# Publishing a Web App to Azure

## Objectives:

- Publish a simple web page
- Set up Continuous Deployment
- Use modern source control tools

## Introduction:

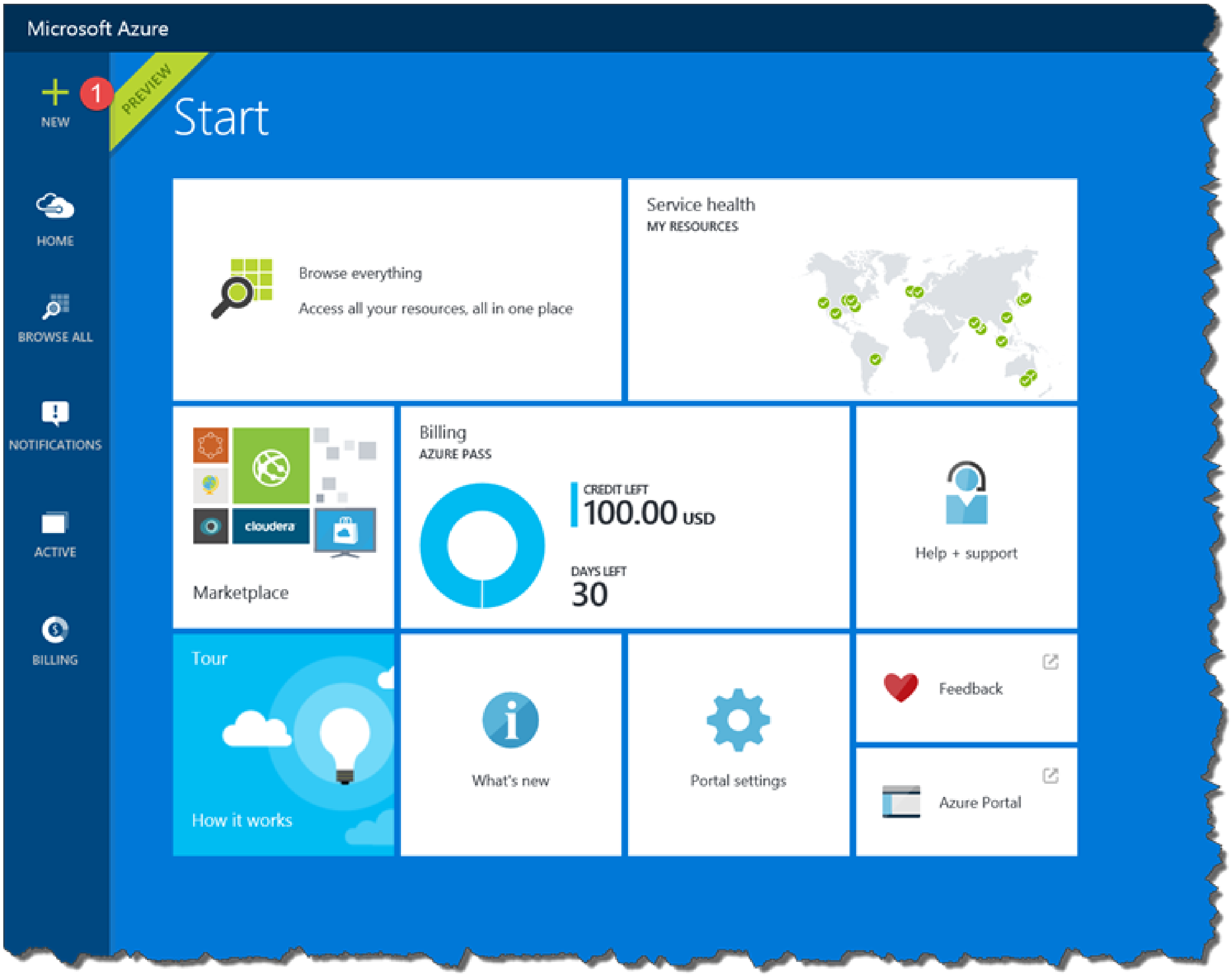
In this module you will learn how to publish a website to Microsoft Azure. We will start by creating a Web App on Azure to host your website for all the world to see. Then we'll set up a place for your source code and watch as Azure publishes your code to your website.

