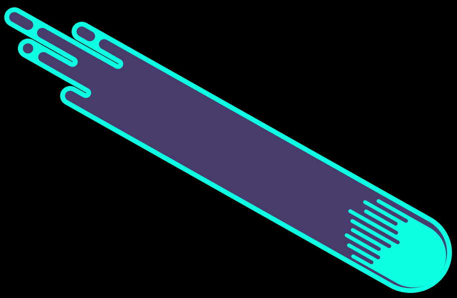
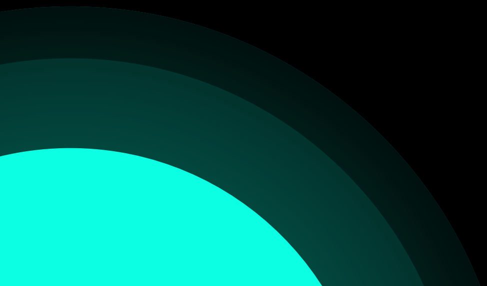
Microsoft Azure

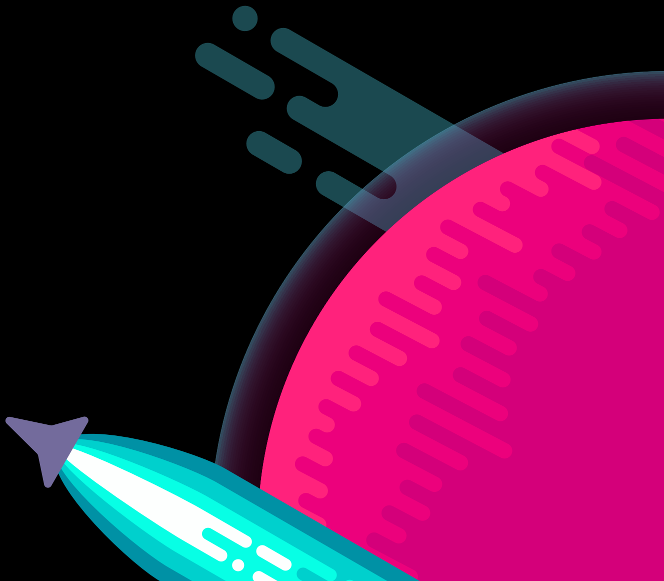


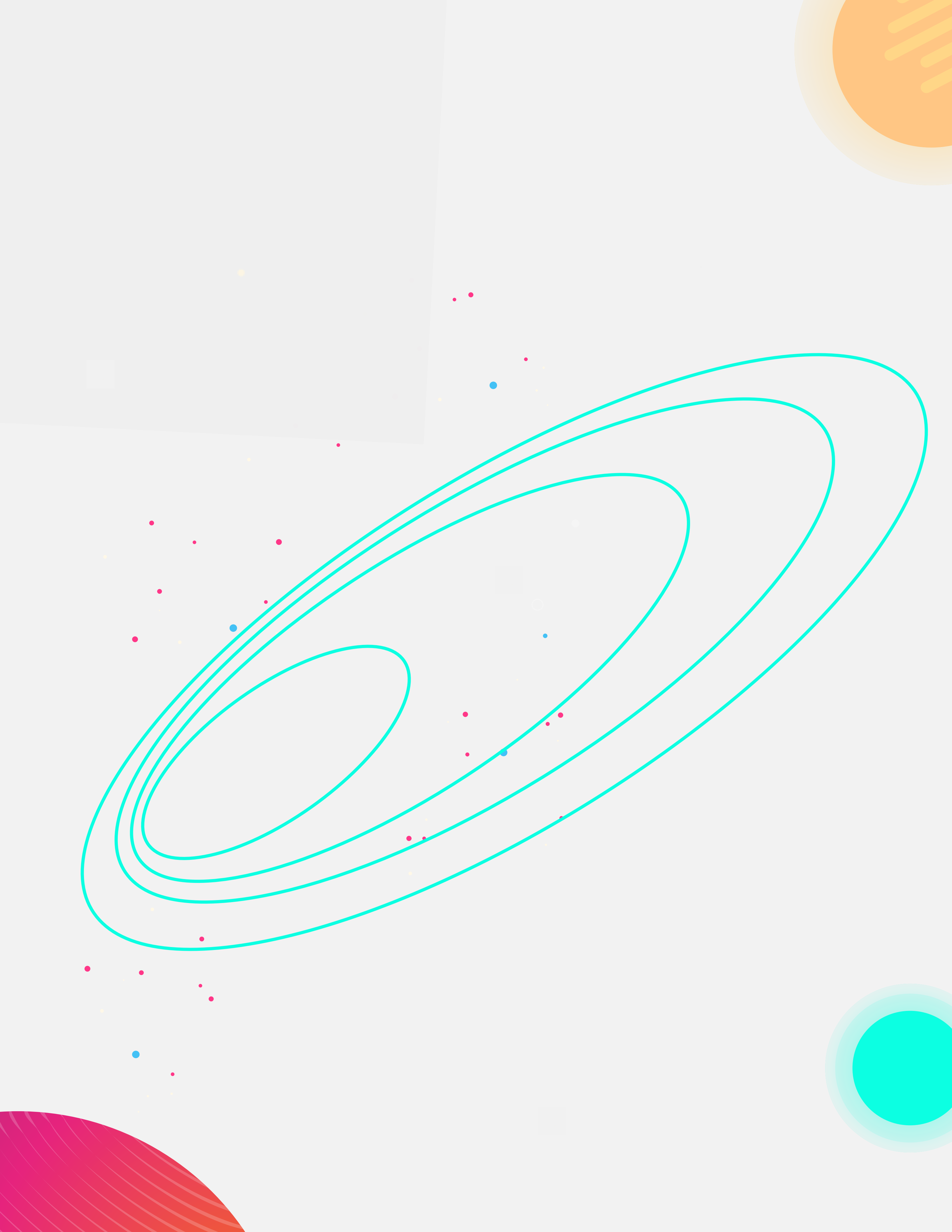
Azure Tips and Tricks: Web

azuredev.tips



ISBN 978-1-7327041-2-1





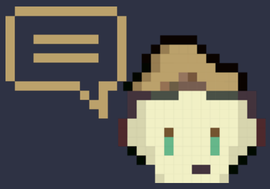
Introduction

Hi, folks!

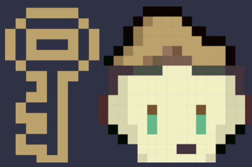
When I reflect back on Azure Tips and Tricks a year ago, I was only thinking that I'd write a couple of posts and move on. Fast-forward to today, the collection has grown to over 150+ tips, as well as videos, conference talks, and now an e-book spanning the entire universe of the Azure platform. What you are currently reading is a special collection of tips based on page views of the entire series over the last year. Before we dive in, you'll notice my pixelated form as you turn each page.

These represent:

Something I found interesting and you may too.



Additional resources to get the most out of this tip.



A key takeaway from the tip.

You can stay up to date with the latest Azure Tips and Tricks at:

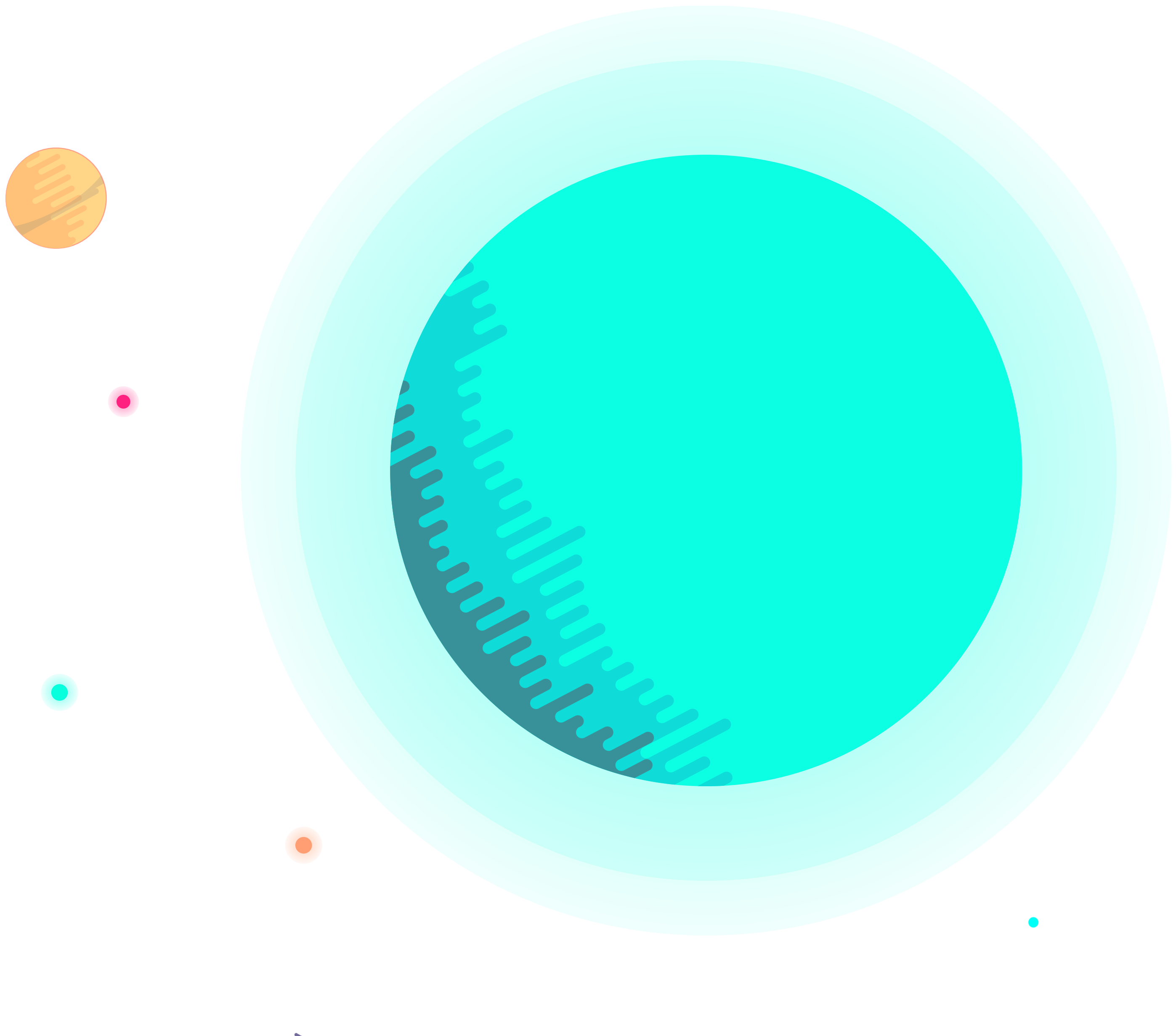
- Blog - azuredev.tips
- Videos - videos.azuredev.tips
- eBook - ebook.azuredev.tips
- Survey - survey.azuredev.tips

I hope you enjoy reading the eBook as much as I did writing it.

