status : Running App Service Plan : ASP-samplewebsitename1-8666 (S1: 1)

Location : Central US FTP/Deployment user... : No FTP/Deployment user set

Subscription (change) : Visual Studio Professional Subscription FTP Hostname : ftp://waws-prod-dm1-103.ftp.azurewebsites.wi...

Subscription ID : 39317181-21dc-4d9f-bd69-e59e5fba3db FTPS Hostname : https://waws-prod-dm1-103.ftp.azurewebsites...

Tags (change) : Click here to add tags

Diagnose and solve problems Our self-service diagnostic and troubleshooting experience helps you identify and resolve issues with your web app.

Application Insights Application Insights helps you detect and diagnose quality issues in your apps, and helps you understand what your users actually do with it.

App Service Advisor App Service Advisor provides insights for improving app experience on the App Service platform. Recommendations are sorted by freshness, priority and impact to your app.

Http 5xx

Data In

Data Out

Dashboard > sample-website-name1 > ASP-samplewebsitename1-866e | Apps > sample-website-name1

sample-website-name1

App Service

Search (Ctrl+ /) < Browse Stop Swap Restart Delete Get publish profile Reset publish profile

Overview

App Service has installed a patch that changes cross-site and iframe cookie handling due to upcoming changes in the new version of Chrome. Developers relying on these scenarios need to update their apps to handle these changes. Click to learn more.

Resource group (change) : sample-website-name1 URL : https://sample-website-name1.azurewebsites....

Status : Running App Service Plan : ASP-samplewebsitename1-866e (S1: 1)

Location : Central US FTP/deployment userna... : No FTP/deployment user set

Subscription (change) : Visual Studio Professional Subscription FTP hostname : ftp://waws-prod-dm1-103.ftp.azurewebsites.wi...

Subscription ID : 39317181-21dc-4d9f-bd69-e59e5ffba3db FTPS hostname : https://waws-prod-dm1-103.ftp.azurewebsites....

Tags (change) : Click here to add tags

Diagnose and solve problems Our self-service diagnostic and troubleshooting experience helps you identify and resolve issues with your web app.

Application Insights Application Insights helps you detect and diagnose quality issues in your apps, and helps you understand what your users actually do with it.

App Service Advisor App Service Advisor provides insights for improving app experience on the App Service platform. Recommendations are sorted by freshness, priority and impact to your app.

Http 5xx 100, 90, 80, 70, 60, 50

Data In 5kB, 4kB, 3kB

Data Out 3kB, 2.5kB, 2kB

Dashboard > sample-website-name1 > ASP-samplewebsitename1-866e | Apps > sample-website-name1 | Configuration

sample-website-name1 | Configuration

App Service

Search (Ctrl+ /) < Refresh Save Discard

General settings

Stack settings

Stack : .NET Core

Platform settings

Platform : 32 Bit

Managed pipeline version : Integrated

FTP state : All allowed FTP based deployment can be disabled or configured to accept FTP (plain text) or FTPS (secure) connections. [Learn more](#)

HTTP version : 1.1

Web sockets : On Off

Always on : On Off Prevents your app from being idled out due to inactivity. [Learn more](#)

ARR affinity : On Off Improve performance of your stateless app by turning Affinity Cookie off, stateful apps should keep this setting on for compatibility. [Learn more](#)

Debugging

Dashboard > sample-website-name1 > ASP-samplewebsitename1-866e | Apps > sample-website-name1 | Console

sample-website-name1 | Console

App Service

Search (Ctrl+ /) <>

Push

MySQL In App

Properties

Locks

Export template

App Service plan

App Service plan

Quotas

Change App Service plan

Development Tools

Clone App

Console

Advanced Tools

App Service Editor (Preview)

Resource explorer

Extensions

API

API Management

API definition

CORS

```
Manage your web app environment by running common commands ('mkdir', 'cd' to change directories, etc.) This is a sandbox environment, so any commands that require elevated privileges will not work.

D:\home\site\wwwroot>dotnet --info
.NET Core SDK (reflecting any global.json):
  Version: 2.2.109
  Commit: 586f23c400

Runtime Environment:
  OS Name: Windows
  OS Version: 10.0.14393
  OS Platform: Windows
  RID: win10-x86
  Base Path: D:\Program Files (x86)\dotnet\sdk\2.2.109\

Host (useful for support):
  Version: 3.1.1
  Commit: a1388f194c

.NET Core SDKs installed:
  1.1.14 [D:\Program Files (x86)\dotnet\sdk]
  2.1.509 [D:\Program Files (x86)\dotnet\sdk]
  2.2.109 [D:\Program Files (x86)\dotnet\sdk]

.NET Core runtimes installed:
  Microsoft.AspNetCore.All 2.1.14 [D:\Program Files (x86)\shared\Microsoft.AspNetCore.All]
  Microsoft.AspNetCore.All 2.1.15 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.All]
  Microsoft.AspNetCore.All 2.2.8 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.All]
  Microsoft.AspNetCore.App 2.1.14 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 2.1.15 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 2.2.8 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 3.0.1 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 3.0.2 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 3.1.0 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 3.1.1 [D:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.NETCore.App 1.0.16 [D:\Program Files (x86)\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 1.1.13 [D:\Program Files (x86)\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 2.0.9 [D:\Program Files (x86)\dotnet\shared\Microsoft.NETCore.App]
```

Dashboard > sample-website-name1 > ASP-samplewebsitename1-866e | Apps > sample-website-name1 | Extensions > Add extension > Choose extension

Add extension

*Choose Extension

Configure required settings

*Legal Terms

Accept legal terms

Choose extension

Extension	Version
.NET Datadog APM [Beta] Datadog	0.2.5-prerelease
.NET Raygun APM Raygun	1.0.1118
.NET Retrace APM Stackify	1.5.3
.NET Synopsys Seeker IAST Extension Synopsys	20.1.0
.NET elmah.io for Azure elmah.io	3.1.0
APM .NET Agent for Azure App Services CA Technologies	20.1.0.44
APM Insight .NET Agent Site24x7	4.7.1
APM Insight .NET Core Agent Site24x7	4.5.2
ASP.NET Core 2.2 (x64) Runtime Microsoft	2.2.0-preview3...
ASP.NET Core 2.2 (x86) Runtime Microsoft	2.2.0-preview3...
ASP.NET Core 3.0 (x64) Runtime Microsoft	3.0.3

OK

[Autogenerated] In this demo, we'll create a Web app using the azure portal. You can also do this using the Ascher CLI or Power Shell. I'll start by going to create a resource at the top left of the Azure portal. Then let's search for Web

app. You could search for any of the other APP service types from here to like the AP I AP Mobile app or function appetites. You can see there are various marketplace templates here for creating different Web APS, pre configured for PHP or No. J s Web APS with databases already configured or with platforms like Julia or Drew Paul already installed. We just want a basic Web app, though, so let's select that we get brought to a screen that gives us more information on the solution details and let's click create. To get to the configuration screen from this screen, we need to choose some options before we can create the APP service Web app. First of all, each APP service needs to be contained within a resource group. We could create a new resource group from hair, which is just a matter of giving it a name at this point. Well, Let's choose an existing one in my subscription. Next, we need to give the Web app. Name will be assigning a custom name later in this course, but right now we need to give it a default name that's unique across Azure because it will get Suffolk's with asher websites dot net And the same is true when you create a new A P I mobile or function app. A default UL is created with azure websites dot net at the end. The next choice is whether you'll be deploying code or deploying a custom docker image. If it's code, you need to choose a runtime stack, which is the platform that your code will run on like dot net Know js or Java. Let's see what happens when we select docker image. The runtime stack selection disappears because you'll be choosing a custom doctor image you've uploaded that has the platforms you want already configure. A new button appears at the bottom to select the doctor image. Here, you can choose a single container or a doctor composed configuration file. You can choose where the images stored, which could be the Ascher Container Registry Doctor Hub or a private registry. We won't be getting into Dr Images in this course, so let's go back to the previous screen. Now we've still got Dr Image selected and Lennox is the default operating system. We can choose to deploy to any one of the azure regions with this configuration. But if we change to Dr Image on the Windows operating system, we only have a select number of regions. While this is in preview. Now let's switch back to a more traditional deployment, where we will be deploying code to an APP service in Azure, we get the runtime stack drop down list back, and here we can choose from a variety of run times like dot net core sp dot net Jabba No js php, python and ruby. What's interesting here is that if you're following the order of selections down the screen, you're choosing the platform before choosing the operating system. So it may be a little confusing that we had Windows selected, but all these Jabba options are available. This drop down list isn't filtered by the operating system. You need to first choose a platform. Let's choose Java 11 s E. And then the operating system selection changes to show which OS is available for that platform. In this case on Lee, Lennox is available to host this run time. If I choose sp dot net, only Windows is available. But if I choose dot net core, I can select either the Windows or Linux operating systems. Now let's choose the region where we want this APP service deployed now in this region. I already have a NAP service plan created for Lennox Apse, so the APP service plan defaults to that. Remember the APP service plan to finds the underlying infrastructure that the APP service's will run on. We could, of course, create a new APP service plan here, too. And if I change to the Windows OS, I don't have an existing APP service plan in this region, so by default it will create one for me. But I'd like to name this myself, so I'll click to create a new APP service plan and give it a name. Now we can choose the size of the APP service plan, which defines the cost and features available. I'll click change size, and this is where we choose pricing tier for the APP service plan. We'll be able to create multiple APP. Service is within this plan, and we can always resize the plan later. If I choose the Dev test group at the top, I could select the shared plan, which is the least expensive moving to the D one plan. We get the ability to use custom domains and at the basic pricing tier, weaken scale, the underlying infrastructure to multiple VMs up to three instances. But we have to do that manually. There's no auto scale at this pricing tier, so let's go back to the production grouping and expand all the options I already know. Well, at least need the standard here in order to use custom domains, SSL and deployment slots. So I'll choose that and click apply. Next. We can choose to enable application insights to provide deep monitoring into our application, but we won't be covering that in this course, so I'll disable APP insights. Next, we can create tags on the APP service, and these can help you identify your APP service later for billing purposes or for automation scripts. Next week in review and create the Web app, we get brought to the deployment screen where we can watch the status of the deployment. It can take around two minutes for the resources to get created. Swell, positive recording. And now the APP service and APP service plan has been created. We could navigate to them from here, but let's goto all service's and search for the APP service plan. From my list of APP service plans, I can