## Create an Azure Web App

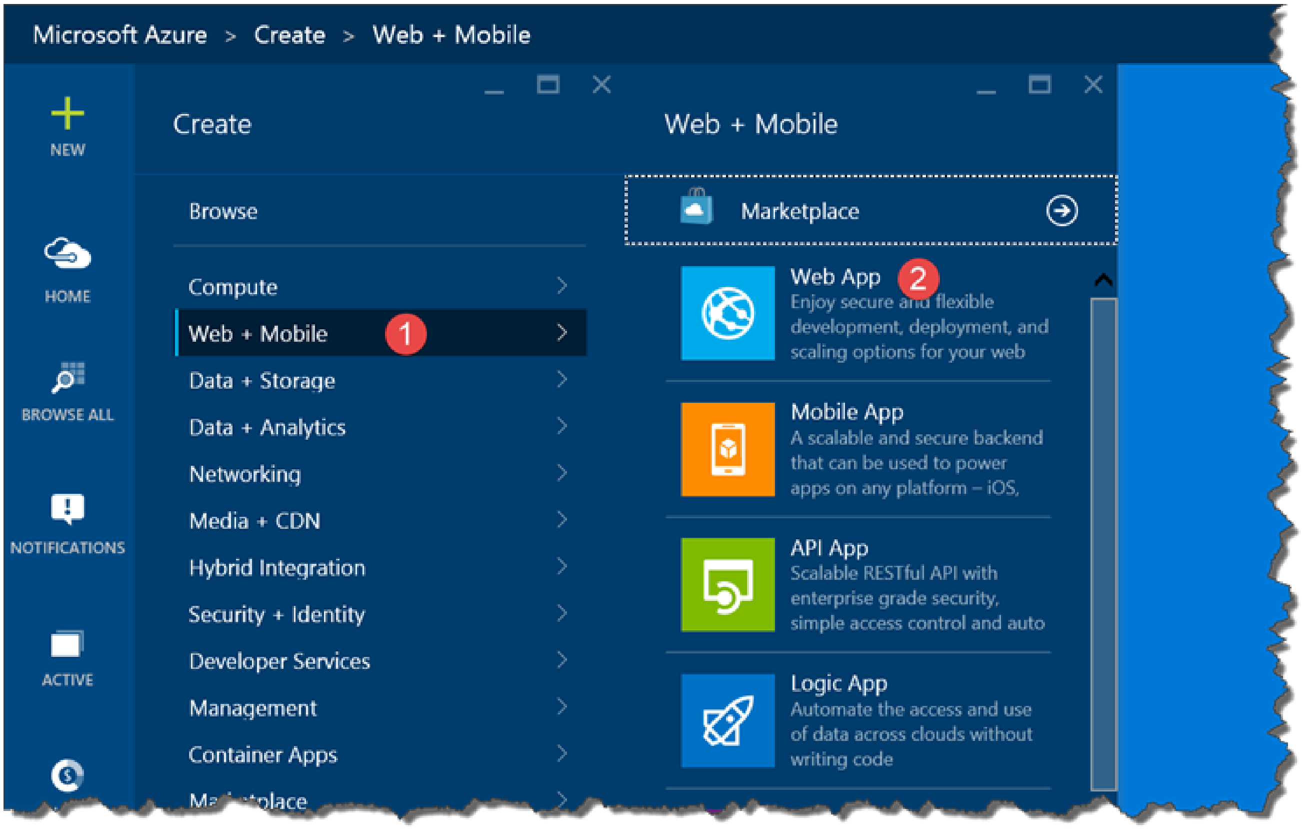
First, let's get started with creating a place to put your new website. At this point, you need to have an active Azure subscription and a machine set up with Visual Studio Code and Git.

We are going to start at the Azure portal. With your browser, navigate to http://portal.azure.com and sign in if you need to. Go ahead and do that now and you should arrive at the portal home page.

In the upper left corner, click the plus sign or "New" button (1) to create a new Azure resource.



In the Create blade that comes up, click the "Web + Mobile" (1) option and then select "Web App" (2) to begin creating a Web App resource for your website.



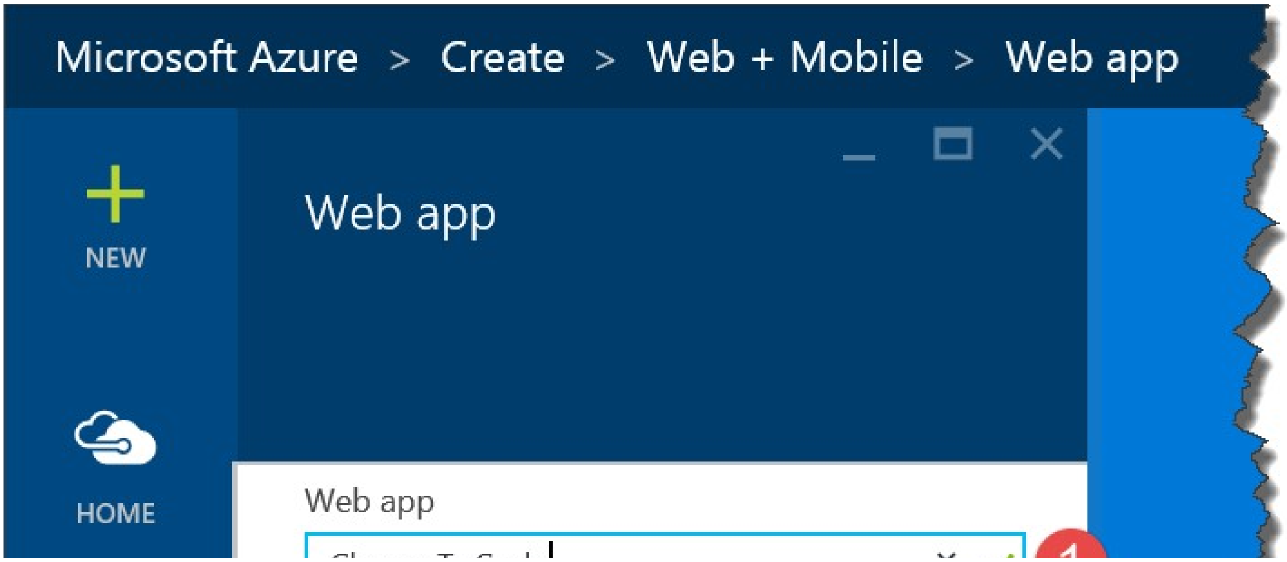
Enjoy secure and flexible development, deployment, and scaling options for your web