Thanks,
Michael Crump (@mbcrump)

WEB

If you've used Azure, you've more than likely used Azure App Service to easily host web applications, REST APIs, and mobile back ends. In this e-book, I've pulled out the **top 6 tips** since the creation of Azure Tips and Tricks for Azure App Service. They include easily working with files in the console, easily setting up staging environments and swapping between them, and routing traffic to different versions of your app to "Test in Production". I'll also cover how you can implement performance testing, best practices for App Settings in Azure App Service, and cloning a web app that is especially helpful if you have customers all over the world.



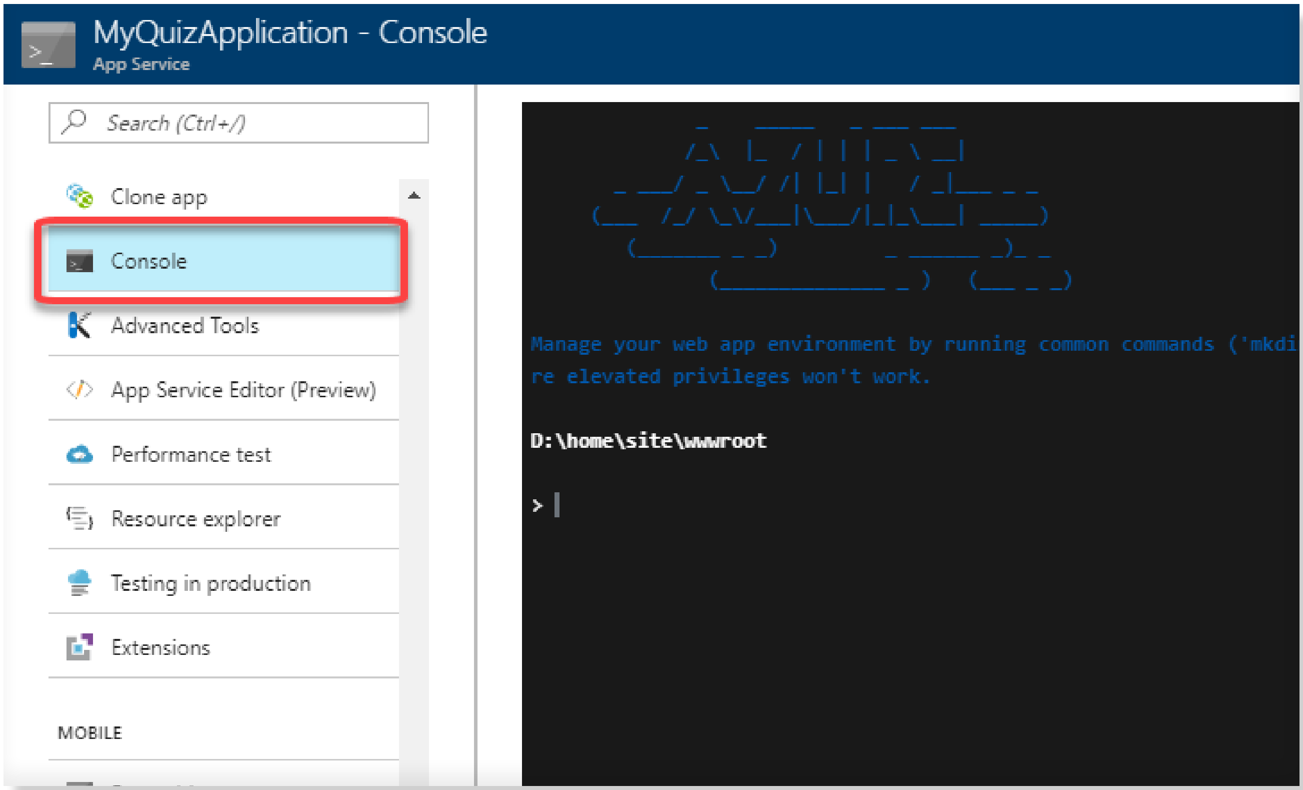
You can learn more about Azure App Service here

Working with Files in Azure App Service

We'll take a look at the files inside an Azure App Service web site and how you can easily work with them.

Console Access to my App Service

Go to the Azure Portal and select my App Service. Click on **Console** under **Development Tools** to have a command prompt to quickly work with my Azure App Service.



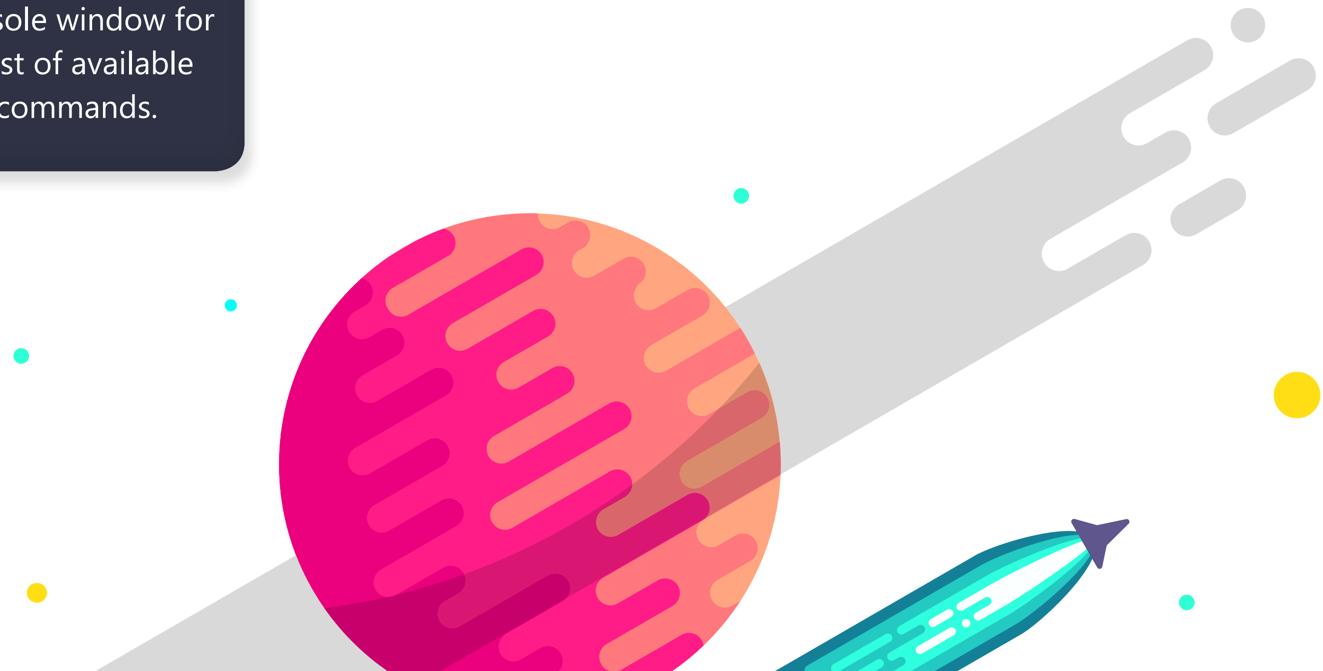
```
Manage your web app environment by running common commands ('mkdi
re elevated privileges won't work.

D:\home\site\wwwroot

>
```

As you can tell from the screenshot, I start in D:\home\site\wwwroot. I can type dir to see a current directory listing.

```
Volume in drive D is Windows
 Volume Serial Number is FE33-4717

 Directory of D:\home\site\wwwroot

09/21/2017  08:35 PM    <DIR>          .
09/21/2017  08:35 PM    <DIR>          ..
09/20/2017  09:03 PM    <DIR>          css
09/20/2017  09:03 PM             5,351 Default.html
09/20/2017  09:03 PM    <DIR>          js
09/20/2017  09:03 PM             1,950 jsQuizEngine.sln
09/20/2017  09:03 PM               304 jsQuizEngine.userprefs
09/20/2017  09:03 PM            31,744 jsQuizEngine.v12.suo
09/20/2017  09:03 PM    <DIR>          PrecompiledWeb
09/20/2017  09:03 PM    <DIR>          quiz
               4 File(s)         39,349 bytes
               7 Dir(s)   1,072,893,952 bytes free
```

Quick Tip You can type **help** from the console window for a list of available commands.

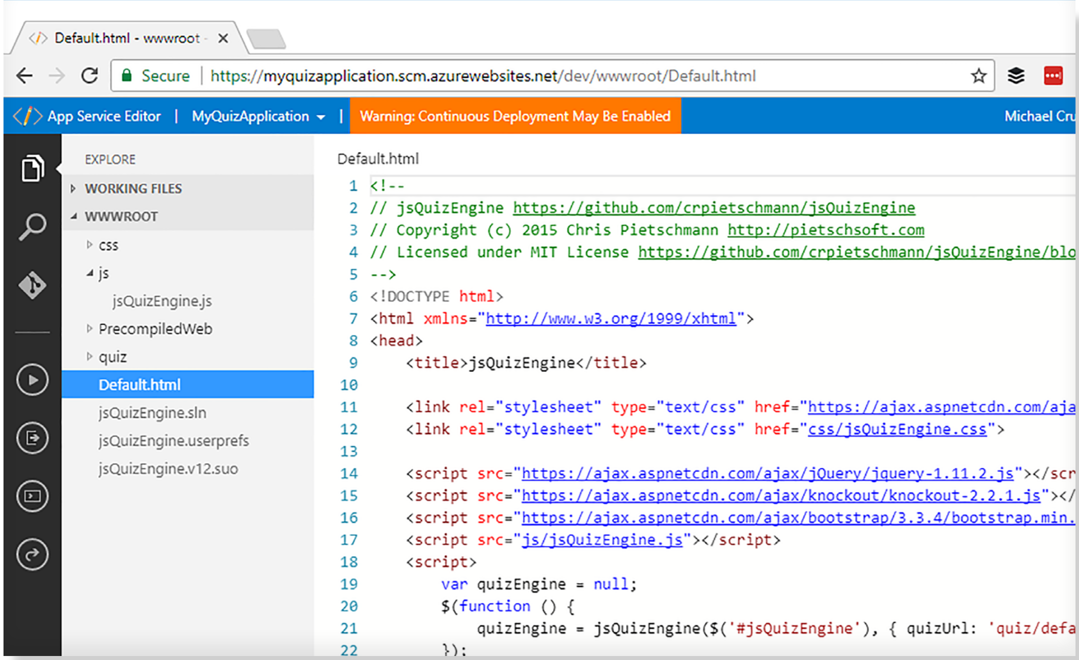
I can do basic commands here and even use **TYPE <FILENAME>** to parse the output of a file to the screen. You can make directory and so forth, but keep in mind that this is a sandbox environment and some commands which require elevated permissions may not work.