see the PS testing plan we created. So let's like that. And when the APP service plan opens, we've got some metrics on the overview page and weaken. Scale the plan up to another pricing tier from here, or scale out the number of instances. Let's click on APS to see all the APP service is running inside the APP service plan. Of course, there's just one app service here, and it's got the icon for a Web app. So let's open it up and on the overview page for the Web app. We've got the U R L to the AP at the top. Right and below that is the FTP information that we can use to deploy our code to the file system across the top. We can stop the app up to make it unavailable. This doesn't prevent you from incurring costs, though it just removes access to the running app. We can restart the app if we're having any performance issues, delete the APP and download the published profile to help with deploying from visual studio. We can also browse to the girl from the browse button. So let's do that to see that the APP is up and running. We haven't deployed anything yet, but we get a test page in the euro in the address bar shows us that our website is running and by default it's using https with a built in certificate for azure websites dot Net in the next clip will configure the settings for this Web app before we deploy any code.

Exploring App service settings:

[Autogenerated] I'm back at the AP Service Plan level, and I've created a few more app. Service is at the top is an A P i AP a mobile app, and at the bottom is the Web app that was created in the previous clip. Let's open up the Web app along the menu. On the left, there's a grouping for deployment features. Below that are the settings, and we'll take a look at configuration settings in a little bit. We can also add custom domains in SSL settings from here, and this is also the section where we can add Web jobs. If I click to add a new Web job, I can upload a ZIP file with the code or scripts to be run and choose whether it should be run continuously or triggered. Let's exit out of this and see what else is on the menu. We can access the APP service plan from here and development tools. The Advanced Tools opens up the coup portal for this Web app. Then below that are the easy tables and easy AP eyes for mobile APS in the a p l settings that let us configure the location of the swagger document in course settings. Now let's look at the application settings, which are accessible from the configuration menu. General settings is where we configure the framework that her app will be running on. There are multiple dot net frameworks available here, and I'll show you shortly how to check the version of dot net core that's installed. You can choose a different application stack for your code to run on. You could choose PHP in the version. You can enable Python and choose the version. And if you choose Jabba, you get a few more options for the major version minor version and the server type that you want your Jaffa code to run on. Let's discard these changes, since I'll be uploading a dot net app later in this module. Then you can select whether your Web app runs in a 32 or 64 bit environment, and you have to be on at least the basic pricing tier in order to be able to enable 64 bit. The free and shared tears always run on a 32 bit environment. The managed pipeline version is an I. I s setting. That you should leave is integrated unless you're running older. APS basically integrated mode combines the request processing models of I I s and SP dot net into a unified process. So things like authentication aren't duplicated. You should only change this. If your APP isn't working in integrated mode and you've narrowed down the problem to this setting, then we can select whether we want to allow FTP access or not and whether FTP needs to be done over a secure FTP s connection. Below that, you can set the http version Most modern browsers support http 2.0, which has some optimization cz for downloading and browsers and forced t l s encryption Below that you can choose whether you want to enable Web sockets and you'll need this if you want to use signal, are in the dot at framework or dot net core. He always on setting is here because when a Web app is idle for a while, it gets unloaded to preserve resources. If you're running a Web job that's continuous, you'll want to set always onto true and the same is true for function APS Below that, you can turn off session affinity when this is on and your APP service plan is scaled out to multiple server instances. The browser gets a cookie that roots the session back to the same server instance on subsequent calls. If you've designed, you're apt to be stateless, then you don't need this on and you can improve performance by turning it off. Then we've got the settings for remote debugging from visual studio. When we turn this on, we can use the remote to _____ and visual studio to connect to the running Web app in Azure, and this will stay enabled for 48 hours. And at the bottom, you can enable client certificate authentication. You'll need to enable https for the domain, and then Asher will

forward the client certificate in their request to your application to validate. APP Service doesn't actually validate the certificate. It just injects the request header with the client certificate and forwards it to your APP. Your code can then read the certificate. Now let's look at the application settings tab at the top. If you're dot net developer, you're familiar with the APP settings in a Web config file or upsetting start Jason file. These settings here in the portal will override those settings in your deployed code, so you basically have a window into your APP settings to override them here without having to edit the config. File deployed with your code for node Java, python and PHP APS Thes upsetting variables are available in code as environment variables, where this really becomes useful or when you're using deployment slots. Because you could mark a setting as being slot specific, meaning an upsetting or connection string for your staging slot should remain in staging and not overrate the production settings. When you swap your staging code to prod. Well, look at this feature closer in the next module on deployment slots. Below application settings are connection strengths. When I add a new connection string, you can see that the slot setting feature is available for the connection strings as well. Here you can choose the target database type two or choose custom for dot net APS thes connection strings air injected into your configuration settings at runtime, overriding entries where the key equals the database connection. String name for other frameworks. Thes connection strings air available as environment variables. The next tab will look at is for default documents so when a user goes to the root of the site without specifying a document. APP service will go down this list trying to find a matching document in your sight on the path mapping tab handler map ings let you add custom script processors to handle requests for specific file extensions. You'll upload the script process er to the D drive of your APP service and point to it from here. You can pass arguments to the custom process er from here, too. This gives you a lot of control over responding to requests for certain file types like you could write a custom process er, to return RS s formatted XML. When the file extension dot rs s is requested at the bottom, you can configure virtual directories and applications. A virtual directory is a path you specify here for user's to see in the earl. But it actually maps to a physical location on the D drive of your APP service. So you can specify a different name for the virtual directory than for the structure of the physical directory. So users don't see the physical structure of the file system by marking the type as an application you can deploy multiple sites under the same map service. Virtual directories and applications are a feature of I I s So you don't see this section when deploying a Web app on Lennox. Now we mentioned I'd show you how to check the version of dot net core that's installed. If we go down to the console under development tools, I'll open that up. This is like using the command prompt on the underlying Web server. If I type in dot net two dashes in the info parameter, we can see a list of all the SP dot net core and dot net core SD Case and run times that are installed. Notice that s p dot net core 2.2 is the latest final release installed at the time of this course, there's a dot net core three preview available. You can install that from the Extensions menu. I'll click add and I get a list of all the extensions that are available to install on this APP service and SP dot net core 3.0 ca NBI installed from here. So he's got an APP service plan with AP Service is, but we have yet to deploy an actual application. Before we do that. Let's talk about the different deployment options next

App service deployment options:



Code Deployment to Azure App Service

Visual Studio

- Publish dialog - connect to Azure
- Import publish profile

Continuous deployment

- Visual Studio Team Services
- Bitbucket
- GitHub

Local Git deployment

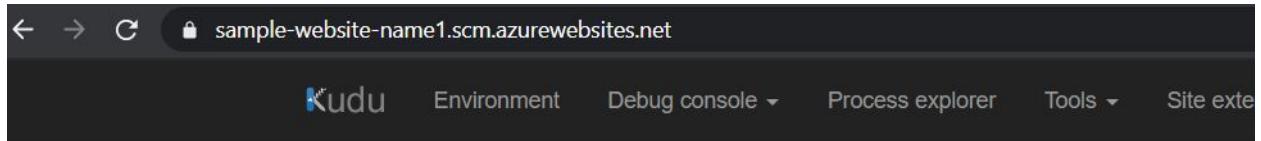
OneDrive/Dropbox deployment

FTP Deployment

Zip/War file deployment

[Autogenerated] Let's review the different options for deploying your code to your APP service. I won't go into too much detail here because I've got a whole module on deployment in my other course on managing Microsoft Azure APP service plans. But let's review the different ways you can deploy your code to APP service. If you're developing in visual studio, you can deploy directly from the published dialogue. If you've got contributor credentials on the APP service, you can connect directly to Azure and choose your AB servers for deployment. If you don't have those credentials on, Administrator can download the publisher's profile from the azure portal and provide to you inside that profile. Our APP level credentials that will allow visual studio to authenticate to the APP service. To deploy the code, you can set up continuous deployment to as your APP service from service is like azure devops's bit bucket and get up. You can deploy from your local git repositories and the KUDU build server and as your APP service will compile your code and deploy it to your APP service. Sometimes, though, there's a separation of responsibilities within an organization whereby developers have to package their code and provide it to an operations team to deploy into the production environment. For those scenarios, there are other deployment options. You can drop your code in a cloud folder in one drive or Dropbox and employer code on demand from within the azure portal. You can deploy the azure up service using FTP using an FTP client, and lastly, you can deploy your app using a zip file or war file for ____ by either dragging it right into the KUDU interface for your app service or pushing it to the rest endpoint programmatically. Let's look at using the KUDU interface in the next clip.

