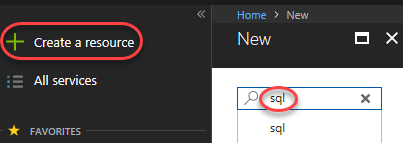
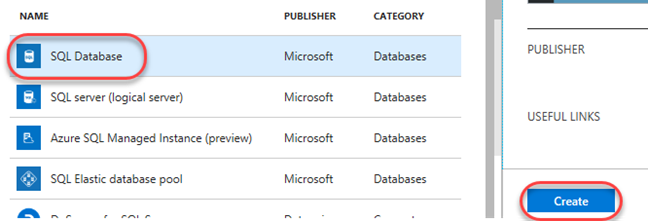
**Setting up Azure resource**

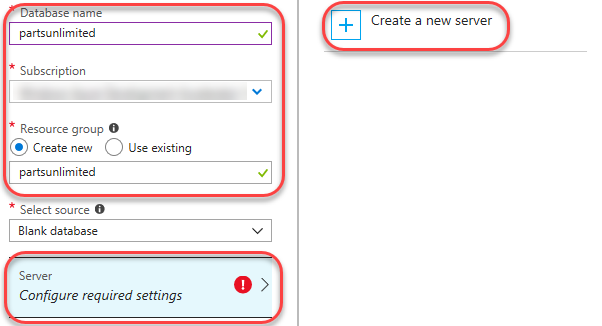
1. Start off by creating the Azure resources needed for this lab. This includes a database and two app services: one for QA and one for production. Log into your account at [https://portal.azure.com](https://portal.azure.com/). For now, we need only one app service for QA
2. Click **Create a resource** and search for **“sql”**.



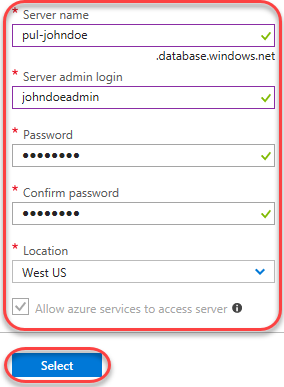
1. Select **SQL Database** and click **Create**.



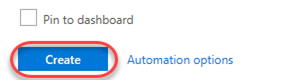
1. Enter **“fitfortis” (or any other name. In example will be used “partsunlimited”)** as the **Database name**. Select a subscription (it doesn’t matter which one, but use the same one for all steps in this lab). Select **Create new** for **Resource group** and enter **“partsunlimited”** as the name. Make sure **Select source** is set to **Blank database** and click **Configure required settings**. If you don’t already have a server you want to use, click **Create a new server**.



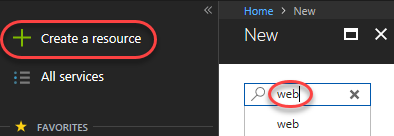
1. Enter a unique name for **Server name**, such as by including your name. Enter an admin username and password you can remember. Note that **“P2ssw0rd”** meets the password requirements. Click **Select** to select these options.



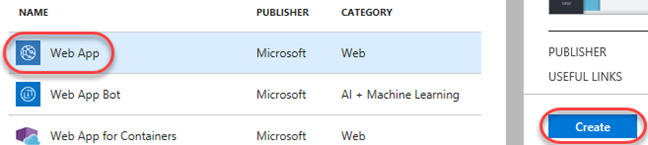
1. Click **Create**. It’ll take some time to complete, but you can move on to the next step while it works in the background.



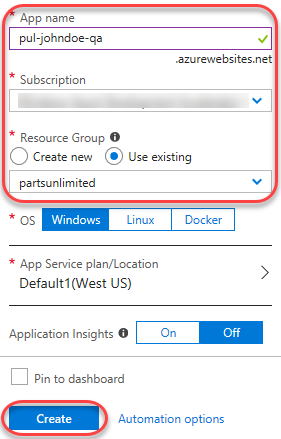
1. Click **Create a resource** and search for **“web”**.



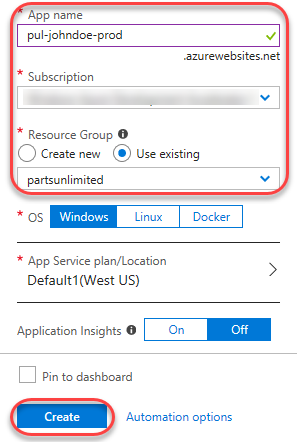
1. Select the **Web App** template and click **Create**.