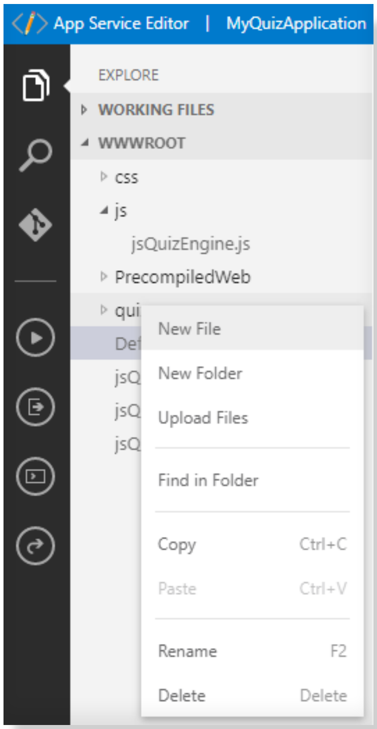
If you're familiar with VS Code, then you'll be right at home as you can explore, search and add to Git. You can also manipulate files from within the window. This makes it easy to add, edit or delete files.

A VS Code Experience to an Azure App Service

A VS Code Experience to an Azure App Service
There is also another option that is called **"App Service Editor"** located just two items down from **"Console"** that you picked before.



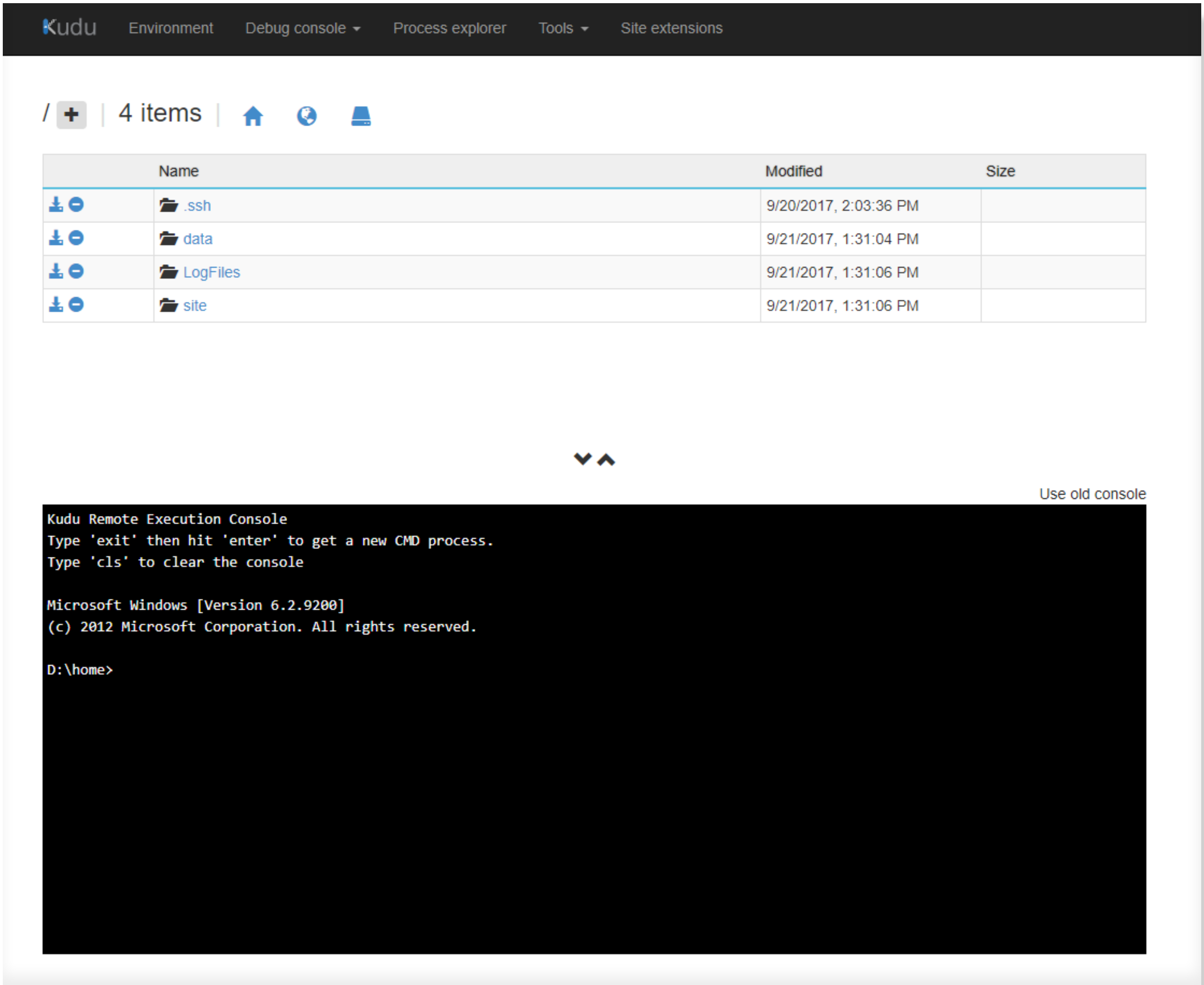
```
<!--
// jsQuizEngine https://github.com/crpietschmann/jsQuizEngine
// Copyright (c) 2015 Chris Pietschmann http://pietschsoft.com
// Licensed under MIT License https://github.com/crpietschmann/jsQuizEngine/blo
-->
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>jsQuizEngine</title>

    <link rel="stylesheet" type="text/css" href="https://ajax.aspnetcdn.com/aja
    <link rel="stylesheet" type="text/css" href="css/jsQuizEngine.css">

    <script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.11.2.js"></scr
    <script src="https://ajax.aspnetcdn.com/ajax/knockout/knockout-2.2.1.js"></
    <script src="https://ajax.aspnetcdn.com/ajax/bootstrap/3.3.4/bootstrap.min.
    <script src="js/jsQuizEngine.js"></script>
    <script>
        var quizEngine = null;
        $(function () {
            quizEngine = jsQuizEngine($('#jsQuizEngine'), { quizUrl: 'quiz/defa
        });
```



Just like in VS Code, you can modify your settings and even change your theme.

Kudu Diagnostic Console

No App Service tutorial is complete without mentioning Kudu Diagnostic Console. You can access it from within the **App Service Editor** under your **app name -> Open Kudu Console** or through the portal under **Advanced Tools.**



Name	Modified	Size
.ssh	9/20/2017, 2:03:36 PM	
data	9/21/2017, 1:31:04 PM	
LogFiles	9/21/2017, 1:31:06 PM	
site	9/21/2017, 1:31:06 PM	

```
Kudu Remote Execution Console
Type 'exit' then hit 'enter' to get a new CMD process.
Type 'cls' to clear the console

Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

D:\home>
```

Quick Tip The App Service Editor is a great choice if ever in doubt and you can access it directly here

You can just click on the folder name to navigate or type in the command. You can also easily manipulate the files, but I like the App Service Editor better for that functionality.

Editor is perfect for lightweight work such as editing files whereas Kudu puts you deep into the weeds with debugging information, file manipulation and more.

The main reason that I typically come to the Kudu Diagnostic Console is to download files.

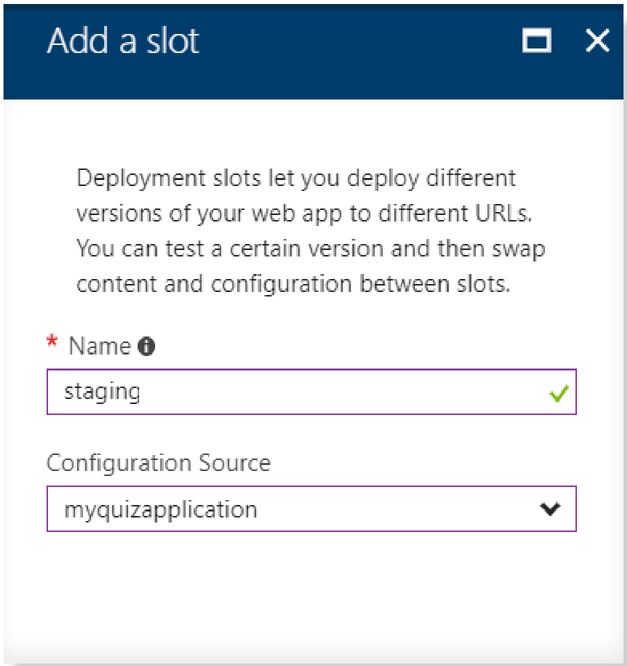
Test Web Apps in Production with Azure App Service

You can learn more about Azure Deployment Slots here

We'll take a look at the files inside an Azure App Service web site and how you can easily work with them.

Creating Deployment Slot

Deployment slots let you deploy different versions of your web app to different URLs. You can test a certain version and then swap content and configuration between slots.
Go to the Azure Portal and select my App Service and click on **Deployment Slots** under **Deployment** to get started.
Then click on the **Add Slots** button. Give it a name such as staging then use an existing configuration source. We'll use our "production" web app. You know, the cool quiz application. Aka.ms/azuretips/myquizapp



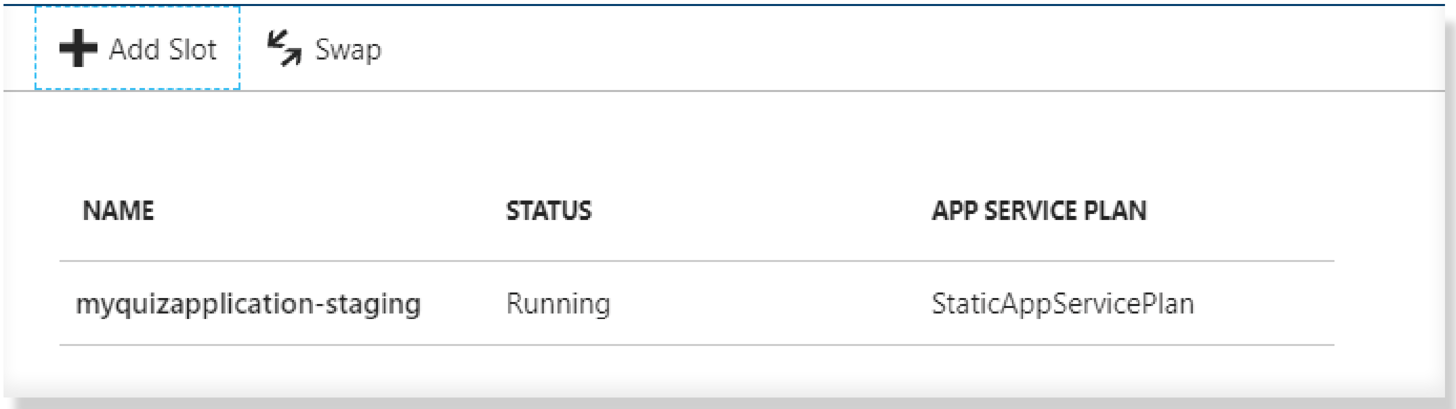
Add a slot

Deployment slots let you deploy different versions of your web app to different URLs. You can test a certain version and then swap content and configuration between slots.