Web App deployment :



The screenshot shows the Kudu environment page for a website named 'sample-website-name1.scm.azurewebsites.net'. The page has a dark header with navigation icons and the URL. Below the header, there's a navigation bar with links: 'Kudu', 'Environment', 'Debug console', 'Process explorer', 'Tools', and 'Site exte'. The main content area is titled 'Environment' and contains the following data:

Build	85.11226.4297.0 (0a03cecc9e)
Azure App Service	87.0.7.74 (master-de1559597a4)
Site up time	00:00:41:33
Site folder	D:\home
Temp folder	D:\local\Temp\

Environment

Build	85.11226.4297.0 (0a03cecc9e)
Azure App Service	87.0.7.74 (master-de1559597a4)
Site up time	00:00:41:33
Site folder	D:\home
Temp folder	D:\local\Temp\

REST API (works best when using a JSON viewer extension)

- [App Settings](#)
- [Deployments](#)
- [Source control info](#)
- [Files](#)
- [Log streaming \(use curl, not browser!\)](#)
- [Processes and mini-dumps](#)
- [Runtime versions](#)
- [Site Extensions: installed | feed](#)
- [Web hooks](#)
- [WebJobs: all | triggered | continuous](#)
- [Functions: list | host config](#)

More information about Kudu can be found on the [wiki](#).

Add .scm in url of the website to go to KUDO portal console.goto tools>zip web push deploy.
Now drag and drop published zip file of .net app.

/wwwroot  | 1 items

Name	Modified	Size
   hostingstart.html	3/29/2020, 2:09:52 PM	4 KB



Deployment Id : b9a006b786fa45b9899e086b9a948c40

- 2020-03-29T09:26:05.5867569Z : Created via a push deployment
 - 2020-03-29T09:26:06.1054283Z : Updating submodules.
 - 2020-03-29T09:26:06.230758Z : Preparing deployment for commit id 'b9a006b786'.
 - 2020-03-29T09:26:06.4960795Z : Generating deployment script.
 - 2020-03-29T09:26:07.683624Z : Running deployment command...
 - 2020-03-29T09:26:11.4335784Z : Running post deployment command(s)...
 - 2020-03-29T09:26:11.5273858Z : Triggering recycle (preview mode disabled).
 - 2020-03-29T09:26:11.6211371Z : Deployment successful.

← → C  sample-website-name1.azurewebsites.net

Website1 Home Privacy

Welcome

Learn about [building Web apps with ASP.NET Core](#).

now go back to the website url to see the deployed website.

Configuring Authentication:



Azure App Service Authentication

Users log in to Azure Active Directory

App Service passes user identity to app

Server-directed authentication flow

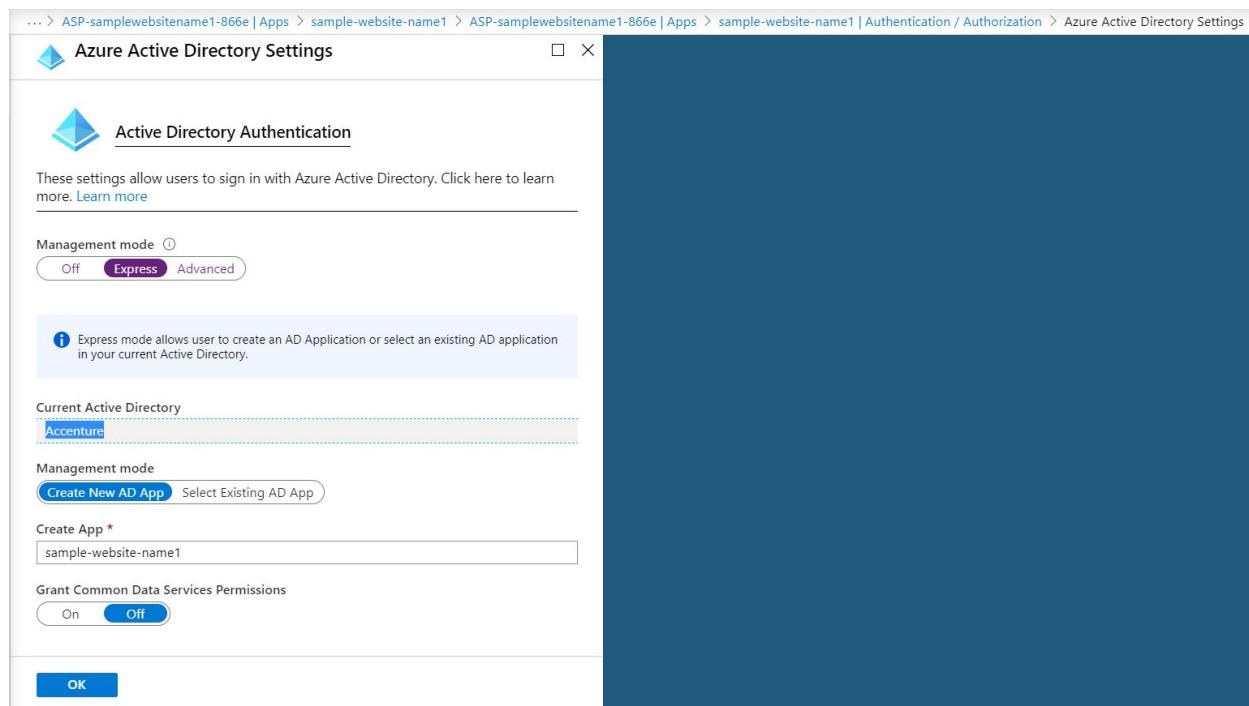
- HTTP redirects user to authenticate

Client-directed authentication flow

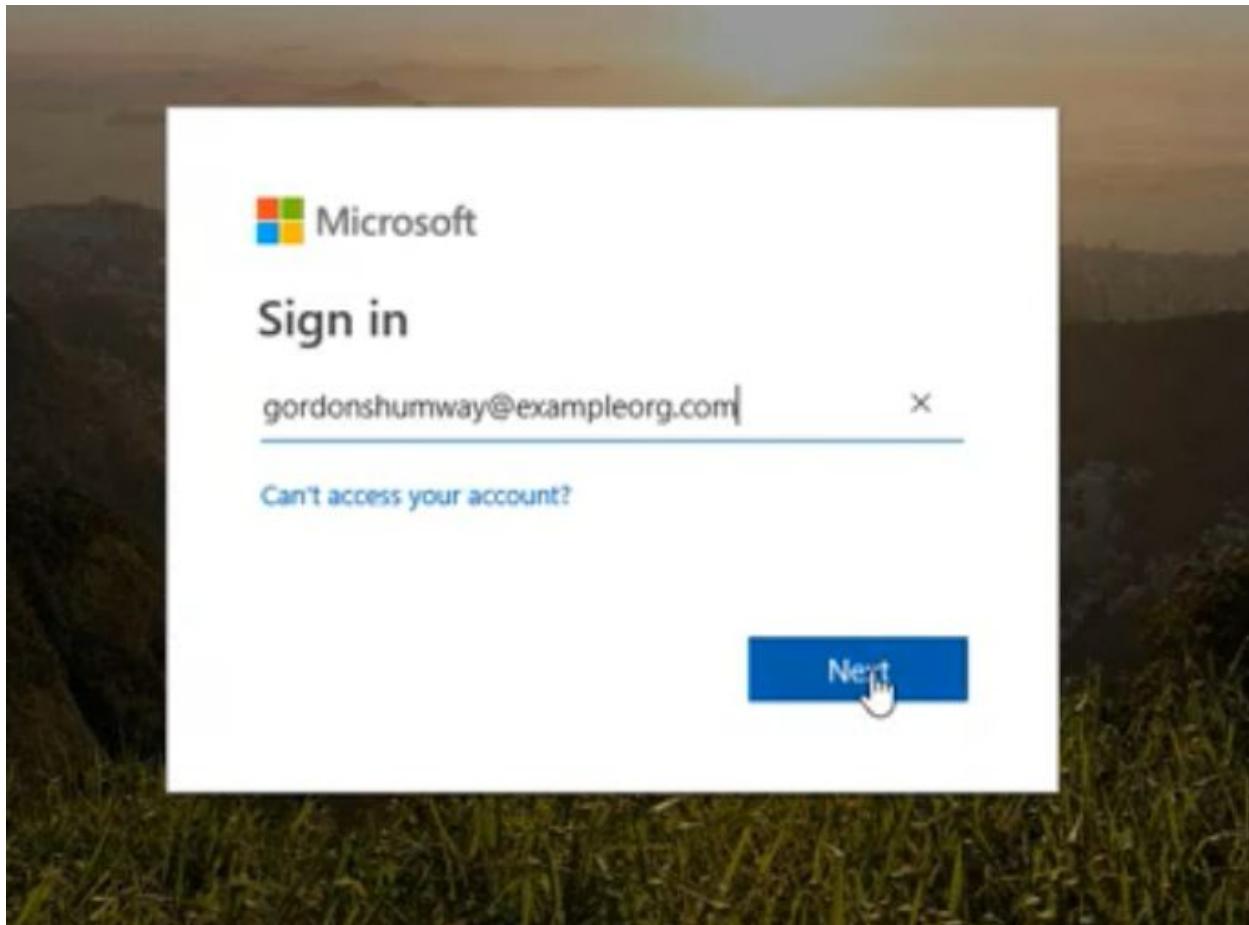
- Client applications pass authentication token to App Service

