Lab 04

Release Management

Hands on Lab



**Conditions and Terms of Use**

**Microsoft Confidential - For Internal Use Only**

This training package is proprietary and confidential, and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the 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.

**Copyright and Trademarks**

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in 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.

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.

For more information, see Use of Microsoft Copyrighted Content at  
[*http*://www.microsoft.com/about/legal/permissions/](http://www.microsoft.com/about/legal/permissions/)

Microsoft®, Internet Explorer®, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

© 2016 Microsoft Corporation. All rights reserved.

Contents

[Introduction 4](#_Toc465695853)

[Exercise 1 – Setup your Azure account (MSDN or Personal) 4](#_Toc465695854)

Exercise 2 – Creating an Azure App Service and Web App 5

[Exercise 3 – Creating a Release Definition in VSTS 8](#_Toc465695856)

[Exercise 4 – Setting up Continuous Delivery 13](#_Toc465695857)

[Exercise 5 – Using variables 15](#_Toc465695860)

[Exercise 6 – Triggering the complete workflow (end to end scenario/review Git) 16](#_Toc465695861)

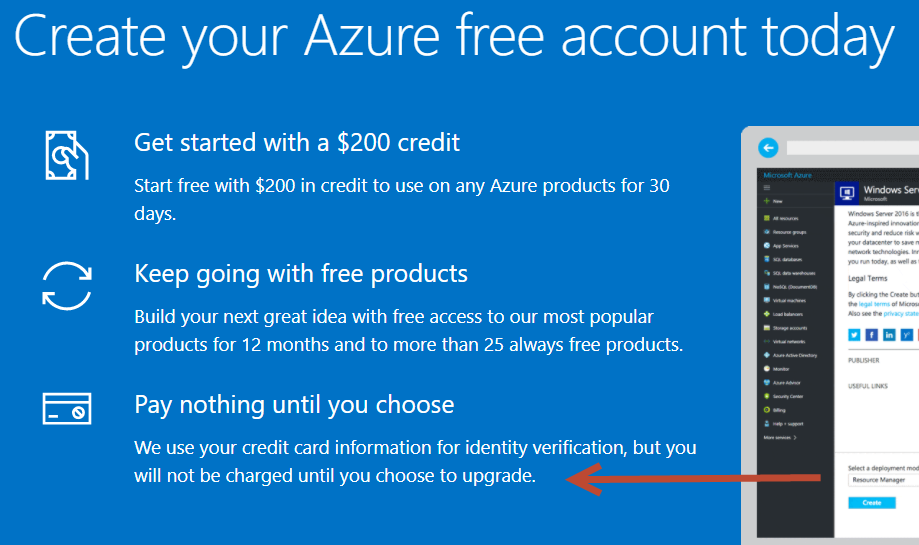
# Introduction

**Duration: 45 minutes**

Goal: Introduction to Release Management

# Exercise 1 – Setup your Azure account

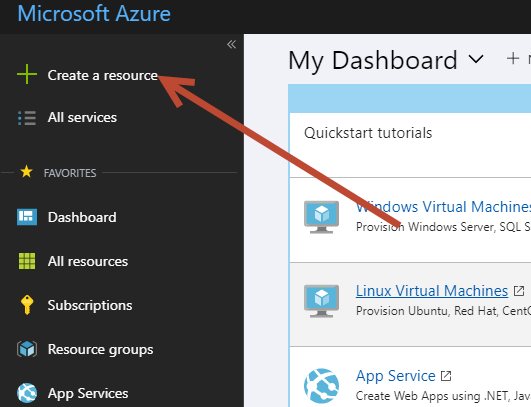
You can use either a free MSDN account or create a personal trial account. The personal trial account will last for 30 days and give you $200 in credits. It asks for a credit card, however, it will not charge you anything unless you specifically activate the account to become a paid account. If your time or credits run out, it will simply deactivate/disable your account at that time.