* Name: staging

Configuration Source: myquizapplication

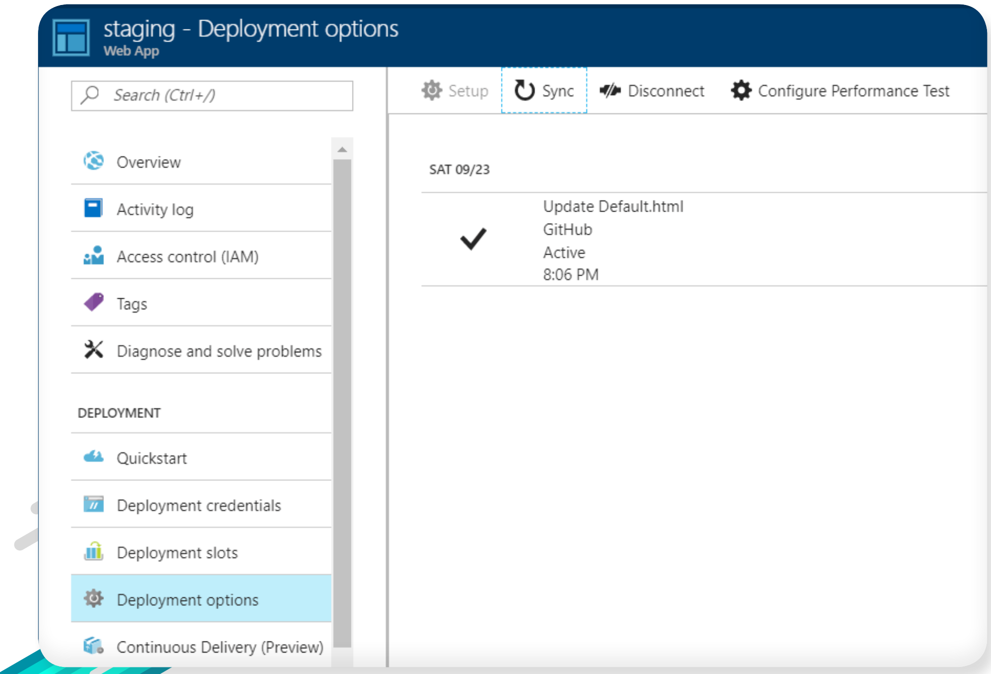
Great, now if we go back to Deployment Slots, we should see it running.



NAME	STATUS	APP SERVICE PLAN
myquizapplication-staging	Running	StaticAppServicePlan

Click on the new staging site that we just created and you'll notice that it has appended the word **staging.** You'll also notice we have a new site: Aka.ms/azuretips/quizsourcegit

We need to push a new version of our existing quiz application to this staging slot. Go to **Deployment Options** and select **External Repository.** Give it the following URL: Aka.ms/azuretips/quizsource and hit OK."You might have to hit Sync, and you'll eventually see the following:



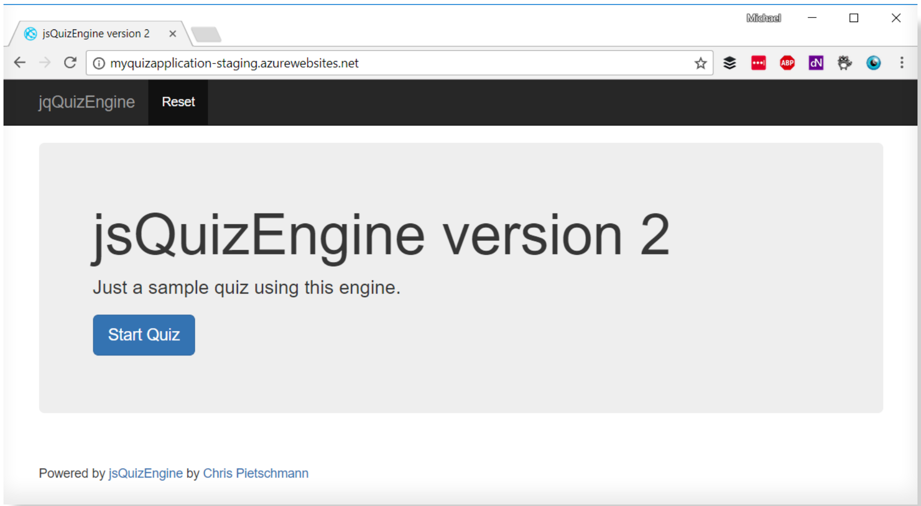
SAT 09/23

✓ Update Default.html
GitHub
Active
8:06 PM

We could now return to the original app service that we created and swap between the two sites that we have. For example, you might want to move the **staging** site over to the **production** site and vice versa. The power of this is that your users don't experience a downtime and you can continue working in your preferred space until ready to move to production.

Source Code The source code to the staging environment can be found here

Give it a couple of minutes until you see that it has completed pulling down your code from Git and then go to the new URL of your site. You can find the URL on your overview page. In my case it is, http://myquizapplication-staging.azurewebsites.net/



jsQuizEngine version 2

Just a sample quiz using this engine.

Powered by jsQuizEngine by Chris Pietschmann

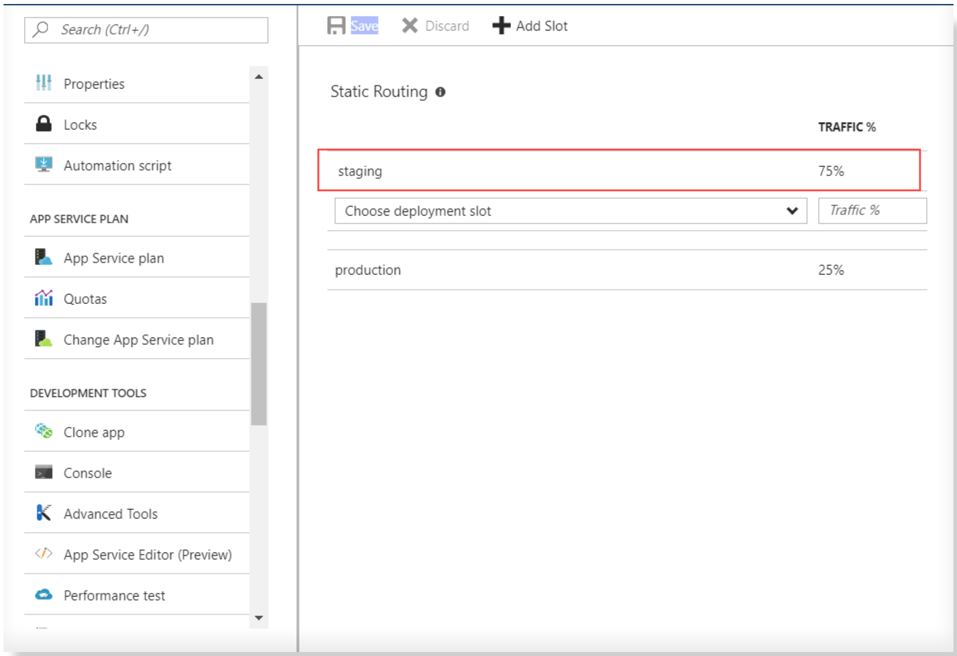
Success! This is our new site as indicated by the awesome large font that says **jsQuizEngine version 2.**

In this tip, we'll look at a feature called Testing in Production which allows you to test your application in production. Not scary at all!

Testing Web Apps in Production with Azure App Service

Go to the Azure Portal and select my App Service and click on Testing in Production under **Development Tools** to get started. The first thing you'll see is **Static Routing** and you'll notice that it's looking for a deployment slot and traffic percentage.

We'll want to split the traffic to our site into two groups to test our new site and see if customers like it. Since this is just a demo, I want to send a large number of folks to our new **staging** site as shown below.



Static Routing

	TRAFFIC %
staging	75%
Choose deployment slot	Traffic %
production	25%

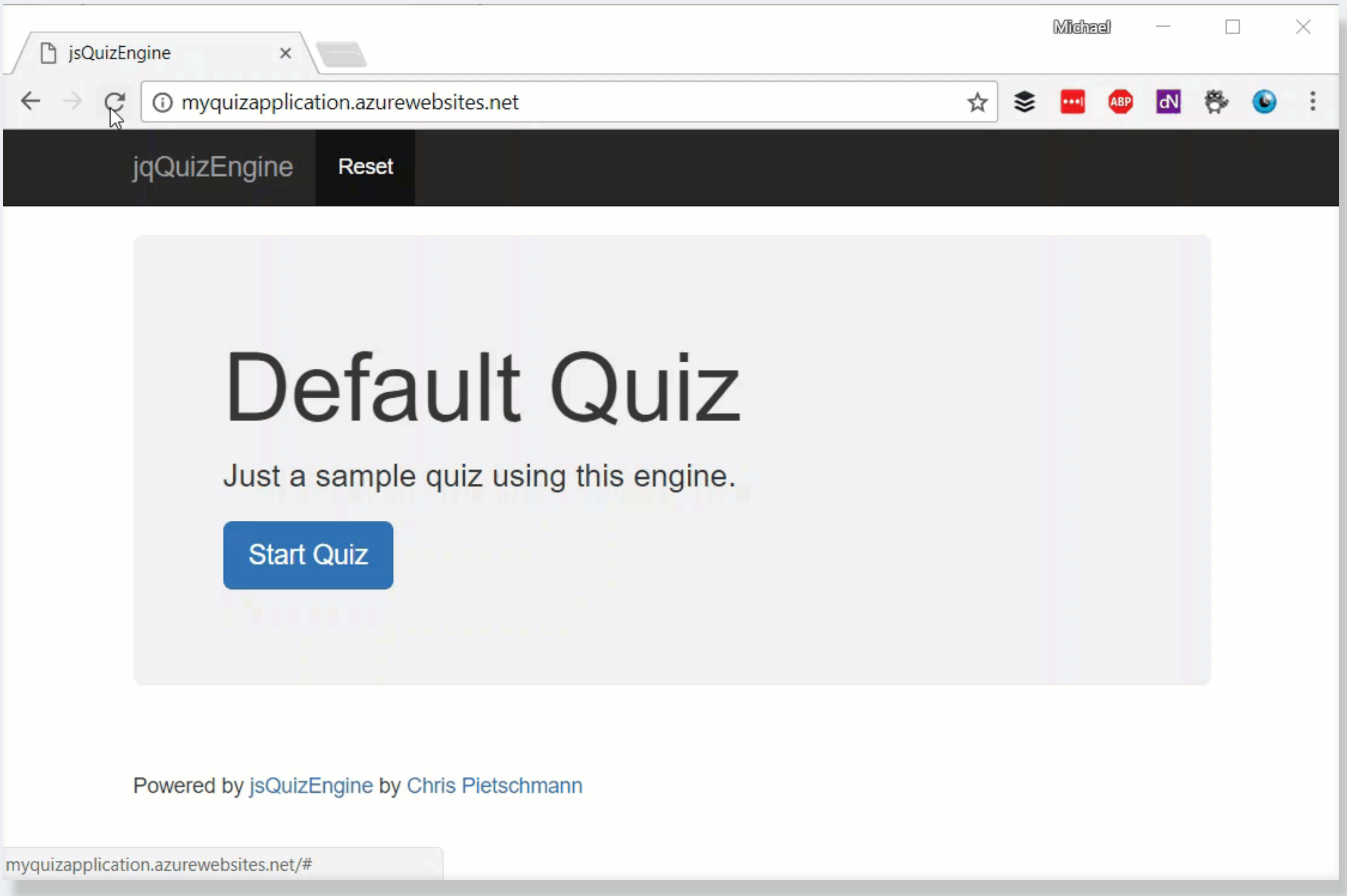
Great! Now keep in mind that we have two versions of our site: one that is **production** and one that is **staging**. They are identical except for the staging site has a large font that says **jsQuizEngine version 2.**

We don't want to **swap** sites, we just want to **distribute** traffic between the two sites.

