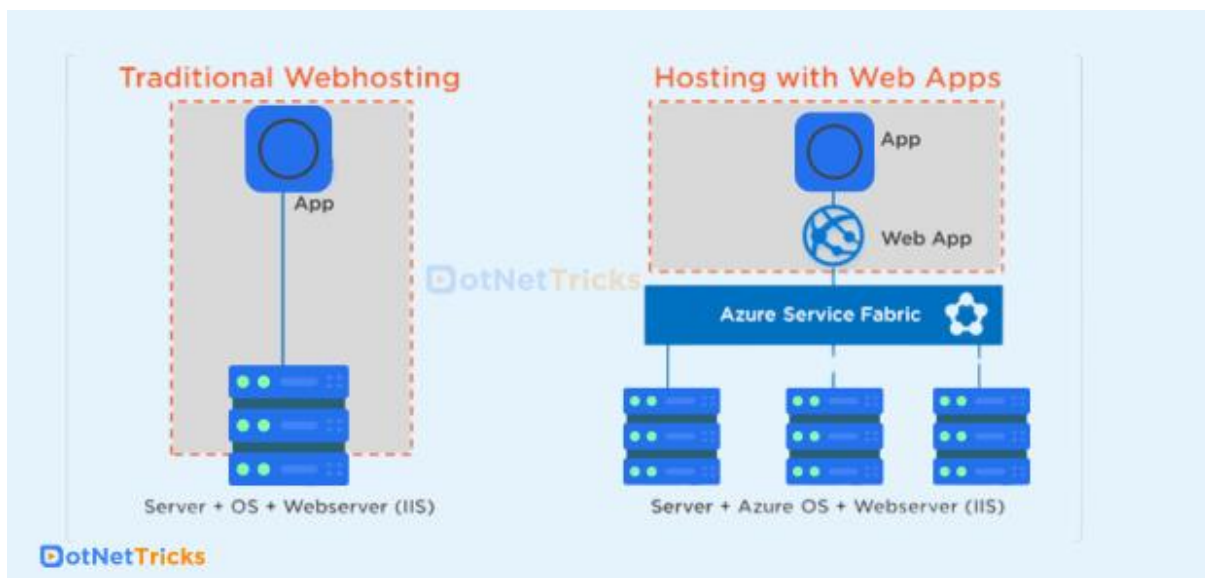


Azure Web App: An Overview

Web Apps are one of the most widely used Azure services that enable us to host our web applications in Azure. It can host a user-interactive application or even a backend service like a WCF Service or Web APIs. With Web Apps, we can leverage the power of [Microsoft Azure](#) and build multi-functional, immensely scalable, highly secured, and seamlessly accessible internet or intranet-based applications.

What we need to do is to push our application package into the web app and Azure App Service will take care of anything and everything required to keep it running. The most important advantage of using a web application is that we need not worry about the infrastructure plumbing that may be required to host the application. In traditional web hosting, we need to make sure that the Server is up, the OS is updated and the IIS is running. But in the case of web apps, Azure Service Fabric makes sure that the web app is allocated with all required computing power and resources.

Azure Web Apps provides an extensive platform to build an app in Azure without having to deploy, configure, and maintain your own [Azure virtual machines](#). We can build any web app using different languages such as ASP.NET, PHP, Node.js, and Python.



We can also build our applications on-premise and migrate them to Azure Web Apps.

Features of Web Apps

1. Auto Scale

Azure promises 99.95% availability in its SLA for web apps. In order to have a seamless experience for the users even in high demand, we can scale out or scale up the web apps instances to support the load. We can also configure auto-scale to web apps based on some metrics like CPU percentage, RAM Utilization, etc.

2. Easy Deployments

With Web apps, we can make use of the DevOps feature and we can deploy our applications directly from source code repositories like Visual Studio Team Services, Git Hub Bitbucket, etc.

3. Language Choice

Web apps support applications built on many languages and frameworks like ASP.Net, .Net Core, Java, PHP, Python, and Node.js.

The Azure web apps support multiple programming languages and frameworks, and some of the widely used languages are ASP.NET, Java, ASP.NET Core, Ruby on Rails, Nodejs, PHP, and Python. The PowerShell and other certain scripts or executables can also be used as background services.

4. Security

We can integrate user identity provider services and enhance the capability of web applications. For example, we can use Azure AD to maintain a user database, authenticated login, and authorization.

The Azure app service complies with the security standards such as ISO, SOC, and PCI thus the level of security is extremely high. Apart from that the end-user or customer can log in to their account by using Azure Active Directory, Facebook, Google, Twitter, or the Microsoft accounts.

5. Diagnostics

Azure App Services provides diagnostics logs for both application level and server level. It supports

- **Application logging** - This is basically the logs written using the code or even uncaught exceptions.
- **Web server logging** -It contains the logs of HTTP requests which hit the web app.
- **Detailed error messages** - Detailed error messages contain detailed information about each failed request with an HTTP status code.
- **Failed request tracing** - This keeps track of IIS steps and modules the request goes and captures generally the requests that respond with HTTP Status code 500 or above.

Web App Deployment Slots

We can deploy the web apps in separate deployments slots to ensure that the application runs as expected before it goes live in production. Deployment slots can be created in standard or premium app service plans. Deployment slots are containers to host the apps with their own IP address and content and configuration.

Benefits of Deployment Slots

1. **Minimum downtime for the production website:** With the Staging slot in place we can push our changes to the staging slot and test if everything works as expected and then easily swap the staging slot with production. Since Azure does this activity by changing the virtual IP of slots, it completes in seconds and the latest code changes reflect in the live website with minimum impact on the business.

Managing the minimum downtimes comes with an advantage where we have a huge scope for better service catering capabilities to the customer all the time without any downtime of the other server-related issues.

2. **Rollback Deployment:** Now that we deployed a new version of the app by swapping the staging slot to production, we can easily revert by again swapping the production slot with the staging slot.