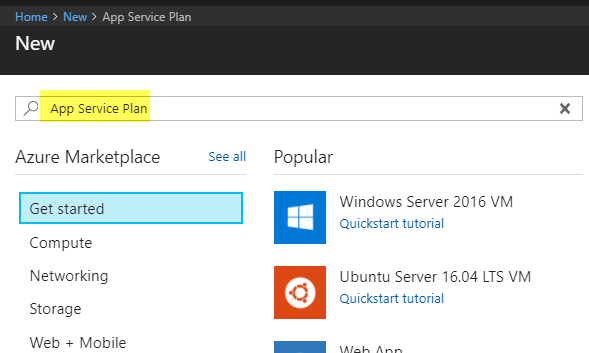
* Make sure you use the same email address that you used for your VSTS account!!!! Otherwise it will not be synced up with your Azure resources for deployment.
* If you are a MSDN (MyVisualStudio) subscriber, go to this URL to activate your monthly free Azure credit: <https://azure.microsoft.com/en-us/pricing/member-offers/credit-for-visual-studio-subscribers/>
* If you do not have MSDN, go to here: <https://azure.microsoft.com/en-us/free/>
* You will fill out a series of forms with your contact info. As mentioned above, it will ask for your credit card info but nothing will be charged unless you specifically choose to upgrade. By doing nothing with the account, nothing can accidentally get charged to you so there is nothing to worry about.
* Once you have activated you Azure account, make sure you can login and see your Azur Portal at <https://portal.azure.com/>

# Exercise 2 – Creating an Azure App Service and Web App

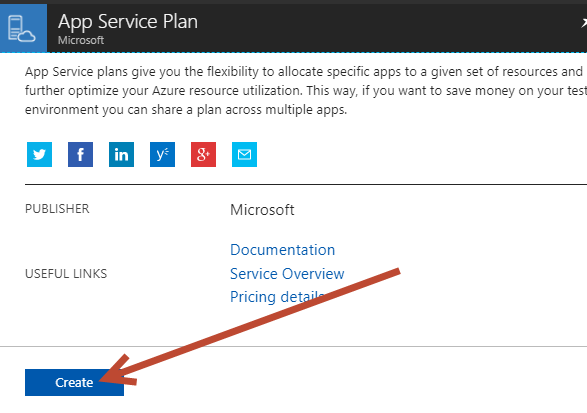
* Go to your Azure Poral
* Hit “Create a Resource” on the left menu.



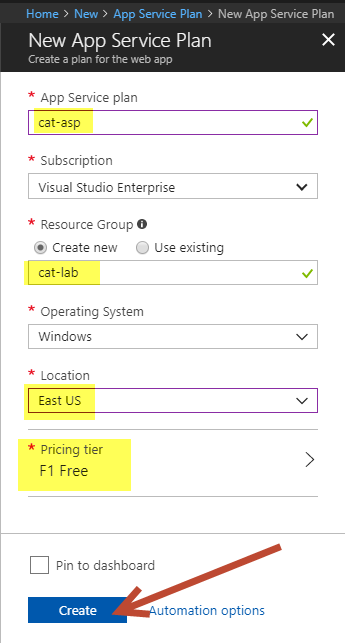
* Seach for an: App Service Plan, hit enter..



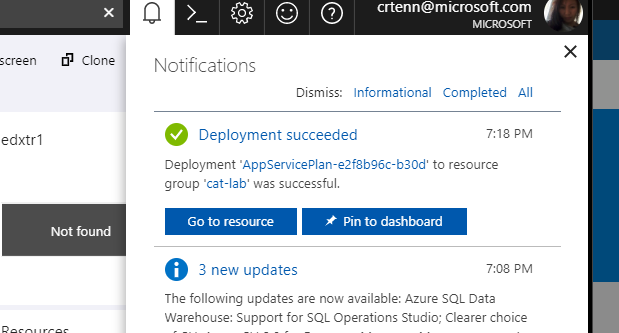
* You will land on this next page. Hit Create.



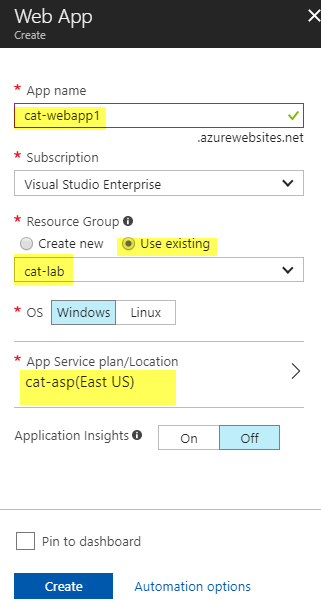
* Add a name for your plan, create a new resource group, and add the location nearest to you. Make sure you click on Pricing tier and change it from the default S1 to F1 free. Then hit Create. The App Service Plan is a holder for related web apps / API apps / function apps.