I can test this by going to my production URL and refreshing the site until the staging site is shown with the production URL.

Hold up! You'll want to take a look at the deployment slots in the previous tip if you haven't worked with deployment slots before.

What is Static Routing This section lets you control how traffic is distributed between your production and other slots. This is useful if you want to try out a new change with a small percentage of requests and then gradually increase the percentage of requests that get the new behavior.

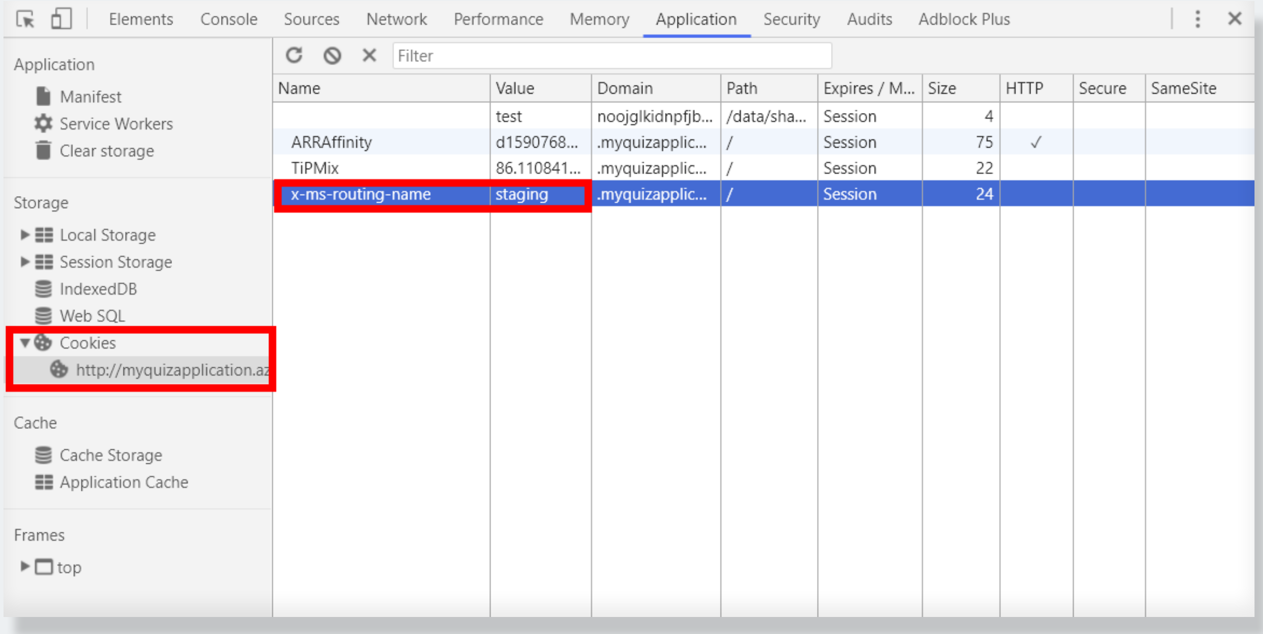

jsQuizEngine
Michael
myquizapplication.azurewebsites.net
jqQuizEngine
Reset
Start Quiz
myquizapplication.azurewebsites.net/#

Default Quiz

Just a sample quiz using this engine.

Powered by jsQuizEngine by Chris Pietschmann

Success! It works, but what happens when they leave the site? We actually store a cookie that keeps track of it. You can find this cookie yourself by inspecting the site and looking for the cookie shown on the next page.



Name	Value	Domain	Path	Expires / M...	Size	HTTP	Secure	SameSite
	test	noojglkidnpfjb...	/data/sha...	Session	4			
ARRAffinity	d1590768...	.myquizapplic...	/	Session	75	✓		
TiPMix	86.110841...	.myquizapplic...	/	Session	22			
x-ms-routing-name	staging	.myquizapplic...	/	Session	24			

You could actually force the old production site by setting the x-ms-routing-name cookie to self or providing it in the URL query string such as http://myquizapplication.azurewebsites.net/?x-ms-routing-name=self You could even use the URL to let your users test different versions of your site. For example, I could use http://myquizapplication.azurewebsites.net/?x-ms-routing-name=staging to let users try my new website before I push it live. This is very neat stuff, folks!

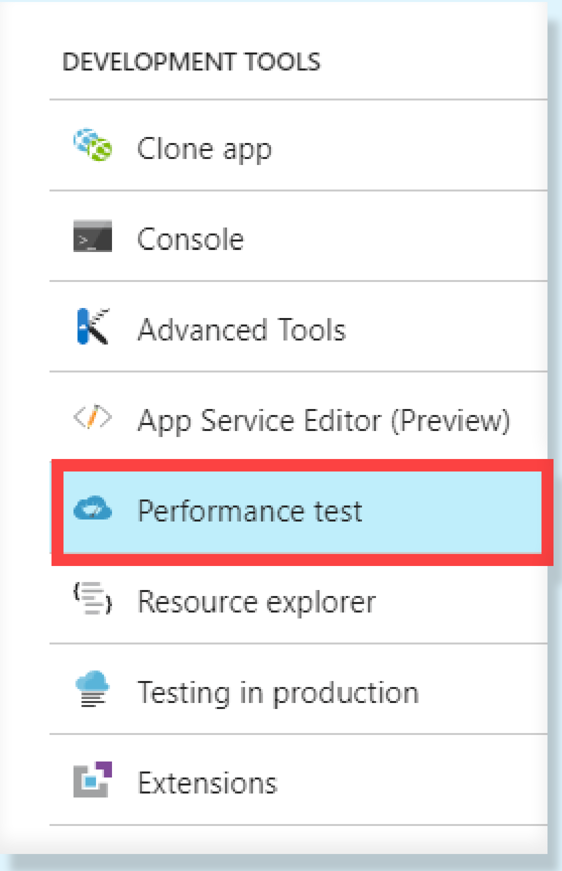
Learn more about load testing at Aka.ms/azuretips/vsts

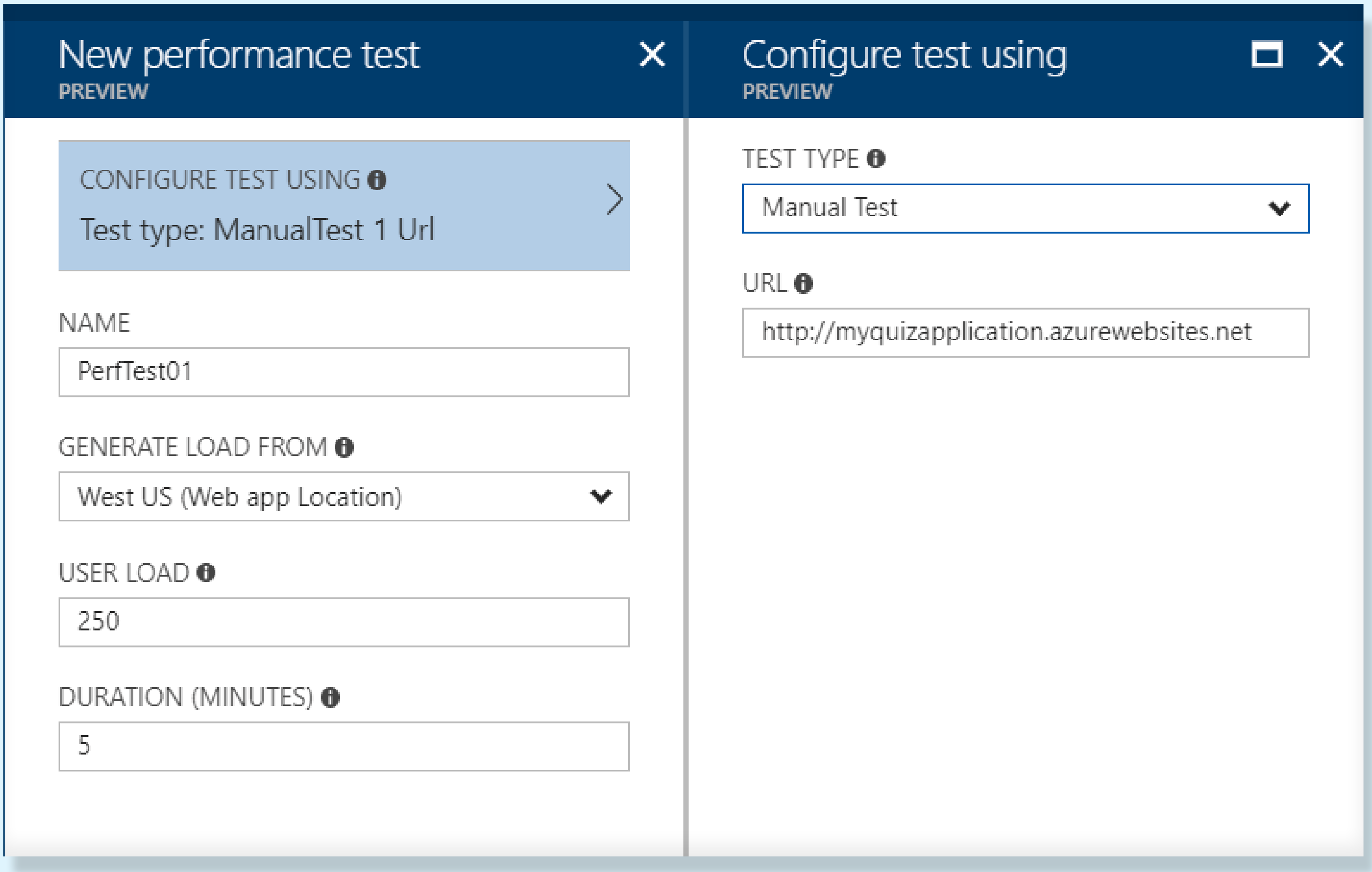
In this tip, we'll look at a simple and quick way to perform load testing of your web app.