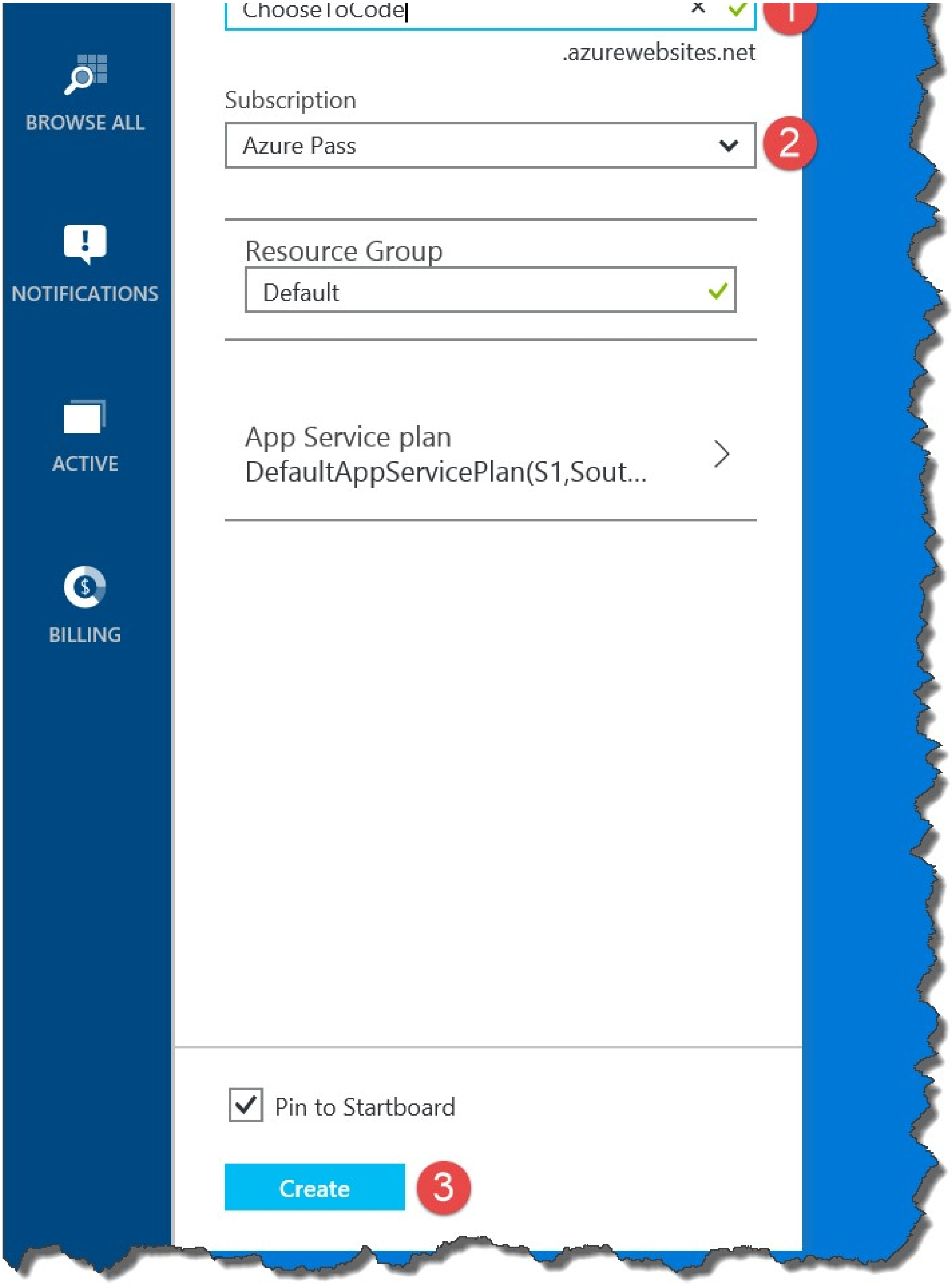
A scalable and secure backend that can be used to power apps on any platform – iOS,

Scalable RESTful API with enterprise grade security, simple access control and auto

Automate the access and use of data across clouds without writing code

Next, in the "Web App" blade that comes up, we will enter a little bit of information about your website. First, you need a name for your website. The name will be created at ".azurewebsites.net". The name will need to be unique and you should get a green checkmark (1) when you have a valid web app name. Next, you will choose the subscription to be used for your Azure web app (2). The "App Service Plan" allows you to change what region your application is hosted in and the pricing tier for your website. You can change this if you want, but you can also change this later. For now, we will skip that. Finally, leave the checkbox "Pin to Startboard" checked and click "Create" (3) at the bottom of the blade to begin creating your new Azure Web App.