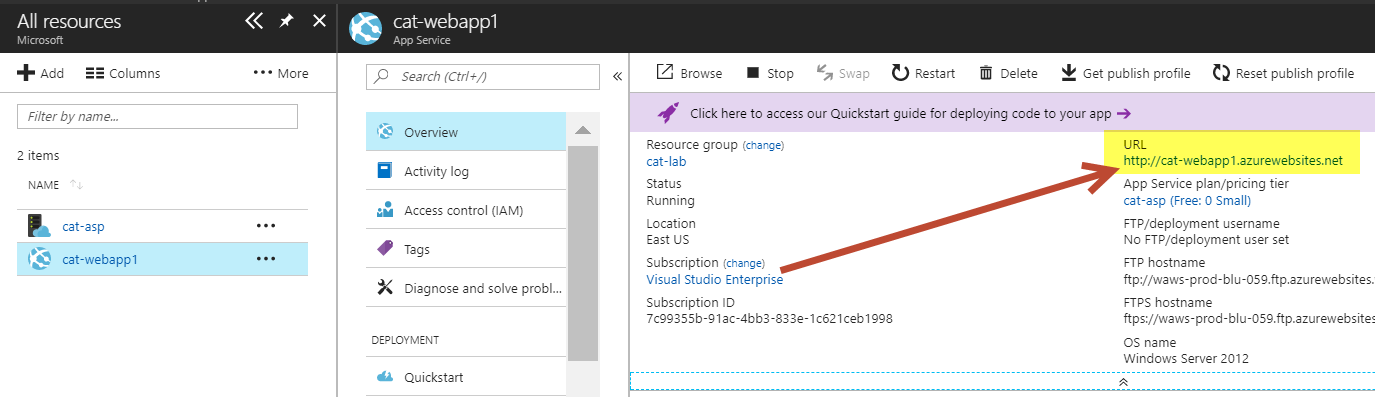
* Wait for the App Service to be created, it should take less than a minuite. You should see a notification on the top right of your screen when it is done.



* Click Create a Resource
* Search for “Web App”. Hit Create. Then fill out the app name. Make sure you use the same resource group as before and that you are attached to the App Service Plan you just created. Hit Create when you are done.

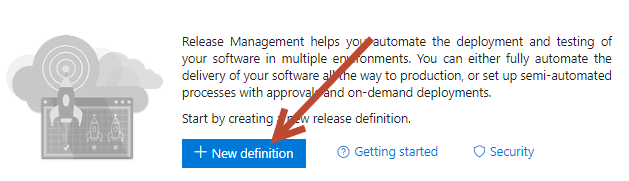


* Once it has created, click All Resources
* You should only have two resources, click on the Web App.
* Save the URL in a Notepad. It is on the Overview page as in the screenshot below. Try hitting it in your browser just to ensure there is nothing there and what it looks like empty.

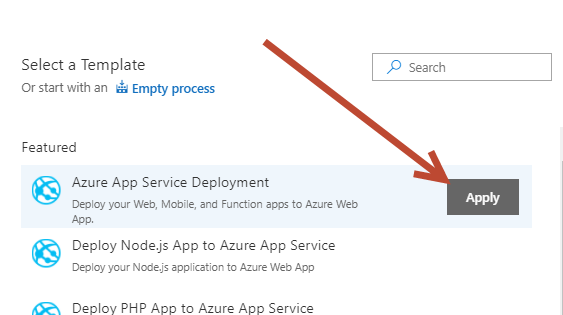


# Exercise 3 – Creating a Release Definition in VSTS

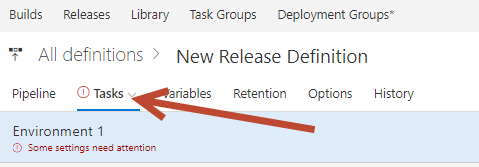
* Go to your VSTS account.
* Click on the Releases tab.
* Click to create a new Release definition.