Load Testing web apps with Azure App Services

Load Testing allows you to test your web app's performance and determine if your app can handle increased traffic during peak times. You can find this tool by logging into your Azure account, going to your App service that you created, and looking under **Development Tools.**

Inside the blade, select New and you will see the following options:





New performance test
PREVIEW

CONFIGURE TEST USING
Test type: ManualTest 1 Url

NAME
PerfTest01

GENERATE LOAD FROM
West US (Web app Location)

USER LOAD
250

DURATION (MINUTES)
5

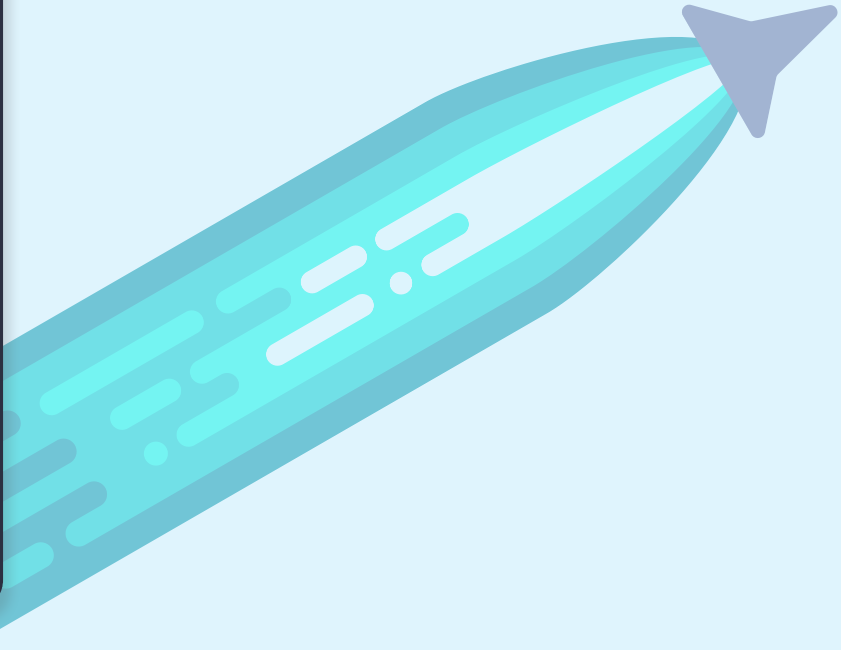
Configure test using
PREVIEW

TEST TYPE
Manual Test

URL
http://myquizapplication.azurewebsites.net

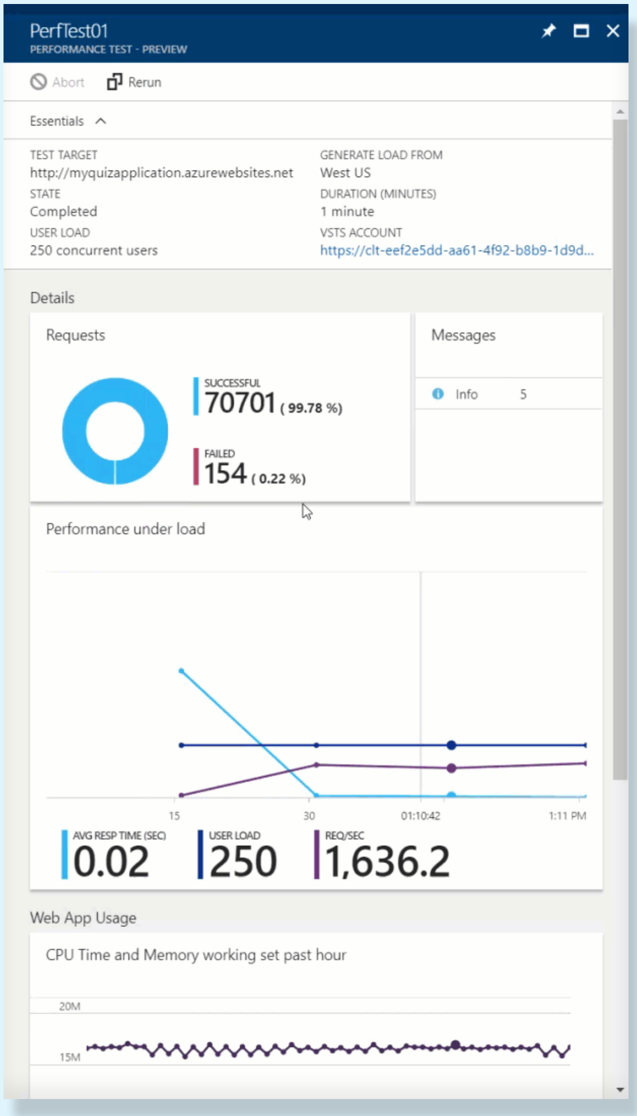
You have the option to **Configure Test** and you can leave this as **Manual Test** or **Visual Studio Web Test**. The main difference between the two is that with the latter you can select multiple URLs and even use a HTTP Archive file (such as one created by Fiddler). Leave the testing option as manual and select a name and location, and make sure you leave the defaults as 250 users for 5 minutes.

Use Case Scenario Suppose you have a web app and you have something for sale. You have an upcoming promo that last year had 175 users connected for 5 minutes. Users complained that the site was slow and since your site has grown, you want to improve customer satisfaction by reducing the page load time and test your web app with a load of 250 users for 5 minutes. Let the test run and you'll be presented with the following information once it has completed:



Look out! Keep in mind that there is a charge for performing a load test in terms of virtual users as indicated in the screenshot.

We were able to do this without writing code and with just a couple of clicks in the portal.



PerfTest01
PERFORMANCE TEST - PREVIEW

TEST TARGET
http://myquizapplication.azurewebsites.net

STATE
Completed

USER LOAD
250 concurrent users

GENERATE LOAD FROM
West US

DURATION (MINUTES)
1 minute

VSTS ACCOUNT
https://clt-eef2e5dd-aa61-4f92-b8b9-1d9d...

Details

Requests

SUCCESSFUL
70701 (99.78 %)

FAILED
154 (0.22 %)

Messages

Info 5

Performance under load

AVG RESP TIME (SEC)
0.02

USER LOAD
250

REQ/SEC
1,636.2

Web App Usage

CPU Time and Memory working set past hour

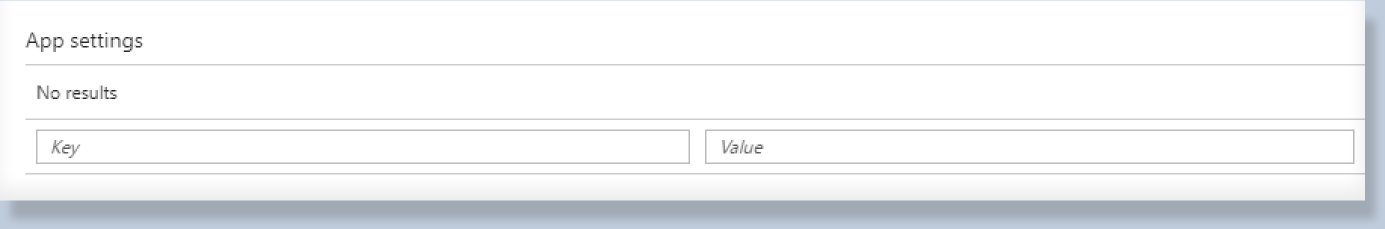
Learn more about App Settings at Aka.ms/azuretips/appservconfig

In this post, we'll take advantage of App Settings to store a Key/Value pair securely in Azure and access it in your web app.

Working with App Settings and Azure App Services

App Settings are used to store configurable items without making any changes to the code. The key/value pairs are stored behind the scenes in a configuration store, which is nice because sensitive information never shows up in a web.config, etc. file. In order to take advantage of this, you'll need to log into your Azure account and go to your App Service that you created and look under Development Tools then you will see Application Settings.

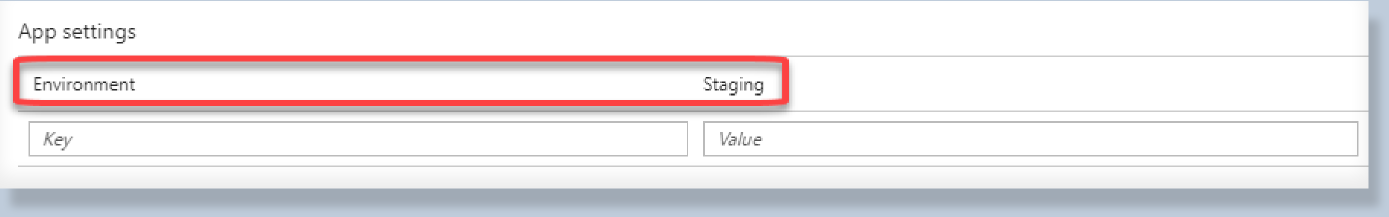
Open it and scroll down and you'll see App Settings as shown below.



App settings

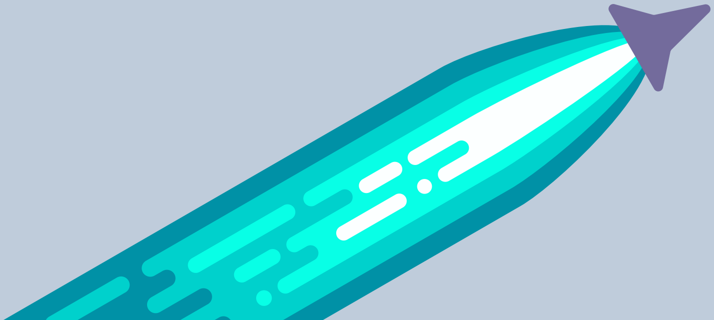
No results

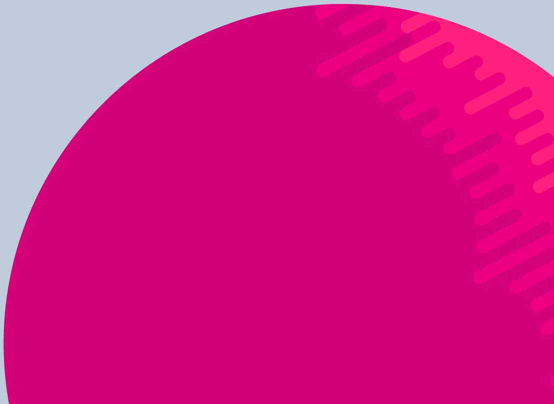
We're going to add an App Setting in Azure. I added one with the key of Environment and the value is set to Staging.