1. For **App name**, enter a unique name, such as by using your name as part. Since this will be for our QA deployment, append the name with **“-qa”**. Select the same **Subscription** and **Resource group** as before. If required to create an **App Service plan**, accept the defaults. Click **Create** to create.



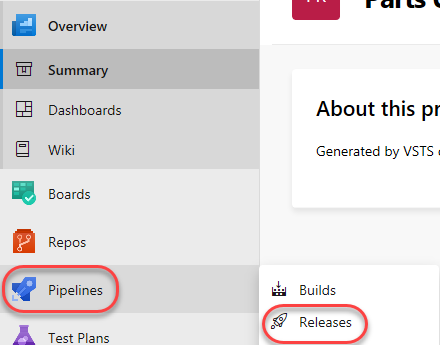
1. Repeat the process above to create a second app service for the production stage. This time, append it with **“-prod”** instead.



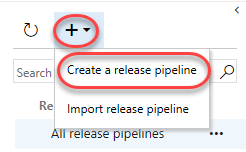
1. It may take a few minutes for all of your new Azure resources to become available, so continue on to the next task. Leave this browser tab open for later.

**Creating a configuration to the QA stage**

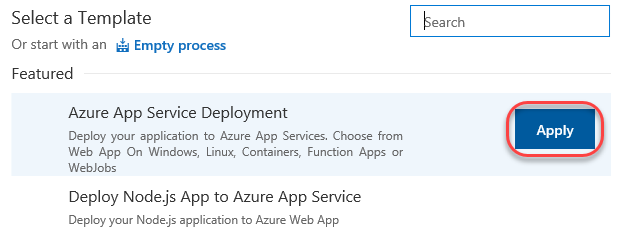
1. Navigate to your team project on Azure DevOps in a new browser tab.
2. Navigate to **Pipelines | Releases**.



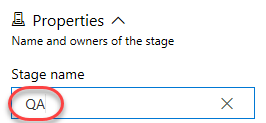
1. From the **New** dropdown, select **Create a release pipeline** to create a new release pipeline.



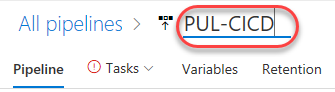
1. There are many starting templates to choose from, or you can even begin with an empty process template. In this case, select the **Azure App Service Deployment** and click **Apply**.



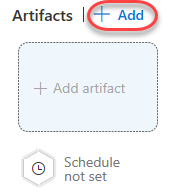
1. Rename the default stage to **“QA”**. This template will deploy to QA, and then to a production stage. We’ll set up this one first.



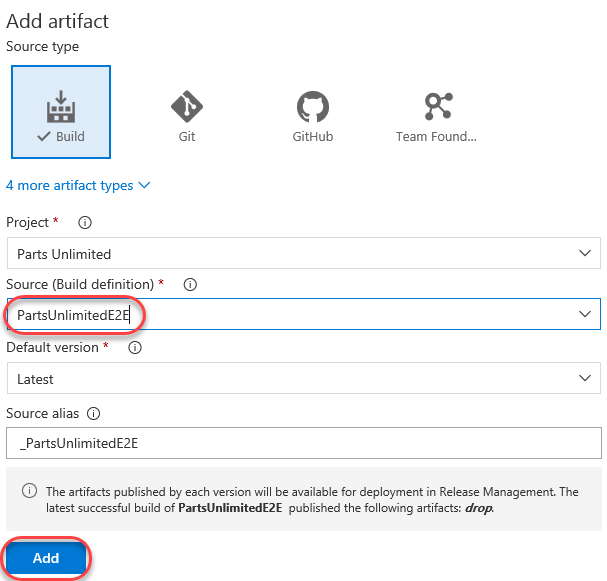
1. Rename the release pipeline to **“PUL-CICD”**.



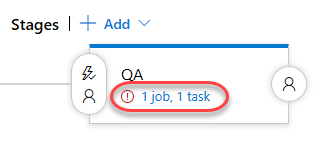
1. The first thing to define is exactly what should be deployed. Click **Add** in the **Artifacts** section to specify the artifact to deploy.



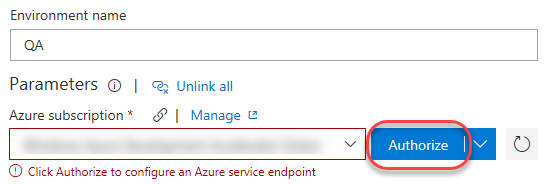
1. There are many types of artifacts, but this one will be pretty simple: a project built from the **FitFortis** build pipeline that already exists in this team project. Click **Add**.