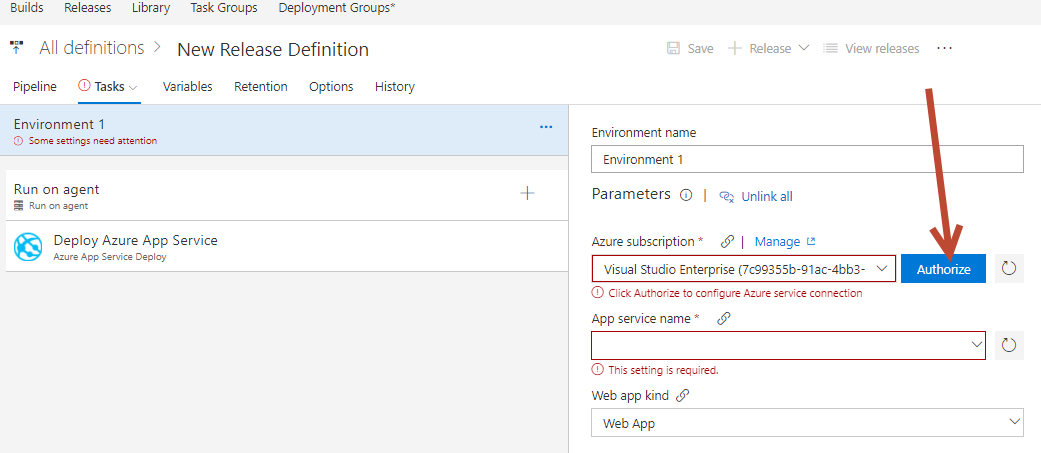
* Choose the Azure App Service template option. Hit Apply.



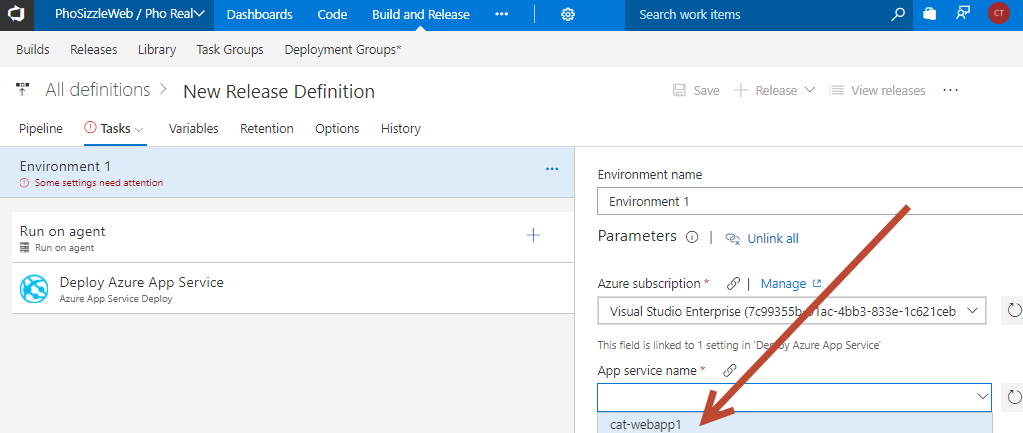
* Close the modal to the right, leave it as Environment 1.
* Click on Tasks.



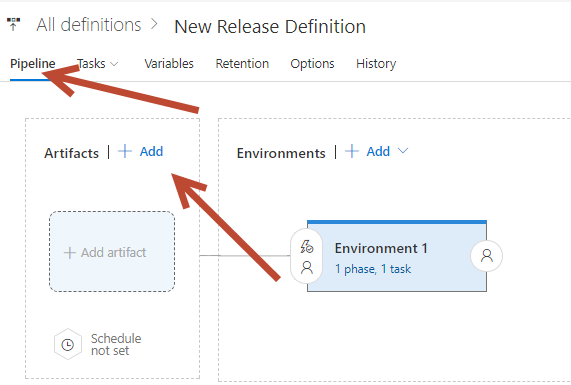
* First, choose your Azure subscription from the dropdown. Then click Authorize. \*This is why you have to use the **same** email account for you Azure account as yoru VSTS. Otherwise it will not recognize.