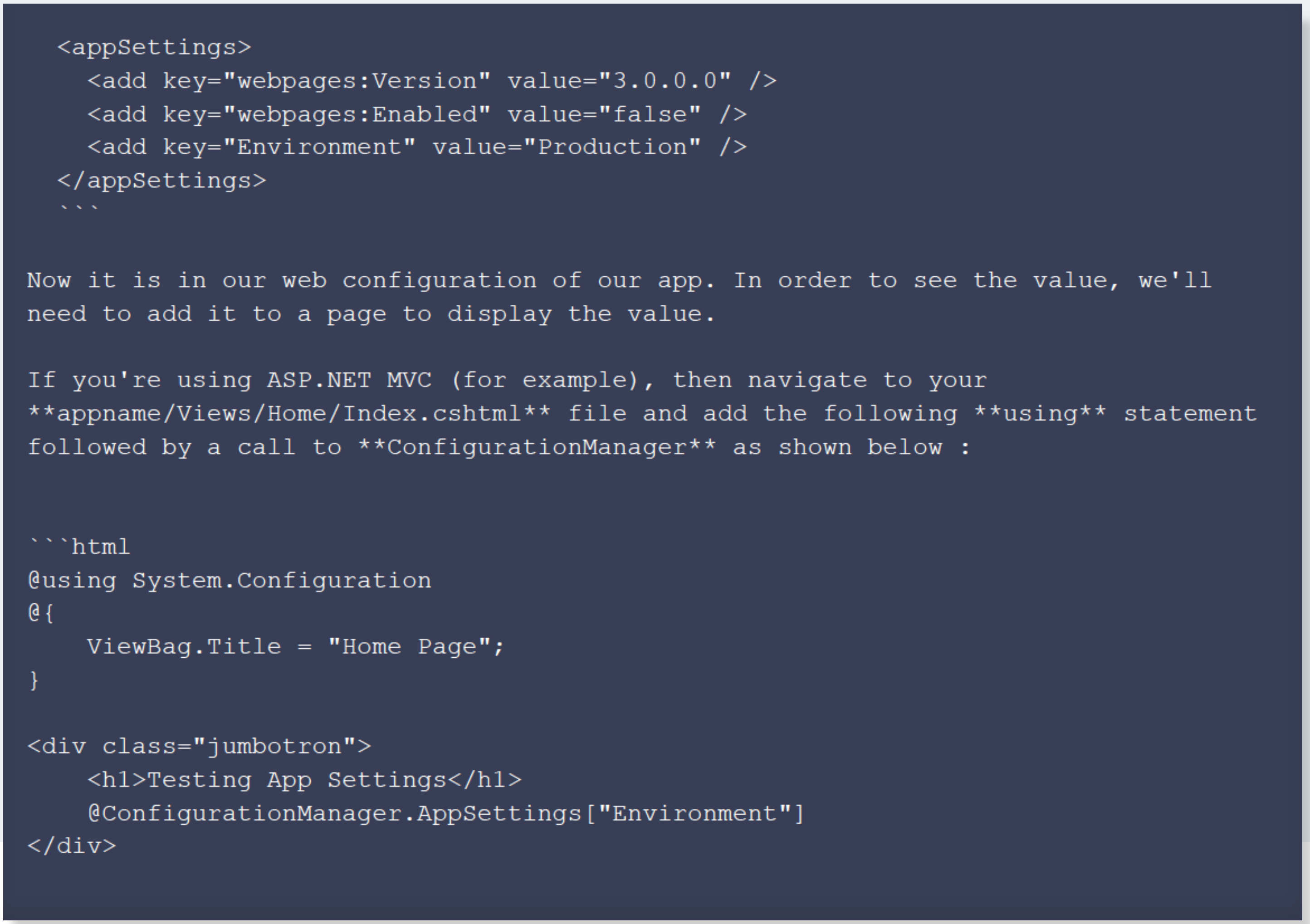
App settings

Environment Staging



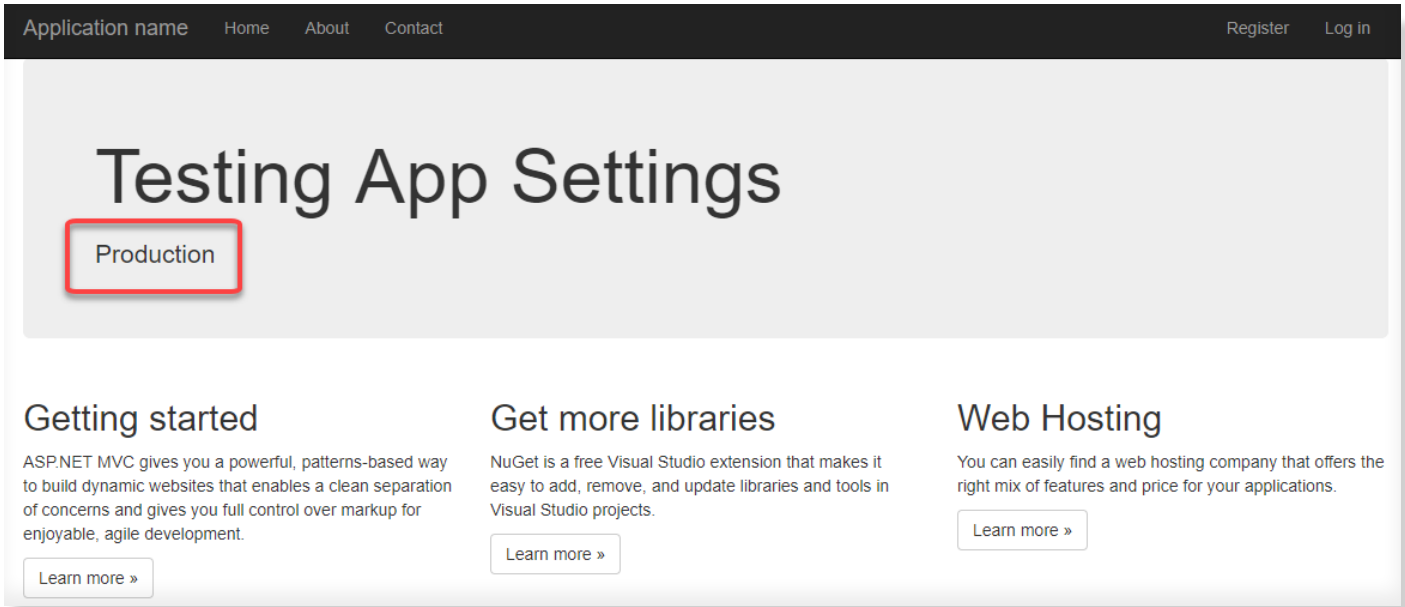


Open or create your ASP.NET MVC app and modify the appSettings section of the web.config file to add our Environment key/value pair as shown below:



~~~
  <appSettings>
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="Environment" value="Production" />
  </appSettings>
  ```

Now it is in our web configuration of our app. In order to see the value, we'll
need to add it to a page to display the value.

If you're using ASP.NET MVC (for example), then navigate to your
**appname/Views/Home/Index.cshtml** file and add the following **using** statement
followed by a call to **ConfigurationManager** as shown below :


```html
@using System.Configuration
@{
    ViewBag.Title = "Home Page";
}

<div class="jumbotron">
    <h1>Testing App Settings</h1>
    @ConfigurationManager.AppSettings["Environment"]
</div>
~~~

If you run the application locally, then you'll see Production as it is coming from the web.config file, but if you run it inside of Azure, then you'll see Staging as it is coming from the Apps Settings configuration store located in Azure. Neat stuff!



Testing App Settings

Production

Getting started

ASP.NET MVC gives you a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and gives you full control over markup for enjoyable, agile development.

Learn more »

Get more libraries

NuGet is a free Visual Studio extension that makes it easy to add, remove, and update libraries and tools in Visual Studio projects.

Learn more »

Web Hosting

You can easily find a web hosting company that offers the right mix of features and price for your applications.

Learn more »

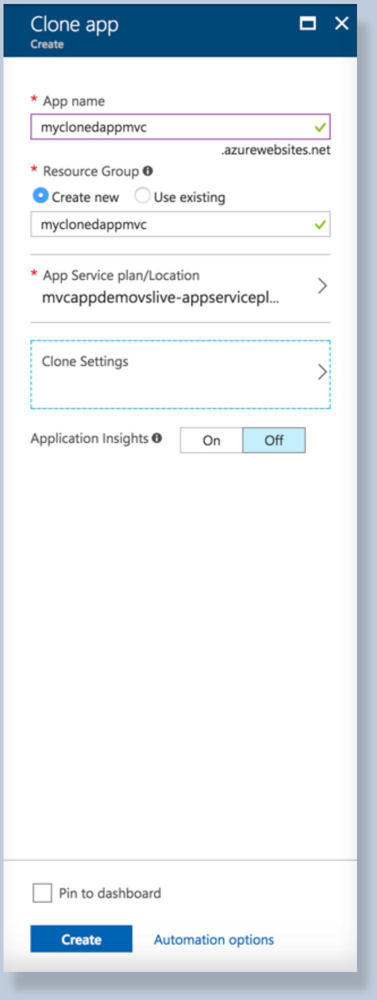
Connection Strings vs. App Settings You may have noticed Connection Strings right below the App Settings option and wonder when to use it. A general rule of thumb is to use Connection Strings for database connection strings and App Settings for key/value pair application settings. If you examine your web.config file, then you'll see there is also a section for *connectionStrings* just as there is a section for *appSettings*.

Cloning is the ability to duplicate an existing Web App to a newly created app that is often in a different region. This will enable customers to deploy a number of apps across different regions quickly and easily.

Cloning Web Apps Using and Azure App Services

Scenario: A company has an existing web app in West US, they would like to clone the app to East US to serve folks that live on that site with better performance such as latency.
To do this, log into your Azure account and go to your App Service that you created. Look under Development Tools and find Clone App.

Open it and
you'll see the following:



Clone app
Create

* App name
myclonedappmvc ✓
.azurewebsites.net

* Resource Group
● Create new ○ Use existing
myclonedappmvc ✓

* App Service plan/Location
mvcappdemovslive-appservicepl...