[Autogenerated] in this demo will set up APP service Authentication to use as your active directory will be securing a Web app so users will get directed to log into Azure A D. Once this setting is in place and then APP service authentication will pass headers to the APP with user identity information that you can use to do your own authorization within the APP. Setting this up for a P i APs in mobile lapses similar in the portal. But the calling APS need to get the authentication tokens first and pass them to APP service. That's beyond the scope of this course, but I just want to point out to you that there's a server directed authentication flow, which will be doing here. And there's also a client directed authentication flow. Either way, you can leverage APP service authentication to protect your server side application. I'm inside the Web application and it configures authentication. You scroll down to the settings section of the menu and click authentication authorization. First thing you need to do is turn it on. There are five providers here that we can enable as your A D, Facebook, Google, Twitter and Microsoft accounts, which are Hotmail and outlook dot com accounts. In other words, user identities that are stored outside the azure active directory tenant for your subscription at the bottom there advanced settings that enable Token store, which is storage for the authentication tokens that are returned by these services is in

case your app needs to do anything against. One of the service is like post to users. Facebook Page. You can retrieve the token from the store, allow external redirect your L's has to do with more advanced scenarios, like using the mobile client SDK to authenticate a user to a specific provider like Google or Facebook and direct them back to your APP. First, we need to decide what action to take when a request is not authenticated. Choosing any of the log in options forces a user to log in before they can access the app, but it ties you to a specific provider. If you want to allow users to log in using one of several providers, like either as your Ad Facebook or Twitter, then you choose the first option shin. To allow anonymous requests, you still have to configure the providers you want, and then provide links on the home page of your app that lets the user choose. And then the links would redirect them to the built in endpoints used by APP service. Again, that's beyond the scope of this course, but that's one of the situations where you'd use this setting. So let's choose. Log in with Azure active directory and click to configure it. If we choose the advanced option, we have to set up the registration of this application in Azure Active Directory First. We've got a pretty simple authentication scenario here, though, so let's choose the express settings. This will create the APP registration for us in Azure A. D and make all the necessary connections will leave. Create New a. D AP selected, and we could change the name of the registered app in Azure a. D. If we wanted to show us something more friendly, let's just leave this. We'll click okay and save these authentication settings. Now let's go back to the overview page for the APP service and copy the URL for the Web app. Now I'll open a new private window so we don't need to worry about any cashed as your a D settings for my azure portal Log in and I'll paste in the Girl. I'll log in with the user that I know doesn't have any permissions in this Web app. And the first time I access this application as this user, I need to accept the permission request that the Apple wants to sign me in and read my profile information from Azure A. D. So I'll accept this and I brought into the application. Remember, I didn't have to write any code in my application to set this up. I'd have to write a little code If I want to get the user information from the headers, though, or in the case of a SP dot net, those credentials get passed into the user principal object, and you can use that in code.



Now website url asks for login



CREATING DEPLOYMENT SLOTS:



Understanding deployment slots

Configure deployment slot settings

Create a new deployment slot

**Deploy new version of application to
staging deployment slot**

Swap staging slot to production

Understanding deployment slots:



Swapping deployment slots

- Deploy v2 to staging slot
- Verify v2 in staging
- Swap staging (v2) with production (v1)
- v1 becomes staging

Can revert back (swap again) if issues

Warm up deployed application before swapping with production

The site is already warmed up when we swap staging to production , because swapping does not actually copy any files.it just swaps the DNS pointers.like below



Auto-swap

- Warms up deployed code
- Automatically swaps configured deployment slots

Traffic routing

- Route % of production traffic to another deployment slot
- Test features with limited users

Below settings also gets swapped along with the code

Deployment Slots Swap

Settings that Swap

General Settings (framework version, web socket setting, etc.)
Handler mappings
Monitoring and diagnostic settings
WebJobs content
Application Settings *
Connection Strings *

Setting that DO NOT Swap

Publishing endpoints
Custom domain names
SSL certificates
App Service scale settings
WebJob schedulers

* Can be configured to stay with



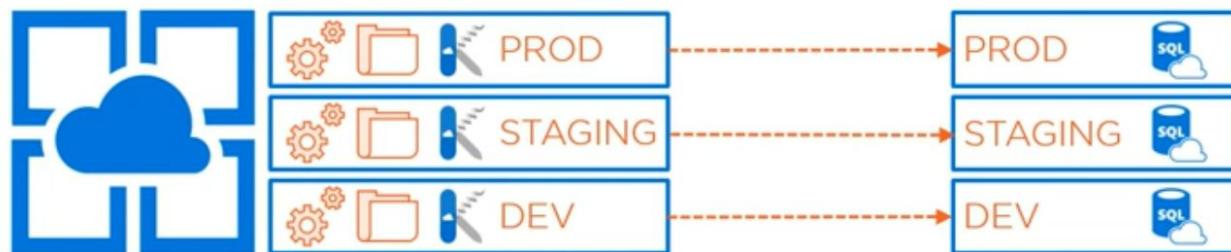
Deployment Slots

Swap with Preview (2 stage swap)

- Stage 1
 - Test staging deployment slot with production settings
- Stage 2
 - Complete swap
 - Or rollback swap

Schema changes in db might make code break in production.so plan for downtime until db changes made.

Updating Related Database Environment

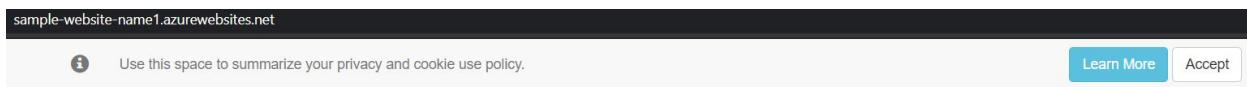


Deployment slots run on the same infrastructure as Production!

Conduct Load Testing in ANOTHER App Service Plan

since staging and prod is in same infrastructure, it can make prod slow when u do load test.so use separate app service plan.

Creating Deployment slots:



A screenshot of the Azure portal showing the 'Application Settings' for a deployment slot. The URL in the address bar is 'sample-website-name1.azurewebsites.net'. A privacy and cookie use policy banner is visible at the top. The 'Application Settings' section shows the following environment variables:

Setting	Value
Environment Name	LocalDev
Level 2 Setting	Dev setting
Non-sticky Setting	Set in DEV

Buttons for 'Learn More' and 'Accept' are visible in the top right corner of the banner.

Application Settings

- . Environment Name:
LocalDev
- . Level 2 Setting:
Dev setting
- . Non-sticky Setting:
Set in DEV

Connection Strings

- . Connection String:
Server=tcp:testdbnm.database.windows.net,1433;Initial Catalog=TESTING;Persist Security Info=False;User ID=uname;Password=pword;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;

In prod

Dashboard > sample-website-name1 > ASP-samplewebsitename1-866e | Apps > sample-website-name1 > ASP-samplewebsitename1-866e | Apps > sample-website-name1 | Configuration

sample-website-name1 | Configuration

App Service

Search (Ctrl+ /) Refresh Save Discard

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Security

Deployment

Quickstart Deployment slots Deployment Center

Settings

Configuration Authentication / Authorizati... Application Insights Identity Backups Custom domains

ANCM_ADDITIONAL_ERROR_PAGE_LINK APPINSIGHTS_INSTRUMENTATIONKEY APPLICATIONINSIGHTS_CONNECTION_STRING ApplicationInsightsAgent_EXTENSION_VERSION EnvironmentName Level1_Level2 NonStickySetting XDT_MicrosoftApplicationInsights_Mode

Hidden value. Click show values button App Config PROD Prod Setting Set in PROD Hidden value. Click show values button App Config

EnvironmentName PROD App Config ✓

Level1_Level2 Prod Setting App Config ✓

NonStickySetting Set in PROD App Config

XDT_MicrosoftApplicationInsights_Mode Hidden value. Click show values button App Config

Connection strings

Connection strings are encrypted at rest and transmitted over an encrypted channel.

New connection string Hide values Advanced edit Filter

Name	Value	Type	Deployment...	Delete
MyDBConnection	Server=PROD;Database=PROD;User Id=myUsername;Password=myPassword;	SQLAzure	✓	trash

Settings in azure will override those in the application

sample-website-name1.azurewebsites.net

Use this space to summarize your privacy and cookie use policy. Learn More Accept

Application Settings

- Environment Name: PROD
- Level 2 Setting: Prod Setting
- Non-sticky Setting: Set in PROD

Connection Strings

- Connection String: Server=PROD;Database=PROD;User Id=myUsername;Password=myPassword;

Create Deployment slot for staging:

sample-website-name1 | Deployment slots

Deployment Slots

Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots.