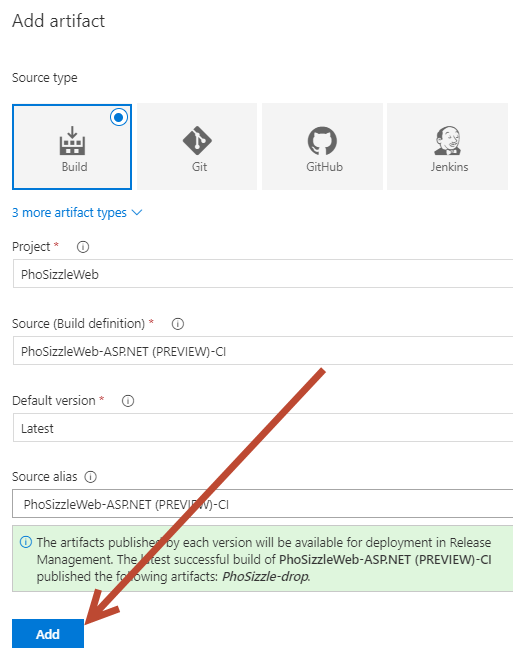
* It will take a minute to authorize. Once it does, then the dropdown for App Service should be populated. Choose your App Service.



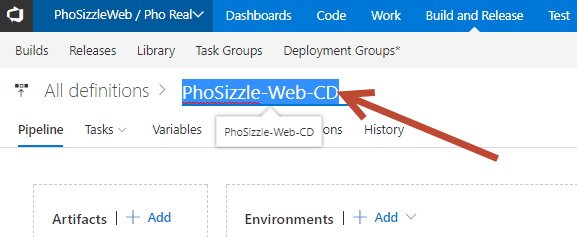
* Click Save on the Release (top right). There will be a modal popup for a comment, you can add one in then hit OK.
* Click on Pipeline. Click Add next to Artifact.



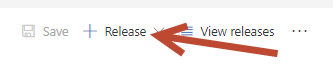
* Choose either of your Build definitions. You should have two identical ones from the previous exercise.



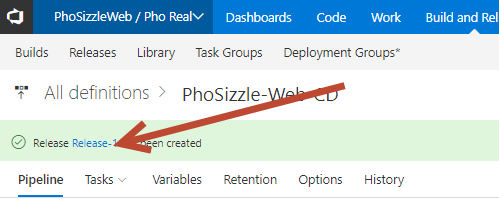
* Click on the name of the Release to change the text



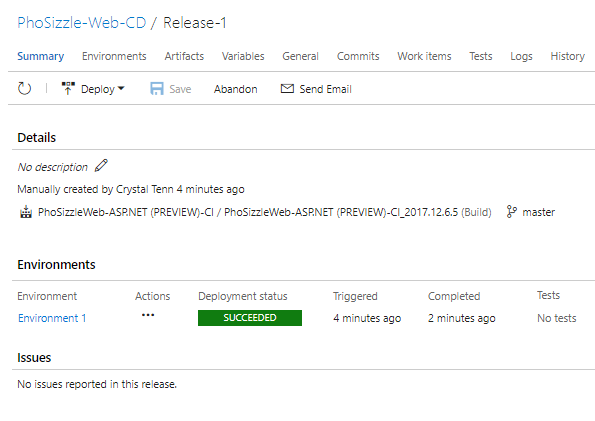
* Click Save.
* Click Release > Create a Release. On the popup, don’t change anything, just hit Create one more time.