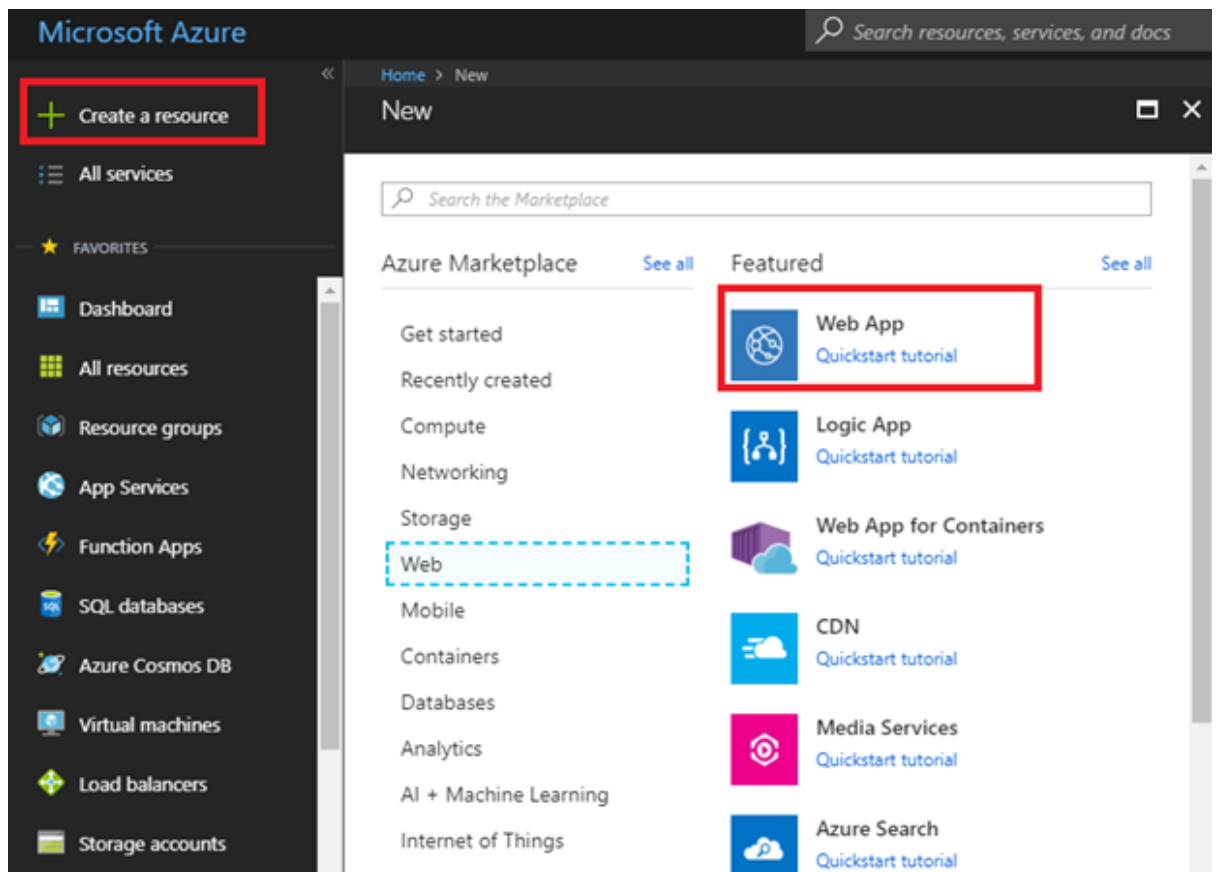
Rollback allows us to quickly swap the previous working version of the web in a live environment hence the way of managing infrastructure is capable of all the rollback deployment capabilities.

3. **Effective Testing:** Deployment slots provide a real-time environmental setup like that of a production website in Azure so, we can do effective testing on the staging environment and there will be fewer chances of issues where anything breaks in production.

The number of deployments slots is different for each app service plan. We do not have a deployment slots option with Free, basic, and Shared App Service plans. The standard App Service plan supports maximum deployment slots up to 5 whereas this can go as high as 20 in the Premium app service plan.

Step by Step: Creating a Web App

1. Log in to the Azure portal.
2. Click on Create a resource and select web from the marketplace.



3. Click on Web App.
4. Fill in the name, resource group, and app service plan.

FAVORITES

Dashboard

All resources

Resource groups

App Services

Function Apps

SQL databases

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

* App name

dotnettrickswebapp

.azurewebsites.net

* Subscription

Visual Studio Enterprise – MPN

* Resource Group

☒ Create new
 ☐ Use existing

dotnettrickswebapp

* OS

WindowsLinuxDocker

* App Service plan/Location

ServicePlan78e09da7-a6f8(Centr...

Application Insights

OnOff

CreateAutomation options

- Go to All resources in the left panel to see the list of created resources.

Create a resource

All services

FAVORITES

Dashboard

All resources

Resource groups

App Services

Function Apps

SQL databases

Azure Cosmos DB

Home > All resources

All resources

Subscription filtering behavior has now changed. To learn more click here.

Subscriptions: 1 of 2 selected - Don't see a subscription? Open Directory + Subscription settings

Filter by name:

Visual Studio Enter...

All resource groups

All types

All locations

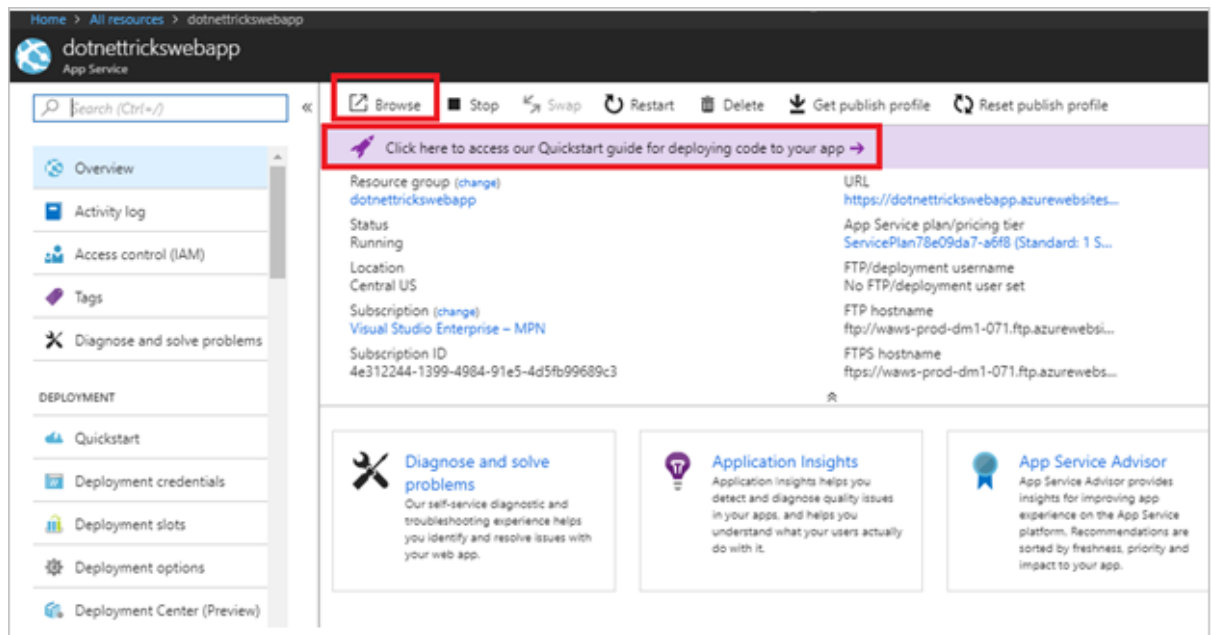
No filter

No grouping

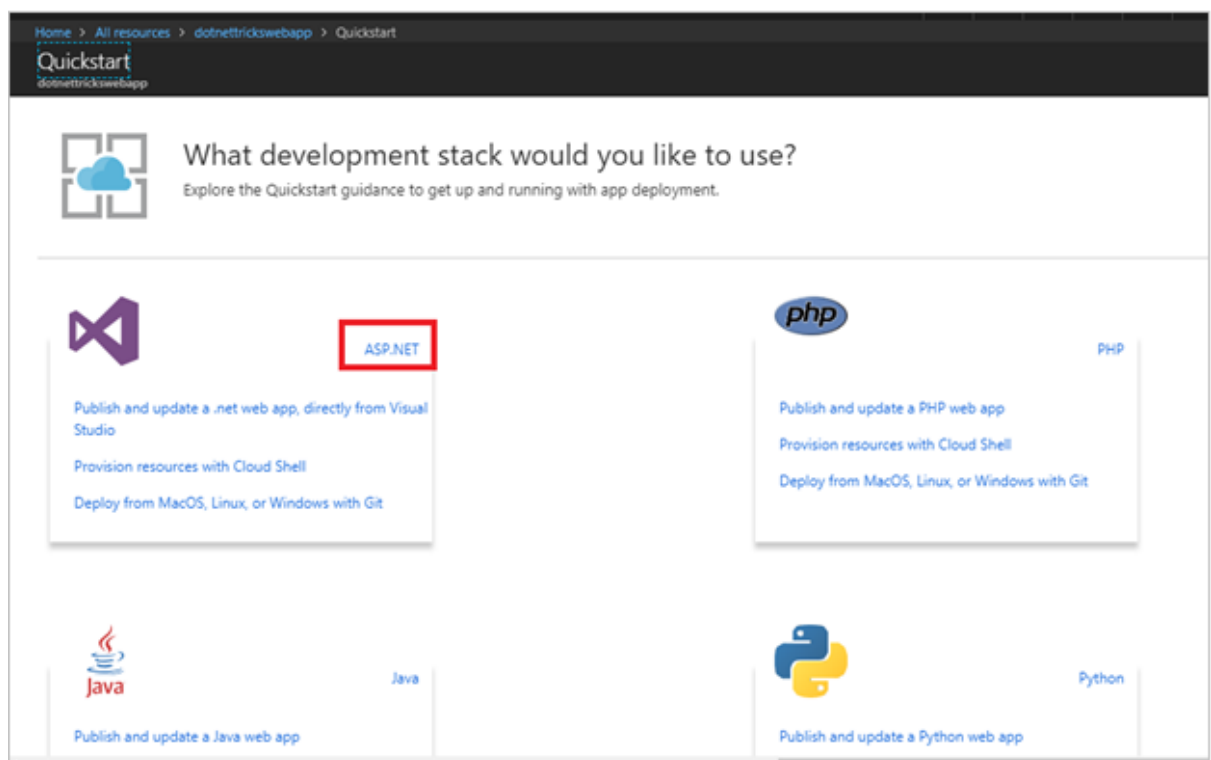
2 items ☐ Show hidden types

	NAME	TYPE	RESOURCE GROUP	LOCATION	SUBSCRIPTION
<input type="checkbox"/>	dotnettrickswebapp	App Service	dotnettrickswebapp	Central US	Visual Studio Enterprise - ...
<input type="checkbox"/>	ServicePlan78e09da7-a6f8	App Service plan	dotnettrickswebapp	Central US	Visual Studio Enterprise - ...

- Click on the Web app name.
- Then click on the Browse button to open the web app in the browser.



8. Click on the purple ribbon as highlighted above to deploy a source code into the web app.



9. Click on your desired development stack and follow the guide.

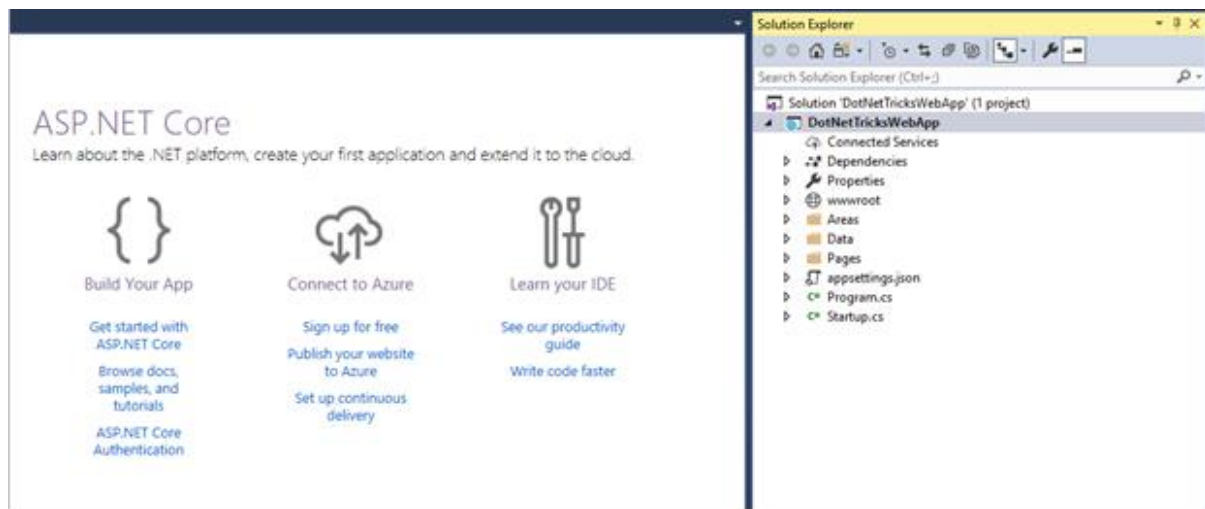
Deploying Asp.Net Core App to WebApp

We can deploy an Asp.Net Core app in multiple ways

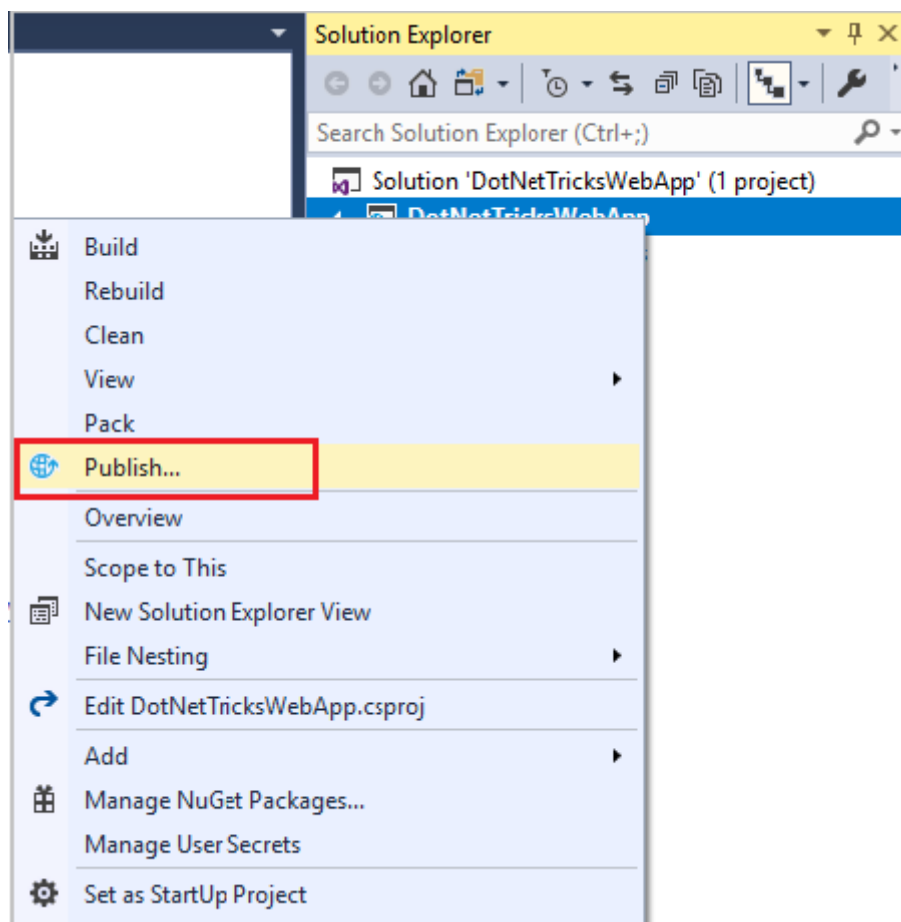
1. Publishing to Azure through Visual Studio

We can directly deploy an application without going to the Azure portal.

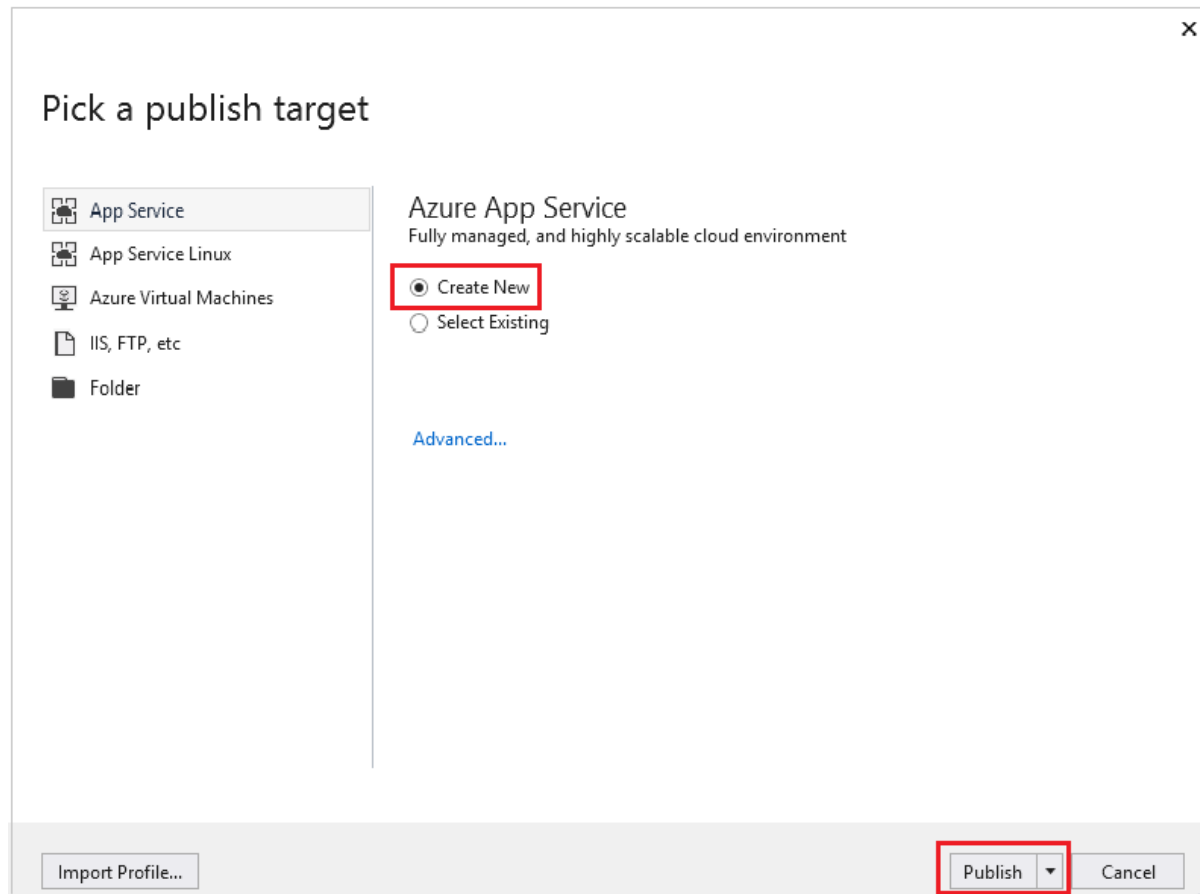
- Create a new Asp.Net Core application in Visual Studio.



- Right-click on the project file and select publish.



- In Publish target window Select the 'Create New' option and click on Publish button.



- In Create App Service dialogue box, fill in the Azure Subscription.

Create App Service
Host your web and mobile applications, REST APIs, and more in Azure

App Name
DotNetTricksWebApp20180902091930

Subscription
Visual Studio Enterprise - MPN

Resource Group
cloud-shell-storage-centralindia (centralindia) [New...](#)

Hosting Plan
DotNetTricksWebApp20180902091930Plan* (Central US, S1) [New...](#)

[Export...](#) [Create](#) [Cancel](#)

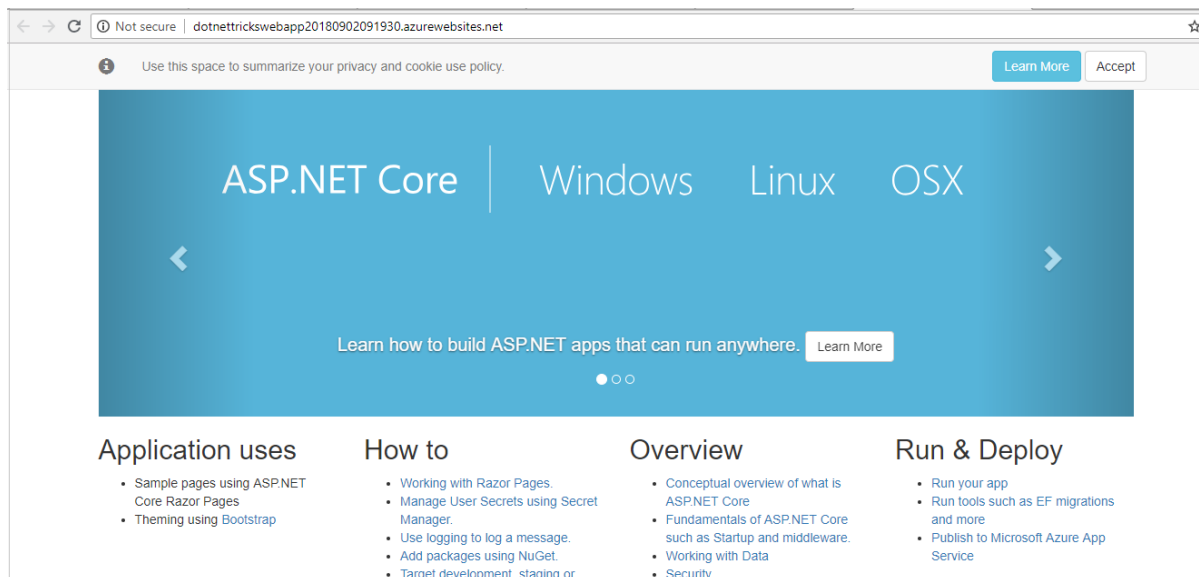
Explore additional Azure services

- [Create a SQL Database](#)
- [Create a storage account](#)

Clicking the Create button will create the following Azure resources

- Hosting Plan - DotNetTricksWebApp20180902091930...
- App Service - DotNetTricksWebApp20180902091930

- The Visual studio starts creating the web app in Azure and deploys the application.



2. Continuous Deployment with Visual Studio

- Create a web app in the Azure portal as described in step by step section above.
- Create an Asp.Net Core application in Visual Studio.
- Login to Azure Portal and select the web app created in the previous section.
- In the deployment, blade Select the Deployment option.

Home > All resources > DotNetTricksWebApp20180902091930

DotNetTricksWebApp20180902091930

App Service

Search (Ctrl+*/*)

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

DEPLOYMENT

- Quickstart
- Deployment credentials
- Deployment slots
- Deployment options**
- Deployment Center (Preview)

Click here to access Application Insights for monitoring and profiling for your ASP.NET Core app. →

Resource group (change) cloud-shell-storage-centralindia	URL https://dotnettrickswebapp20180902091930...
Status Running	App Service plan/pricing tier DotNetTricksWebApp20180902091930Pla...
Location Central US	FTP/deployment username No FTP/deployment user set
Subscription (change) Visual Studio Enterprise – MPN	FTP hostname ftp://waws-prod-dm1-099.ft.azurewebsites...
Subscription ID 4e312244-1399-4984-91e5-4d5fb99689c3	FTPS hostname https://waws-prod-dm1-099.ft.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.

○ C

Home > All resources > DotNetTricksWebApp20180902091930 > Deployment option > Choose source

Deployment option

Set up deployment option

Choose Source
Configure required settings

Visual Studio Team Services
By Microsoft

OneDrive
By Microsoft

**Local Git Repository
By Git**

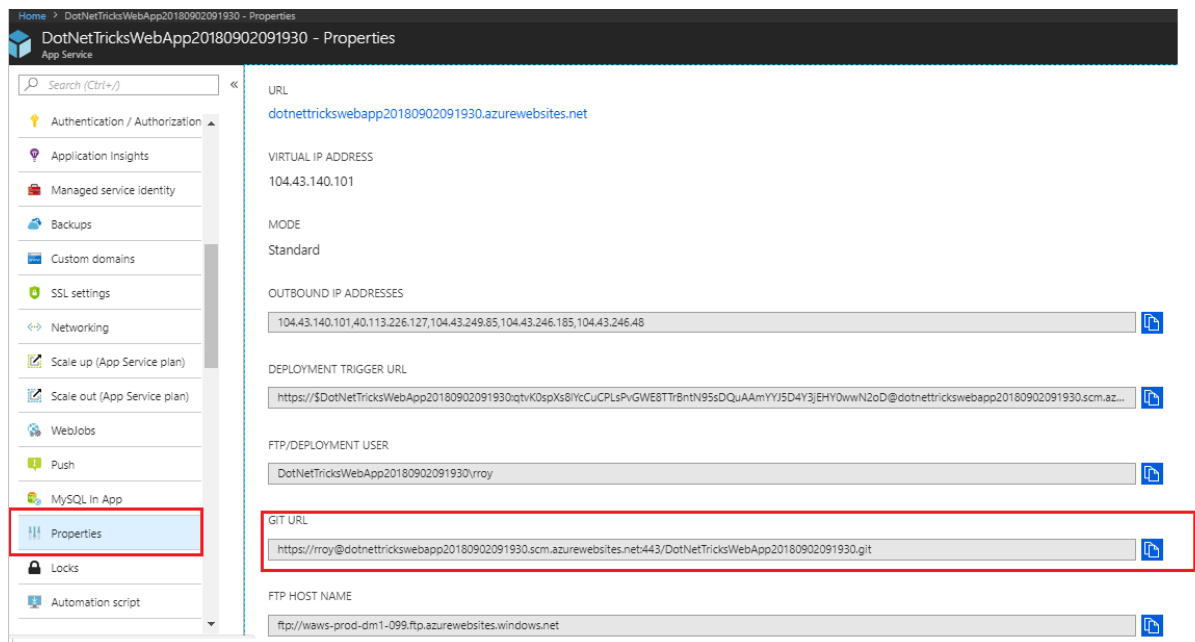
GitHub
By GitHub

Bitbucket
By Atlassian

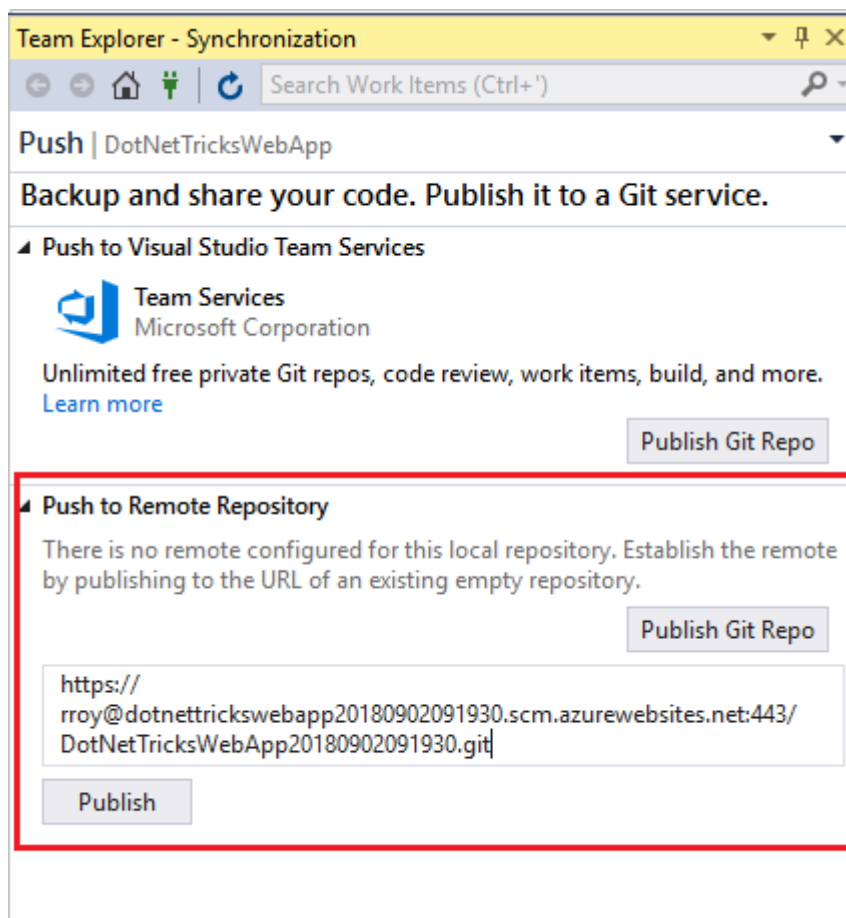
Dropbox
By Dropbox

External Repository

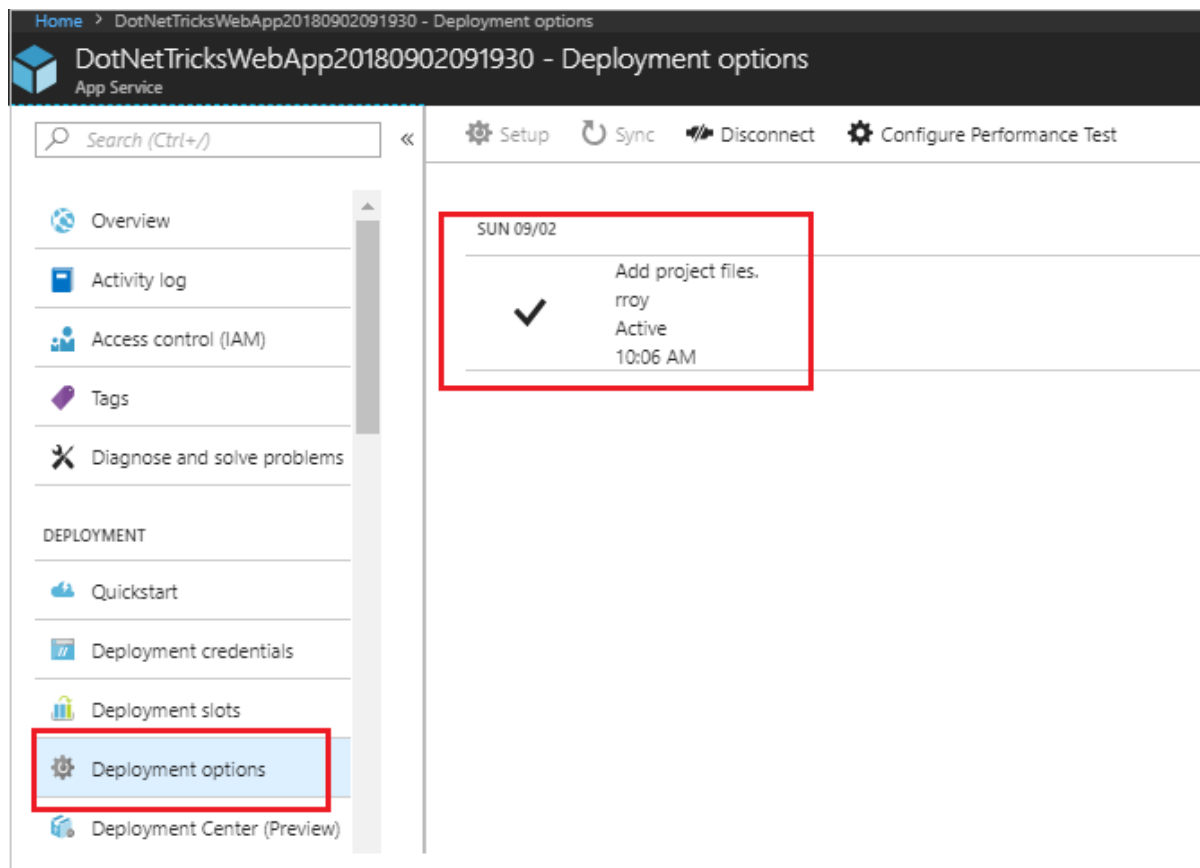
- Set up deployment credentials and click ok.
- In the web app blade, go to the Settings section and click on properties.
- Copy the Git URL.



- Configure the git repository for the local source code in Visual Studio by adding a remote repository Git URL.



- Verify that the web app is configured with a local git repository.



- Now we can make changes in our source code and push the changes to the Git repository. As soon as we push changes, the web app will reflect the updated version.

3. Continuous Deployments to Azure with VSTS

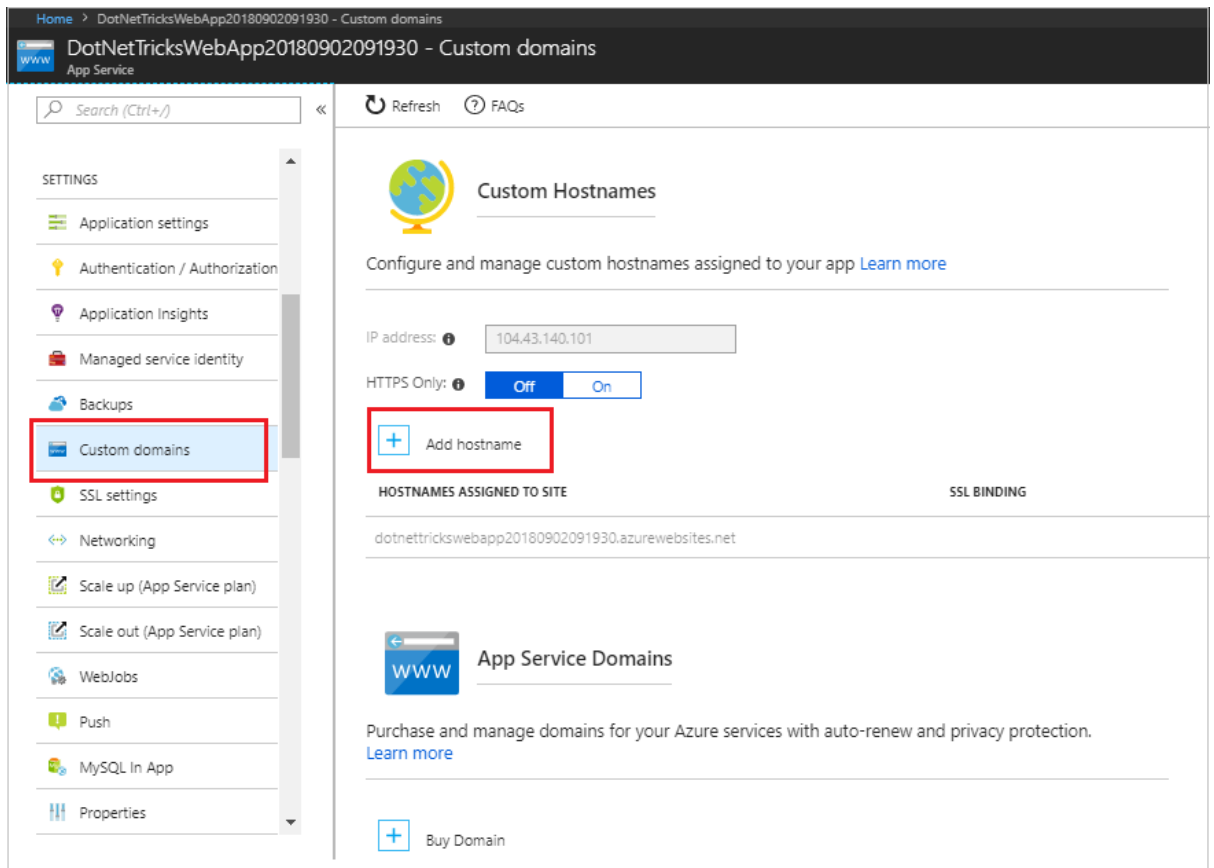
Visual Studio Team Services is Microsoft's DevOps Solution to configure continuous integration (CI) and continuous delivery (CD) pipeline for an application. With a DevOps Project in Azure, we can easily configure the CI and CD of an application and deploy an application into the web app with every check into the VSTS repository. This will be discussed in detail in a separate article.