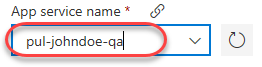
1. Now that the artifact has been defined, it’s time to configure the deployment to QA. Click **1 job, 1 task** in the **QA** stage.



1. Select the **Azure subscription** you used earlier to create the resources and click **Authorize**. If you need to create a connection to an Azure account associated with a different Microsoft account, click **New** and follow that workflow before continuing.



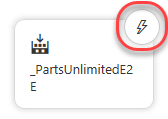
1. Follow the workflow to authorize access to your Azure account.
2. Enter the **App service name** used earlier when creating the QA app service.



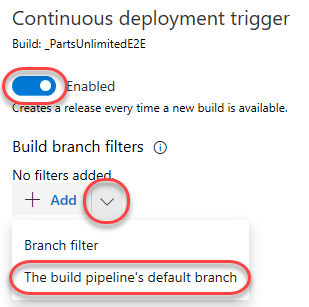
1. Return to the **Pipeline** tab.



1. Click the **Triggers** button to define what triggers will invoke this deployment.



1. **Enable** the **Continuous deployment trigger**. Add a **Build branch filter** that points at the **The build pipeline’s default branch**. This will kick off the deployment when the build completes.

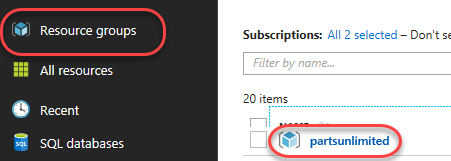


1. **Save** the release pipeline.



**Configuring the Azure app services**

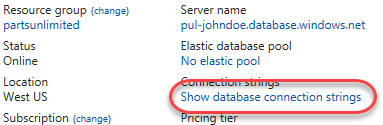
1. Return to the browser tab open to the Azure portal.
2. Click the **Resource groups** tab from the left menu. Locate and click the **yourname** group created earlier.



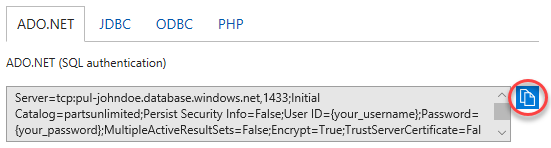
1. Click your SQL database (**pul-johndoe/partsunlimited**). Make sure you click the database you created and not the server. Note that it may take a few minutes for the database and server to become available, so click the **Refresh** button every once in a while to check in.



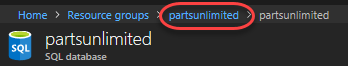
1. In the new blade, click **Show database connection strings**.



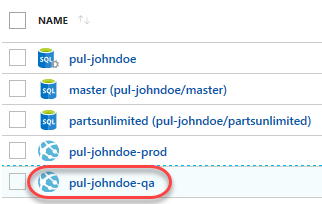
1. This will provide you with a list of connection strings based on platform. Copy the **ADO.NET** string to your clipboard so you can configure your new web site to use it. Close this blade.



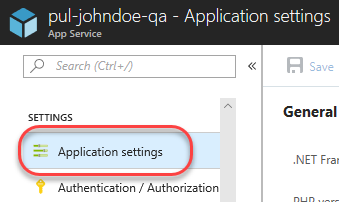
1. Open a new instance of **Notepad** and paste the connection string into it. This will make it easier to edit and retrieve later on in case anything happens to the clipboard copy.\
2. Use the breadcrumb navigation to return to the **partsunlimited** resource group.



1. Click the **QA** app service created earlier.



1. Select the **Application settings** tab from the **Settings** section.

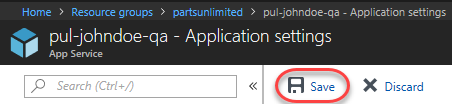


1. On this blade you can configure settings for your app, such as connection strings. Locate the **Connection strings** section and add a new entry with the key **“DefaultConnectionString”** and the value pasted from the clipboard. You’ll need to locate the “{your\_username}” and “{your\_password}” sections and replace them (including braces) with the actual SQL credentials entered earlier. Press **Enter** to complete.

Server=tcp:pul-johndoe.database.windows.net,1433;Initial Catalog=partsunlimited;Persist Security Info=False;User ID={your\_username};Password={your\_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;



1. Click **Save** from the toolbar to commit.



Original source: https://www.azuredevopslabs.com/labs/azuredevops/continuousdeployment/#task-1-setting-up-azure-resources