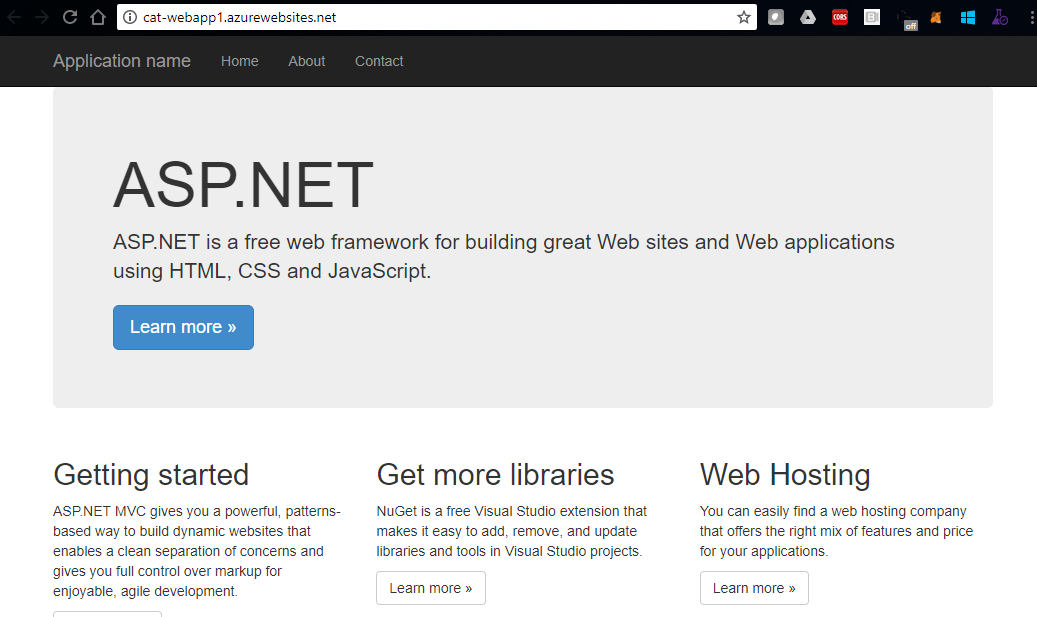
* Click on the Release note blue text.



* Once the build succeeds click on the summary:



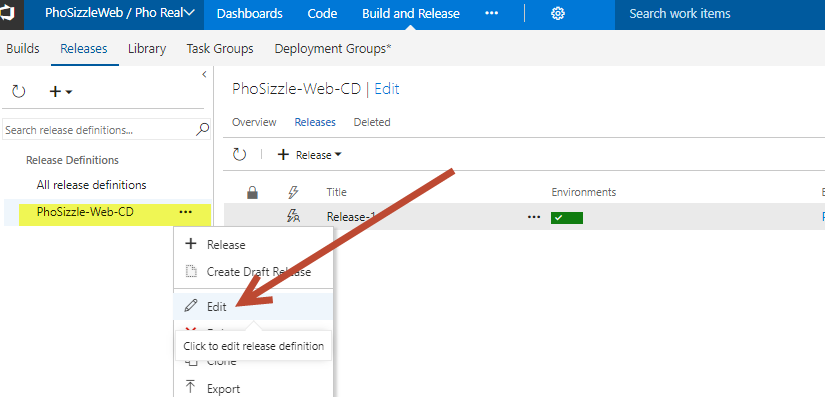
* Remember the URL you saved from you Azure Web App? Navigate to it now: You should see yoru MVC ASP.NET default page.



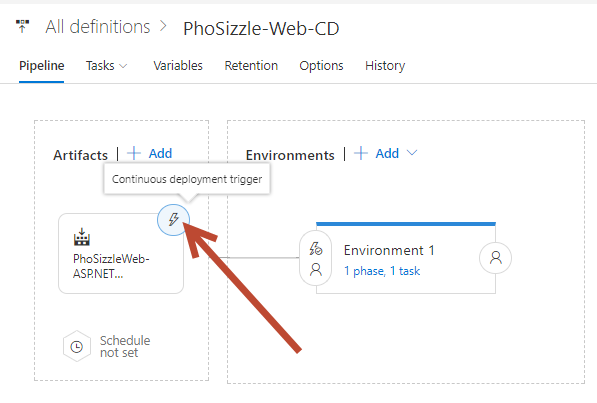
# Exercise 4 – Setting up Continuous Delivery

The build definition that was created in the previous steps is linked to our CI build which is setup for continuous integration. However, it still requires manual triggering. Let’s change that.