NAME	STATUS
sample-website-name1	PRODUCTION

Add Close

above > we have cloned settings from prod. We can also avoid if u want.

This will create new deployment slot, which is a new website on the same infrastructure and access that site from the same app service.

sample-website-name1 | Deployment slots

Deployment Slots

Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots.

NAME	STATUS	APP SERVICE PLAN	TRAFFIC %
sample-website-name1	PRODUCTION	ASP-samplewebsitename1-866e	100
sample-website-name1-Staging	Running	ASP-samplewebsitename1-866e	0

Staging (sample-website-name1/Staging)

Overview

Resource group (change) : sample-website-name1

Status : Running

Location : Central US

Subscription (change) : Visual Studio Professional Subscription

Subscription ID : 39317181-21dc-4d9f-bd69-e59e5ffba3db

Tags (change) : Click here to add tags

Authentication settings won't get copied/swapped from prod or between slots so staging not have authentication as production.

Name	Value	Deployment slot overriding	Delete	Edit
ANCM_ADDITIONAL_ERROR_PAGE_LINK	Hidden value. Click show values button ↗			
APPINSIGHTS_INSTRUMENTATIONKEY	Hidden value. Click show values button ↗			
APPLICATIONINSIGHTS_CONNECTION_STRING	Hidden value. Click show values button ↗			
ApplicationInsightsAgent_EXTENSION_VERSION	Hidden value. Click show values button ↗			
EnvironmentName	STAGING	✓		
Level1_Level2	Staging Setting	✓		
NonStickySetting	Set in STAGING			
XDT_MicrosoftApplicationInsights_Mode	Hidden value. Click show values button ↗			

Connection strings

Connection strings are encrypted at rest and transmitted over an encrypted channel.

+ New connection string Hide values Advanced edit Filter

Name	Value	Type	Deployment...	Delete	Edit
MyDBConnection	Server=STAGING;Database=STAGING;User Id=myUsername;Password=myPassword	SQLAzure	✓		

above>change settings in the staging slot

Application Settings

- Environment Name:
STAGING
- Level 2 Setting:
Staging Setting
- Non-sticky Setting:
Set in STAGING

Connection Strings

- Connection String:
Server=STAGING;Database=STAGING;User Id=myUsername;Password=myPassword;

now settings show different in staging slot , since settings in azure overrides the one with in the app.

Swapping Deployment Slots:

Objective: move staging code to production

PROD:

sample-website-name1.azurewebsites.net

 Use this space to summarize your privacy and cookie use policy.

Application Settings

- Environment Name:

PROD

- Level 2 Setting:

Prod Setting

- Non-sticky Setting:

Set in PROD

Connection Strings

- Connection String:

Server=PROD;Database=PROD;User Id=myUsername;Password=myPassword;

STAGING:

sample-website-name1-staging.azurewebsites.net

 Use this space to summarize your privacy and cookie use policy.

Application Settings

- Environment Name:

STAGING

- Level 2 Setting:

Staging Setting

- Non-sticky Setting:

Set in STAGING

Connection Strings

- Connection String:

Server=STAGING;Database=STAGING;User Id=myUsername;Password=myPassword;

SWAP:

Dashboard > sample-website-name1 > ASP-samplewebsitename1-866e | Apps > sample-website-name1

sample-website-name1
App Service

Search (Ctrl+ /) < Swap Stop Restart Delete Get published

Overview Activity log Access control (IAM) Tags Diagnose and solve problems Security

Deployment Quickstart Deployment slots Deployment Center

Settings Configuration Authentication / Authorization Application Insights Identity Backups Custom domains

Resource group (change) : sample-website-name1

Status : Running

Location : Central US

Subscription (change) : Visual Studio Professional Subscription

Subscription ID : 39317181-21dc-4d9f-bd69-e59e5ffba3db

Tags (change) : Click here to add tags

Diagnose and solve problems Our self-service diagnostic and troubleshooting experience helps you identify and resolve issues with your web app.

Application Insights Application insight diagnose quality issues with your web app.

Http 5xx Swap

Data In Swap

Source Changes Target Changes

SETTING	TYPE	OLD VALUE	NEW VALUE
NonStickySetting	AppSetting	Set in PROD	Set in STAGING

only NonStickySetting will get swapped since it is not set as deployment slot setting.

AFTER SWAP:

PROD:

sample-website-name1.azurewebsites.net

 Use this space to summarize your privacy and cookie use policy.

Application Settings

- Environment Name:

PROD

- Level 2 Setting:

Prod Setting

- Non-sticky Setting:

Set in STAGING

Connection Strings

- Connection String:

Server=PROD;Database=PROD;User Id=myUsername;Password=myPassword;

STAGING:

sample-website-name1-staging.azurewebsites.net

 Use this space to summarize your privacy and cookie use policy.

Application Settings

- Environment Name:

STAGING

- Level 2 Setting:

Staging Setting

- Non-sticky Setting:

Set in PROD

Connection Strings

- Connection String:

Server=STAGING;Database=STAGING;User Id=myUsername;Password=myPassword;

MANAGING CUSTOM DOMAINS:



Custom Domains

Overview of Custom Domains

Map a root domain to App Service using external DNS

Map a domain alias to App Service using external DNS

Purchase a Custom Domain in App Service

Purchase through a Domain Registrar

Purchase through Azure App Service

Custom Domains require paid pricing tier
>= Shared Tier



Domain Registrar



exampleorg.com ➔ 104.45.152.60





Azure DNS

Delegate DNS management to Azure
For hosting and managing DNS records
Point external registrar to Azure DNS
Azure Portal credentials, tools, billing
RBAC and Activity Logs
Purchase Custom Domain in App Service
 - **Automatically managed by Azure DNS**

App service is a multi tenant service where network infrastructure is shared.so, its only when u move to isolated envi tiers,u get dedicated networking.

IP Addresses in Azure App Service



App Service Environment

Isolated Pricing Tier

Static inbound/outbound IP
addresses



App Service

Inbound/outbound IPs can be different

App Service Plan apps share IP address

New static IP for app by creating IP-
based SSL binding

Dashboard > sample-website-name1 > ASP-samplewebsitename1-866e | Apps > sample-website-name1 | Properties

sample-website-name1 | Properties

App Service

Search (Ctrl+ /) < Status
Running

Push

MySQL In App

Properties

Locks

Export template

App Service plan

App Service plan

Quotas

Change App Service plan

Development Tools

Clone App

Console

Advanced Tools

App Service Editor (Preview)

Resource explorer

Extensions

Virtual IP address
104.43.254.102

Mode
Standard

Outbound IP addresses
104.43.254.102,23.99.191.39,104.43.218.218,23.99.178.238,104.43.209.26

Additional Outbound IP Addresses
104.43.254.102,23.99.191.39,104.43.218.218,23.99.178.238,104.43.209.26,23.99.177.6,23.99.180.234

Deployment Trigger URL
[https://\\$sample-website-name1:MbL4tnsFGXhiGSBirWqPHkEGw3sC2uJFN9pnvHYBtx3Eufrlp0affmZhTb8@sample-website-name1.scm.azurewebsites.net/deploy](https://$sample-website-name1:MbL4tnsFGXhiGSBirWqPHkEGw3sC2uJFN9pnvHYBtx3Eufrlp0affmZhTb8@sample-website-name1.scm.azurewebsites.net/deploy)



DNS Records

Root domain

- e.g. exampleorg.com

Subdomain

- admin.exampleorg.com
- www.exampleorg.com

Domain Name System (DNS) Servers

- “A Record” maps domain to an IP Address
- “CNAME Record” maps subdomain to another domain/subdomain
- “TXT Record” stores strings for any purpose

sample-website-name1 | Custom domains

Custom Domains

Configure and manage custom domains assigned to your app [Learn more](#)

IP address: 104.43.254.102

HTTPS Only: Off

Add custom domain

Status Filter: All (1) Not Secure (0) Secure (1)

SSL STATE: ASSIGNED CUSTOM DOMAINS

Secure: sample-website-name1.azurewebsites.net

SSL Binding

App Service Domains

Purchase and manage domains for your Azure services with auto-renew and privacy protection. [Learn more](#)

Buy Domain

register this ip, azure domain in ur domain registrar as 'Aname' record and txt record. Then keep ur custom domain in add section and validate.

easyDNS DOMAINS USER : Neil Morrissey HELP RENEW

Edit A Records :: exampleorg.com

A Records map host names to IP addresses. You can have more than one entry per host to activate round-robin service for that host.

Enter an "at-sign" (i.e. "@") on the left to indicate a A record for the root domain (e.g. "example.com"), or an asterisk (i.e. "*") to indicate a wild-card A record for your domain.

You can also enter "PARK" as the IP address to point your domain to our standard Parked Domain page.

When you have finished editing the domain information below, click on the **[next]** button.

hosts

A records

HOST	IP ADDRESS	DEL
@	exampleorg.com 104.45.152.60	
	exampleorg.com	

easyDNS DOMAINS USER : Neil Morrissey HELP RENEW

Edit TXT Records :: exampleorg.com

For assistance with setting up SPF or DKIM records, [please see our help page on TXT records](#).

When you have finished editing the domain information below, click on the **[next]** button.

TXT records

HOST	TEXT	DEL
@	.exampleorg.com ptestappnm.azurewebsites.net	

below>add cname records for staging

Edit CNAME Records :: exampleorg.com	
aliases	
CNAME records	
HOST	ADDRESS
test	exampleorg.com ptestappnm-staging.azurewebsites.net.
www	exampleorg.com exampleorg.com
www.test	exampleorg.com test.exampleorg.com

Add custom domain in staging app also like in prod

Below >add www. Subdomain in prod also

Home > App Services > ptestappnm - Custom domains

ptestappnm - Custom domains
App Service

Search (Ctrl+)

Refresh FAQs

SETTINGS

- Application settings
- Authentication / Authorization
- Application Insights
- Managed service identity
- Backups
- Custom domains**
- SSL settings
- Networking
- Scale up (App Service plan)
- Scale out (App Service plan)

Custom Hostnames

Configure and manage custom hostnames assigned to this site

IP address: 104.45.152.60

HTTPS Only: Off

Add hostname

HOSTNAMES ASSIGNED TO SITE

- exampleorg.com
- ptestappnm.azurewebsites.net

Add hostname

Add hostname
ptestappnm

* Hostname
www.exampleorg.com

Validate

Hostname record type
CNAME (www.example.com or any subdomain)

CNAME configuration

A CNAME record is used to specify that a domain name is an alias for another domain. In your scenario, that would be mapping www.exampleorg.com to ptestappnm.azurewebsites.net [Learn More](#)

CNAME
ptestappnm.azurewebsites.net

Add hostname

We can also buy custom domain in azure

This will be created under DNS zones in azure

mobileexample.com

DNS zone

Search (Ctrl +/)

Record set Move Delete zone Refresh

Resource group (change) ptestingrg

Subscription (change) Visual Studio Professional

Subscription ID [REDACTED]

Name server 1 ns1-06.azure-dns.com.

Name server 2 ns2-06.azure-dns.net.

Name server 3 ns3-06.azure-dns.org.

Name server 4 ns4-06.azure-dns.info.

Tags (change)

Click here to add tags

Search record sets

NAME	TYPE	TTL	VALUE
104.45.152.60	A	3600	ns1-06.azure-dns.com.
ns1-06.azure-dns.com.	NS	172800	ns2-06.azure-dns.net.
ns2-06.azure-dns.net.			ns3-06.azure-dns.org.
ns3-06.azure-dns.org.			ns4-06.azure-dns.info.
ns4-06.azure-dns.info.			Email: azuredns-hostmaster.microsoft.com
Email: azuredns-hostmaster.microsoft.com			Host: ns1-06.azure-dns.com.
Host: ns1-06.azure-dns.com.			Refresh: 3600
Refresh: 3600			Retrv: 300
Retrv: 300	SOA	3600	

mobileexample.com

one

Search (Ctrl +/)

Record set Move Delete zone Refresh

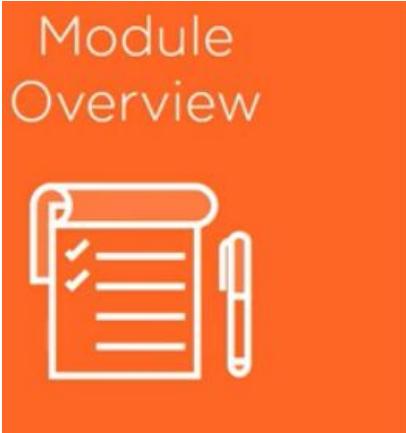
Tags (change)

Click here to add tags

Search record sets

NAME	TYPE	TTL	VALUE
104.45.152.60	A	3600	ns1-06.azure-dns.com.
ns1-06.azure-dns.com.	NS	172800	ns2-06.azure-dns.net.
ns2-06.azure-dns.net.			ns3-06.azure-dns.org.
ns3-06.azure-dns.org.			ns4-06.azure-dns.info.
ns4-06.azure-dns.info.			Email: azuredns-hostmaster.microsoft.com
Email: azuredns-hostmaster.microsoft.com			Host: ns1-06.azure-dns.com.
Host: ns1-06.azure-dns.com.			Refresh: 3600
Refresh: 3600			Retrv: 300
Retrv: 300	SOA	3600	Expire: 2419200
Expire: 2419200			Minimum TTL: 300
Minimum TTL: 300			Serial number: 1
Serial number: 1			www
www	CNAME	3600	psnmobileappnm.azurewebsites.net

ASSIGNING SSL CERTIFICATE:



Review SSL Certificate options

Upload certificate from external certificate authority

Re-enable App Service authentication of web app

Purchase App Service Certificate

below > conditions that certificate should meet



Basic pricing tier or higher

Upload your own certificate

- Signed by trusted certificate authority
- Password protected pfx file
- Private key length of at least 2048 bits
- Must contain all intermediate certificates

Complicated? We can do it easily using App service certificate

SSL Certificate Binding

IP-based SSL

Traditional approach

One certificate per IP address/Port combination

App Service assigns your site a new dedicated IP address

SNI-based SSL

Multiple certificates can secure the same IP address / port combination

Server Name Indication

Client must tell server hostname it's trying to access

Multiple websites can be hosted on same IP without using same certificate

Types of SSL Certificates

yourdomain.com



SSL Certificate

Subject Name = domain name
Subject Alternative Name (SAN)

*.yourdomain.com



Wildcard Certificate

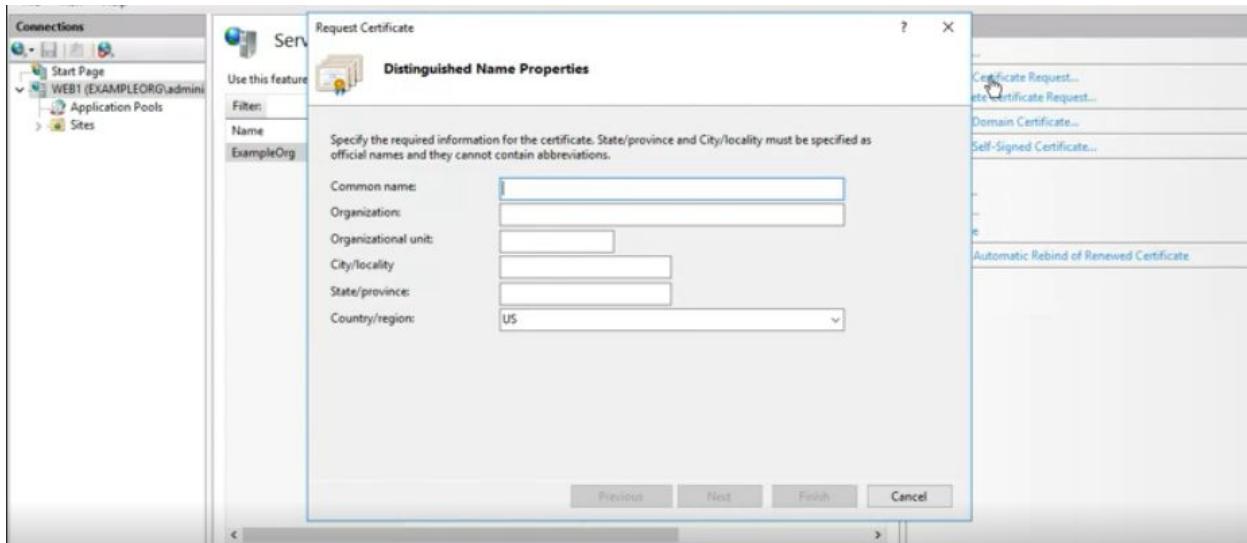
Matches all subdomains of root domain
More expensive up front

Name	Issued To	Issued By	Expiration Date
ExampleOrg	exampleorg.com	RapidSSL RSA CA 2018	9/3/2019 8:00:00 A...

right click>view

The 'View' option in the context menu is highlighted.

we can request a new certificate to concerned authority>they reply by email



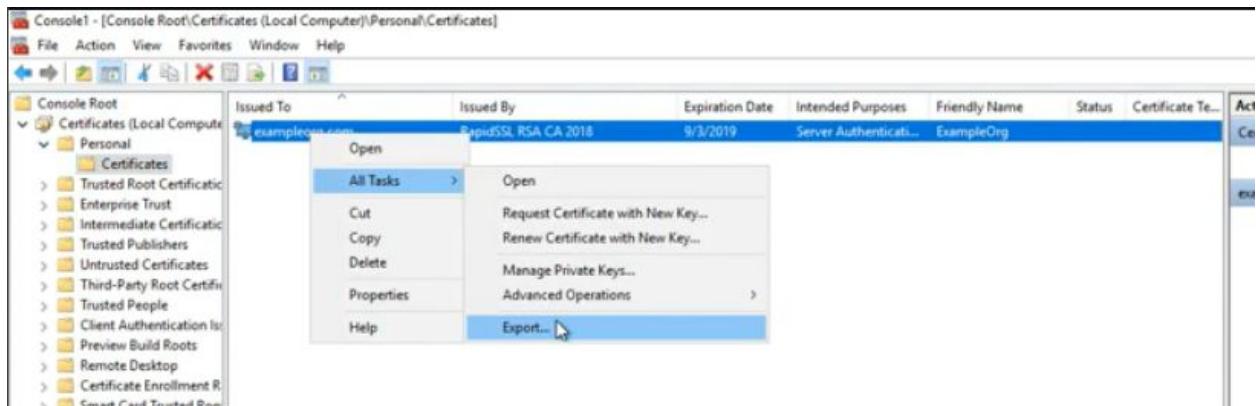
fill the details , it generates text file 1> copy that contents> navigate to the request form for the certificate authority and paste certificate request>they send u a confirmation email for the administrator account at the domain u r requesting the certificate for.>they email new certificate text along with the intermediate certificate>combine 2 certificates in a text file and upload back to the iis by clicking complete certificate request



EXPORT CERTIFICATE:

start>mmc>file>add snapin>certificates>computer account>next>finish

expand>personal store (where certificates are imported into)>select certificate >alltasks >export

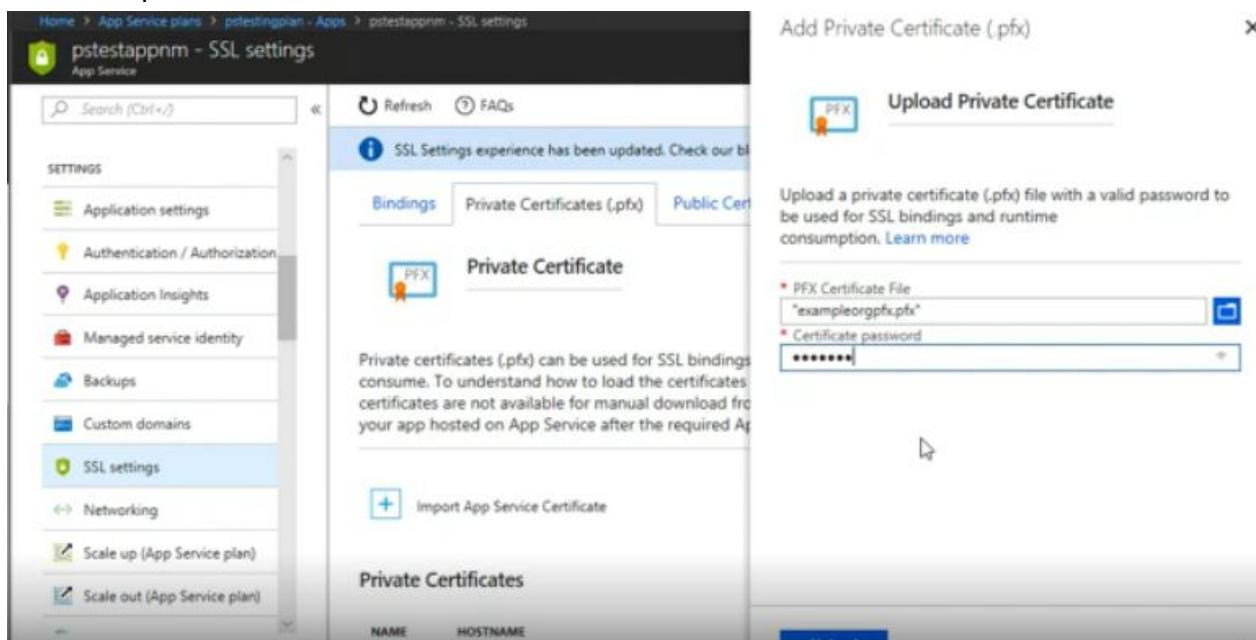


private key>yes

Export all extended properties>yes

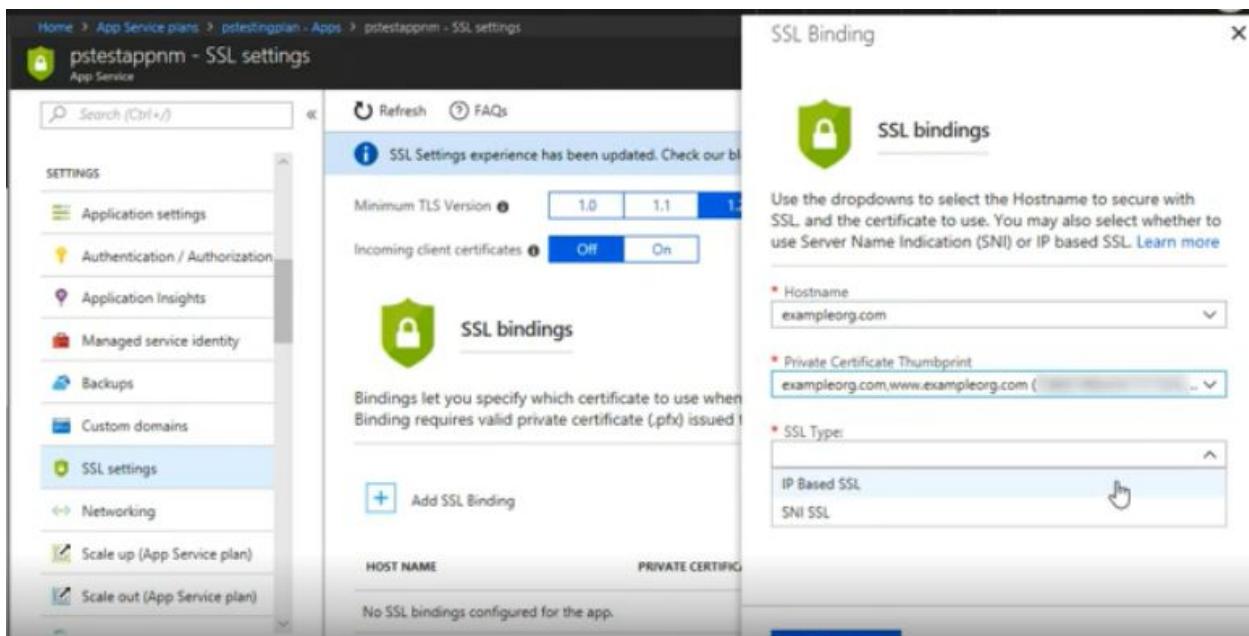
password>yes

Generate .pfx file



now bind this certificate to our custom domain.ip based ssl changes the already existing ip and keep dedicated ip address.

Do below for all domains and subdomains



Home > App Service plans > ptestappn - Apps > ptestappn - SSL settings

pptestappn - SSL settings

SETTINGS

- Application settings
- Authentication / Authorization
- Application Insights
- Managed service identity
- Backups
- Custom domains
- SSL settings**
- Networking
- Scale up (App Service plan)
- Scale out (App Service plan)

SSL Settings experience has been updated. Check our blog for more details →

Minimum TLS Version: 1.0 1.1 1.2

Incoming client certificates: Off On

SSL bindings

Bindings let you specify which certificate to use when binding requires valid private certificate (.pfx) issued to the host name.

Add SSL Binding

HOST NAME	PRIVATE CERTIFICATE
No SSL bindings configured for the app.	

SSL bindings

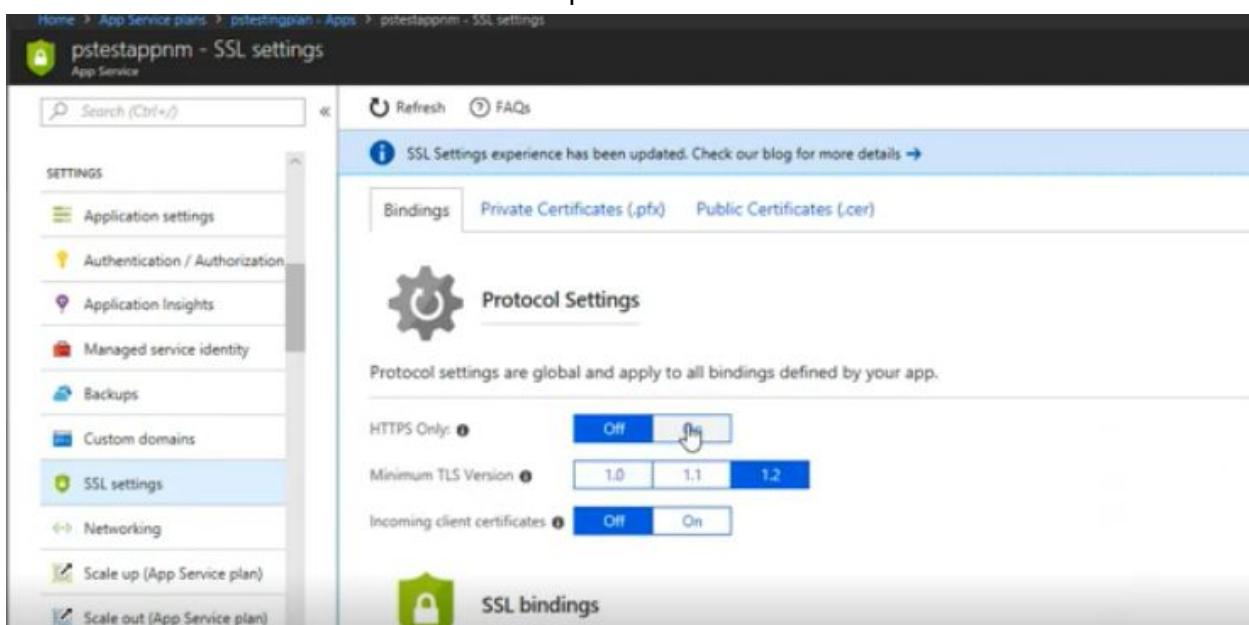
Use the dropdowns to select the Hostname to secure with SSL and the certificate to use. You may also select whether to use Server Name Indication (SNI) or IP based SSL. [Learn more](#)

* Hostname: exampleorg.com

* Private Certificate Thumbprint: exampleorg.com.www.exampleorg.com (exampleorg.com.www.exampleorg.com)

* SSL Type: IP Based SSL SNI SSL

now since we have certificate.so enforce https



Home > App Service plans > ptestappn - Apps > ptestappn - SSL settings

pptestappn - SSL settings

SETTINGS

- Application settings
- Authentication / Authorization
- Application Insights
- Managed service identity
- Backups
- Custom domains
- SSL settings**
- Networking
- Scale up (App Service plan)
- Scale out (App Service plan)

SSL Settings experience has been updated. Check our blog for more details →

Bindings Private Certificates (.pfx) Public Certificates (.cer)

Protocol Settings

Protocol settings are global and apply to all bindings defined by your app.