Setting Custom Domains

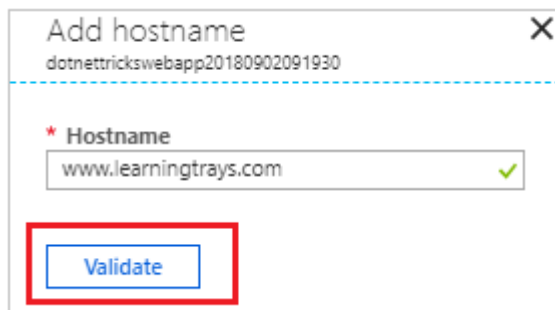
By default, Azure web apps have an URL in the format <web app name>. azurewebsite.net. We can add custom domain names to the web apps. For this, we should buy a domain from one of the domain name service providers like Godaddy.com.

Steps to Add Custom domain to web app

1. Go to the Web App in the Azure portal.
2. In Settings Section on the left-hand blade, Select Custom Domains and click on Add hostname.



3. In the Add hostname panel, provide the domain name and click on validate.



4. If the domain name is available to be mapped to the default domain, the host name record type will be automatically selected as the CNAME type.

Add hostname
dotnettrickswebapp20180902091930

* **Hostname**
 ✓


[Validate](#)

Hostname record type

CNAME (www.example.com or any subdomain) ^

A record (example.com)

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.learningtrays.com to dotnettrickswebapp20180902091930.azurewebsites.net [Learn More](#)


CNAME


In case, we want to map a reserved IP address of an application to map with a custom domain, we would go for the 'A record' type.

5. In the below section, we get a warning to add a CNAME record to the DNS provider site.

 **DNS propagation**

Please be aware that depending on your DNS provider it can take up to 48 hours for the DNS entry changes to propagate. You can verify that the DNS propagation is working as expected by using <http://digwebinterface.com/>. [Learn more](#)

 **Hostname availability**

 **Domain ownership**





Create an CNAME record with your DNS provider using the configuration below. [Learn more](#)

Type	Host	Value
CNAME	www or subdomain	dotnettrickswebapp20180902091930.azurewebsites.net

6. Add a record to DNS Settings in the DNS provider site as depicted below.

Records

Last updated 02-09-2018 11:45 AM

Type	Name	Value	TTL	
A	@	Parked	600 seconds	
CNAME	ftp	@	1 Hour	
CNAME	www	dotnettrickswebapp20180902091930.azurew...	1 Hour	
CNAME	_domainconnect	_domainconnect.gd.domaincontrol.com	1 Hour	
NS	@	ns15.domaincontrol.com	1 Hour	
NS	@	ns16.domaincontrol.com	1 Hour	
SOA	@	Primary nameserver: ns15.domaincontrol.com.	600 seconds	

[ADD](#)

It may take a couple of minutes to propagate the change. Go back to the Azure portal and click on Validate button again.

7. Once the Domain ownership is validated a green tick mark will appear. Click on Add hostname button.

Add hostname

dotnettrickswebapp20180902091930


*** Hostname**

www.learningtrays.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.learningtrays.com to dotnettrickswebapp20180902091930.azurewebsites.net [Learn More](#)

CNAME

dotnettrickswebapp20180902091930.azurewebsites.net

Add hostname

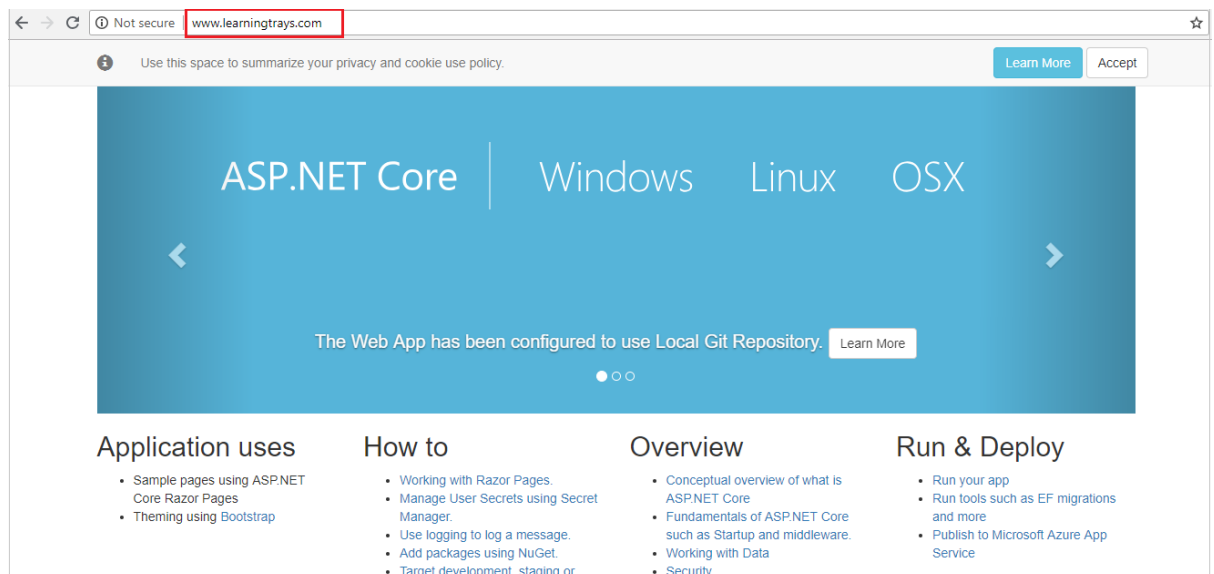
✓

Hostname availability

✓

Domain ownership

- Go to the browser and navigate to the custom URL mapped.



Summary

With Azure features of Auto Scaling, multiple deployment slots, many ways to deploy web applications, detailed diagnostics, and enhanced security along with load balancing with traffic manager, Web Apps have become the obvious choice for many enterprise applications. [Azure's online training](#) program helps you gain proficiency in Microsoft's various Azure services to earn Azure certification.