ChooseToCode .azurewebsites.net

Subscription: Azure Pass

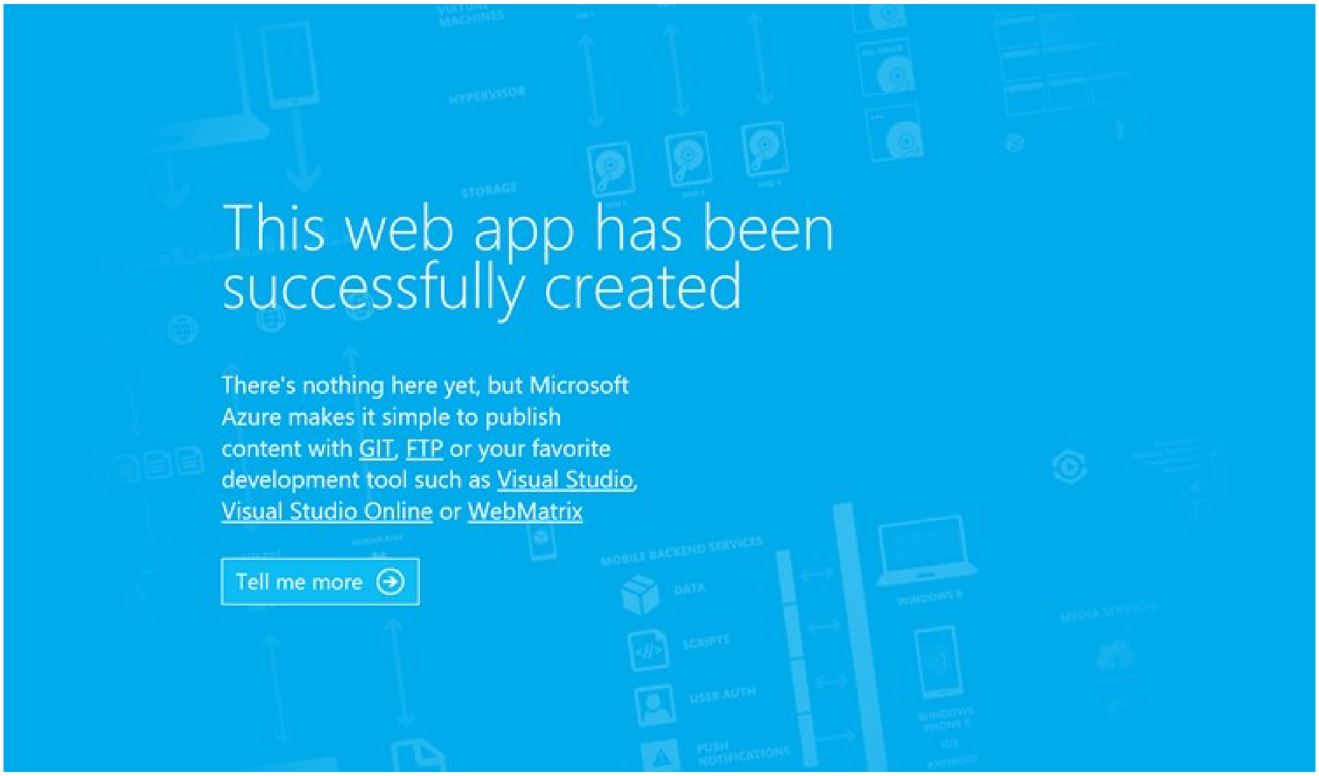
Resource Group: Default

App Service plan: DefaultAppServicePlan(S1,Sout...

☑ Pin to Startboard

You will be taken back to the portal home page and a new tile will show that the web app creation is in process. Once your website is up and running, you can navigate to your Web App management blade by clicking on the tile on the home page. Once you are on the Web App management blade, you can click on the link to go to your website or you can just navigate to the name of your website. For example [http://ChooseToCodeTutorial.azurewebsites.net](http://ChooseToCodeTutorial.azurewebsites.net).

Since you haven't created any content yet, you should see a page that shows your web app was successfully created.



This web app has been successfully created

There's nothing here yet, but Microsoft Azure makes it simple to publish content with GIT, FTP or your favorite development tool such as Visual Studio, Visual Studio Online or WebMatrix