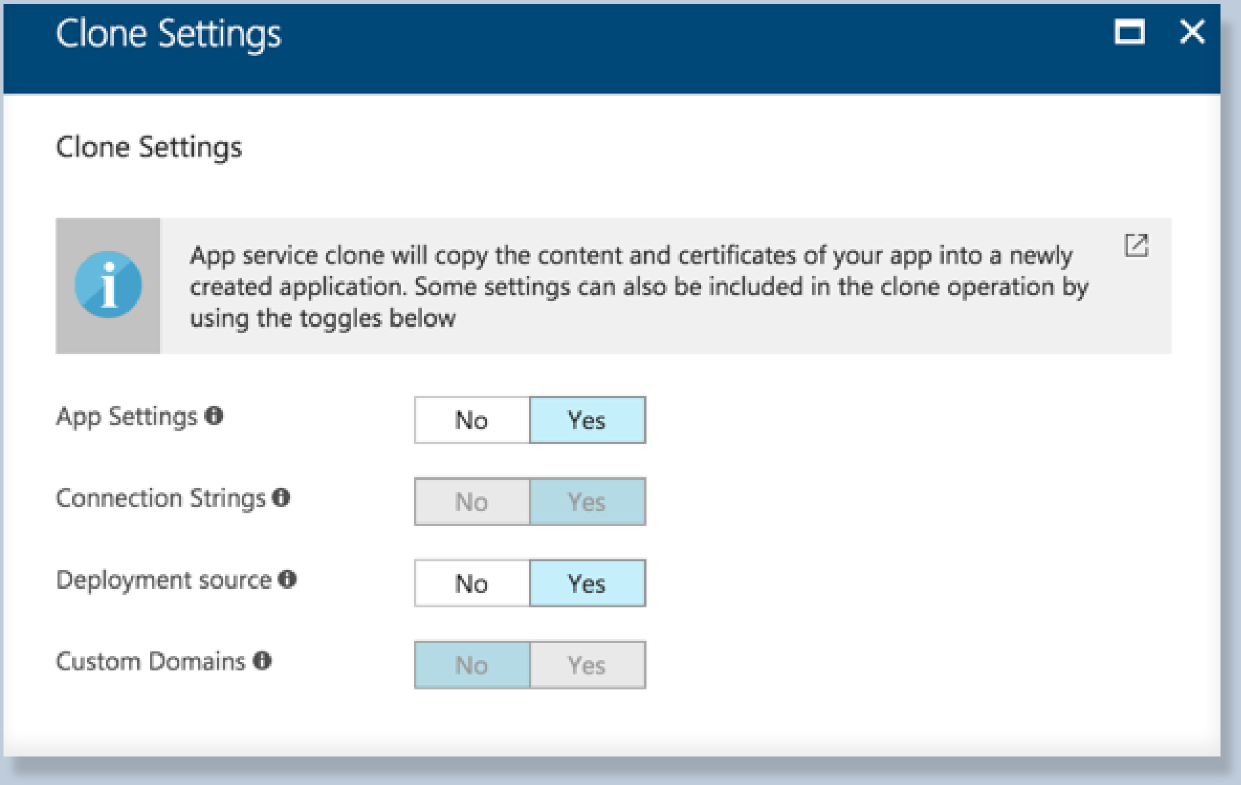
Hold Up Besides changing the location, this is also a great time to determine the plan needed. You might not need all the horsepower to serve this site if you expect very low traffic in that region.

Ensure you give it an:

- App Name - Something unique as this site will live in something.azurewebsites.net
- Resource Group - Create a new one or use an existing one
- App Service Plan/Location - This is a good time to associate a new plan that will determine the location, features, and cost, and compute resources associated with your app.

- **Application Insights** - You can turn it on or off to help you detect and diagnose issues with .NET apps.
- **Clone Settings** - Clone will copy the content and certificates of your app into a newly created application. You can also copy things like App Settings, Connection Strings, Deployment Source, and Custom Domains.

Finally, there is **Automation Options** which brings you to the Azure Resource Manager templates that are so valuable. Aka.ms/azuretips/resourcemanager



Clone Settings

Clone Settings

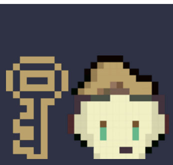
App service clone will copy the content and certificates of your app into a newly created application. Some settings can also be included in the clone operation by using the toggles below

App Settings: No [✓ Yes]

Connection Strings: No [✓ Yes]

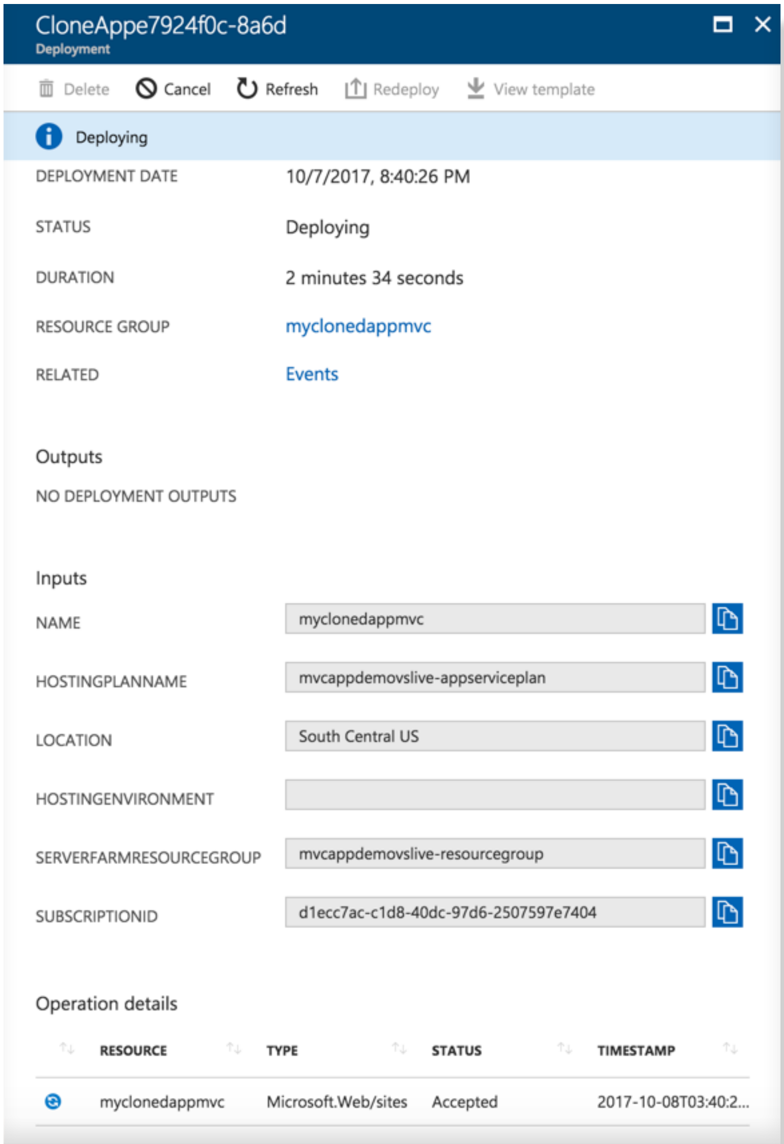
Deployment source: No [✓ Yes]

Custom Domains: [✓ No] Yes



What is a Azure Resource Manager again? Azure Resource Manager enables you to work with the resources in your solution as a group. You can deploy, update, or delete all the resources for your solution in a single, coordinated operation. You use a template for deployment and that template can work for different environments such as testing, staging, and production. Resource Manager provides security, auditing, and tagging features to help you manage your resources after deployment. Aka.ms/azuretips/appservdeploy

Once everything is set up then press **Create** and you'll see the **Deployment in Progress** begin. You can click on it while deploying to see details as shown:



CloneAppe7924f0c-8a6d
Deployment

Deploying

DEPLOYMENT DATE	10/7/2017, 8:40:26 PM
STATUS	Deploying
DURATION	2 minutes 34 seconds
RESOURCE GROUP	myclonedappmvc
RELATED	Events

Outputs

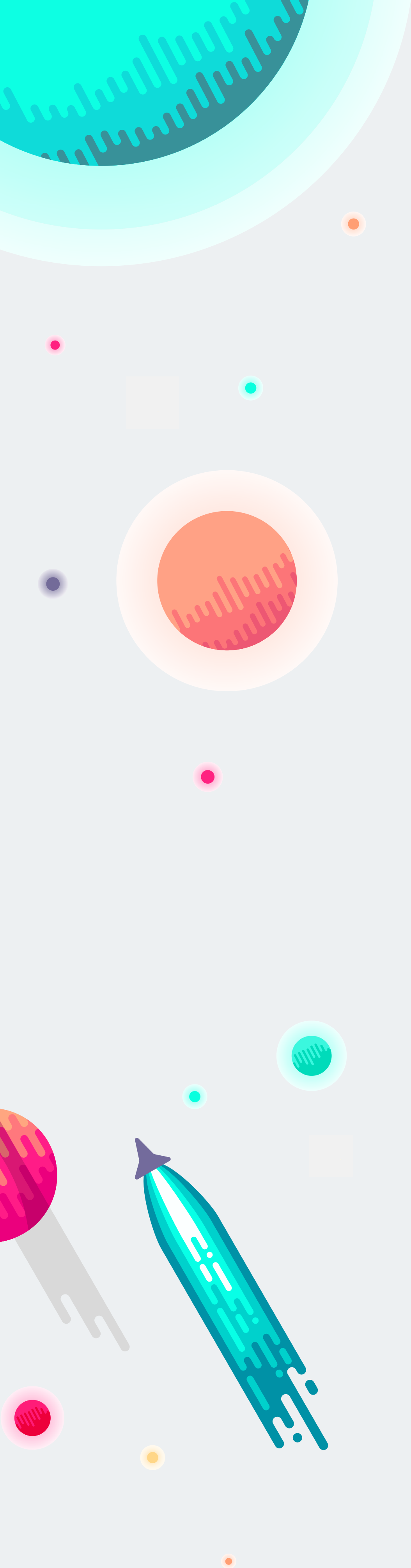
NO DEPLOYMENT OUTPUTS

Inputs

NAME	myclonedappmvc
HOSTINGPLANNAME	mvcappdemovslive-appserviceplan
LOCATION	South Central US
HOSTINGENVIRONMENT	
SERVERFARMRESOURCEGROUP	mvcappdemovslive-resourcegroup
SUBSCRIPTIONID	d1ecc7ac-c1d8-40dc-97d6-2507597e7404

Operation details

	RESOURCE	TYPE	STATUS	TIMESTAMP
	myclonedappmvc	Microsoft.Web/sites	Accepted	2017-10-08T03:40:2...



Conclusion

There are 130+ additional tips waiting on you that cover additional topics such as :

- App Services
- CLI
- Cloud Shell
- Cognitive Services
- Containers
- Cosmos DB
- Functions
- IoT
- Logic Apps
- Portal
- PowerShell
- Productivity
- Storage
- SQL and Search

Find all of these and more at azuredev.tips

Don't forget that if you are modernizing an existing application or building a new app, you can get started
Azure for free and get:

- $200 credit toward use of any Azure service
- 12 months of free services—includes compute, storage, network, and database
- 25+ always-free services—includes serverless, containers, and artificial intelligence

Start free

Until next time,

Michael Crump @mbcrump

signing off...

Azure Tips and Tricks

azuredev.tips

Made with love By Red Door Collaborative.com