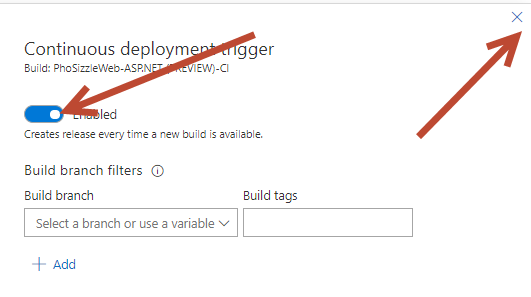
* Click on the Release Tab
* Find your PhoSizzle-Web-CD Release Definition, click on the “…” next to it and hit Edit.



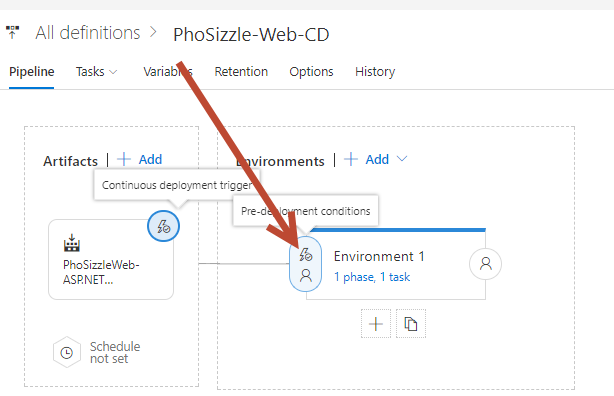
* Click on the icon with the lightning bolt on the Artifact.



* Click to Enable Continuous Deployment. This means that every time a new Build is completed successfully that the Release process will get kicked off. Hit the X on the right of the menu to close it. It will save your changes. You will not set a filter here. In the future when you have many branches you can filter on a specific branch to setup multiple environments.



* Click on the predeployment conditions for the Environment. Try toggling on different options like Schedule or Predeployment Approvers. Schedule will let you deploy at certain times/days that you choose. Predeployment approvals are to require certain users to give an OK before the Release will complete. On a trigger, it will can send out a note to approvers, and when it gets the go-ahead from them then it will complete the Release. This is also useful for if you place multiple environemnts into one Release definition, you can have it Release into multiple environments, but some/all may require different approvers. Close this for now without changing anything, this is just for your future knowledge.

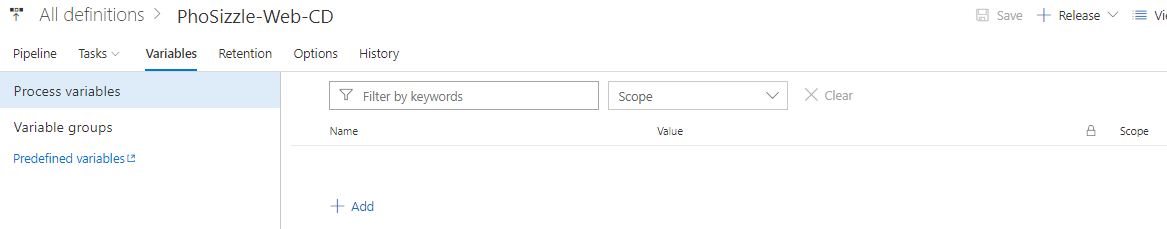


* Hit Save when you are done. We will test that the Trigger works in the final exercise when we do an end to end change.

# Exercise 5 – Using variables

Many tasks of the release definition can use a password or database connection string. Since your Azure subscription is a very simple webpage with no variables to edit, this will simply explain how to use Variables even though we will not change anything here.

* Click on Variables.