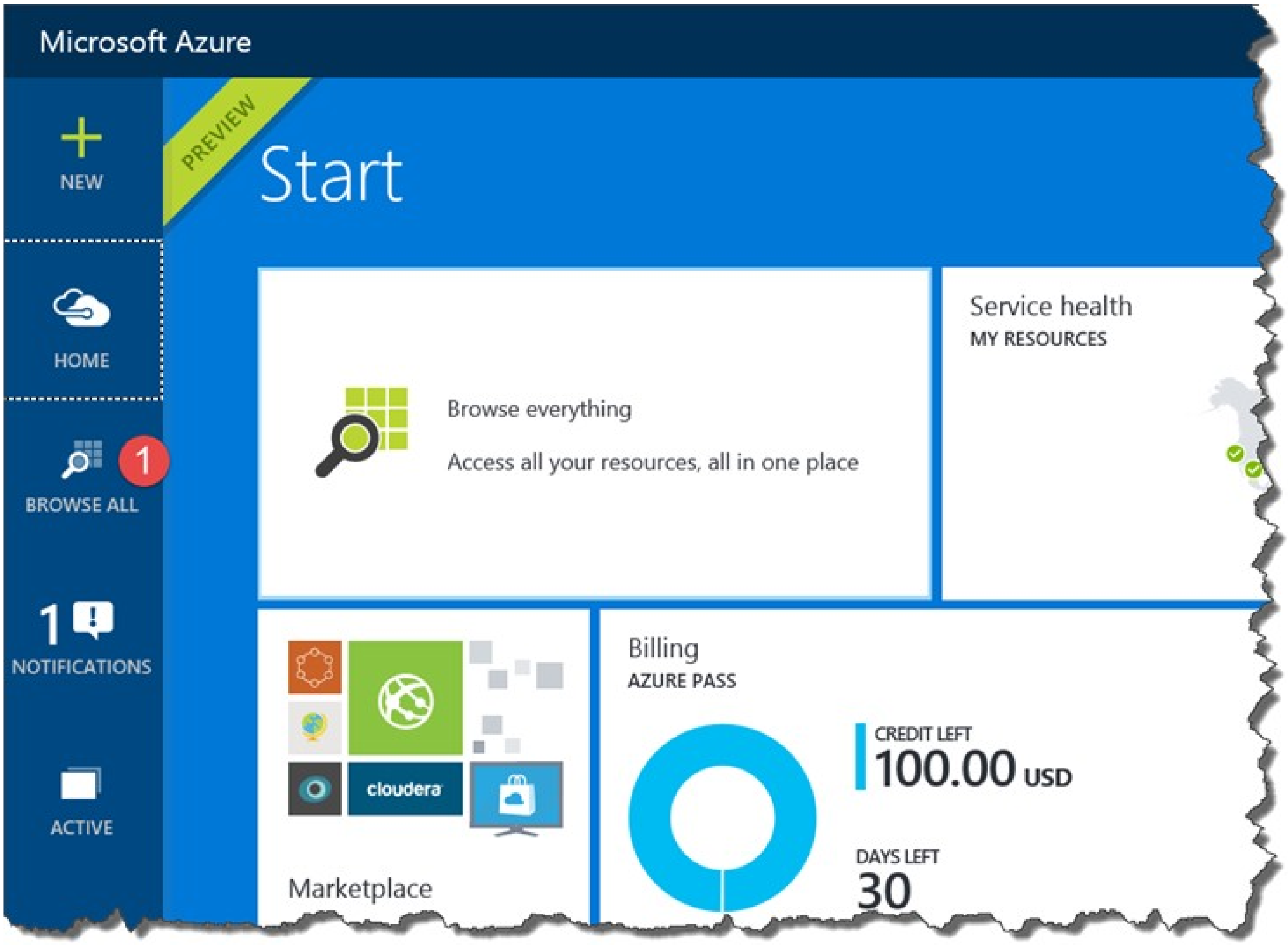
There are a number of tools you can use to start publishing content to your new web application, but we are going to use a modern tool that includes version control as well.

## Publish your first web page

Now that you have a web app created for your content, we need to set you up with a way to publish your code. To do this, we are going to take advantage of a version control system built into Azure called "git".

It is important to keep track of the changes that you are making as you work on your projects. Version Control Systems keep track of those changes and allow you to experiment with your code without losing track of past work. You can always "undo" your work if you need to. This course won't cover version control, but we will use it in order to move your code into your Azure Web App.

Let's go back to the Azure Portal and take a look at the Web App blade for your website. Remember you can get to the management blade by clicking on the tile on the Portal home page. Additionally, you can also click on "Browse All" in the left navigation bar.



Browse everything

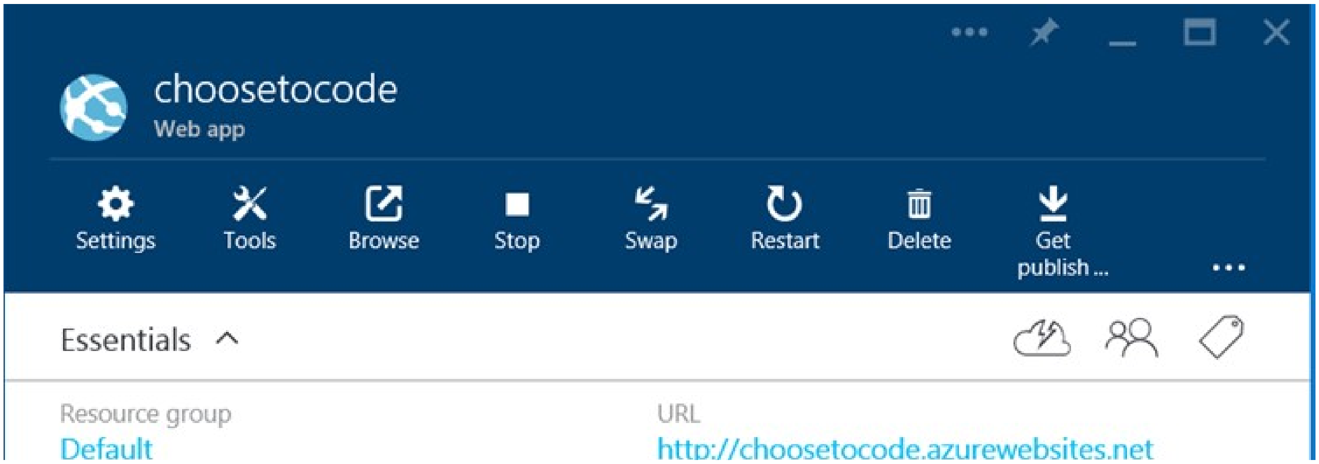
Access all your resources, all in one place

CREDIT LEFT
100.00 USD

DAYS LEFT
30

From here, you can scroll down to Web Apps and then select your website in the blade that comes up.

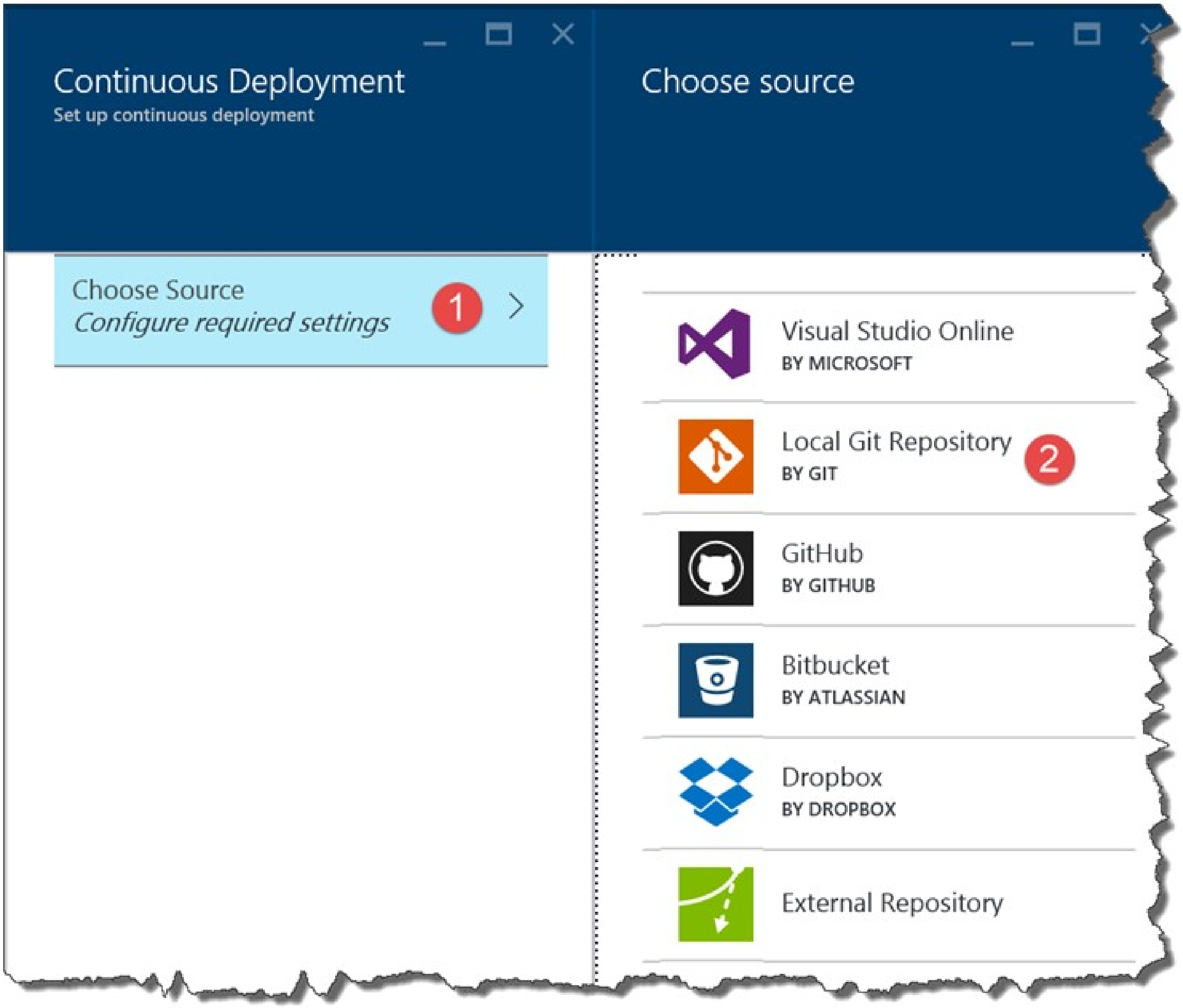
Once you have the Web App management blade up, scroll down to "Deployment". Click the "Set up continuous deployment" to begin setting up your source control.



Resource group
Default

URL
http://choosetocode.azurewebsites.net

In the blade that comes up for "Continuous Deployment", select "Choose Source" (1) and then "Local Git Repository" (2).



Continuous Deployment
Set up continuous deployment

Choose Source
*Configure required settings* 1

Choose source

Visual Studio Online
BY MICROSOFT

Local Git Repository 2
BY GIT

GitHub
BY GITHUB

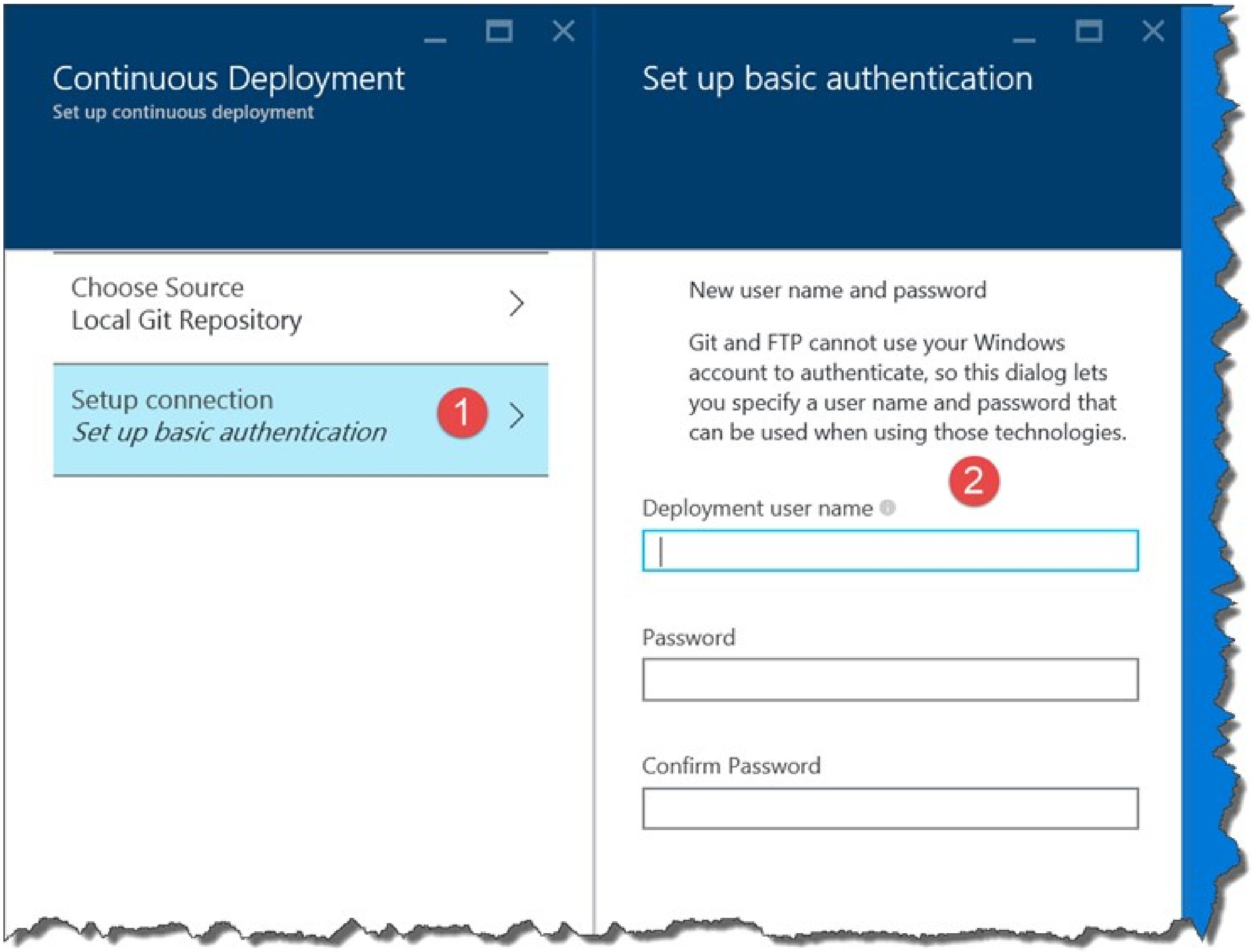
Bitbucket
BY ATLASSIAN

Dropbox
BY DROPBOX

External Repository

When the "Choose Source" closes, you will see a new option for "Setup Connection" (1). Click this option to set up basic authentication for deploying your code. You will need to choose a username (2) and a password.

NOTE: The full deployment username will be the name of your website and the username you entered joined with a "\". For example: "choosetocode\username".



Continuous Deployment

Set up continuous deployment

Choose Source
Local Git Repository

Setup connection
*Set up basic authentication*

Set up basic authentication

New user name and password

Git and FTP cannot use your Windows account to authenticate, so this dialog lets you specify a user name and password that can be used when using those technologies.

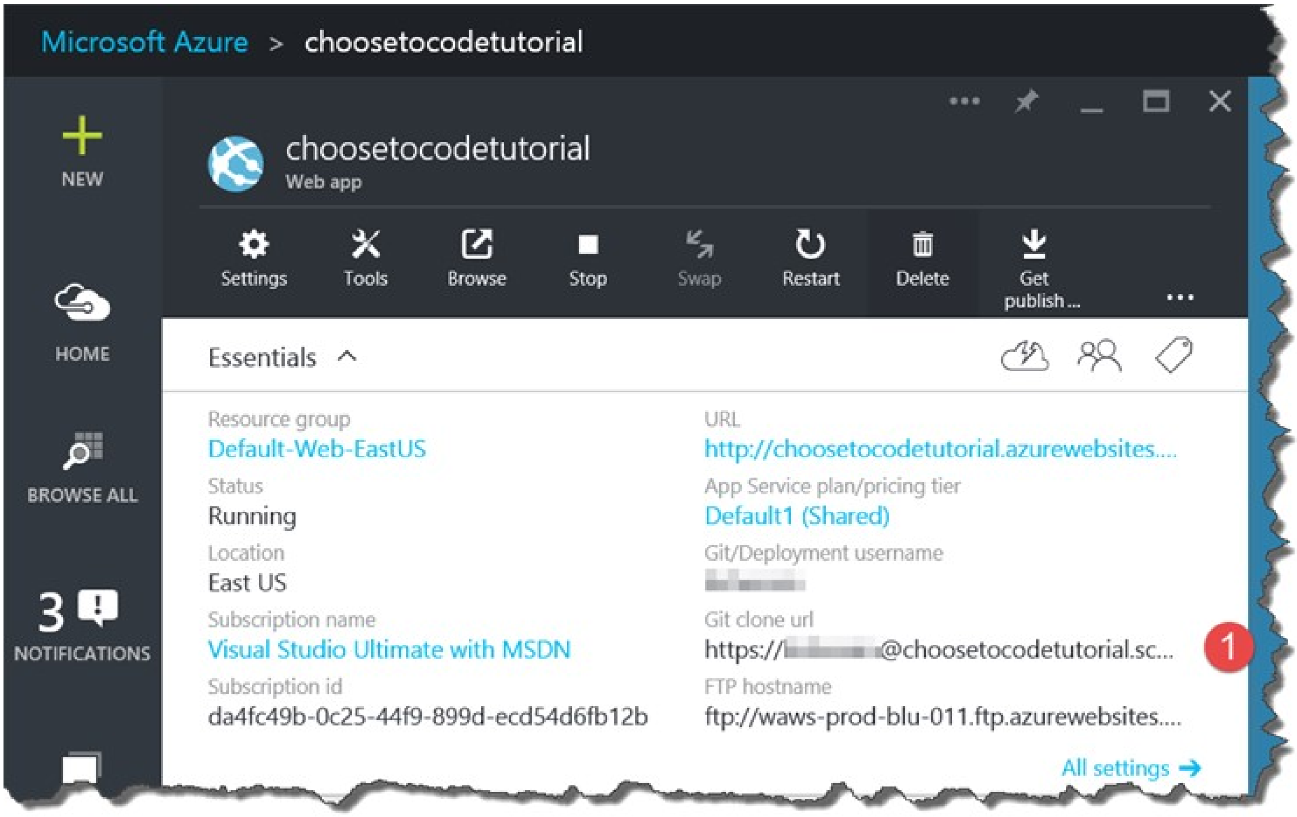
Deployment user name

Password

Confirm Password

So now that you have Azure set up for your source code it's time get some code in there to get published.

The next thing we need to get from the Azure portal is your "git clone url". This is located on your Web App management blade. Copy this URL to your clipboard and we will use it in a minute.



Resource group
Default-Web-EastUS

Status
Running

Location
East US

Subscription name
Visual Studio Ultimate with MSDN

Subscription id
da4fc49b-0c25-44f9-899d-ecd54d6fb12b

URL
http://choosetocodetutorial.azurewebsites....

App Service plan/pricing tier
Default1 (Shared)

Git/Deployment username
[redacted]

Git clone url
https://[redacted]@choosetocodetutorial.sc...

FTP hostname
ftp://waws-prod-blu-011.ftp.azurewebsites....

Now we have everything we need to connect up to Azure.

## Connecting your code to Azure

At this point, we need to get your local code folder ready to push code to Azure. Start up a "Git Shell" and navigate to the folder where you have stored your code.

If you are using a Mac, at your shell prompt you can type

```
cd /path/to/code/directory
```

On Windows, you can use:

```
cd c:/path/to/code/directory
```

Once we get to the directory where you have the basic website we have been working on, we will need to create a local git repository on your machine. We create that repository with the following command:

```
git init
```

This will create an invisible ".git" folder along side your project that will keep track of all of the changes to your code.

You may need to do one more bit of Git configuration if you haven't done this before. We need to give Git some basic user configuration. We'll set this up with some global settings so you only have to do this once. Run the following commands so that git knows your name and email. Be sure to change the commands to reflect the username you specified in Azure when setting up your automated deployment

```
git config --global user.name "YourAzureWebUsername"
git config --global user.email "your_email@whatever.com"
```

At this point, we need to add all of the code that you have created so far to a check in list.

```
git add .
```

The "." is a wildcard and means "add all files". Next, we need to commit all of the files we just added to the check in list.

```
git commit -m "Initial commit"
```

The "-m" means add message and then in quotes we put the message to put on the commit as the files are checked in.

Alright, at this point, we have a git repository in Azure (that is currently empty) and we have a git repository on our local machine. We now need to sync the changes up to azure.

First, we need to tell our local git repository that we want to connect it to the Azure repository. To do this, we add a remote connection. Remember that "git clone url" we copied a few minutes ago? This is where we need to use it. Replace the https:// below with your "git clone url".

```
git remote add azure -f https://username@website.scm.blah.blah
```

This command will add a remote repository and fetch any existing items from it to syncronize our repositories. I use the name "azure" to remind me that I am pushing the code to Azure. Sometimes I will have multiple remote repositories, and they each can be named something different. In the documentation, you will often see this as "origin".

Finally, lets push our code from our local machine to Azure. When we do this, Azure will automatically kick off a deploy and put our code into our website.

```
git push azure master
```

Now, if you navigate to your web site, instead of that blue page, you should see your web site up and running.

Congratulations, you just set up a continuously deployed web site in the cloud using modern source control tools.