

Deploying Migrated IBM® Notes® Applications to the Cloud

A guide on deploying OpenText™ Composer Notes
application to Microsoft® Azure™

Table of Contents

OpenText Composer Notes Deployment in Azure3

Overview3

Azure PaaS versus IaaS for Composer Notes Deployment4

Azure Setup Using Azure PaaS
(Azure Database and Azure Web App).....5

OpenText Composer Notes Deployment in Azure

The purpose of this document is to demonstrate how Composer Notes applications can be deployed in Azure PaaS. The differences between Azure PaaS and IaaS will be highlighted as both can be used and can even be mixed together to suit the needs of the deployment. One of the greatest benefits of deploying Composer Notes applications in Azure is time of deployment, as a complete production environment can be set up in as little as a few hours.

Overview

Composer Notes is a software solution that automates the conversion of complex Notes applications to the Microsoft .Net environment while preserving the business logic and user interface. By preserving the user interface and business logic/workflow, Composer enables the rapid adoption of the Microsoft platform with minimal disruption to the organization and the end users.

The entire Notes application, including the business logic (LotusScript® & Formula Logic,) user interface, security, and document data with attachments is converted to the Microsoft platform. The corresponding Microsoft technologies in the converted application are Visual C #,® ASPX, Active Directory®, and SQL Server®.

The converted application is maintained and extended in Visual Studio®. Since it is a web app, it can be deployed in Azure.

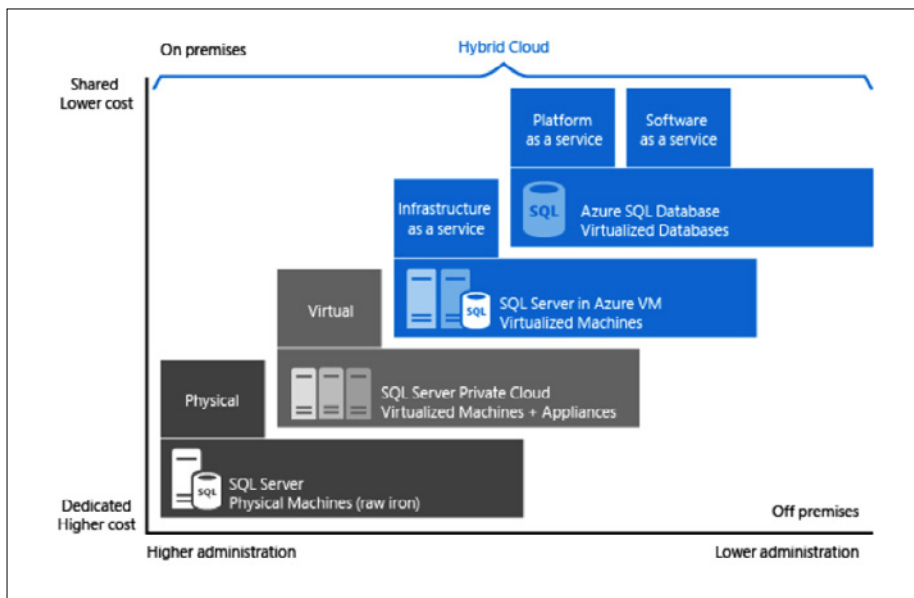


Azure PaaS versus IaaS for Composer Notes Deployment

Azure has two options for hosting a SQL Database. You can either use IaaS (Infrastructure as a Service) or PaaS (Platform as a Service). IaaS requires more administration, as you have to administer the Virtual Machine as well as SQL Server, just as you would in a traditional physical or virtual setup. With Azure SQL database, it is abstracted one level higher and you only administer the SQL server instance, and no longer deal with a virtual machine.

Azure also offers IaaS and PaaS with Azure Web Apps or an Azure VM running IIS. There are a few distinctions between the two, as an Azure Web App is intended to be a publicly accessible site, although security can easily be implemented by restricting to IP address ranges. The front end and back end don't both have to both be IaaS or PaaS. They can be intermixed easily to suit the needs of your business.

This Microsoft illustration shows that hosting the database with PaaS will require less administration and cost less than IaaS. Also, IaaS and PaaS will both be cheaper and require less administration than a physical or a locally hosted virtual deployment.



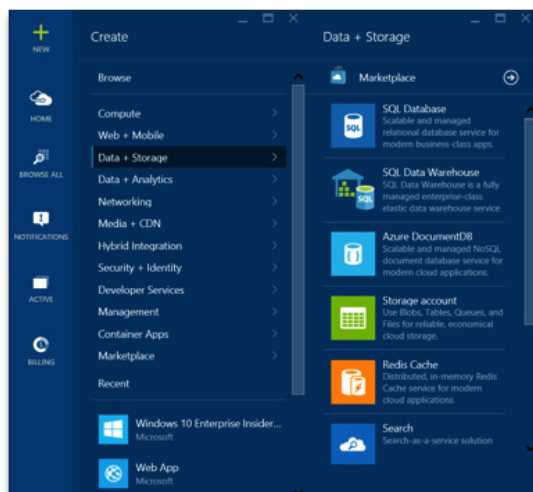
Composer Notes runs identically on Azure IaaS as it would outside of the cloud. The ease of adjusting the performance alleviates any worries over buying too little or too much hardware.

Running Composer Notes in Azure PaaS also performs without issue although the setup is different than configuring to run in IaaS. The setup for PaaS is below.

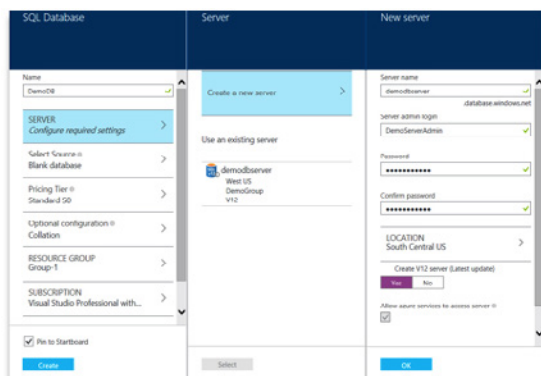
Azure Setup Using Azure PaaS (Azure Database and Azure Web App)

Creating the Azure Database:

- a. In the Azure Portal, select “NEW” on the main menu, then select “Data + Storage” in the “Create” window. Finally select, “SQL Database” from the “Data + Storage” window.



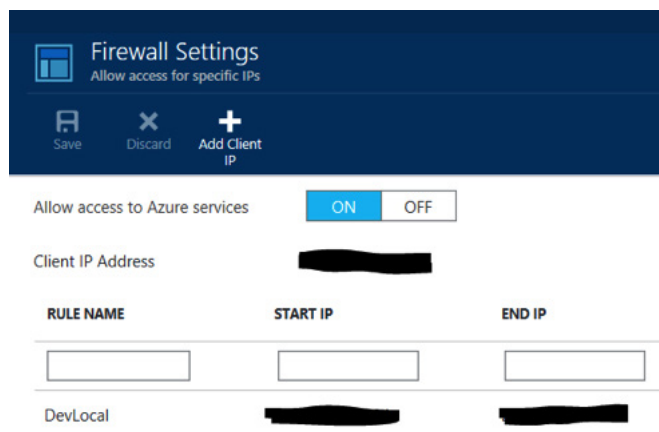
- b. Enter a name for the Database and change the Pricing Tier if needed. If you do not have a SQL Server instance created, you will do it here by selecting “SERVER” then “Create a new server” where you will give a name and enter login credentials.



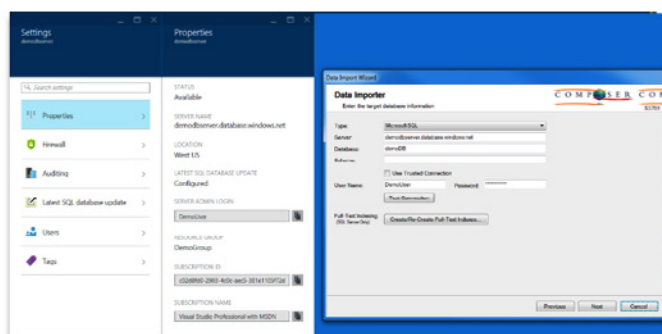
- c. Now the Database Server and SQL Database are created.

Importing Data into the Azure Database:

- a. In order to administer the database, you must go into the Firewall Settings for the database and add the IP address of the computer you wish to administer the database from. You will need to enter a name and an IP address Range.



- b. Open the information for the Azure Database before starting up the Composer Data Importer. Once you get to the connection tab in the Data Importer, enter all the settings from the Azure information pane for the database then execute the importation of the data and your back end will be completely set up.

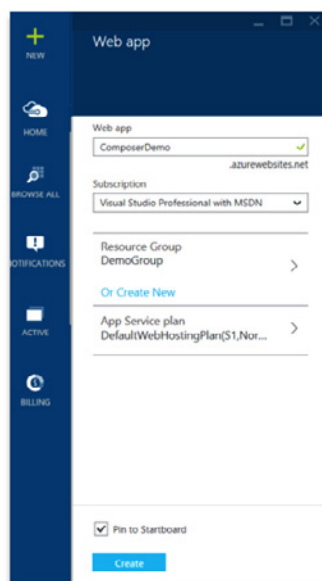


Creating the Azure Web App:

- a. In the Azure portal, click on “NEW” in the main menu, then “Web + Mobile”. Next, click on “Web App.”



- b. Next, name the Web App and adjust the App Service plan if needed. Now all that is needed is to publish a website to this Web App.



Publishing the Website to an Azure Web App:

Before starting this section, if you don't want the website to be publicly accessible, you will need to add a section in the web.config to screen IP addresses in order to only allow ones from a certain range. To do this, add the code below to <system.webServer> in web.config, modifying the IP ranges as needed.

<security>

<ipSecurity allowUnlisted="false" denyAction="NotFound">

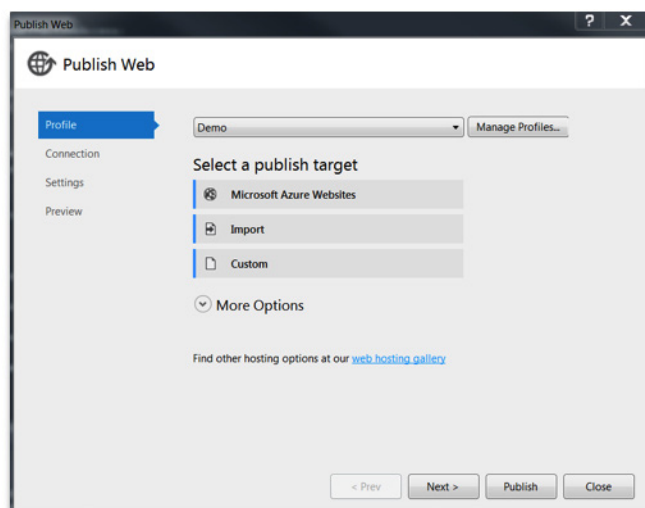
<add allowed="true" ipAddress="10.0.0.0" subnetMask="255.255.0.0"/>

<add allowed="true" ipAddress="192.168.1.18"/>

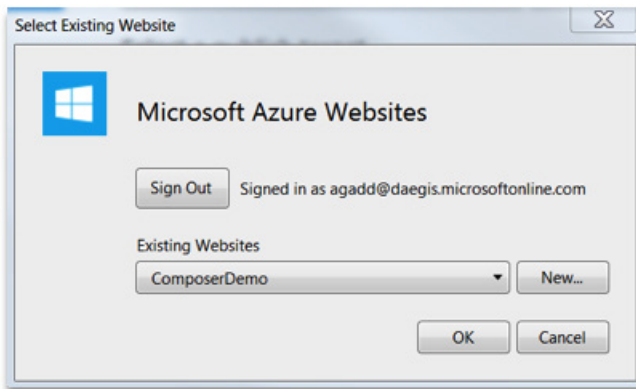
</ipSecurity>

</security>

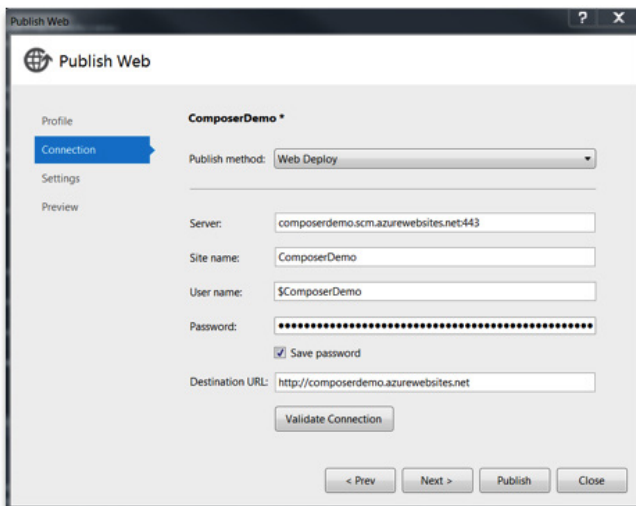
- a. In Visual Studio, select Build->Publish Website. In the dialog box, select Microsoft Azure Websites as the publish target.



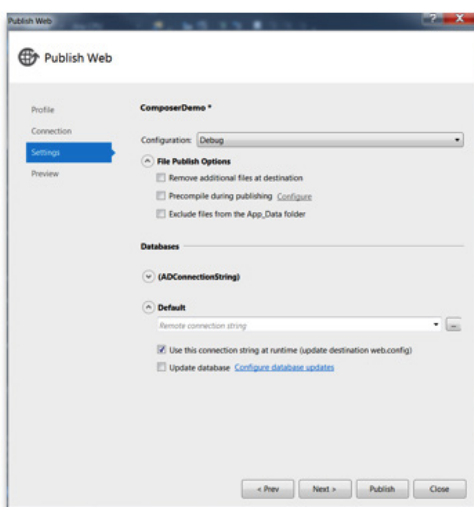
- b. Sign into your Microsoft Azure account then select the Web App that you wish to publish to and click “OK.”



- c. Click on the Connection tab and check that all the information is correct and filled in. Make sure you validate the connection as well.



- d. Next, click on the Settings tab and add connection strings for Active Directory and/or the database



No changes are needed in the preview tab, and at this point you will need to click publish to finish.