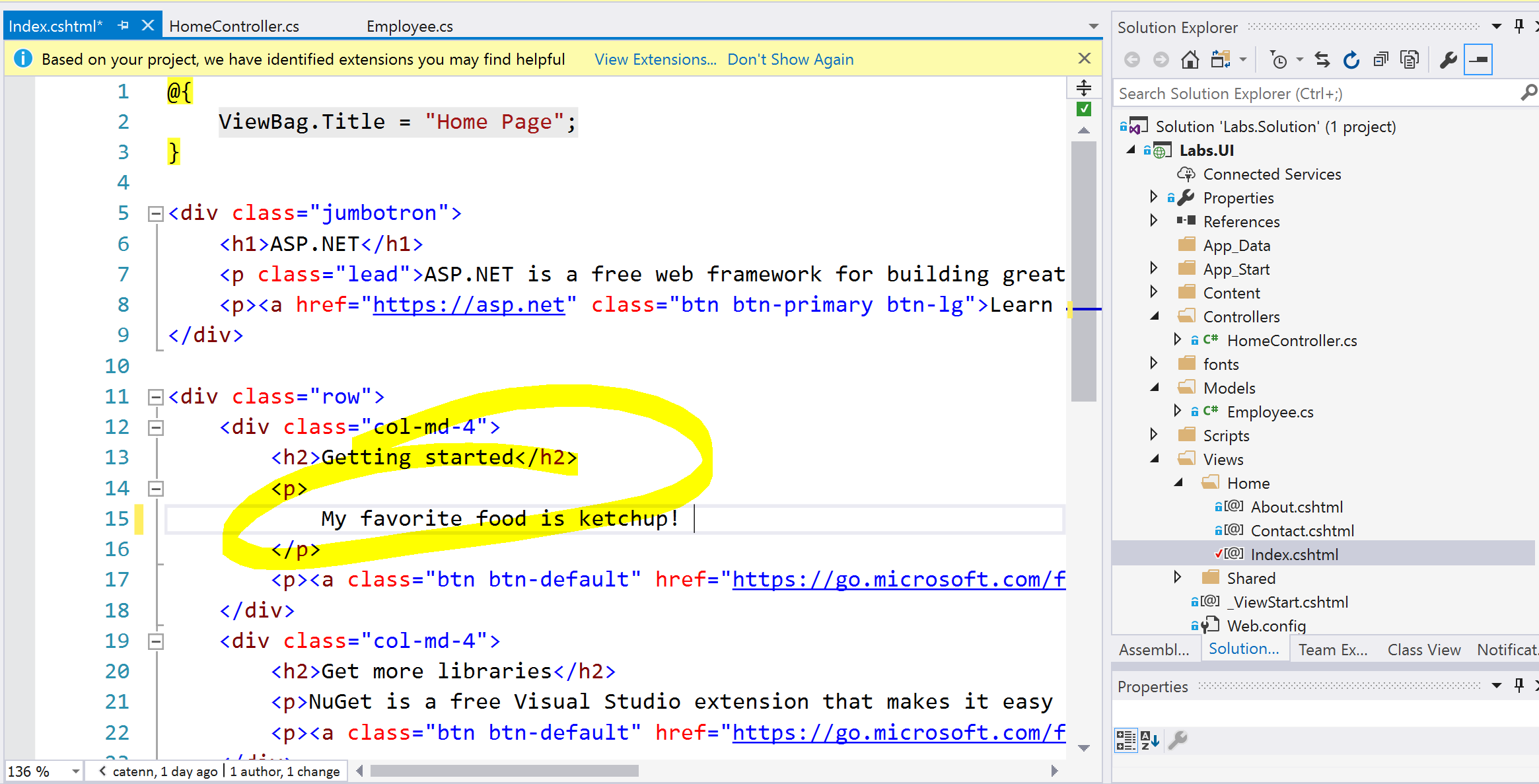


* If you use a Name/Value key pair such as Name: AdminPasword and Password: 123, then you can use the name in your Release definition anywhere needed by using this: $(AdminPassword)
* Please use this page as a reference for default variables, and variables in general:

<https://docs.microsoft.com/en-us/vsts/build-release/concepts/definitions/release/variables?tabs=batch>

# Exercise 6 – Triggering the complete workflow

* Now you should have a complete Build and Release cycle with triggers. This means that if you check in code via Visual Studio and complete a Pull Request (or make a code change in VSTS Code tab and commit it), then it should trigger a full change to the Azure website.
* Please reference Lab 1 if needed for any steps, this is a great chance to review everything you have learned so far!
* Go to Visual Studio
* Either use any exisitng feature branch, or create a new one.
* Go to the Home Index.cshtml page. Edit it to something custom.



* Commit the change
* Sync the change
* Create a Pull request from your feature branch to master branch
* Complete the Pull request which should merge into dev without any issues
* Check that a Build is kicked off
* When the Build succeeds, check that a Release is kicked off
* Ensure that the change shows up in your Azure website once the Release is complete.
* Your final website should look like this:

