



Microsoft Cloud Workshop

Azure Stack

Hands-on lab step-by-step

May 2018

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2018 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <https://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/Usage/General.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Contents

Azure Stack hands-on lab step-by-step	1
Abstract and learning objectives.....	1
Overview	1
Solution architecture.....	2
Help References.....	3
Before the hands-on lab.....	4
Task 1: Create a virtual machine to execute the lab.....	4
Task 2: Install the Azure Stack Developer Kit.....	6
Task 3: Install PowerShell for Azure Stack.....	7
Task 4: Download the Latest Azure Stack Tools.....	9
Task 5: Register Azure Stack with Azure AD.....	10
Task 6: Download VM Images to Azure Stack Marketplace.....	11
Exercise 1: Configure Azure Stack.....	15
Task 1: Deploy the Azure Stack App Service Resource Provider	15
Task 2: Deploy the Azure Stack SQL DB Resource Provider.....	29
Task 3: Create Azure Stack Deployment Taxonomy for Tenets	35
Exercise 2: Deploy the SQL Hosting Server and DB on Azure Stack.....	45
Task 1: Create SQL Hosting Server.....	45
Task 2: Register resource providers in the tenant subscription.....	47
Task 3: Deploy SQL DB on Azure Stack.....	47
Exercise 3: Deploy Contoso Financial Web Application	52
Task 1: Create the Web App.....	52
Task 2: Provision an Azure Storage Account.....	53
Task 3: Update the configuration strings.....	56
Task 4: Publish the Contoso Financial Web Application	57
Exercise 4: Deploy the customer offers Web API	59
Task 1: Provision the offers Web API App.....	59
Task 2: Deploy the Contoso.Apps.Financial.Offers project.....	60
Task 3: Update the Application Settings of the Web App with the API URL	61
Exercise 5: Automating backend processes with Azure functions	64
Task 1: Create an Azure function to generate PDF receipts.....	64
Exercise 6: Deploy Contoso Finance Admin website.....	69
Task 1: Provision the Contoso Finance Admin Web App.....	69
Task 2: Deploy the call center admin Web App from Visual Studio.....	70
After the hands-on lab	75

Azure Stack hands-on lab step-by-step

Abstract and learning objectives

This workshop is designed to teach attendees how to design a hybrid cloud architecture using a combination of the Azure public cloud and Azure Stack. This functional architecture will enable customers to leverage their investments in Azure as a "cloud platform," rather than Azure as a "place." Attendees will learn to determine which systems are good candidates for the Azure public cloud and which are better suited on Azure Stack.

After completing workshop, attendees will be better able to recommend and design hybrid cloud systems that leverage one application and deployment model: Azure.

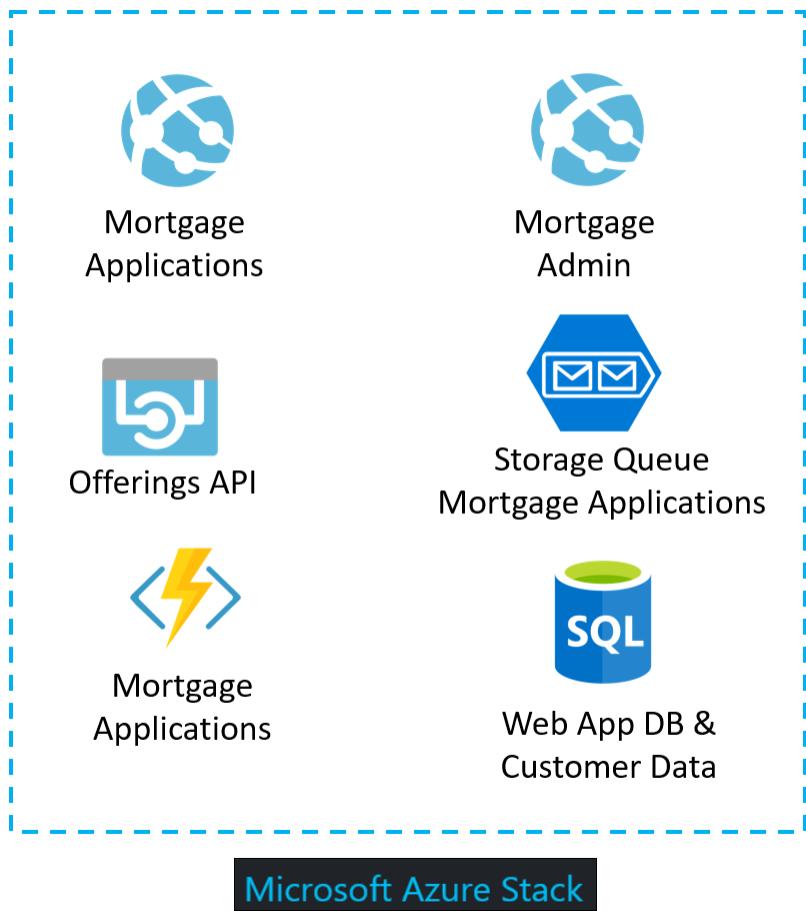
In addition, attendees will learn to:

- Understand when the Azure public cloud versus Azure Stack is appropriate based on customer requirements.
- Describe possible integrations between Azure public cloud solutions and Azure Stack
- Understand the taxonomy of Azure Stack: tenants, regions, subscriptions, offers, plans, services and quotas.
- Describe the resource providers that are available for use with Azure Stack
- Design and deploy hybrid connectivity between Azure public cloud and Azure Stack.

Overview

In this hands-on lab, you will deploy the Azure Stack Development Kit and deploy the SQL Database and Azure App Service resource providers as well as download several virtual machine images from the Azure Stack Marketplace. From there, you will implement a full taxonomy in Azure Stack consisting of a region, subscription, plan, offer, and quotas. After Azure Stack is configured, you will then deploy Azure SQL Database, Web and API apps and then deploy the Contoso application.

Solution architecture



Help References

Description	Links
Azure Stack overview	https://azure.microsoft.com/en-us/overview/azure-stack/
Azure Stack use cases	https://azure.microsoft.com/en-us/overview/azure-stack/use-cases/
Azure Stack features	https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-key-features
Azure Stack planning considerations	https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-planning-considerations
Azure Stack documentation	https://docs.microsoft.com/en-us/azure/azure-stack/
Azure Stack operator documentation	https://docs.microsoft.com/en-us/azure/azure-stack/
Azure Stack networking	https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-network-overview
Azure Stack to Azure Global VPN	https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-connect-vpn
Register Azure Stack with your subscription	https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-register
Azure Stack with App Service – Before you get started	https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/azure-stack/azure-stack-app-service-before-you-get-started.md
Deploy App Service on Azure Stack	https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-app-service-deploy
Deploy the Azure Stack SQL Resource Provider	https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-sql-resource-provider-deploy#deploy-the-resource-provider
White paper	https://azure.microsoft.com/en-us/resources/azure-stack-an-extension-of-azure/
PowerShell for Azure Stack	https://docs.microsoft.com/en-us/azure/azure-stack/user/azure-stack-powershell-install
Azure Stack marketplace	https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-marketplace-azure-items

Before the hands-on lab

Duration: 6-7 hours

To execute this lab, you will have two options: you can use your own Azure Stack Development Kit that is already installed, or you will need to follow the instructions below to setup your own in Microsoft Azure using nested virtual machines.

For help with installation of the Azure Stack Development Kit, review the following article: <https://azure.microsoft.com/en-us/overview/azure-stack/development-kit>.

Task 1: Create a virtual machine to execute the lab

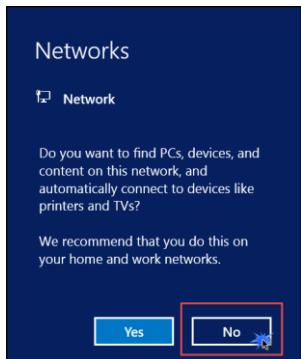
1. Launch a browser, and navigate to <https://github.com/opsigility/cw-azure-stack> and click the **Deploy to Azure** button.
2. Specify a resource group name **AzureStack** and deploy the template.

NOTE: Please wait for the virtual machine to be provisioned prior to moving to the next step.

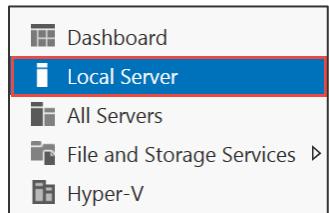
3. After the VM is provisioned, **connect** to establish a new Remote Desktop Session.
4. Depending on your Remote Desktop Protocol Client and browser configuration, you will either be prompted to open an RDP file, or you will need to download it and then open it separately to connect.
5. Log in with the credentials specified during creation:
 - a. User: **administrator**
 - b. Password: **demo@pass123**
6. You will be presented with a Remote Desktop Connection warning because of a certificate trust issue. Click **Yes** to continue with the connection.



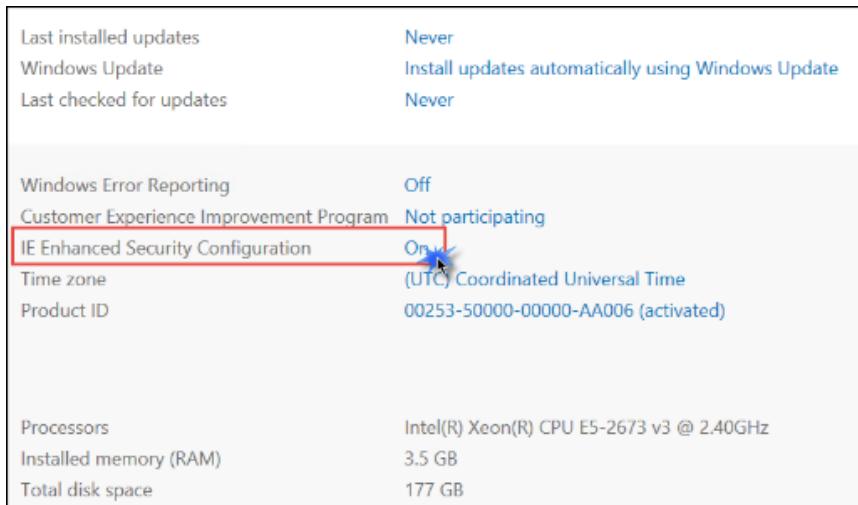
7. When logging on for the first time, you will see a prompt on the right asking about network discovery. Click **No**.



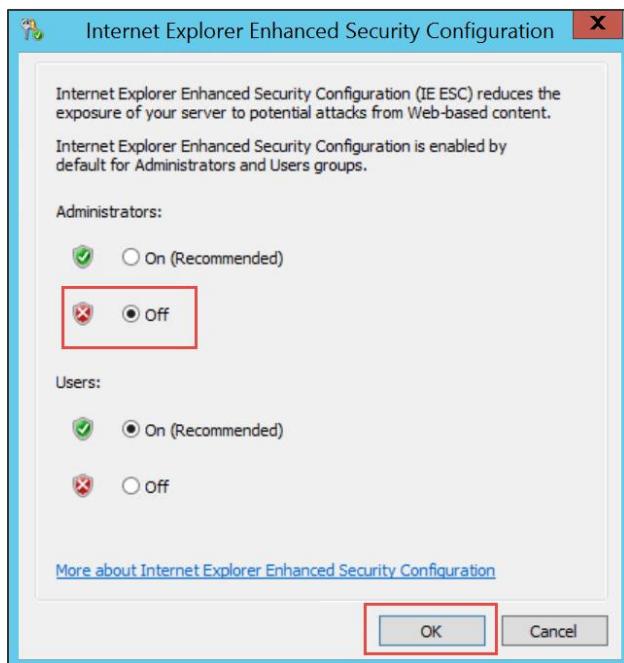
8. Notice that Server Manager opens by default. On the left, click **Local Server**.



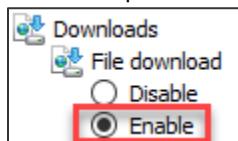
9. On the right side of the pane, click **On** by **IE Enhanced Security Configuration**.



10. Change to **Off** for Administrators and click **OK**.



11. Launch Internet Explorer, and right click at the top of the browser and check Menu bar. Then open Tools -> Internet Options -> Security -> Custom Level. Change File download to **Enable**.



Task 2: Install the Azure Stack Developer Kit

1. Launch an elevated PowerShell console and execute the following commands:

```
Cd C:\AzureStackOnAzureVM  
.\\Install-ASDK.ps1
```

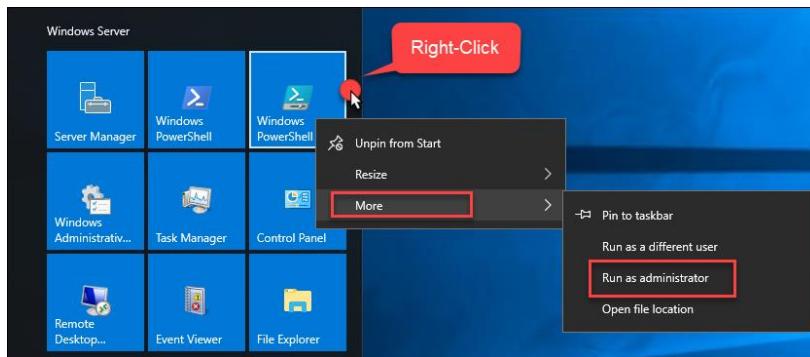
When prompted enter:

- Password for administrator: **demo@pass123**
- Enter Azure AD User: Specify your Azure subscription user account
- Select ASK Version **1803** and press **C** to continue

2. It will take up to 6 hours to successfully install the Azure Stack developer kit.
3. After the installation reboots the virtual machine, login to the AzSHost-1 virtual machine using RDP to monitor the installation progress. You will need to use the account:
 - a. Username: **azurestack.local\azurestackadmin**
 - b. Password: **[your Azure Stack password]**
4. Once connected open Server Manager. On the left, click **Local Server**.

Task 3: Install PowerShell for Azure Stack

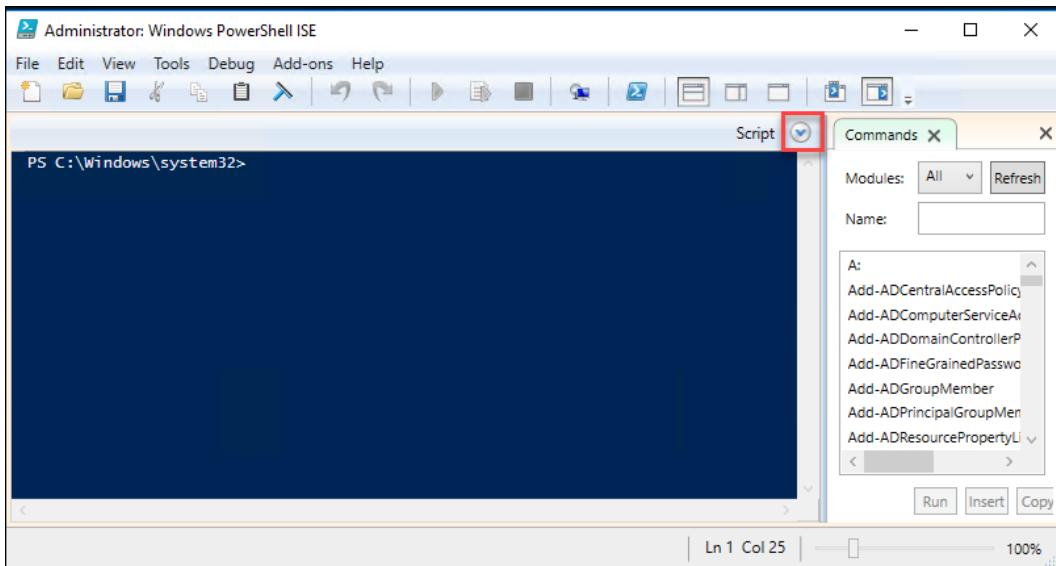
1. Using your LABVM connect to your Azure Stack Host using RDP. You will need to use the account:
 - a. Username: **azurestack.local\azurestackadmin**
 - b. Password: **[your Azure Stack password]**
2. Open PowerShell ISE as an administrator. You will need to **right-click the link** on the start menu, click **More**, and **Run as administrator**.



3. Click **Yes** when prompted.



4. On the PowerShell ISE window, open the script pane by clicking the > mark next to the word **Script**.



5. Execute the following command to trust the PSGallery repository:

```
Set-PSRepository  
  -Name "PSGallery"  
  -InstallationPolicy Trusted
```

6. Next, execute the following command to remove any previous versions of Azure PowerShell:

```
Get-Module -ListAvailable | where-Object {$_.Name -like "Azure*"} |  
Uninstall-Module
```

Note: You may see several errors, you can ignore these.

7. Remove any folders in the following folder that begin with Azure (if any).

```
C:\Users\AzureStackAdmin\Documents\WindowsPowerShell\Modules
```

8. Finally, execute the following command to install the Azure Stack PowerShell cmdlets:

```
# Install the AzureRM.Bootstrapper module. Select Yes when prompted to
# install NuGet
Install-Module `
    -Name AzureRm.BootStrapper

# Install and import the API Version Profile required by Azure Stack into
# the current PowerShell session.
Use-AzureRmProfile `
    -Profile 2017-03-09-profile -Force

Install-Module `
    -Name AzureStack `
    -RequiredVersion 1.2.11
```

Task 4: Download the Latest Azure Stack Tools

1. Using the previous elevated PowerShell console execute the following commands to download and extract the Azure Stack tools from GitHub

```
# Change directory to the root directory.
cd \

# Download the tools archive.
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
Invoke-WebRequest `
    https://github.com/Azure/AzureStack-Tools/archive/master.zip `
    -OutFile master.zip

# Expand the downloaded files.
Expand-Archive master.zip `
    -DestinationPath . `
    -Force

# Change to the tools directory.
cd AzureStack-Tools-master
```

Task 5: Register Azure Stack with Azure AD

In this task, execute all the commands from an elevated PowerShell console on the Azure Stack host.

1. CD to the registration folder of the AzureStack-Tools-master folder

```
CD C:\AzureStack-Tools-master\Registration
```

2. Execute the following command to login:

```
Add-AzureRmAccount -EnvironmentName AzureCloud
```

3. Next, execute the following command to view the Azure subscriptions available:

```
Get-AzureRmSubscription
```

4. Next, execute the following command and specify the subscription ID you want to register Azure Stack to.

```
Select-AzureRmSubscription -SubscriptionId [your subscription id here]
```

5. Register the Azure Stack resource provider with your subscription by executing the following command:

```
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.AzureStack
```

6. Next, execute the following commands to register your Azure Stack installation. When prompted, login with the **azurestack.local\azurestackadmin** account.

```
Import-Module .\RegisterWithAzure.psm1
$AzureContext = Get-AzureRmContext
$CloudAdminCred = Get-Credential
Set-AzsRegistration `
    -PrivilegedEndpointCredential $CloudAdminCred `
    -PrivilegedEndpoint Azs-ERCS01 `
    -BillingModel Development
```

7. Open the Azure portal and navigate to your resource groups. Notice a new resource group named **azurestack** was created. If you open this resource group, you will see the registration resource.

The screenshot shows the Azure Stack Admin portal interface. On the left is a navigation sidebar with links for Overview, Activity log, Access control (IAM), Tags, SETTINGS, and Quickstart. The main area shows a resource group named 'azurystack' with a status bar indicating 'Subscription (change) OTA-PRD-295' and 'Deployment No deployments'. Below this are filters for 'Filter by name...', 'All types', and 'All locations'. A table lists '1 items' with columns for NAME, TYPE, and LOCATION. One item is listed: 'AzureStack-b5ee041d-' of type Microsoft.AzureStack/registrati... global.

- To verify the registration was successful on the Azure Stack Host, navigate to the admin portal at <http://adminportal.local.azurestack.external>. Sign in using your Azure AD credentials used during the Azure Stack Development Kit installation.

Task 6: Download VM Images to Azure Stack Marketplace

In this task, you will download the following images and artifacts which are needed for the resource providers installed later:

- SQL Server 2017 on Windows Server 2016 Image
- SQL IaaS Extension
- Windows Server 2016 – Server Core
- Windows Server 2016 Data Center

- From within the Azure Stack Admin portal, click **Marketplace management**.

The screenshot shows the Microsoft Azure Stack - Administration dashboard. The sidebar on the left includes links for New, Dashboard, All resources, Resource groups, Marketplace management (which is highlighted with a red box), Virtual machines, Plans, Offers, Recent, and More services. The main dashboard area displays 'Region management' with 1 critical and 11 warning alerts, and 'Update' status showing 11 idle items. The bottom of the dashboard indicates Version: 1.0.171122.1.

- You should see the following message: **You have no items downloaded to your Azure Stack marketplace yet. Click "Add from Azure" to add items.**

The screenshot shows the 'Marketplace management' screen. At the top left is a '+ Add from Azure' button. Below it is a search bar with a magnifying glass icon. A red arrow points from the text above to this search bar. Underneath the search bar is a table header with columns for 'NAME', 'PUBLISHER', and 'TYPE'. A message at the bottom states: 'You have no items downloaded to your Azure Stack marketplace yet. Click "Add from Azure" to add items.'

3. Click **+Add from Azure** and if you see the Marketplace Items, you have registered successfully.

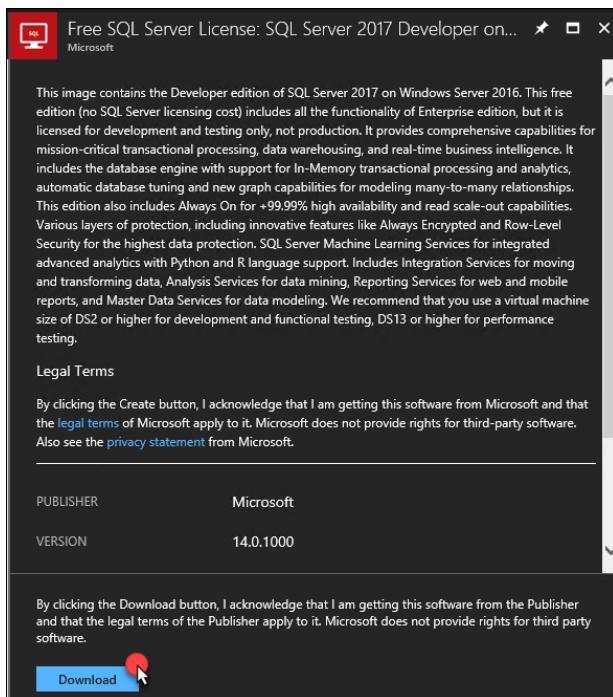
The screenshot shows the 'Add from Azure' interface. At the top left is a '+ Add from Azure' button with a red circle highlighting the cursor. Below it is a search bar with the placeholder 'Search to filter items...'. A red arrow points from the text above to this search bar. The main area is a table listing various marketplace items, with the first few rows shown:

NAME
Acronis Backup for Linux (preview)
Acronis Backup for Windows (preview)
Ubuntu Server 14.04 LTS
Ubuntu Server 14.04 LTS
Ubuntu Server 16.04 LTS
Ubuntu Server 16.04 LTS
Ubuntu Server 17.10
CoreOS Linux (Stable)
Free License: SQL Server 2016 SP1 Developer on Windows Server 2016

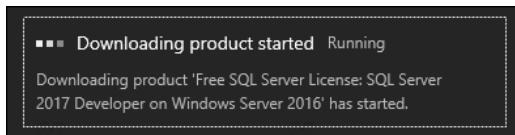
1. From the Marketplace management area of the Azure Stack Admin portal, select the **Developer edition of SQL Server 2017 on Windows Server 2016**.

The screenshot shows a license offer for 'Free SQL Server License: SQL Server 2017 Developer on Windows Server 2016'. To the left of the text is a small icon of a computer monitor with a database symbol on the screen.

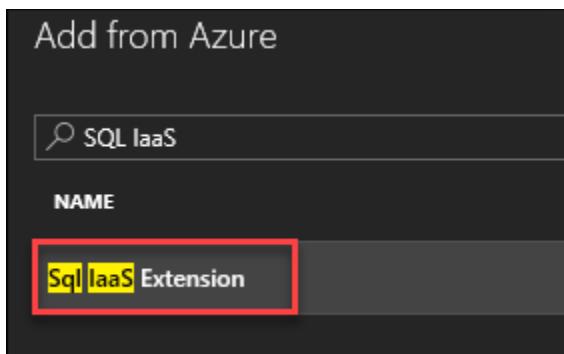
2. Click **Download**.



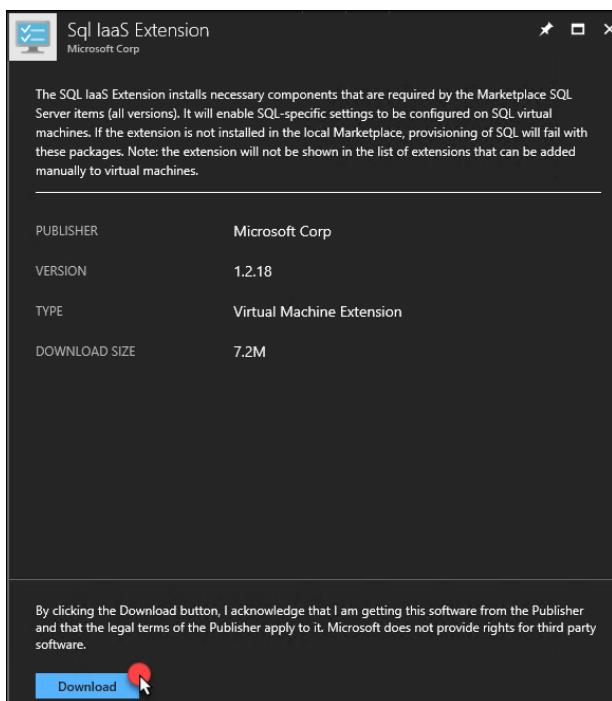
3. A notification will pop-up notifying you the product is being downloaded to Azure Stack.



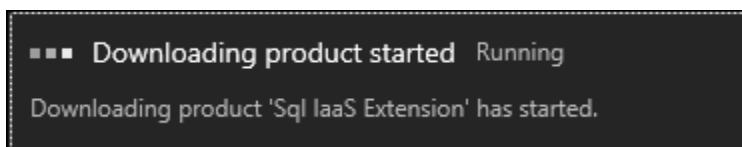
4. In the Add from Azure search bar, type SQL IaaS, and press enter. The SQL IaaS Extension will appear in the search results. Click its name.



- When the SQL IaaS Extension appears, click **Download**.



- A notification will pop-up notifying you the product is being downloaded to Azure Stack.

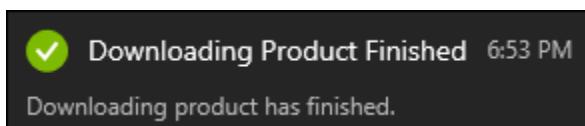


- In the Add from Azure search bar, type **Windows Server**, and press **enter**.

Download **both** of the following images:

- Windows Server 2016 Datacenter
- Windows Server 2016 Datacenter - Server Core

- Once the products are downloaded, you will receive a notification.



Note: These downloads will take some time depending upon your Azure Stack connectivity. Wait until they complete before proceeding to the next task.

Exercise 1: Configure Azure Stack

Duration: 2-3 hours

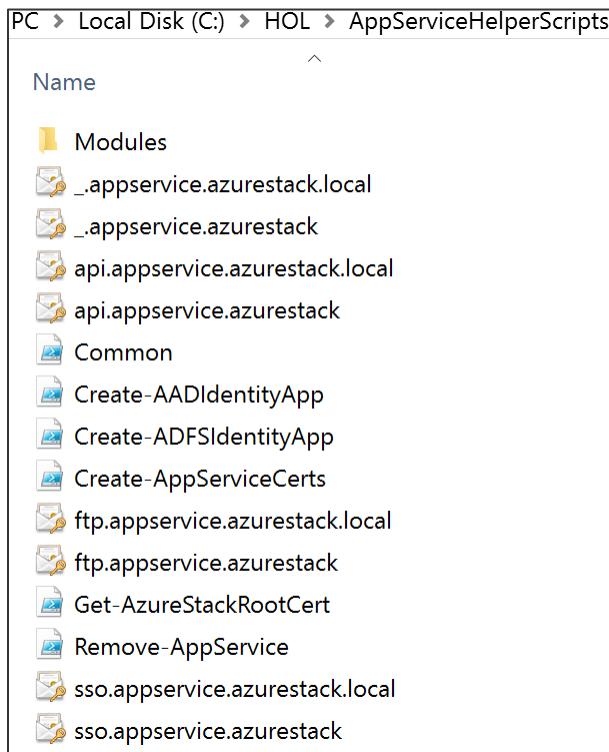
In this exercise, you will configure the Azure Stack environment for the lab. You will install the resource providers for Azure SQL Database and Azure App Service, from there you will configure the taxonomy for the Azure Stack hands-on lab.

Tip: To minimize prompts from PowerShell, set your execution policy to bypass
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope CurrentUser

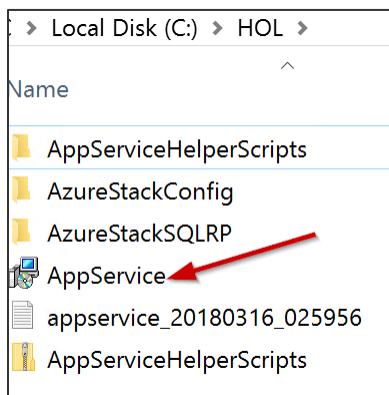
Task 1: Deploy the Azure Stack App Service Resource Provider

Sub Task 1: Download the Installer and Scripts

1. Download the installer executable from <https://aka.ms/appsvconmasinstaller> and helper scripts zip file from <https://aka.ms/appsvconmashelpers> to the **C:\HOL** folder on the Azure Stack Host. The following files and folder structure appear after extracting the helper scripts zip:



The following screenshot shows the installer executable:



Sub Task 2: Create App Service Scripts

1. Change to the **AppServiceHelperScripts** folder by executing the following command in an elevated PowerShell console:

```
CD C:\HOL\AppServiceHelperScripts
```

2. Create certificates that the Azure App service resource provider will need by running the following command:

```
.\Create-AppServiceCerts.ps1
```

When prompted use the following values:

- .pfx password use **demo@pass123**
- DomainName: **local.azurestack.external**

The script creates four certificates in the C:\HOL\AppServiceHelperScripts folder. The four certificates created by this script are shown in the following table:

File name	Use
_appservice.local.azurestack.external.pfx	App Service default SSL certificate
Api.appservice.local.azurestack.external.pfx	App Service API SSL certificate
ftp.appservice.local.azurestack.external.pfx	App Service publisher SSL certificate
Sso.appservice.local.azurestack.external.pfx	App Service identity application certificate

3. Get the Azure Stack Root Certificate by executing the following command:

```
.\Get-AzureStackRootCert.ps1
```

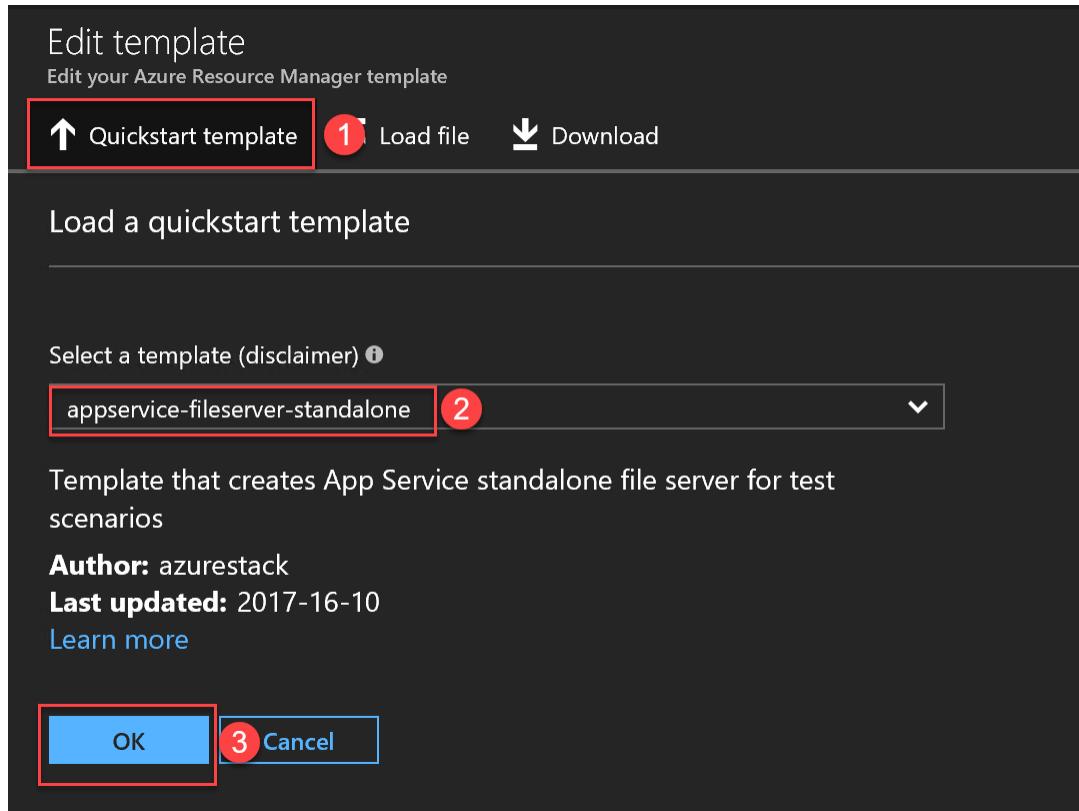
When prompted:

- PrivilegedEndpoint: AzS-ERCS01
- User name: azurestack\cloudadmin
- Password: demo@pass123

Sub Task 3: Deploy Supporting File Server

Note: Azure App Service requires the use of a file server. You will use an ARM template to create a standalone file server for App Service.

1. Launch the Azure Stack Administration portal if it isn't open.
2. Deploy the template to the Azure Stack portal by clicking New and search for **Template Deployment**, click the result and then click the **Create button**. Within the new template deployment blade click **Edit Template**.
3. Within the Edit Template blade, click **Quickstart template**, and then in the dropdown list choose **appservice-fileservice-standalone**, click **OK** and then **Save**.



4. Specify **demo@pass123** for the three password parameters.

The screenshot shows the 'Custom deployment' screen in the Azure Stack Administration portal. The 'Parameters' section is highlighted with a red box. Inside this box, the 'ADMINPASSWORD' parameter is set to a secure string represented by five asterisks ('*****'). Other parameters shown include 'FILESERVERVIRTUALMACHINESIZE' (Standard_A2), 'IMAGEREFERENCE' (MicrosoftWindowsServer | WindowsServer...), 'DNSNAMEFORPUBLICIP' (appservicefileshare), 'ADMINUSERNAME' (fileshareowner), 'FILESHAREOWNER' (fileshareowner), 'FILESHAREOWNERPASSWORD' (securestring with five asterisks), 'FILESHAREUSER' (fileshareuser), 'FILESHAREUSERPASSWORD' (securestring with five asterisks), and 'VMEXTENSIONSCRIPTLOCATION' (https://raw.githubusercontent.com/Azure/azur).

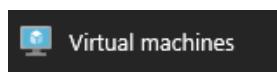
5. Use **AzSHOLFS** as the new resource group name and click **Create**. It may take 10-15 minutes to complete provisioning.

NOTE: The ARM template automatically creates the local users needed for the App Service Resource Provider.

6. After the template deployment is complete, open the new virtual machine from clicking on **Virtual Machines**, **FileServerVM** and Note the Public IP address/DNS name label for later reference

The screenshot shows the 'FileServerVM' virtual machine details page. The 'Public IP address/DNS name label' field is highlighted with a red box and contains the value '192.168.102.32/appservicefileshare.local.cloudapp.azurestack.external'. Other visible fields include 'Computer name' (FileServerVM), 'Operating system' (Windows), 'Size' (Standard A2 (2 vcpus, 3.5 GB memory)), and 'Virtual network/subnet' (FileServerVnet/FileServerSubnet).

7. Connect to the new file server virtual machine by clicking **Virtual Machines**, **FileServerVM**, and then **Connect** and login using the following credentials
- o User: fileshareowner
 - o Password: demo@pass123



 Connect

- Next you need to provision the content share on the file server. Launch an elevated console from within the VM by opening the run dialog and entering **cmd** then pressing **enter**, and then execute the following commands.

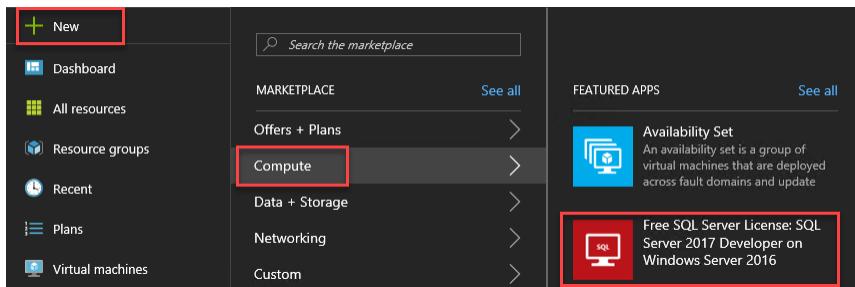
```
set WEBSITES_SHARE=WebSites
set WEBSITES_FOLDER=C:\WebSites
md %WEBSITES_FOLDER%
net share %WEBSITES_SHARE% /delete
net share %WEBSITES_SHARE%=%WEBSITES_FOLDER% /grant:Everyone,full
```

- Next up is to configure access control to the shares. To configure access run the following commands at an elevated command prompt on the file server. Replace values in italics with values that are specific to your environment.

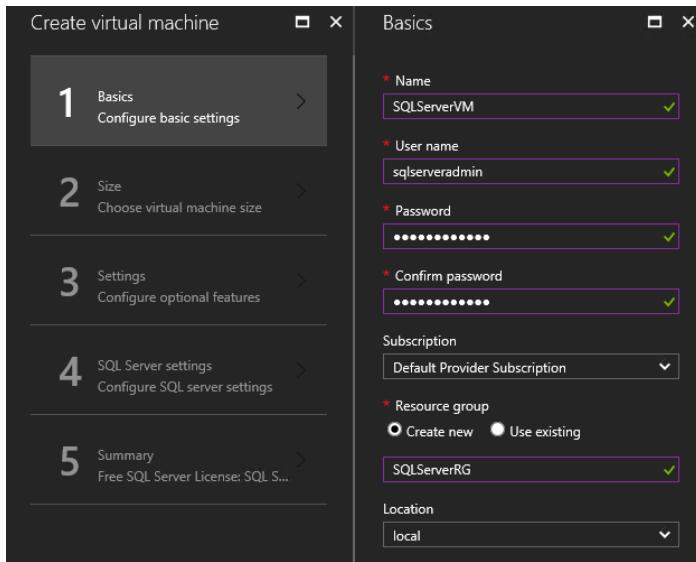
```
set WEBSITES_FOLDER=C:\WebSites
icacls %WEBSITES_FOLDER% /reset
icacls %WEBSITES_FOLDER% /grant FileShareOwner:(OI)(CI)(F)
icacls %WEBSITES_FOLDER% /inheritance:r
icacls %WEBSITES_FOLDER% /grant FileShareUser:(CI)(S,X,RA)
icacls %WEBSITES_FOLDER% /grant *S-1-1-0:(OI)(CI)(IO)(RA,REA,RD)
```

Sub Task 4: Deploying a Supporting SQL Server

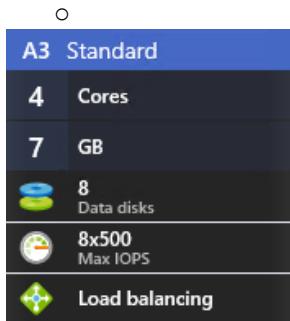
- From the Azure Stack Admin Portal, click **+ New, Compute, Free SQL Server License: SQL Server 2017 Developer on Windows Server 2016**, and then click **Create**.



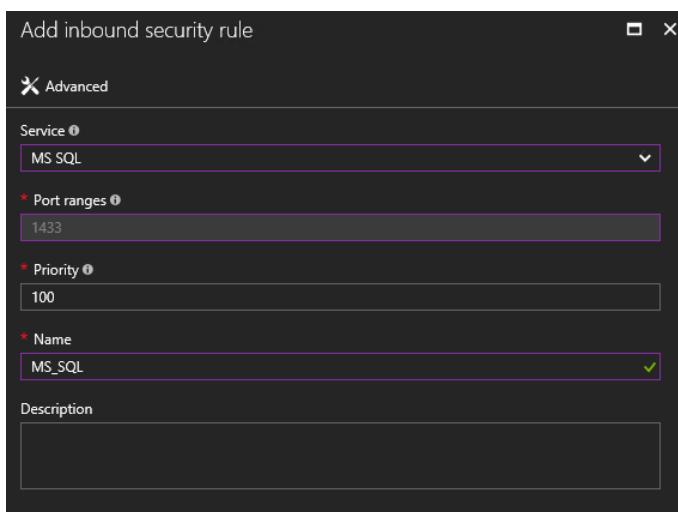
- Specify the following configuration on the Basics blade and click OK:
 - Name: SQLServerVM
 - User name: sqlserveradmin
 - Password: demo@pass123
 - Resource Group: SQLServerRG



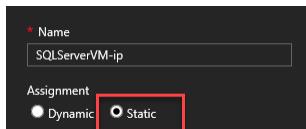
3. Find and select A3 Standard as the virtual machine size and click **Select**.



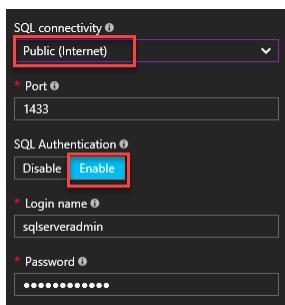
4. On the settings blade, click **Network security group (firewall)** and then click **+ Add an inbound rule**. Change the service to **MS SQL** and click **OK** until you are back to the settings blade.



5. Click **Public IP address**, and then click **Create New**. Change the assignment to **Static**.



6. Click **OK** on the settings blade to continue.
7. On the **SQL Server settings** blade make the following changes and then click **OK**.
 - SQL connectivity: Public (Internet)
 - SQL Authentication: Enable



8. Click **OK** on the summary blade to create the VM (this VM may take 10-15 minutes to provision).
9. Note the Public IP address of the VM for later reference.

Sub Task 5: Setup Application Identity

1. Switch back to the Azure Stack host to configure an Azure AD Service Principal for Azure Functions.
2. Launch an elevated PowerShell instance (ensure you are logged in as azurestack\AzureStackAdmin) and run the following commands:

```
C:\HOL\AppServiceHelperScripts  
.\\Create-AADIdentityApp.ps1
```

When prompted specify the following:

- DirectoryTenantName: [your tenant name].onmicrosoft.com
- AdminArmEndpoint: adminmanagement.local.azurestack.external
- TenantArmEndpoint: management.local.azurestack.external
- CertificateFilePath: C:\\HOL\\AppServiceHelperScripts\\sso.appservice.local.azurestack.external.pfx
- CertificatePassword: demo@pass123
- Login with your Azure credentials when prompted

- Note the application ID created (yours will be different).

**Please note Application Id: 398277f8-d1c9-4934-86ce-ff70485e726b
Sign in to the Azure portal as Azure Active Directory Service Admin -> Search for Application Id and grant permissions.
398277f8-d1c9-4934-86ce-ff70485e726b**

- Switch the Azure Portal (<https://portal.azure.com>) and click Azure Active Directory on the navigation or under All Services.
- Select **App Registrations**, and change the drop down to All apps, then search for the application ID noted earlier.

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

DISPLAY NAME	APPLICATION TYPE
AS App Service	Web app / API

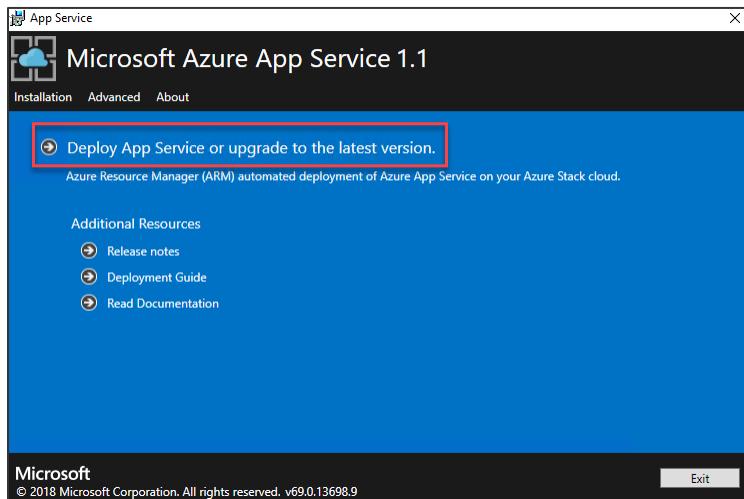
- Select the **App Service** returned from search results and then click **Settings -> Required Permissions**.

The screenshot shows the 'Settings' blade for an 'App Service' registered app. The 'Required permissions' link under the 'API ACCESS' section is highlighted with a red box.

- Select **Grant Permissions** and click **Yes**.

Sub Task 6: Install the App Service Resource Provider

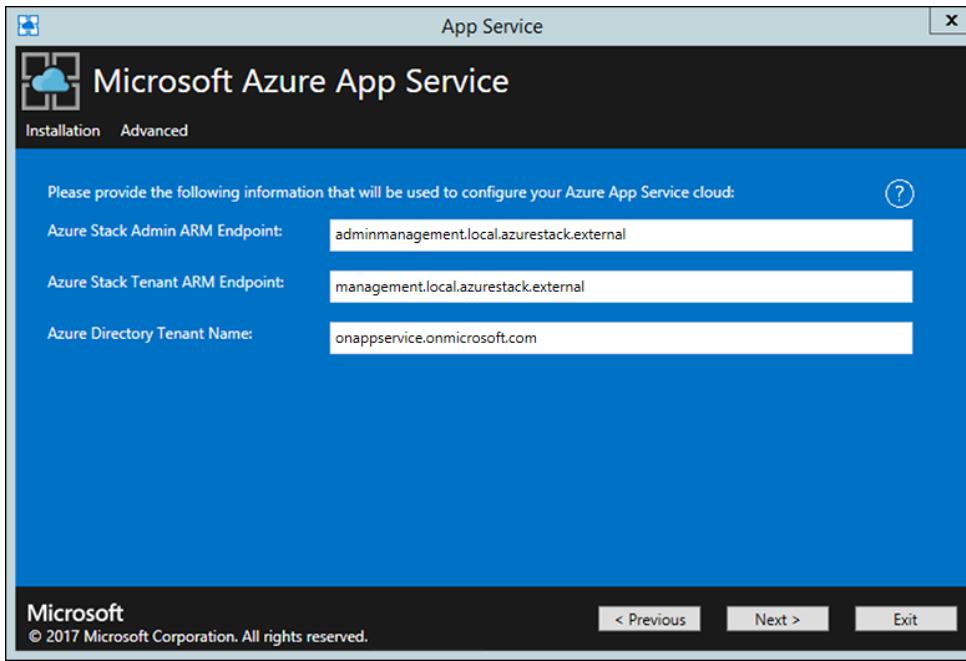
- Next start the Azure Stack App Service resource provider deployment by navigating to C:\HOL folder using File Explorer, and double click the **AppService.msi** file to start the installation.
- Click **Deploy App Service or upgrade to the latest version**.



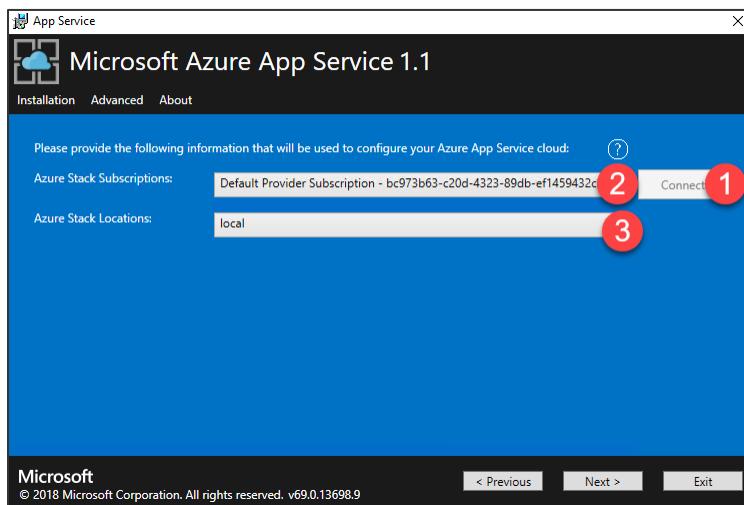
3. Review and accept the Microsoft Software License Terms and then click **Next**.

4. Review and accept the third-party license terms and then click **Next**.

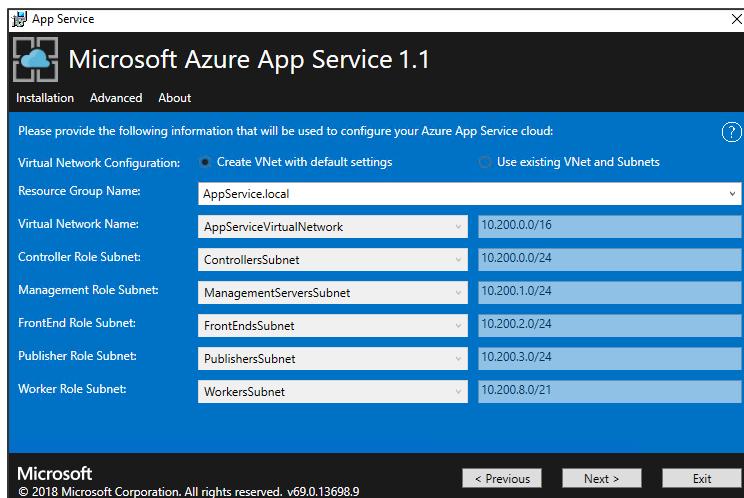
5. Review the settings and click **Next**.



6. Click Connect, and then specify your Azure subscription information. After logging in, click the dropdown by the subscription and location dropdown to specify the correct configuration.

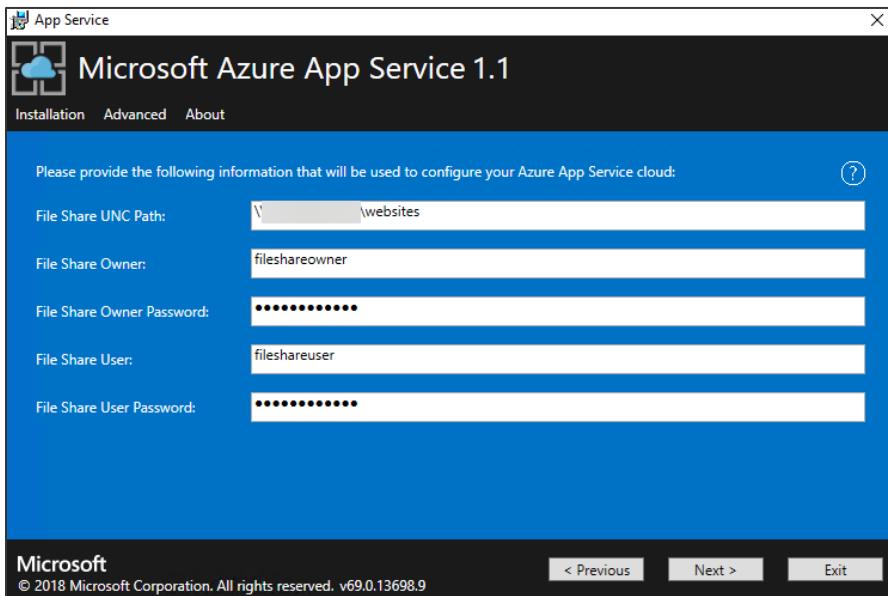


7. Accept the defaults on the network configuration by clicking next.

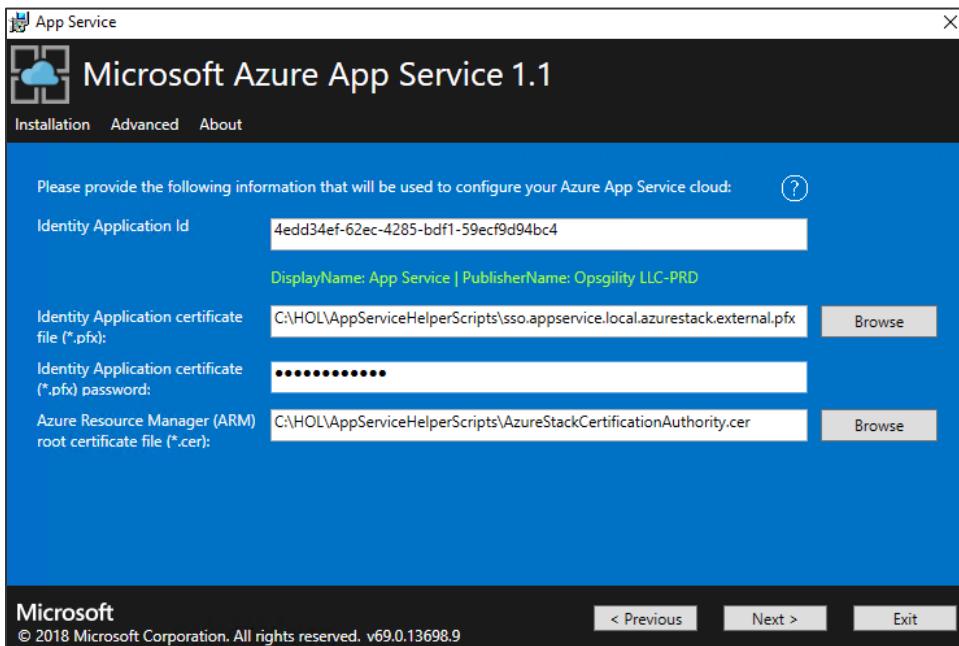


8. Specify the following configuration for the file share:

- File Share UNC Path: \File Server IP\websites (use the public IP from your file server noted earlier)
- File Share Owner: fileshareowner
- File Share Owner Password: demo@pass123
- File Share User: fileshareuser
- File Share User Password: demo@pass123

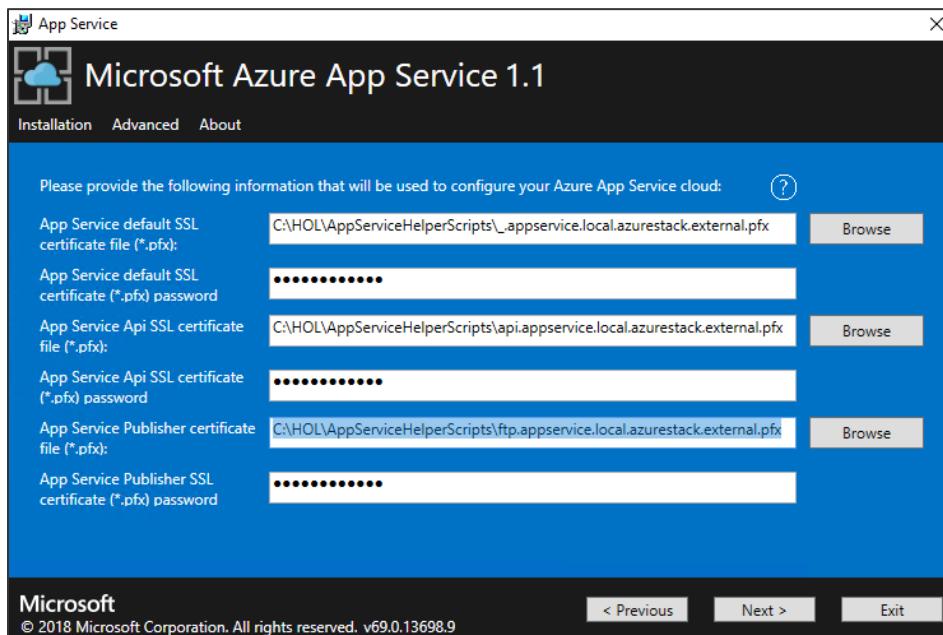


9. On the next screen, paste in the application id created earlier and specify the following certificate information:
- Identity Application ID: your application id key created earlier
 - Identity Application Certificate (*.pfx): C:\HOL\AppServiceHelperScripts\sso.appservice.local.azurestack.external.pfx
 - Identity Application Certificate (*.pfx) Password: demo@pass123
 - Azure Resource Manager (ARM) root certificate file (*.cer): C:\HOL\AzureStackCertificationAuthority.cer

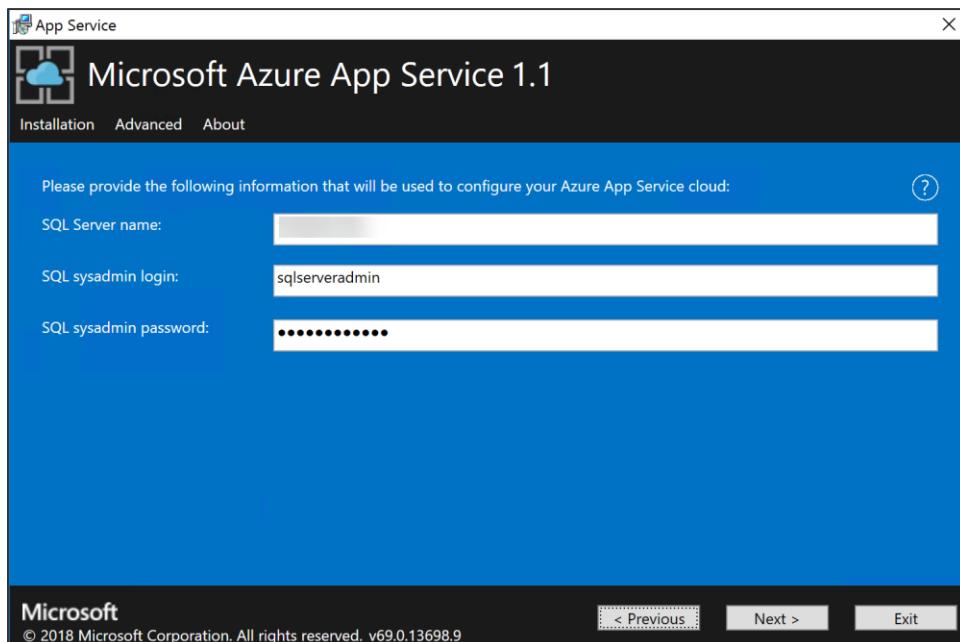


10. On the next page you will specify the remaining certificates:
- App Service default SSL certificate file (*.pfx) : C:\HOL\AppServiceHelperScripts_.appservice.local.azurestack.external.pfx

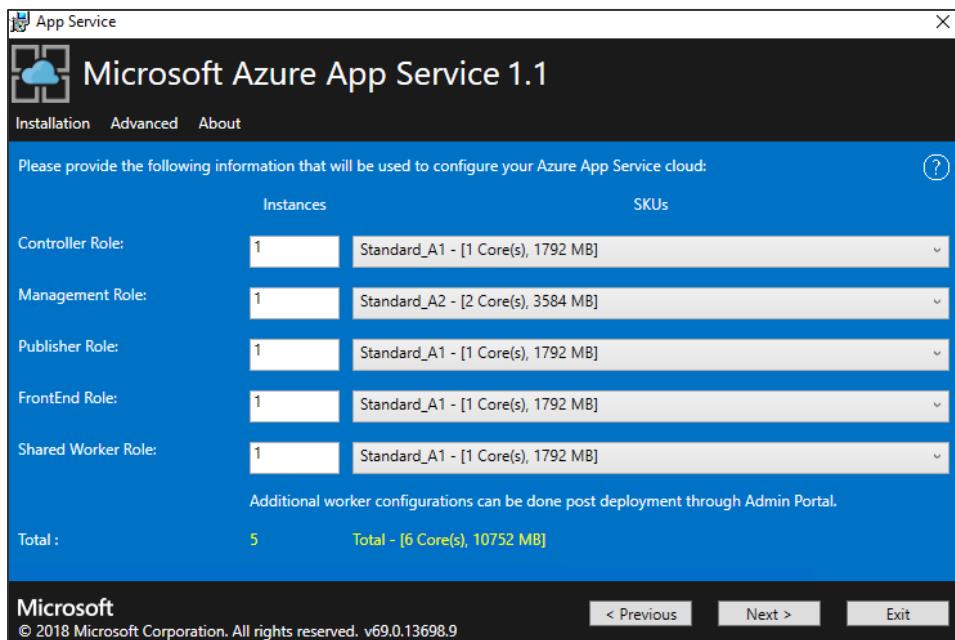
- App Service API SSL certificate file (*.pfx):
C:\HOL\AppServiceHelperScripts\api.appservice.local.azurestack.external.pfx
- App Service Publisher certificate file (*.pfx):
C:\HOL\AppServiceHelperScripts\ftp.appservice.local.azurestack.external.pfx



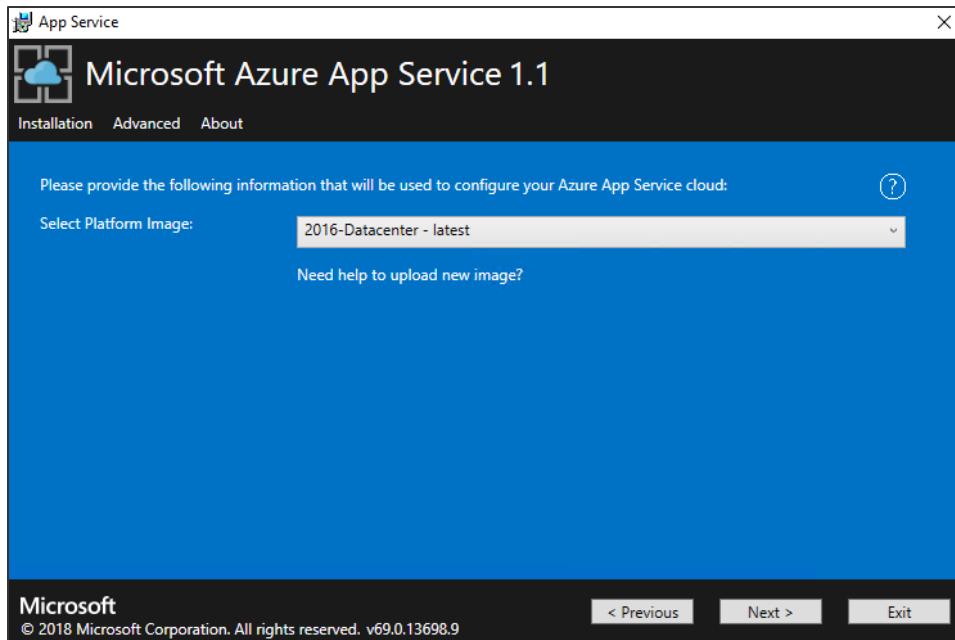
11. On the next screen, specify the public IP and the credentials for your SQL Server VM and click **Next**.



12. Accept the defaults for the VMs to provision for the App Service resource provider by clicking **Next**.

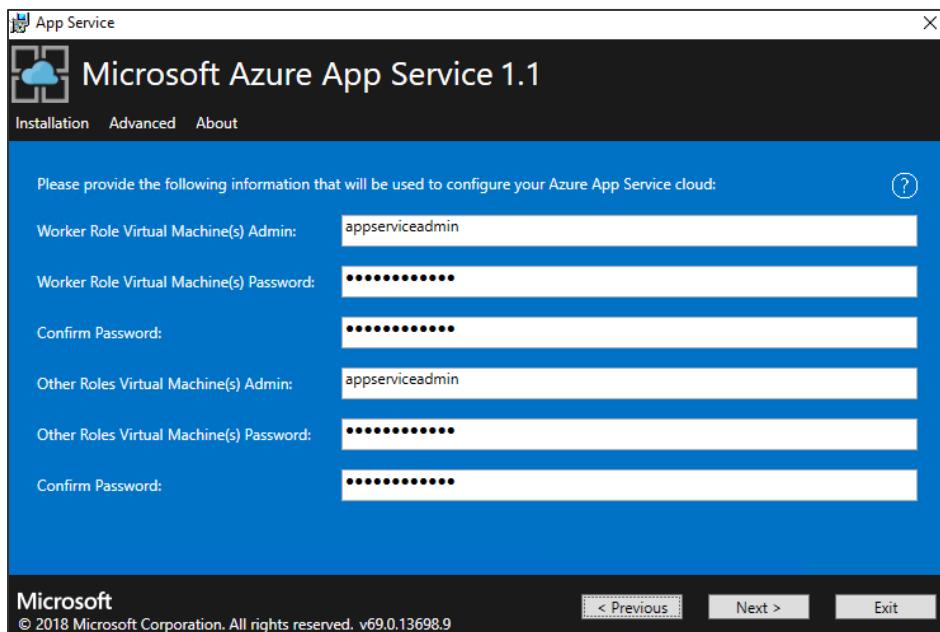


13. Except the default for the platform image by clicking **Next**.

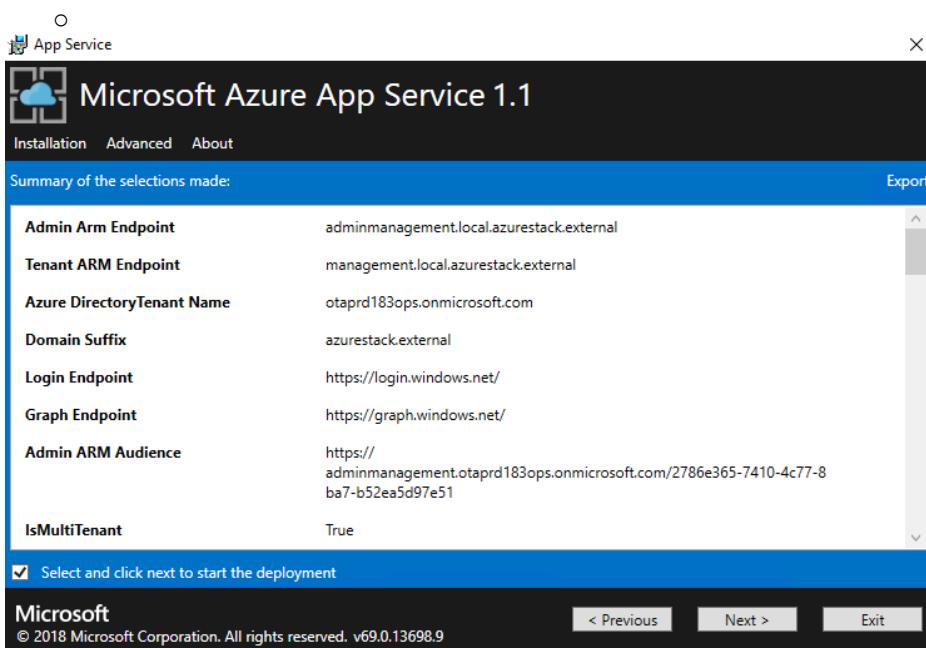


14. Specify the following user account for both administrator accounts and click Next.

- User name: appserviceadmin
- Password: demo@pass123



15. Click the Checkbox next to Select and click next to start the deployment and then click **Next**.



16. The final step is to validate the App Service on Azure Stack installation. TO validate the App Service installation perform the following 2 steps:

- o In the Azure Stack admin portal, go to **Administration - App Service**.
- o In the overview under status, check to see that the **Status** shows **All roles are ready**.

Task 2: Deploy the Azure Stack SQL DB Resource Provider

- From the Azure Stack Host, you will need to locate the IP address of the Azure Stack "Privileged Endpoint." Click **Start**, **Windows Administrative Tools** followed by **Hyper-V Manager**.



- Once Hyper-V Manager loads, you will see all the virtual machines that make up the Azure Stack Development Kit on your host. Locate and click the virtual machine (VM) with the name **AzS-ERCS01**.

A screenshot of the Hyper-V Manager application. The left pane shows a tree view with 'Hyper-V Manager' and 'AZS-HOST1'. The right pane is titled 'Virtual Machines' and lists ten VMs. A red circle highlights the row for 'AzS-ERCS01', which is currently selected. The table columns are Name, State, CPU Usage, Assigned Memory, and Uptime. The data for the selected VM is: Name = AzS-ERCS01, State = Running, CPU Usage = 0 %, Assigned Memory = 2048 MB, Uptime = 04:14:20.

- In the bottom pane of Hyper-V Manager, you will see the **AzS-ERCS01** VM with four tabs. Click the **Networking** Tab. Then, take note of the IPv4 address. In the case of the example it is **192.168.200.225**.



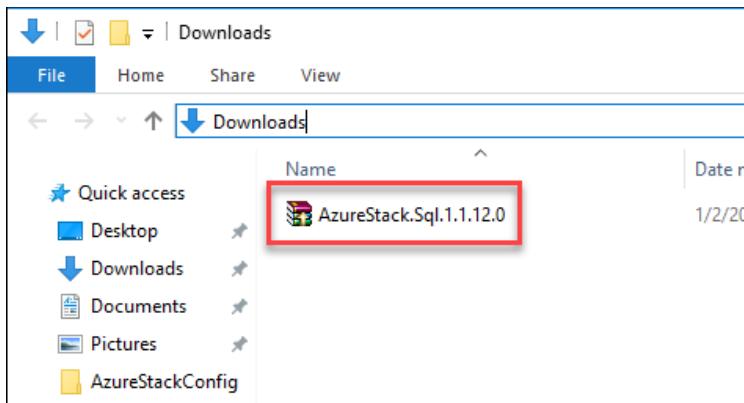
4. From the Azure Stack Host, open a new Internet Explorer tab and connect to the follow address.

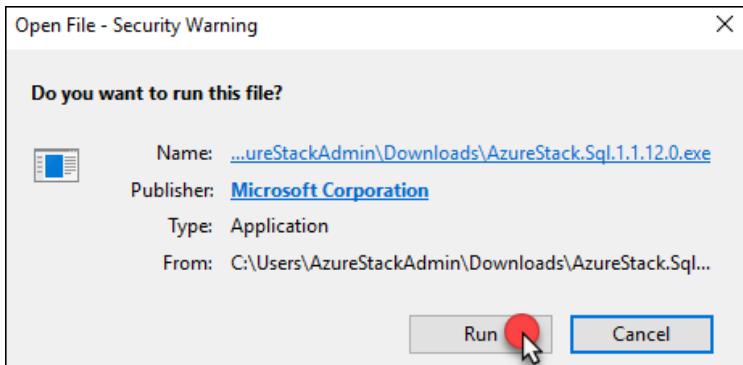
<https://docs.microsoft.com/en-us/azure/azure-stack/azure-stack-sql-resource-provider-deploy>

5. From this webpage, download the version of the SQL Resource Provider that matches your version of Azure Stack.

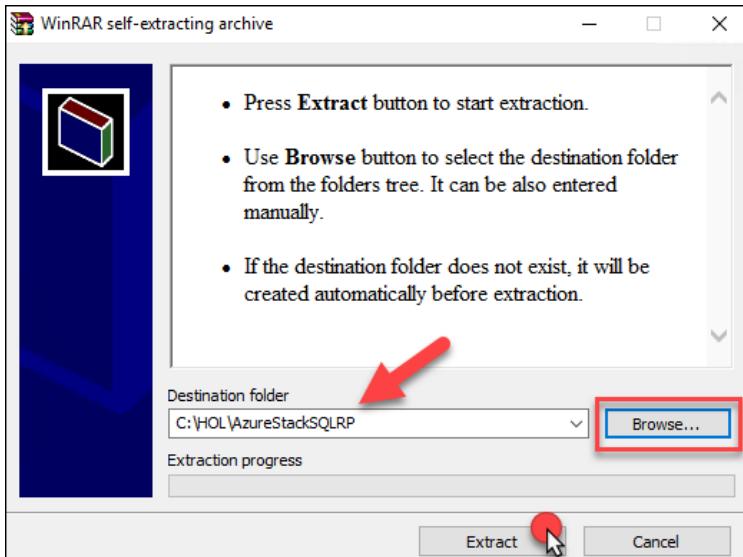
Azure Stack build	SQL resource provider installer
1802: 1.0.180302.1	SQL RP version 1.1.18.0
1712: 1.0.180102.3, 1.0.180103.2 or 1.0.180106.1 (multi-node)	SQL RP version 1.1.14.0
1711: 1.0.171122.1	SQL RP version 1.1.12.0
1710: 1.0.171028.1	SQL RP version 1.1.8.0

6. Once it is downloaded, using Windows Explorer on the Azure Stack Host, navigate to your downloads directory. Double click the file to install the package. You will need to click **Run** at the Security Warning prompt.





7. Click **Accept** and use the **Browse** and **Make New Folder** buttons to change the destination folder to a new folder named: **C:\HOL\AzureStackSQLRP**. Click **Extract**.



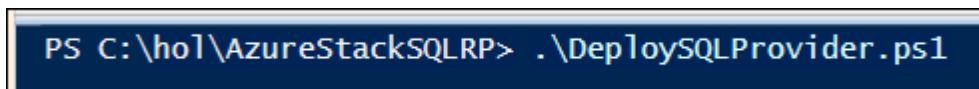
9. From the elevated PowerShell console, move to the directory where you extracted the SQL RP files: **C:\HOL\AzureStackSQLRP**.

```
cd C:\HOL\AzureStackSQLRP
```

10. Move back to your Azure Stack Admin portal and check on the download of Windows Server 2016 Datacenter – Server Core. **This must be downloaded prior to continuing.** If for some reason the download is not complete, you must wait until the image is in the Azure Stack Marketplace prior to moving forward. This is due to the SQL RP using this image.

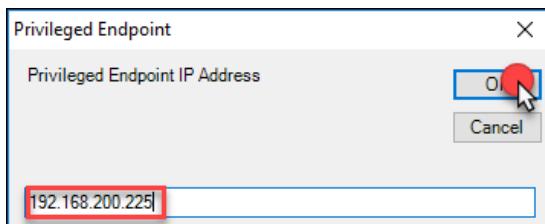
11. From this directory, run the script DeploySQLProvider.ps1 by executing the following command.

```
.\DeploySQLProvider.ps1
```



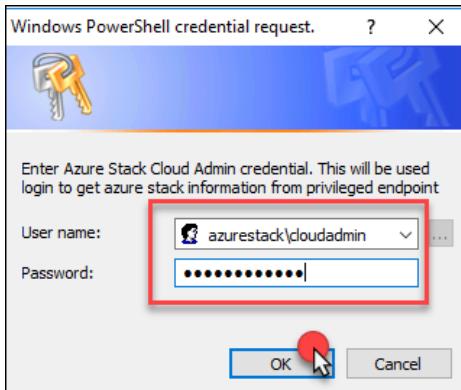
```
PS C:\hol\AzureStackSQLRP> .\DeploySQLProvider.ps1
```

12. At the **Privileged Endpoint** prompt, enter the **IP Address that you found using Hyper-V Manager**, and click **OK**. In the example the IP is **192.168.200.225**.

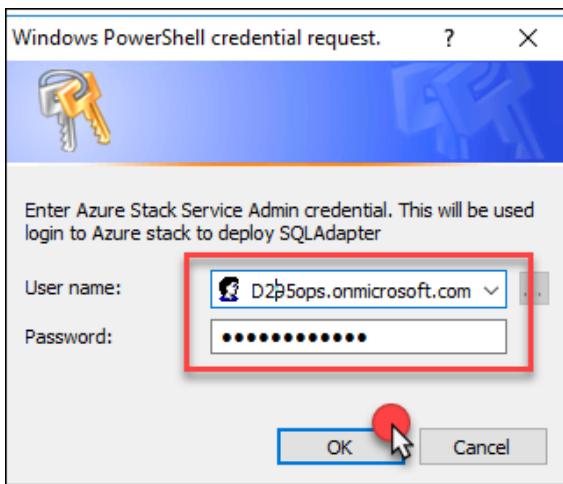


Note: The popup may appear the PowerShell ISE

13. Next, you will be prompted for the **Azure Stack Cloud Admin** credential. Enter the following user name along with your password:
- Username: **azurestack\clouadmin**
 - Password: [your password – should be the same as your Azure Stack Host Local Admin].

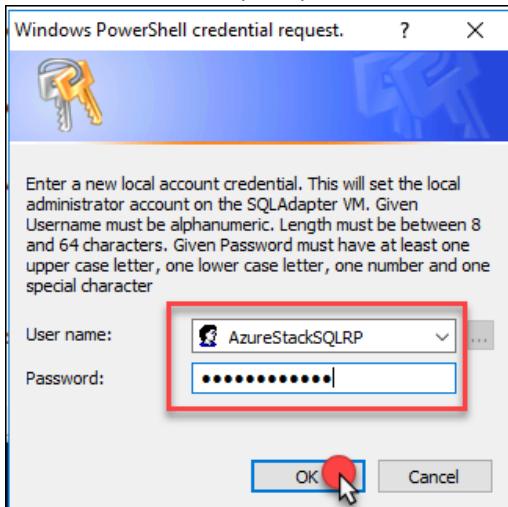


14. Next, you will be prompted for the **Azure Stack Service Admin** credential. Enter the following user name along with your password:
- Username: **username@azuredomain.onmicrosoft.com**
 - Password: [your password]



15. Next, you will be prompted for a new local account to be used for the local SQL Adapter VM. Enter the following user name along with your password:

- Username: **AzureStackSQLRP**
- Password: [your password]



16. You will then be prompted for the Password for the Default SSL Certificate. **Use the same password as the Azure Stack Host Administrator account** and press **OK**.



Note: The script will run for at least 25 mins. Please wait until it completes prior to moving on to the next step.

17. Once the SQL RP is installed, the following message will appear. Close any Azure Stack portal sessions and restart them to update the portal UI.

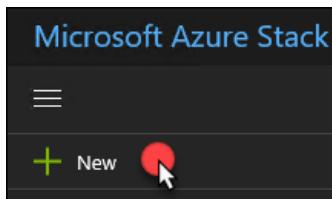
```
*****
*SQLAdapter Installation was successful. Please refresh all the opened Azure Stack portal
*****
```

18. Open the Azure Stack Admin portal <https://adminportal.local.azurestack.external>, and navigate to Resource Groups. Notice the new **system.local.sqladapter** resource group. Click on this group.

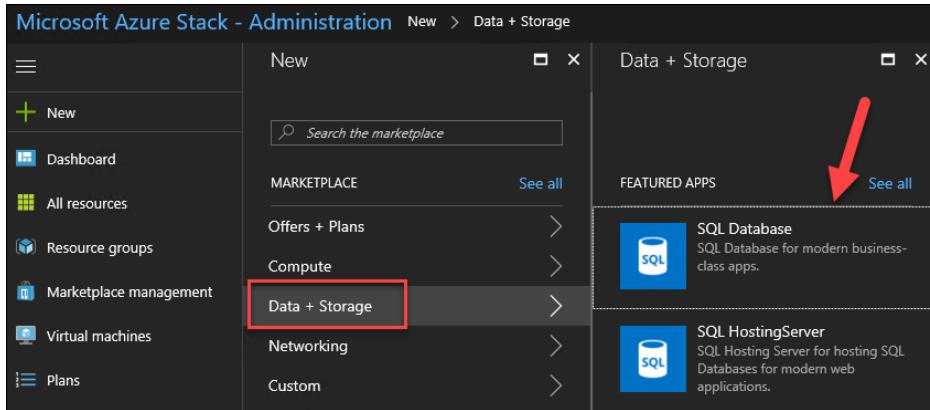
The screenshot shows the Microsoft Azure Stack - Administration portal. On the left, there's a sidebar with various navigation options like Dashboard, All resources, Resource groups, Marketplace management, Virtual machines, Plans, Offers, Recent, and More services. The main area is titled 'Resource groups' and shows a list of 8 items under 'Subscriptions: Default Provider Subscription'. The items are listed in a table with columns for NAME and TYPE. The 'NAME' column contains: 'azurestack-activation', 'system.keyvault', 'system.local', 'system.local.adminkeyvault', 'system.local.dbadapter.dns', 'system.local.keyvault', 'system.local.sqladapter' (which is highlighted with a red box), and 'System.UsageConnection'. The 'TYPE' column shows icons representing each resource type.

19. This will load the resource group where you can see all the different resources provisioned to allow for the use of the SQL RP.

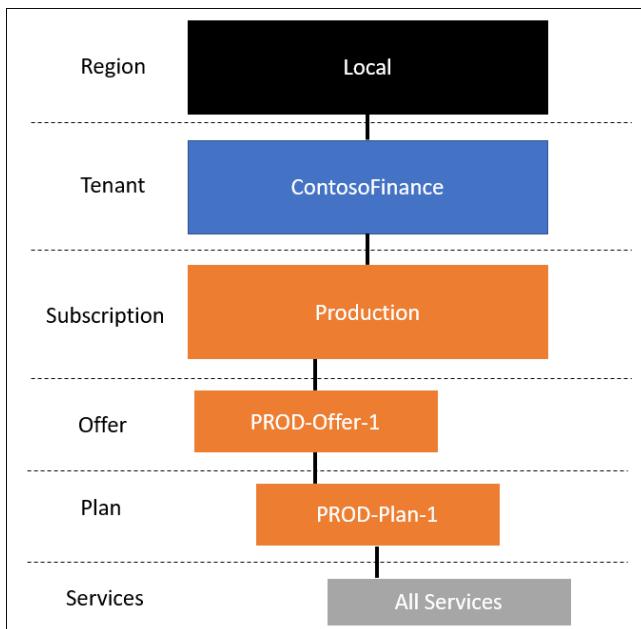
20. Click **+New** in the Azure Stack Admin portal.



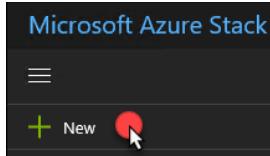
21. Click **Data + Storage** and notice that the **SQL Database** and **SQL HostingServer** now appear as options for you to provision.



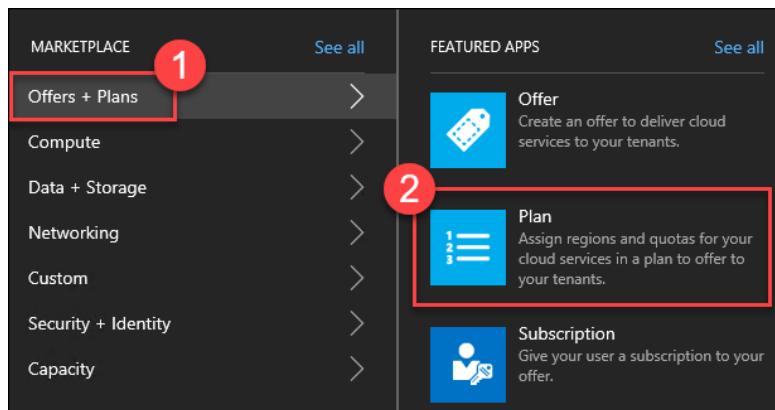
Task 3: Create Azure Stack Deployment Taxonomy for Tenets



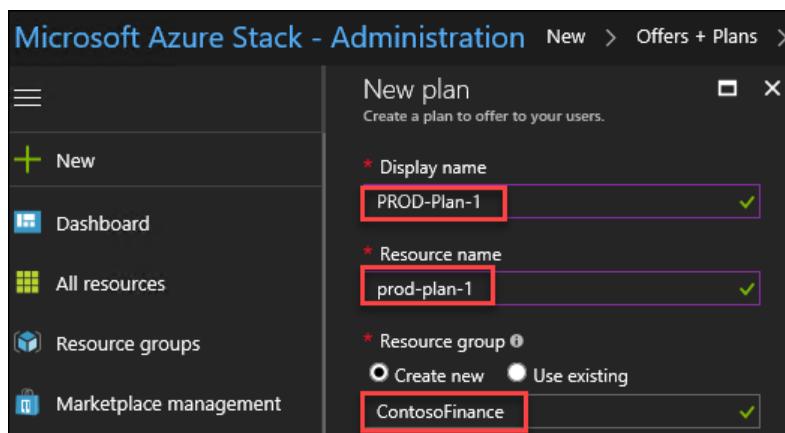
1. Click +New in the Azure Stack Admin portal.



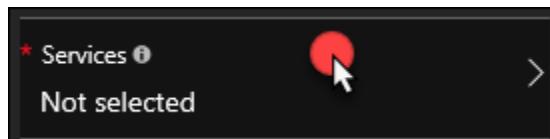
2. Click **Offers + Plans** followed by **Plan**.



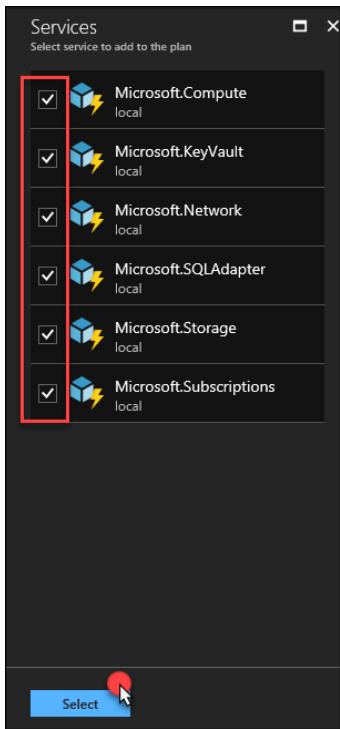
3. In the New Plan blade, provide the following inputs:
 - Display name: **PROD-Plan-1**
 - Resource name: **prod-plan-1**
 - Resource group: **ContosoFinance**



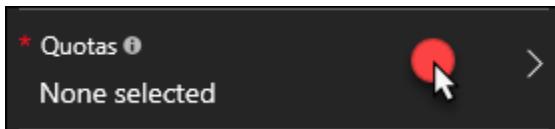
4. Click **Services**.



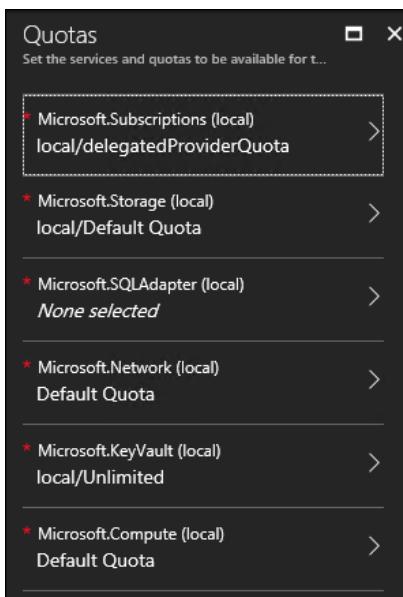
5. Next, check each of the **Services** listed, and click **Select**.



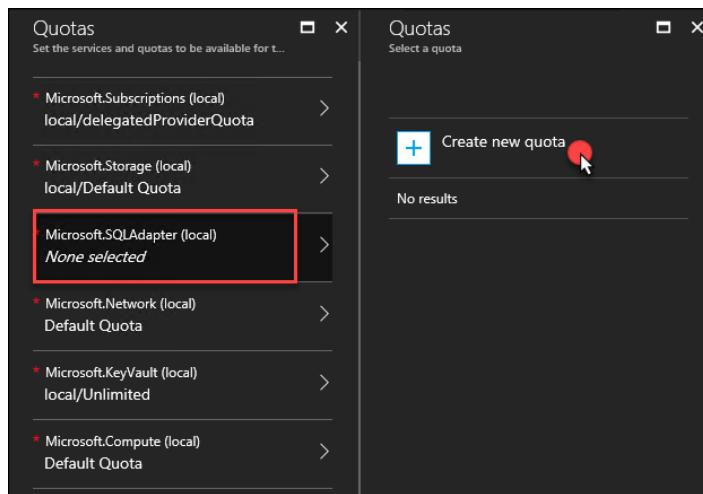
6. Click **Quotas**.



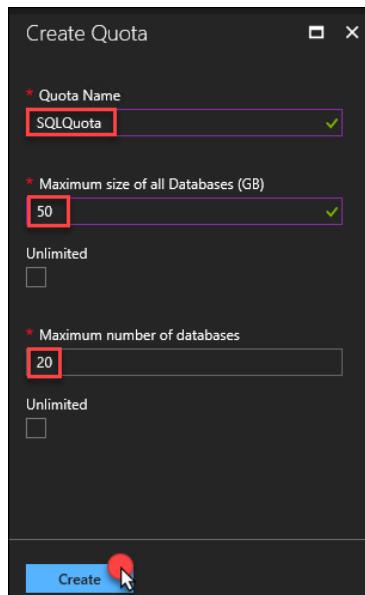
7. For Quotas, click through and select the default quota presented skipping the Microsoft.SQL Adapter. Once this is complete, clicking the blade will resemble this example.



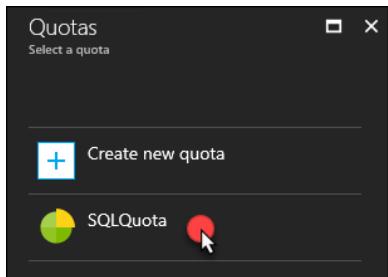
8. Click the **Microsoft.SQLAdapter**, followed by **Create new quota**.



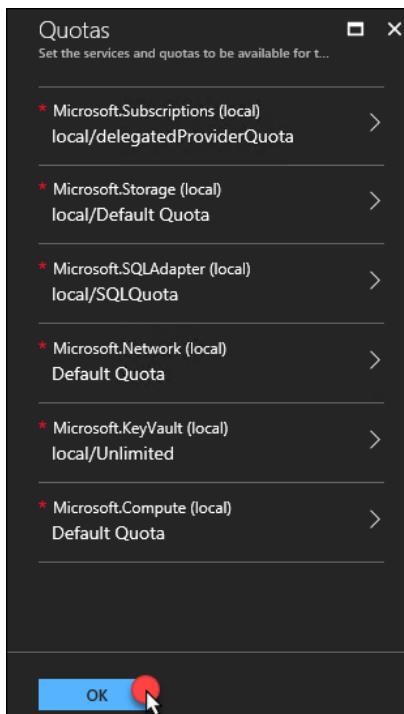
9. Complete the **Create Quota** blade using these inputs. Then, click **Create**.
 - a. Quota Name (no spaces allowed): **SQLQuota**
 - b. Maximum size of all databases (GB): **50**
 - c. Maximum number of databases: **20**



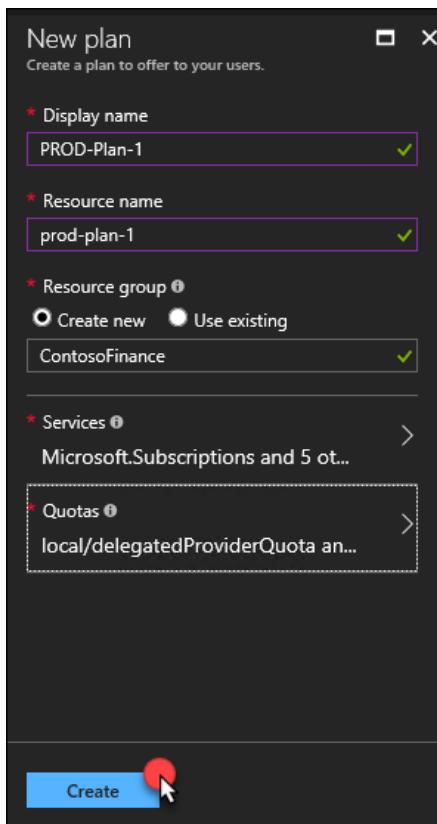
10. Click the **SQLQuota** to assign this to the Microsoft.SQLAdapter.



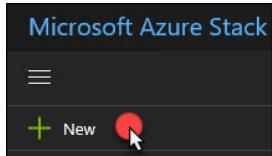
11. Click **OK** on the **Quotas** blade to assign these to **Prod-Plan-1**.



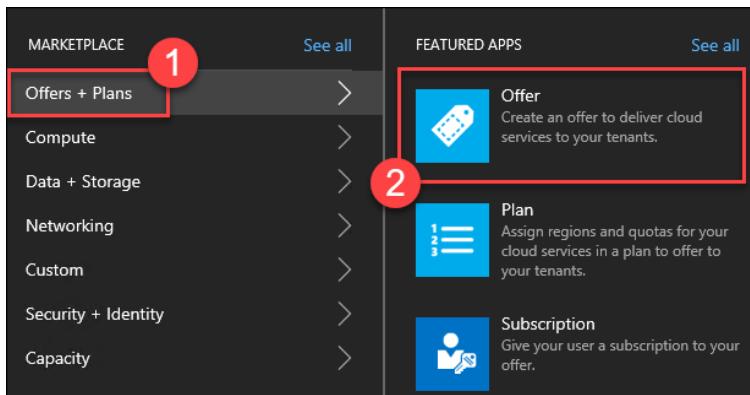
12. Click **Create** to provision the New Plan: **PROD-Plan-1**.



13. The Plan will deploy immediately. Click **+New** in the Azure Stack Admin portal.

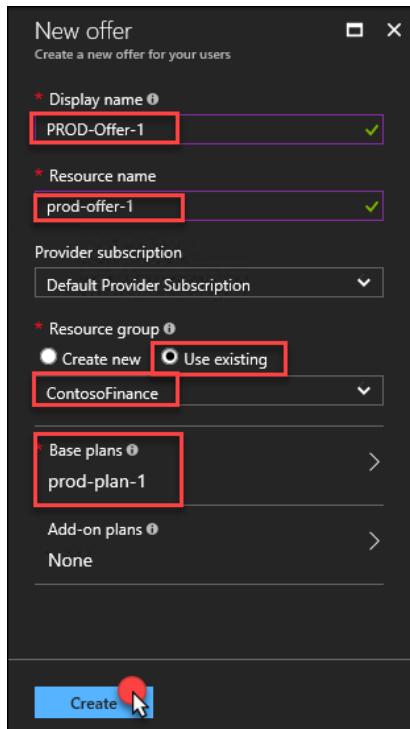


14. Click **Offers + Plans** followed by **Offer**.



15. Update the New offer blade using the following inputs. Then, click **Create**.

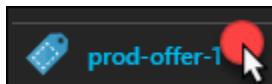
- a. Display Name: **PROD-Offer-1**
- b. Resource Name: **prod-offer-1**
- c. Resource group: **Use existing / ContosoFinance**
- d. Base plans: **PROD-Plan-1**



16. In the Azure Stack Admin portal, navigate to the **ContosoFinance** resource group. Here, you can review the new Offer and Plan just created.

The screenshot shows the 'ContosoFinance' resource group in the Azure Stack Admin portal. The 'Overview' tab is selected. The 'Essentials' section displays the subscription information. The 'Items' section shows two items: 'prod-offer-1' and 'prod-plan-1'. A red arrow points to the 'prod-offer-1' item.

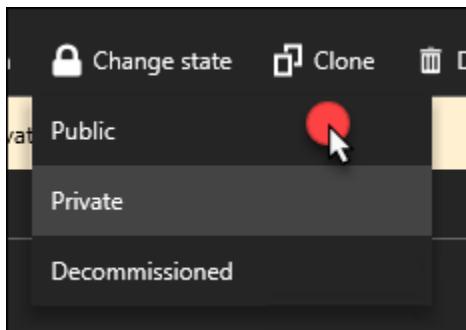
17. Click **prod-offer-1**.



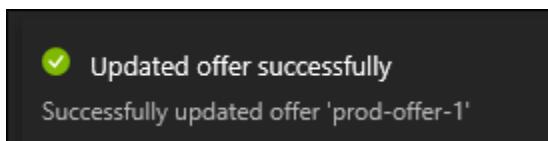
18. Notice the portal shows a warning stating: "**This offer is private, and users cannot see it.**" To fix this, click the **Change state** button.

The screenshot shows the 'prod-offer-1' offer card in the Azure Stack portal. On the left is a sidebar with options like Overview, Access control (IAM), Delegated providers, Offer settings, and Properties. The main area has a 'User subscription' button, a 'Change state' button (which is the target of a red arrow), a 'Clone' button, and a 'Delete' button. A yellow warning bar at the top says 'This offer is private and users cannot see it'. Below the bar, under 'Essentials', there are details: Resource group 'ContosoFinance', Location 'local', Subscription name 'Default Provider Subscription', and Subscription ID (redacted). The 'Offer settings' section shows 'Public' selected (indicated by a red circle on the mouse cursor).

19. Select **Public**.



20. The portal will immediately provide a notification about the update to the offer.



21. Next, open a new browser tab, and navigate to Azure Stack tenant portal <https://portal.local.azurestack.external>. This is the User portal where Contoso Finance will use to provision and manage their Azure Stack service.

Microsoft Azure Stack

Dashboard + New dashboard Edit dashboard Fullscreen Clone Delete

Marketplace

All resources ALL SUBSCRIPTIONS No resources to display Create resources

Get a subscription

Feedback

Quickstart tutorials

Windows Virtual Machines Provision Windows Server, SQL Server, SharePoint VMs

Linux Virtual Machines Provision Ubuntu, Red Hat, CentOS, SUSE, CoreOS VMs

App Service Create Web Apps using .NET, Java, Node.js, Python, PHP

Functions Process events with a serverless code architecture

SQL Database Managed relational SQL Database as a Service

22. Click **Get a subscription**.



23. Give it the name: **Production** and select the **PROD-Offer-1**. Click **Create**.

a.

Get a subscription

Display name Production

* Offer Select an offer

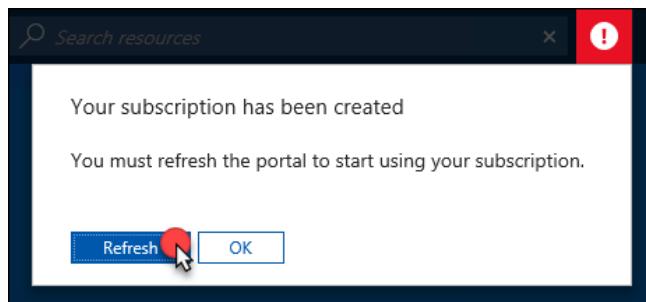
Note: After your subscription is created, you must refresh the portal to start accessing the new services in your subscription.

Choose an offer

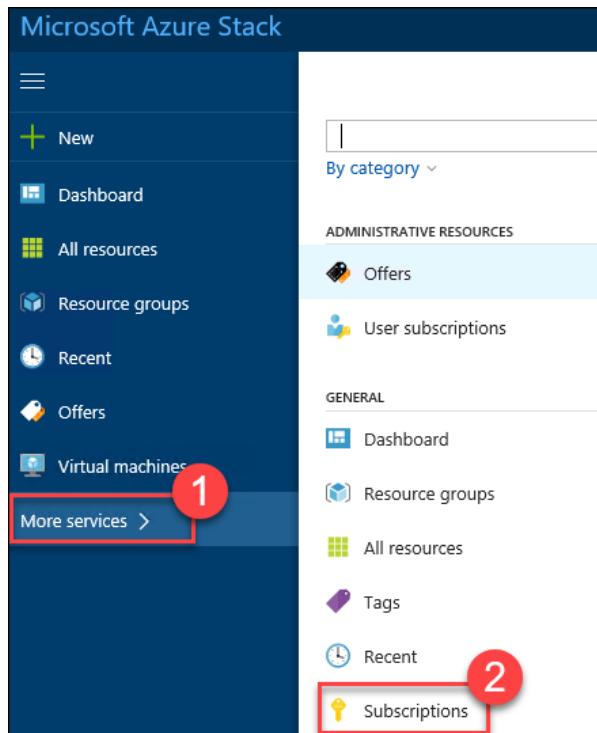
PROD-Offer-1

Create

24. You will need to Refresh the window to start using the new Subscription.



25. Once the portal refreshes, click **More Services > Subscriptions**.



26. The Production Subscription will load, and you can click to review.

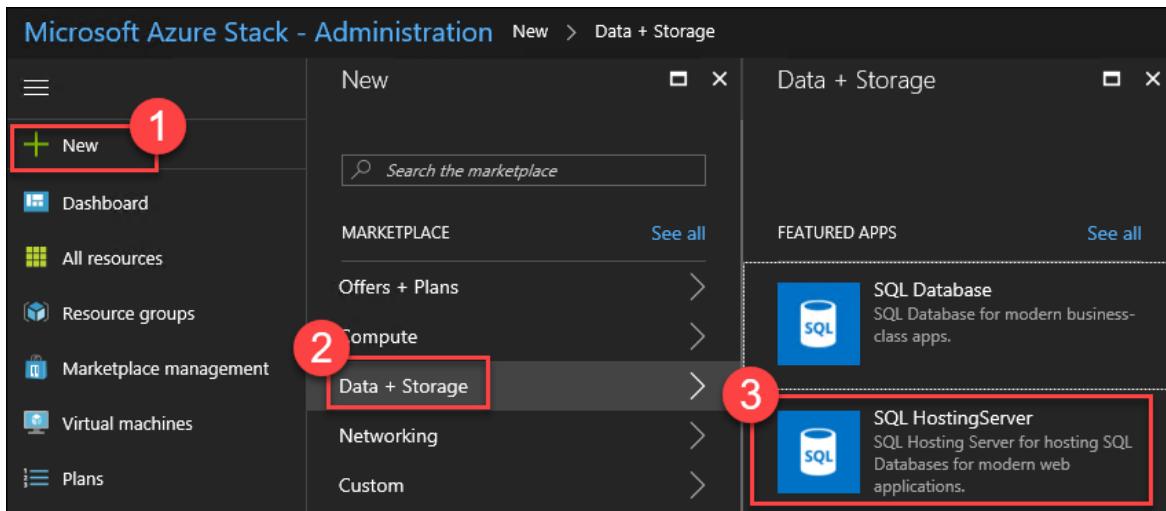
Exercise 2: Deploy the SQL Hosting Server and DB on Azure Stack

Duration: 60 minutes

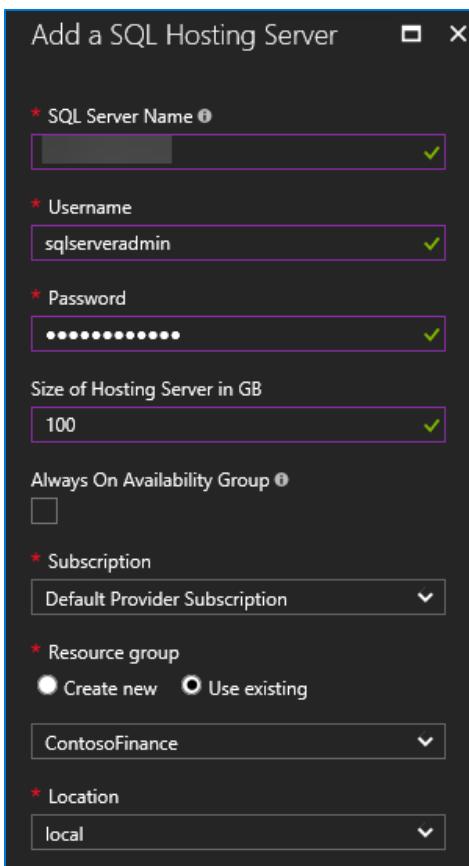
The first step to getting the website up and running is to configure the SQL Database on Azure Stack. This requires you to setup a SQL VM, configure a SQL Hosting Server and finally deploying the SQL database.

Task 1: Create SQL Hosting Server

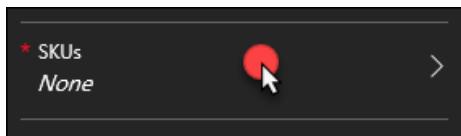
1. In the Azure Stack Admin portal, click **+New, Data + Storage** and then **SQL HostingServer**.



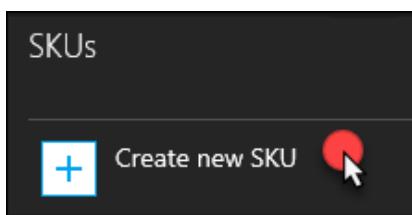
2. Complete the **Add a SQL Hosting Server** blade using the following inputs:
 - a. SQL Server Name: **IP Address of the SQL Server VM created for the App Service Account**
 - b. User Name: **sqlserveradmin**
 - c. Password: **demo@pass123**
 - d. Size of Hosting Server in GB: **100**
 - e. Resource group: **ContosoFinance**
 - f. Location: **Local**



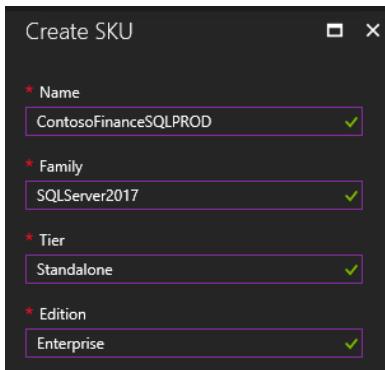
3. Next, click **SKUs**.



4. Click **Create new SKU**.



5. Update the **Create SKU** blade with the following input and click **OK**.
 - Name: **ContosoFinanceSQLPROD**
 - Family: **SQLServer2017**
 - Tier: **Standalone**
 - Edition: **Enterprise** – Tip in a production environment the edition should reflect the actual capabilities



6. Review the Add a **SQL Hosting Server** Blade and click **Create**.

Note: SKUs **can take up to an hour** to be visible in the portal. Users cannot create a database until the SKU is fully created.

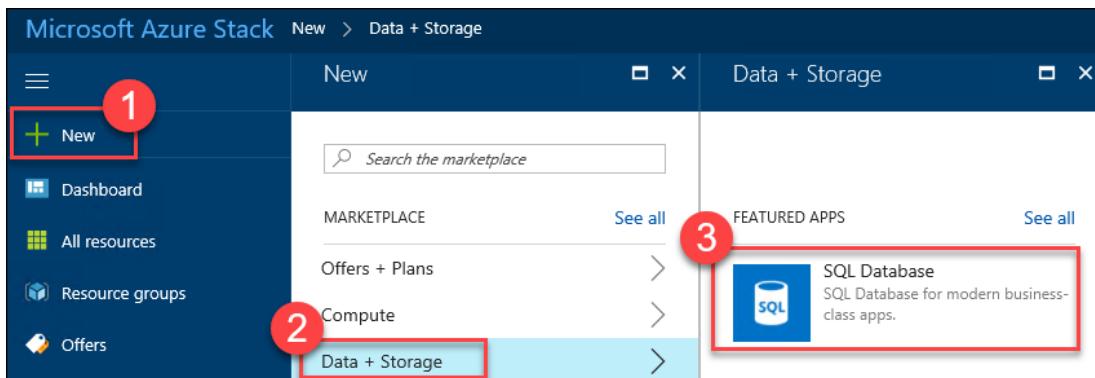
Task 2: Register resource providers in the tenant subscription

1. Launch the Azure Stack tenant portal and click More services -> search for Subscriptions -> click the subscription and then click Resource Providers.
2. Click Register by each of the unregistered resource providers except Microsoft.Resources. After the RPs are registered your screen should look like this:

PROVIDER	STATUS	
Microsoft.Authorization	Registered	Re-register
Microsoft.Commerce	Registered	Re-register
Microsoft.Compute	Registered	Re-register
Microsoft.Gallery	Registered	Re-register
Microsoft.Insights	Registered	Re-register
Microsoft.KeyVault	Registered	Re-register
Microsoft.Network	Registered	Re-register
Microsoft.SQLAdapter	Registered	Re-register
Microsoft.Storage	Registered	Re-register
Microsoft.Subscriptions	Registered	Re-register
Microsoft.Web	Registered	Re-register
Microsoft.Resources	NotRegistered	Register

Task 3: Deploy SQL DB on Azure Stack

1. In the Azure Stack Tenant portal, click **+New, Data + Storage** followed by **SQL Database**.

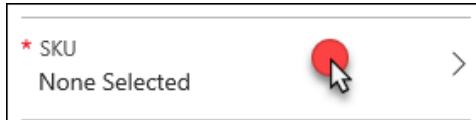


2. Complete the **Create Database** using the following inputs.
 - a. Database Name: **ContosoFinanceWebDB**
 - b. Max Size in MB: **250**
 - c. Resource group: **AzureStackContosoFinance**
 - d. Location: **Local**

The 'Create Database' dialog box is shown. It contains the following fields:

- * Database Name: ContosoFinanceWebDB
- * Collation: SQL_Latin1_General_CI_AS
- * Max Size in MB: 250
- * Subscription: Production
- * Resource Group: Use existing (radio button highlighted with a red box) Create new
AzureStackContosoFinance (dropdown menu highlighted with a red box)
- * Location: local

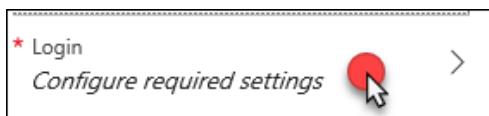
3. Next, click **SKU**.



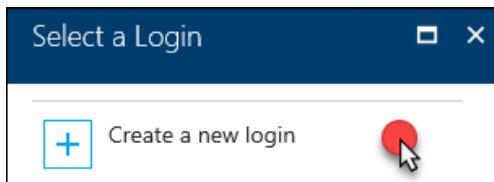
4. Select the **ContosoFinanceSQLPROD** SKU.



5. Click **Login**.

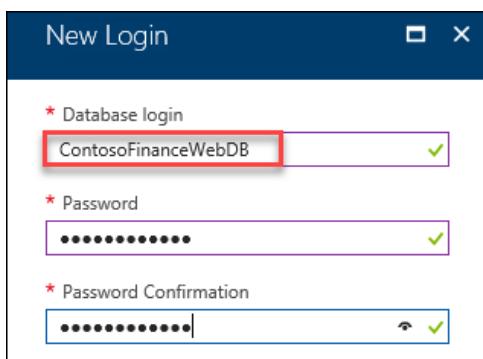


6. Click **Create a new login**.



7. Complete the **New Login** blade using these inputs and click **OK**.

- Database Login: **ContosoFinanceWebDB**
- Password: **Demo@pass123 - Note: Upper case D**



8. Review the **Create Database** blade and click **Create**.

Create Database

* Database Name
ContosoFinanceWebDB ✓

* Collation ⓘ
SQL_Latin1_General_CI_AS

* Max Size in MB
250 ✓

* Subscription
Production

* Resource Group
 Create new Use existing
AzureStackContosoFinance

* Location
local

* SKU
ContosoFinanceSQLPROD >

* Login
ContosoFinanceWebDB >

- Once the deployment completes, use the Azure Stack User portal to locate the **ContosoFinanceWebDB** in the **AzureStackContosoFinance** resource group. Click to examine the details of the new SQL DB running in Azure Stack PaaS.

AzureStackContosoFinance

Resource group

Search (Ctrl+ /)

Add Columns Delete resource group Refresh Move

Overview

Activity log

Access control (IAM)

Tags

SETTINGS

Quickstart

Deployments

Essentials ^

Subscription name (change)
Production

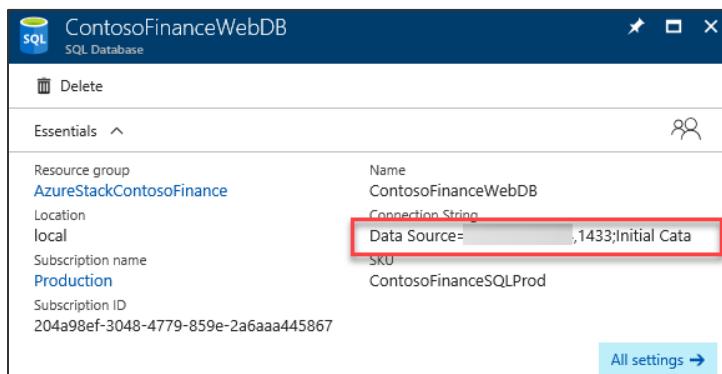
Subscription ID

Filter by name...

13 items

NAME	TYPE
ContosoFinanceWebDB	SQL Database
ContosoFinanceWebDB	SQL Login

- On the **ContosoFinanceWebDB**, the details highlight the connection string and copy to the clipboard. Retain this text for later in the lab by copying to notepad.



Note: If the clipboard copy does not work, you can use the following sample text for your environment. You will need to alter this text to match your configuration.

```
Data Source=X.X.X.X,1433;Initial Catalog=ContosoFinanceWebDB;User  
ID=ContosoFinanceWebDB;Password=demo@pass123
```

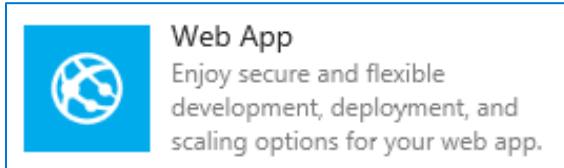
Exercise 3: Deploy Contoso Financial Web Application

Duration: 15-30 minutes

In this exercise, you will provision a website using the Azure Stack portal. The Web App will leverage the SQL DB running in Azure Stack. This is the front-end website that customers will see when browsing for a Mortgage or other financial services products.

Task 1: Create the Web App

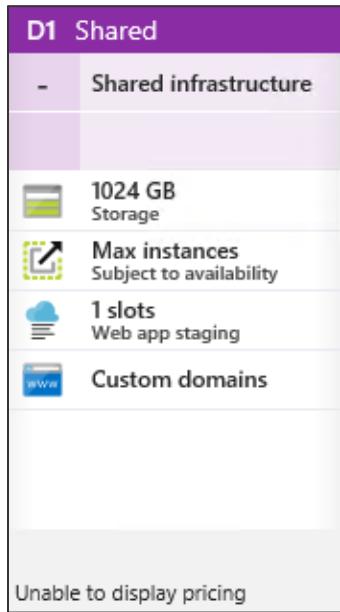
1. From within the Azure tenant portal, click New -> Web + Mobile -> Web App.
2. On the **Everything** blade, select **Web App** followed by **Create**.



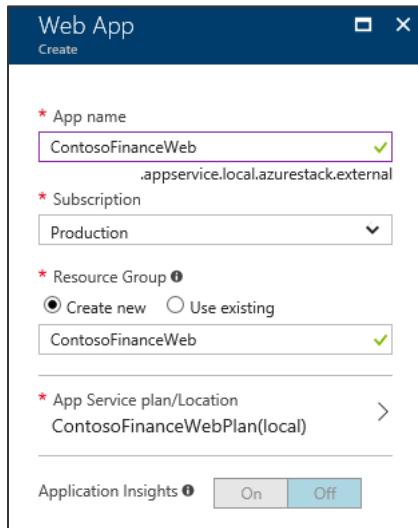
3. On the **Web App** blade, select **App Service plan/Location**.



4. Create a new App Service plan called **ContosoFinanceWebPlan** with the **D1 Shared** pricing tier and click **OK**.



5. On the **Web App** blade, specify the following configuration, and click **Create**.
 - a. App Name: **Specify a unique and valid URL (until the green check mark appears)**
 - b. Resource group: **ContosoFinanceWeb**



Task 2: Provision an Azure Storage Account

1. In the Azure Tenant portal, click **+New, Data + Storage**, and **Storage account**.

The screenshot shows the Microsoft Azure Stack interface. On the left, there's a sidebar with options like 'Dashboard', 'All resources', 'Resource groups', etc. A red box highlights the 'New' button. The main area has a search bar and a 'MARKETPLACE' section with 'Offers + Plans', 'Compute', 'Data + Storage', 'Networking', and 'Custom' options. A red box highlights 'Data + Storage'. To the right, under 'FEATURED APPS', there's a 'SQL Database' entry and a 'Storage account - blob, file, table, queue' entry, which is also highlighted with a red box.

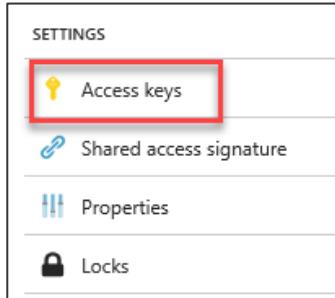
2. On the **Create storage account** blade, specify the following configuration options:
 - a. Name: **unique value for the storage account (ensure the green check mark appears)**
 - b. Resource Group: **AzureGlobalContosoFinance**

The screenshot shows the 'Create storage account' blade. It includes fields for 'Name' (contosofinance), 'Account kind' (General purpose), 'Performance' (Standard), 'Replication' (Locally-redundant storage (LRS)), 'Subscription' (Production), 'Resource group' (ContosoFinanceWeb), and 'Location' (local). Each field has a validation checkmark icon next to it.

3. Click **Create**.

Create

4. After the storage account has completed provisioning, open the storage account by clicking **More services**, **Storage accounts**, and storage account name.
5. On the **Storage** account blade, scroll down, and select the **Access keys** option.



6. On the **Access keys** blade, click the copy button by **key1** to copy the **Key**. Put the value in notepad for later reference.

A screenshot showing the 'Default keys' section of the Access keys blade. It displays a table with two columns: 'NAME' and 'KEY'. The 'key1' row shows the key value 'ARcu0xweetMFG0QLjdK2MXFMLsNa5quP60uxeos6FdtPLxu+PefKpZ...'. A blue copy-to-clipboard icon is located next to the key value. Below this, a Notepad window titled 'Untitled - Notepad' shows the same key value 'ARcu0xweetMFG0QLjdK2MXFMLsNa5quP60uxeos6FdtPLxu...'.

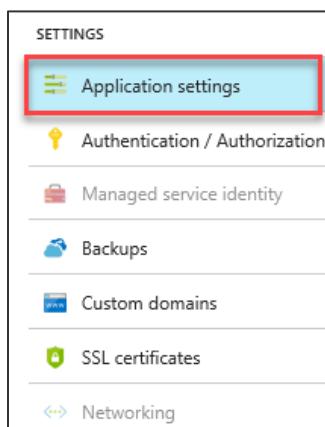
7. On the **Access keys** blade, click the copy button by **key1** on the **Connection string**. Put the value in notepad for later reference.

A screenshot showing the 'DefaultEndpointsProtocol=https;AccountName=' connection string in the Access keys blade. A blue copy-to-clipboard icon is located next to the connection string. Below this, a Notepad window titled 'Untitled - Notepad' shows the connection string 'DefaultEndpointsProtocol=https;AccountName=contosofinanceweb;AccountKey=ARcu0xweetMFG0QLjdK2MXFMLsNa5quP60uxeos6FdtPLxu+'.

Note: If the copy to clipboard button does not work, you may need to highlight the key and copy by right-clicking. Some versions of Internet Explorer have issues with this functionality.

Task 3: Update the configuration strings

1. On the left pane of the **contosofinanceweb** Web App, click on **Application settings**.



2. Scroll down and locate the **App settings** section.

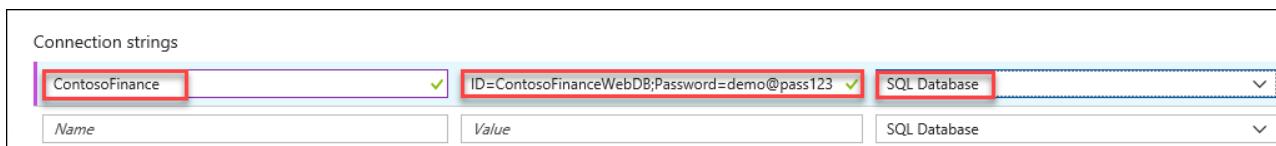
App settings			
WEBSITE_NODE_DEFAULT_V...	6.9.1	<input type="checkbox"/> Slot setting	...
Key	Value	<input type="checkbox"/> Slot setting	...

3. Add a new **App setting** with the following values:
 - a. Key: **AzureQueueConnectionString**
 - b. Value: **enter the Connection String for the Storage Account that was just created**

App settings			
WEBSITE_NODE_DEFAULT_V...	6.9.1	<input type="checkbox"/> Slot setting	...
AzureQueueConnectionString	DefaultEndpointsProtocol=https;AccountName=contosofinance;AccountKey=...	<input checked="" type="checkbox"/> Slot setting	...
Key	Value	<input type="checkbox"/> Slot setting	...

4. Move to your Notepad with the SQL Server Connection string copied from Azure Stack. Update the following items in the string:
 - a. Password: **Demo@pass123**
5. Locate **Connection Strings** below App settings in the Azure tenant portal, add a new **Connection String** with the following values:
 - a. Name: **ContosoFinance**
 - b. Value: **enter the Connection String for the SQL Database in Azure Stack you just updated**

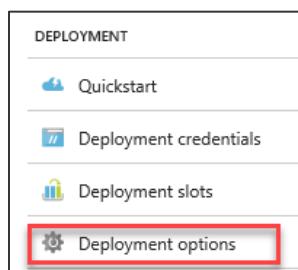
c. Type: **SQL Database**



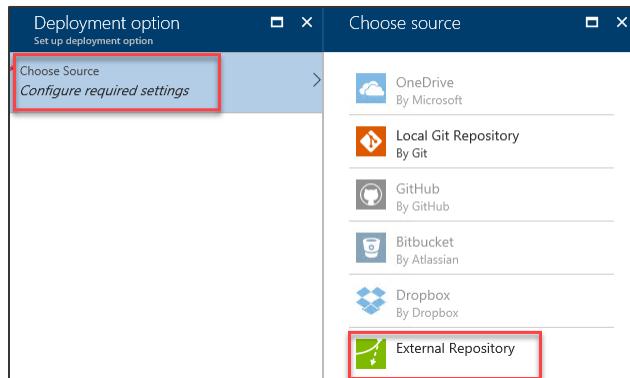
6. Click **Save**.

Task 4: Publish the Contoso Financial Web Application

1. From within the web app blade, click on **Deployment Options**.

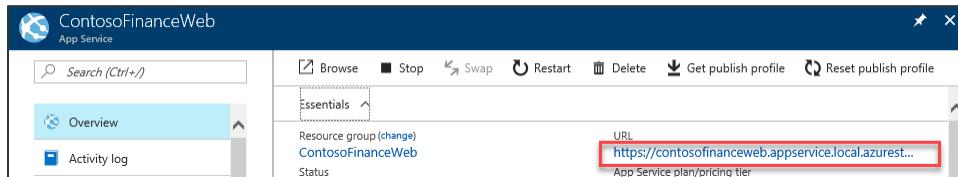


2. Click **Choose Source**, and then **External Repository**.



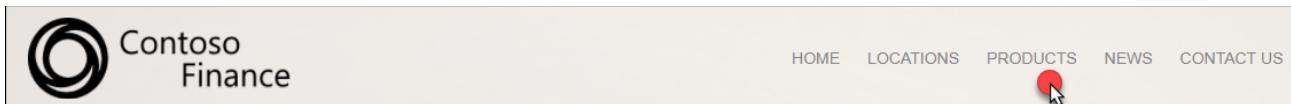
3. Paste <https://github.com/opsgility/contosofinanceweb> as the **Repository URL** and click **OK**.
4. Click on the Deployment options button and monitor until the application is deployed.

5. Click the Overview tab, and then click the URL. You should see the Contoso Finance web app.



Note: You may get an error about CORS. This can be ignored, as it will be configured later in the lab.

6. Validate the website by clicking the **Products** link on the menu. If the products return, the connection to the database is successful.



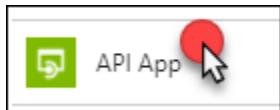
Exercise 4: Deploy the customer offers Web API

Duration: 15-30 minutes

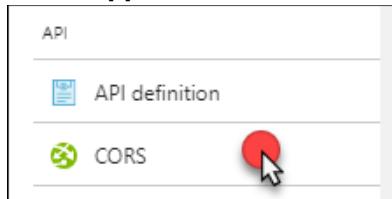
In this exercise, you will provision an Azure API App using the Azure Stack portal. This API application is part of the front-end Web Applications and makes recommendations to the user on products the company wishes to highlight. The API App will leverage the SQL Database deployed previously.

Task 1: Provision the offers Web API App

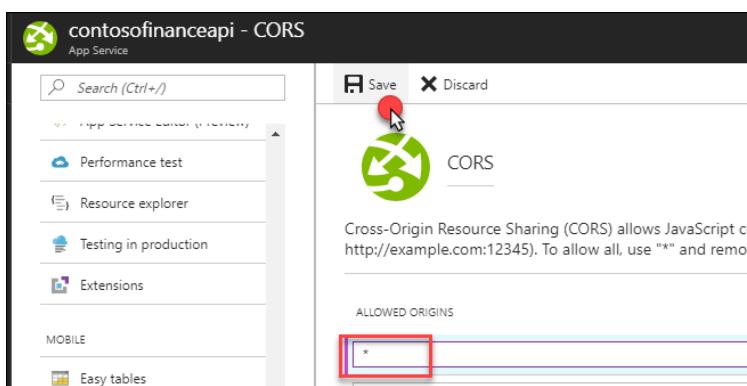
1. Using the Azure Stack Tenant portal, click **+New, Web + Mobile**, and click **API App**.



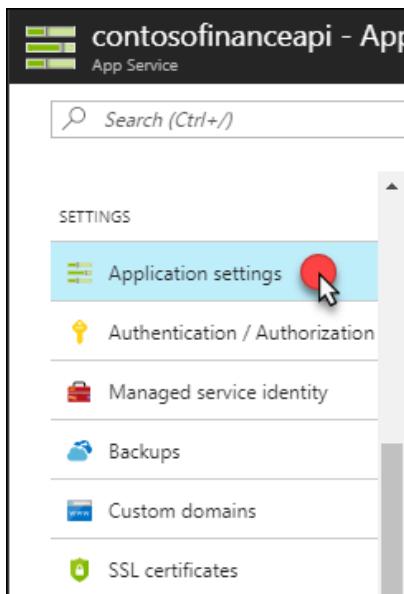
2. Click on **Create**.
3. On the new **API App** blade, **specify a unique name** for the App Name, and ensure the previously used Resource Group and App Service Plan is selected.
4. After the values are accepted click **Create**.
5. On the **App Service** blade, scroll down, and click on **CORS** within the API section of the left pane.



6. In the **ALLOWED ORIGINS** text box specify * and click **Save**.



7. On the **App Service** blade for the Offers API, click on **Application settings**



8. Scroll down, and locate the **Connection strings** section.

A screenshot of the 'Connection strings' section in the Azure Stack portal. It shows a table with columns for Name, Value, Type (set to 'SQL Database'), Slot setting, and an ellipsis button. A message at the top says 'No results'.

9. Locate **Connection Strings** below App settings in the Azure global portal Add a new **Connection String** with the following values:
- Name: **ContosoFinance (must match exactly – case sensitive)**
 - Value: **enter the Connection String for the SQL Database in Azure Stack you just updated**
 - Type: **SQL Database**

A screenshot of the 'Connection strings' section in the Azure global portal. A new row has been added with the following values:

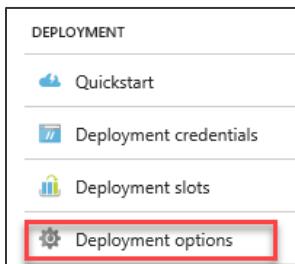
- Name: ContosoFinance (highlighted with a red box)
- Value: ID=ContosoFinanceWebDB;Password=demo@pass123 (highlighted with a red box)
- Type: SQL Database (highlighted with a red box)

The 'Name' and 'Value' fields are empty, and the 'Type' dropdown is set to 'SQL Database'.

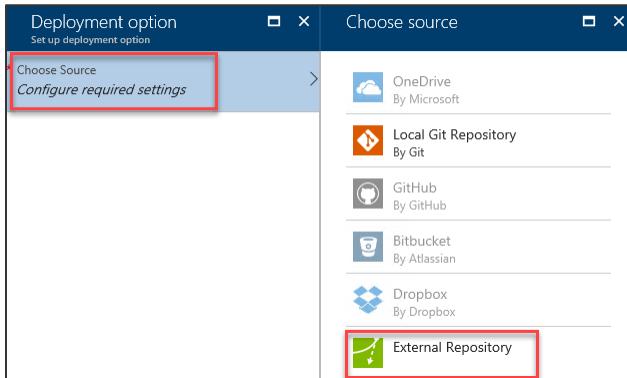
10. Click **Save**.

Task 2: Deploy the Contoso.Apps.Financial.Offers project

- From within the API app blade, click on **Deployment options**.



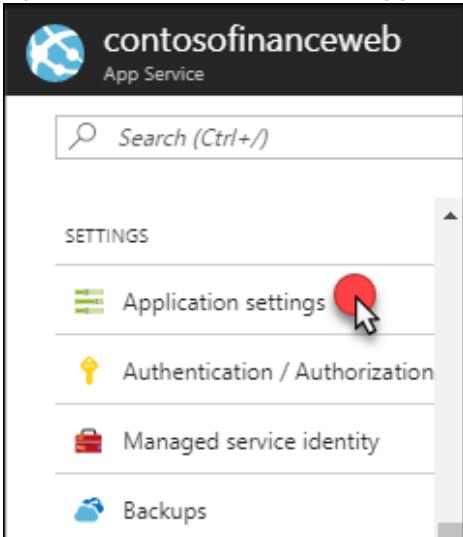
2. Click **Choose Source**, and then **External Repository**.



3. Paste <https://github.com/opsgility/contosofinanceoffers> as the **Repository URL** and click **OK**.
4. Click on the Deployment options button and monitor until the application is deployed.
5. On the **Overview** tab, copy the URL for the web app to the clipboard.

Task 3: Update the Application Settings of the Web App with the API URL

1. Open the ContosoFinanceWeb application in the Azure Stack Tenant portal and click on Application settings.



2. Scroll down and locate the **App settings** section.

Key	Value	Slot setting
WEBSITE_NODE_DEFAULT_VERSION	6.9.1	<input type="checkbox"/> Slot setting
AzureQueueConnectionString	DefaultEndpointsProtocol=https;AccountName=contosofinanceapi;AccountKey=ARcuOxweetMFGOQLjdK2MXFMLsNa5quP60uxeos6FdtPLxu...	<input type="checkbox"/> Slot setting
offersAPIUrl	https://contosofinanceapi.azurewebsites.net/api/get	<input type="checkbox"/> Slot setting

3. Add a new **App Setting** with the following values:
 - a. Key: **offersAPIUrl**
 - b. Value: enter the **HTTPS** URL for the Offers API App with **/api/get** appended to the end. Example:
<https://contosofinanceapi.azurewebsites.net/api/get>

Key	Value	Slot setting
WEBSITE_NODE_DEFAULT_VERSION	6.9.1	
AzureQueueConnectionString	ARcuOxweetMFGOQLjdK2MXFMLsNa5quP60uxeos6FdtPLxu...	
offersAPIUrl	https://contosofinanceapi.azurewebsites.net/api/get	

4. Click on **Save**.

Note: Ensure the API URL is using **SSL** (<https://>), or you will see a CORS errors when loading the webpage.

5. Connect to the URL of the **contosofinanceweb** Web App.

Resource group (change)	OS name
AzureGlobalContosoFinance	Windows Server 2016
Status	URL
Running	https://contosofinanceweb.azurewebsites.net
Location	App Service plan/pricing tier
South Central US	ContosoFinanceWeb (Standard: 1 Small)

6. On the homepage, you should see the latest offers populated from the offers API.

The image shows the homepage of the Contoso Finance website. At the top left is the Contoso Finance logo, which consists of a stylized 'C' icon followed by the text 'Contoso Finance'. The top right features a navigation bar with links for HOME, LOCATIONS, PRODUCTS, NEWS, and CONTACT US, along with a search icon. The main header 'CONTOSO FINANCE' is displayed in a large, bold, black font within a white rectangular box. Below the header is a photograph of a cyclist in a white, red, and blue jersey, wearing a helmet and sunglasses, in a starting position. A red button labeled 'VIEW PRODUCTS' is overlaid on the image. Underneath the main banner, the section 'Today's Offers' is titled 'Today's Offers' in red, with the subtext 'We found these great products just for you!'. Three product cards are shown: '20 Year Fixed' (3.5 %), '3 Year CD' (2 %), and '5 Year CD' (2.45 %). Each card includes a 'PRODUCT DETAILS' button.

Contoso Finance

HOME LOCATIONS PRODUCTS NEWS CONTACT US

VIEW PRODUCTS

CONTOSO FINANCE

Today's Offers
We found these great products just for you!

20 Year **3.5** %
Fixed

3 Year **2** %
CD

5 Year **2.45** %
CD

PRODUCT DETAILS

PRODUCT DETAILS

PRODUCT DETAILS

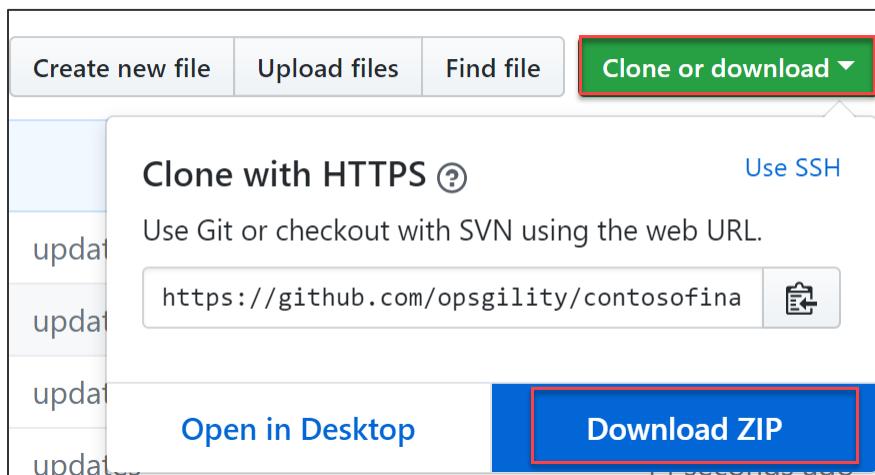
Exercise 5: Automating backend processes with Azure functions

Duration: 15-30 minutes

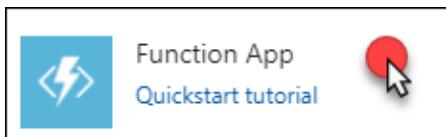
Contoso wants to automate the process of generating applications in PDF format and using Azure Functions. In this exercise, you will provision a Function App using the Azure Stack portal. This Function App will watch the Azure Storage Queue for a message that the web application has submitted, process the application creating a PDF and storing it in Azure Blob Storage.

Task 1: Create an Azure function to generate PDF receipts

- From your Azure Stack Host, navigate to the following repository: <https://github.com/opsgility/contosofinancefunction> and click **Clone or download**, and then **Download ZIP**. Download to the **C:\HOL** folder. After the file is downloaded, right click and extract to **C:\HOL**.



- From the Tenant portal, click **+New, Web + Mobile**, and then click **Function App**.



- Complete the **Create Function App** blade using the following inputs:
 - App name: **specify a unique name**
 - Resource Group: **ContosoFinanceWeb**
 - Hosting Plan: **App Service Plan**
 - App Service plan/Location: **ContosoFinanceWeb**
 - Storage Account: **select your storage account**

Function App Create

* App name: contosofinancefunction .appservice.local.azurestack.external

* Subscription: Production

* Resource Group: ContosoFinanceWeb

* OS: Windows

* Hosting Plan: Consumption Plan

* Location: local

* Storage: contosofinance

Application Insights: Off

This form is used to create a new Function App. It requires filling in fields for App name, Subscription, Resource Group, OS, Hosting Plan, Location, and Storage. Application Insights can be turned On or Off.

4. Click **Create**.
5. Using the Azure Stack portal, open the Function App you just created, click **Functions** and then **New Function**.

contosofinancefunction
Function Apps

All subscriptions

Function Apps

contosofinancefunction

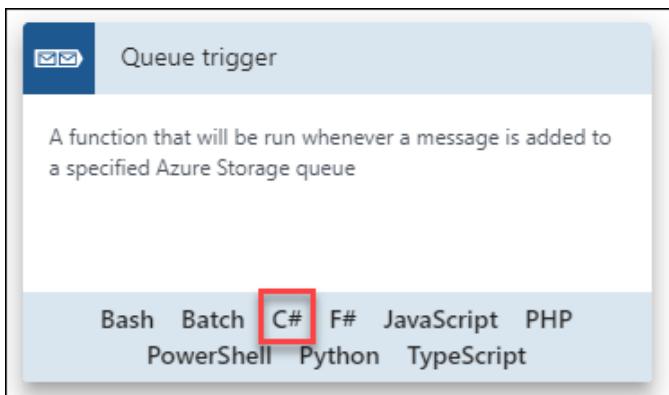
Functions

New function

No results

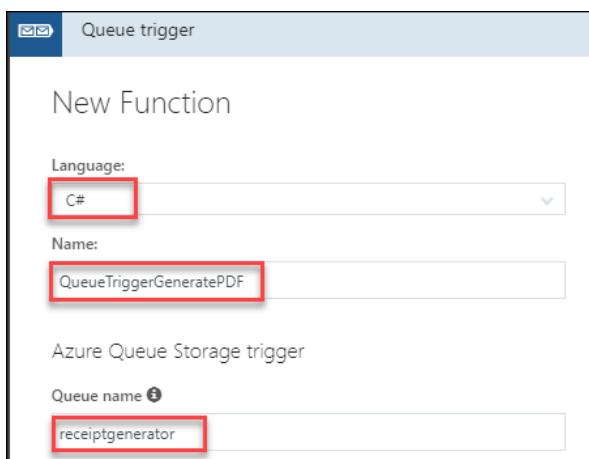
This screenshot shows the Azure Stack portal's Function Apps blade for the 'contosofinancefunction' app. The 'Functions' section is displayed, showing a search bar and a table with columns 'NAME' and 'STATUS'. A red box highlights the '+ New function' button.

6. Locate the **Queue trigger** box and click **C#**.

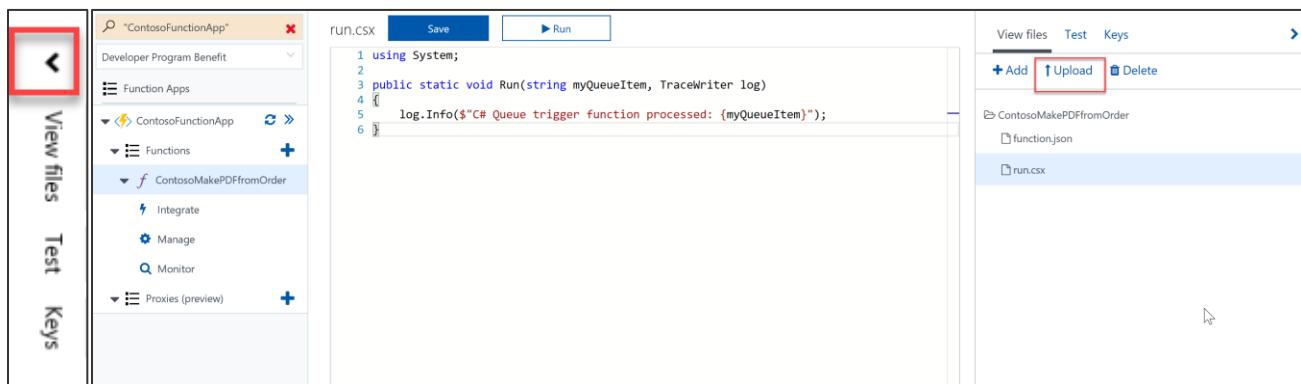


7. Complete the **New Function** blade using the following inputs and click **Create**.

- Language: **C#**
- Name: **QueueTriggerGeneratePDF**
- Queue name: **receiptgenerator** (must be this exact text)



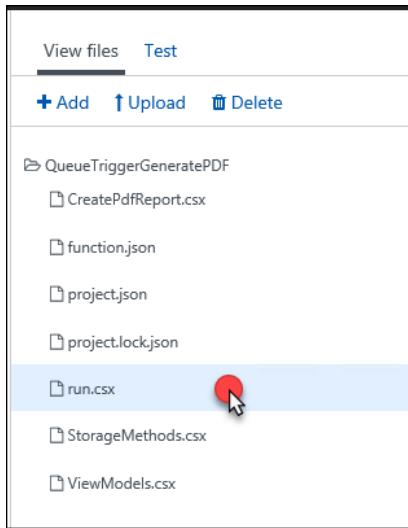
8. Expand the View files area on the right of the code window, and click **Upload**.



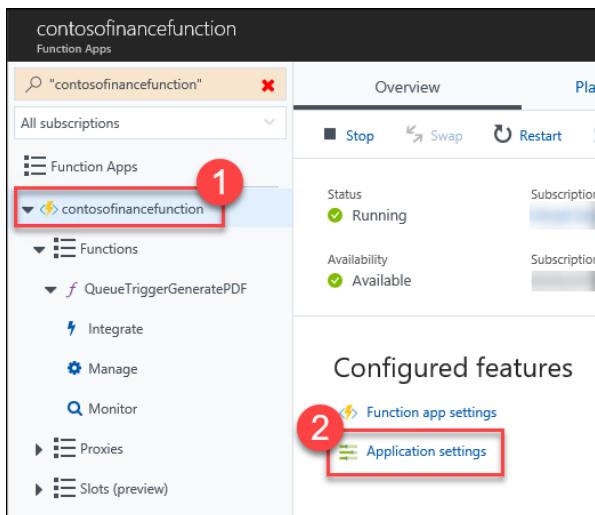
9. Upload the following files from your LABVM in the **C:\HOL\contosofinancefunction-master** folder:

- CreatePdfReport.csx
- Project.json
- run.csx
- StorageMethods.csx
- ViewModels.csx

10. Click on **run.csx** to refresh the code editor.



11. Select the **name of your function app followed by Application settings**.



12. Scroll down to the **Application settings** and click **+Add new setting** to add a storage connection. This storage account will be used to write the PDFs to blob storage.

a. Name: **contosofinancemachine** (must be this name exactly)

- b. Value: **paste the connection string for the storage account created earlier in the lab**

13. Locate **Connection Strings** below Application settings in the Azure tenant portal, and click **+Add a new Connection String** with the following values:
- Name: **ContosoFinance (must match exactly – case sensitive)**
 - Value: **enter the Connection String for the SQL Database**
 - Type: **SQL Azure**

The screenshot shows the Azure tenant portal interface. In the 'Application settings' section, there is a table with several entries. One entry, 'contosofinancestorage', has its value highlighted with a red box. In the 'Connection strings' section, there is a table with one entry, 'ContosoFinance', which also has its value highlighted with a red box.

Application settings		
AzureWebJobsDashboard	DefaultEndpointsProtocol=https;AccountName=contosofinance;AccountKey=NYwiPNAWdcCVuA97fznQKspMRL8qGCWZEEkoCu6lGKgF7k2d...	
AzureWebJobsStorage	DefaultEndpointsProtocol=https;AccountName=contosofinance;AccountKey=NYwiPNAWdcCVuA97fznQKspMRL8qGCWZEEkoCu6lGKgF7k2d...	
FUNCTIONS_EXTENSION_VERSION	~1	
contosofinancestorage	DefaultEndpointsProtocol=https;AccountName=contosofinance;AccountKey=NYwiPNAWdcCVuA97fznQKspMRL8qGCWZEEkoCu6lGKgF7k2d...	

Connection strings		
ContosoFinance	Hidden value. Click to show.	SQLAzure

14. Scroll back up to the top of the blade and click **Save**.

The screenshot shows the Azure Function Apps blade for the 'contosofinancefunction' app. On the left, there is a sidebar with a search bar containing 'contosofinancefunction'. The main area shows an 'Overview' blade with two buttons at the top: 'Save' and 'Discard'. A red circle highlights the 'Save' button. Below the buttons, the text 'General settings' is visible.

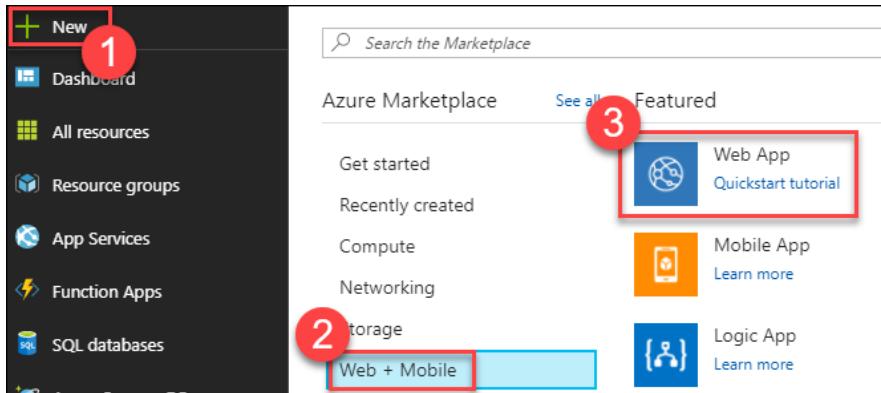
Exercise 6: Deploy Contoso Finance Admin website

Duration: 15-30 minutes

In this exercise, you will provision the admin website to be used by employees to review applications submitted.

Task 1: Provision the Contoso Finance Admin Web App

1. In the Azure tenant portal, click **New, Web + mobile**, and select **Web App**.



2. Specify a **unique URL** for the Web App, ensure the **same App Service Plan** as well as the **ContosoFinanceWeb** resource group you have used throughout the lab are selected.

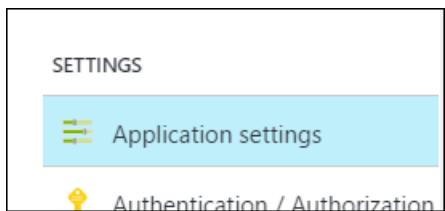
A screenshot of the 'Create Web App' dialog. It shows the following fields:

- App name:** contosofinanceadmin (with a green checkmark)
- Subscription:** Production
- Resource Group:** Use existing (radio button selected), ContosoFinanceWeb
- App Service plan/Location:** ContosoFinanceWebPlan(local)
- Application Insights:** Off (button)

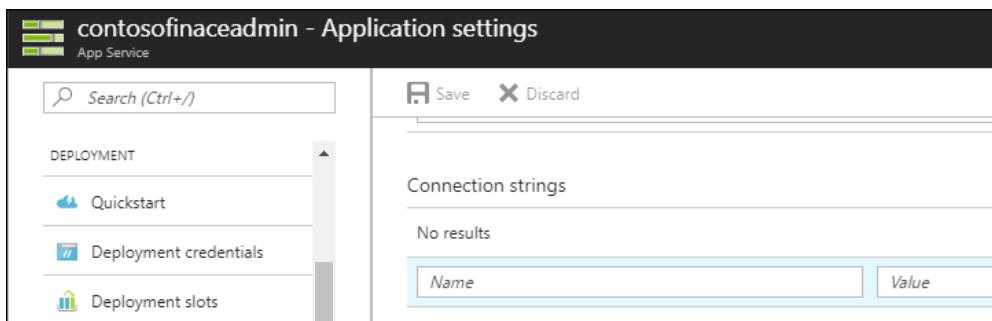
3. After the values are accepted, click **Create**.
4. Navigate to the **App Service** blade for the Admin app recently provisioned.



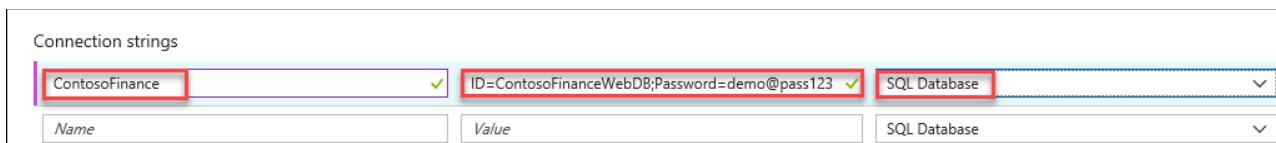
5. On the **App Service** blade, click on **Application settings** in the left pane.



6. Scroll down and locate the **Connection strings** section.



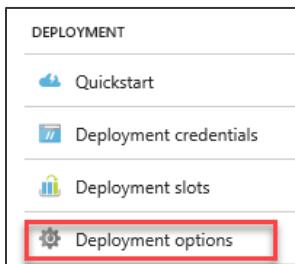
7. Locate **Connection Strings** below App settings in the Azure tenant portal add a new **Connection String** with the following values:
- Name: **ContosoFinance**
 - Value: **enter the Connection String for the SQL Database in Azure Stack you just updated**
 - Type: **SQL Database**



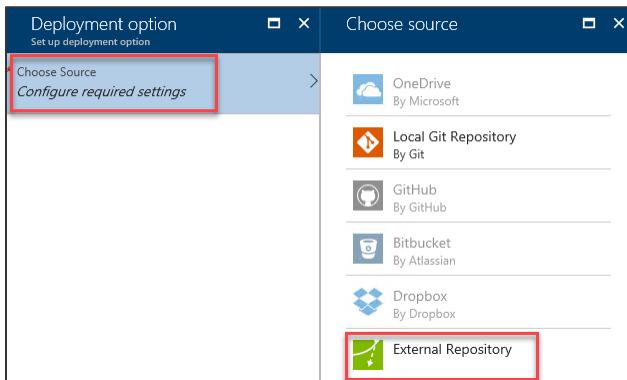
11. Click **Save**.

Task 2: Deploy the call center admin Web App from Visual Studio

1. From within the web app blade, click on **Deployment options**.



2. Click **Choose Source**, and then **External Repository**.

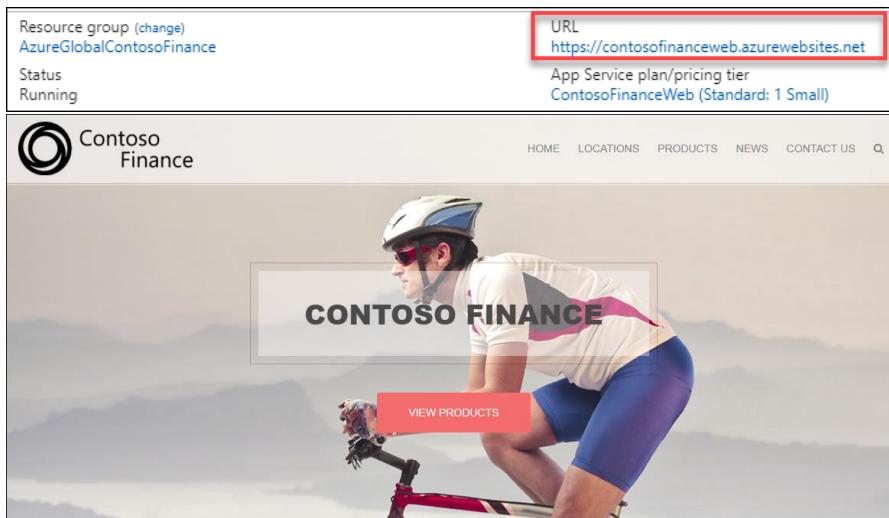


3. Paste <https://github.com/opsgility/contosofinanceadmin> as the **Repository URL** and click **OK**.
4. Click on the Deployment options button and monitor until the application is deployed.
5. On the **Overview** tab, copy the URL for the web app to the clipboard.
6. Connect to the **contosofinanceadmin** portal to see the list of applications that have been completed.

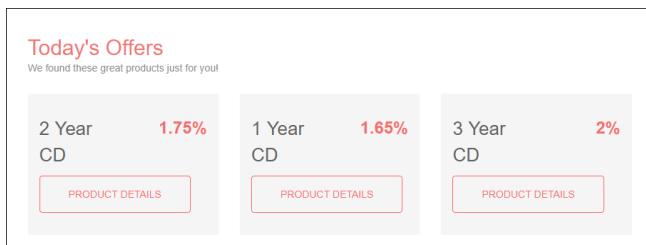
A screenshot of the "Contoso Finance Admin" portal. The top navigation bar includes "Contoso", "Home", "About", and "Sign in". The main content area has a heading "Contoso Finance Admin" and a sub-instruction: "Below is a list of applications, sorted by reverse order date. You may click on an application to see its details and to obtain the generated PDF receipt." A "Learn more »" button is present. Below this is a section titled "Completed Applications" with a table. The table has columns: "Application Details", "Application Date", "First Name", and "Last Name". Under "Application Details", there are three rows, each with a "Details" link. The "Application Date" and "First Name" columns show blurred data, while the "Last Name" column shows a single row of data.

Note: In production this application would be secured using Azure AD for authentication purposes.

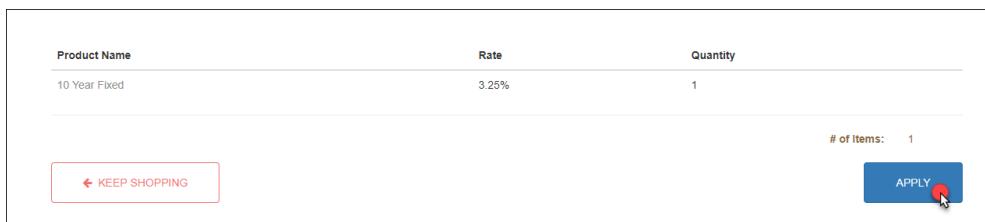
7. Since the application is fully deployed, you will want to see it work end to end. Open the URL for the contosofinanceweb Web App. The application will load in the browser.



8. Notice how the API application loaded the Today's Offers area. Click through to one of the products and add it to your cart.



9. Click **Apply**.



10. Complete the Application and click **Continue** followed by **Complete Application** on the confirmation screen.

City
Highland Village

State
TX

Postal Code
75077

Country
United States

Phone
212-555-1212

Receive SMS Notifications?

11. Now, to act as an employee, open the Admin application to see the submitted applications. Click **Details** on one of the applications.

Resource group (change)
AzureGlobalContosoFinance

Status
Running

URL
<https://contosofinanceadmin.azurewebsites.net>

App Service plan/pricing tier
ContosoFinanceWeb (Standard: 1 Small)

Contoso Finance Admin

Below is a list of applications, sorted by reverse order and to obtain the generated PDF receipt.

Learn more »

Completed Applications

Application Details	Application Date
Details	

12. Notice the details of the application. This data is stored in SQL DB running in PaaS on Azure Stack. Click [Download application to view a sample PDF.](#)

The screenshot shows a web application interface titled "Home ➔ Application Details".

Contact Information

First Name:	John
Last Name:	Adams
Email Address:	jadams@contosofinance.com
Phone Number:	12344554
Address:	1701 Shoal Creek
City:	Highland Village
State/Province:	TX
Postal Code:	75077

Application Details

Application date: [REDACTED]

Item	Qua
15 Year Fixed	1

[Download application](#)

© 2018 - Contoso Finance Admin

After the hands-on lab

Duration: 10 minutes

In this final task you will clean up the Azure Resources that you have created for the hands-on lab. This task is optional.

1. If provisioned using the Azure Stack Developer Kit in an Azure VM, delete the resource group your Azure Stack Host VM is running in.
2. If running on your own Developer Kit, delete all the resource groups from the Azure Stack portal that you created during the execution of this lab.