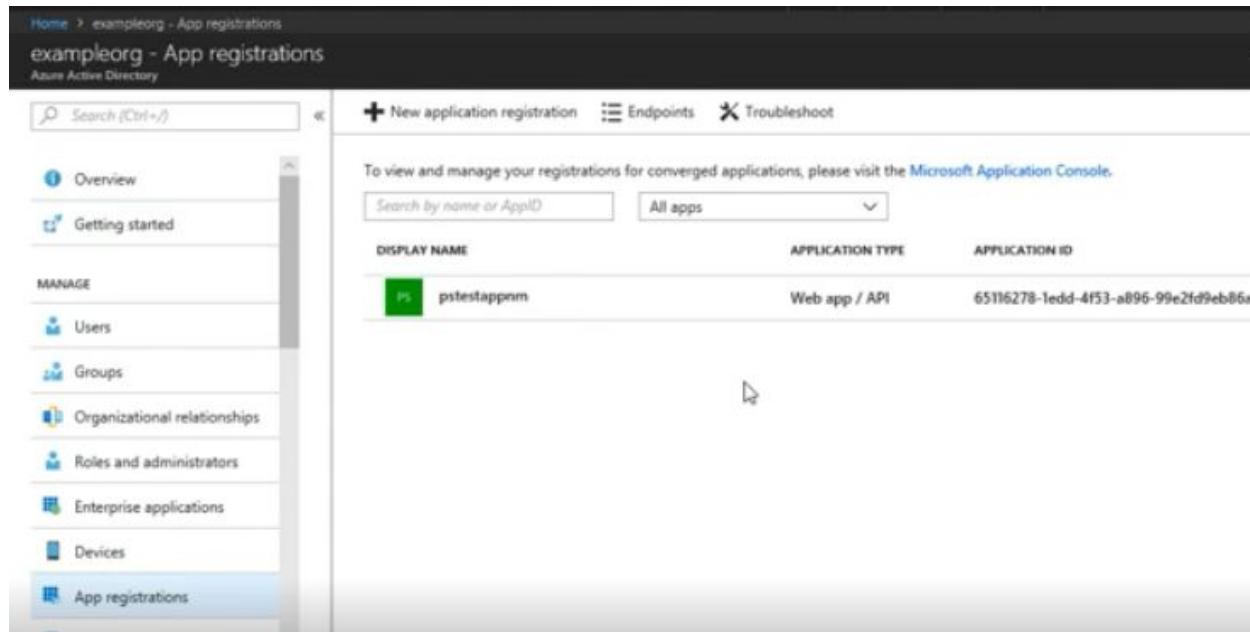
HTTPS Only: **On** Off

Minimum TLS Version: 1.0 1.1 1.2

Incoming client certificates: Off On

SSL bindings

Since we have https , now use azure ad.ie., got to Azure active directory

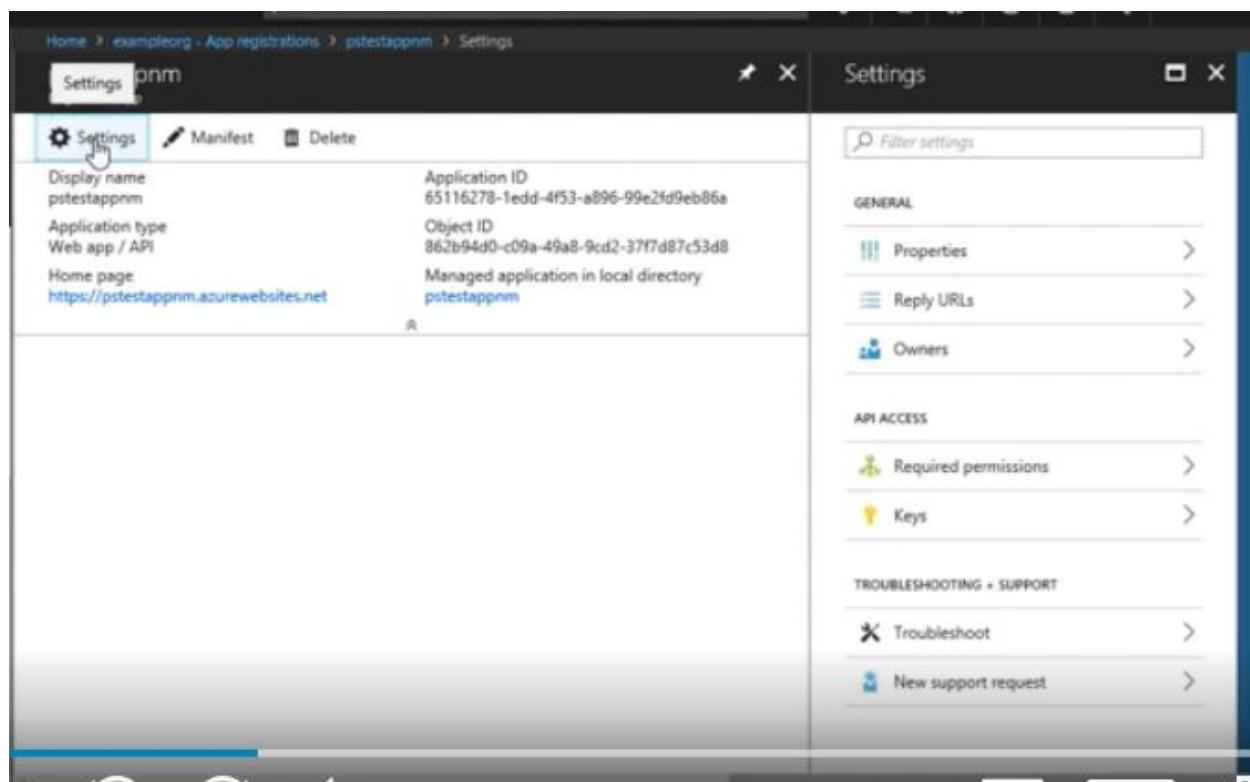


exampleorg - App registrations

New application registration Endpoints Troubleshoot

To view and manage your registrations for converged applications, please visit the [Microsoft Application Console](#).

DISPLAY NAME	APPLICATION TYPE	APPLICATION ID
pptestappnm	Web app / API	65116278-1edd-4f53-a896-99e2fd9eb86a



Home > exampleorg - App registrations > pptestappnm > Settings

Settings pnm

Settings Manifest Delete

Display name	pptestappnm	Application ID	65116278-1edd-4f53-a896-99e2fd9eb86a
Application type	Web app / API	Object ID	862b94d0-c09a-49a8-9cd2-37f7d87c53d8
Home page	https://pptestappnm.azurewebsites.net	Managed application in local directory	pptestappnm

Filter settings

GENERAL

- Properties
- Reply URLs
- Owners

API ACCESS

- Required permissions
- Keys

TROUBLESHOOTING + SUPPORT

- Troubleshoot
- New support request

Settings

Properties

GENERAL

- Properties
- Reply URLs
- Owners

API ACCESS

- Required permissions
- Keys

Name: ptestappnm

Object ID: 862b94d0-c09a-49a8-9c52-37f7d87c53d8

Application ID: 65116278-1edd-4f53-a896-99e2fd9eb86a

App ID URI: https://exampleorg.com

Logo: 

goto properties>

Home page URL: https://exampleorg.com

Home > exampleorg > App registrations > ptestappnm > Settings > Reply URLs

Settings

Reply URLs

Save

Update application urls

Successfully updated application ptestappnm urls

12:22 AM

GENERAL

- Properties
- Reply URLs
- Owners

https://exampleorg.com/auth/login/aad/callback

https://www.exampleorg.com/auth/login/aad/callback

goto azure ad>advanced

Home > ptestappnm > Authentication / Authorization > Azure Active Directory Settings

Azure Active Directory Settings

Management mode: Off

Client ID: 65116278-1edd-4f53-a896-99e2fd9eb86a

Issuer Url: https://sts.windows.net/

Client Secret (Optional): zu21CNgkj7MTLNVl65ZLETp0VC2Rej94zT/OXpVfjiE

ALLOWED TOKEN AUDIENCES

https://exampleorg.com/auth/login/aad/callback

https://www.exampleorg.com/auth/login/aad/callback

OK