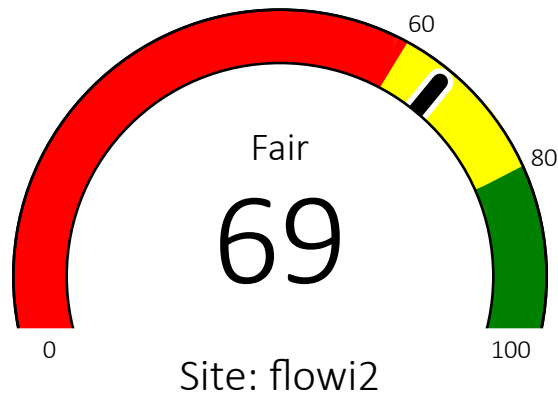


Azure App Service Resiliency Score report for: flowi2

Report generated on: 2023-11-19T20:12:44.026Z

Your Web App's Resiliency Score



This is a weighted calculation based on which best practices were followed. A score of 80 or above is considered highly resilient and it will be marked as green. A score of 100% doesn't mean that the Web App will never be down but rather that it has implemented 100% of our resiliency best practices.

Contributing factors to your score and how you can improve it

| Feature/Site name | flowi2 |
|--|--------|
| <u>Use of multiple instances</u> | ✓ |
| <u>Health Check</u> | ✓ |
| <u>Auto-Heal</u> | ✗ |
| <u>Deploy in Multiple Regions/Zones</u> | ✗ |
| <u>Regional Pairing</u> | ✗ |
| <u>App density</u> | ✓ |
| <u>AlwaysOn check</u> | ! |
| <u>App Service Advisor Recommendations</u> | ✓ |
| <u>ARR Affinity Disabled Check (Recommendation. Not counted against the score)</u> | ✗ |
| <u>Production SKU used</u> | ✓ |

✓ Fully implemented

! Partially implemented

✗ Not implemented

Use of multiple instances

Description

Running your app on only one VM instance is an immediate single point-of-failure. By ensuring that you have multiple instances allocated to your app, if something goes wrong with any instance, your app will still be able to respond to requests going to the other instances. Keep in mind that your app code should be able to handle multiple instances without synchronization issues when reading from or writing to data sources.

Status of verified Web Apps

| Site name | Grade | Comments |
|---------------|-------------|---|
| flowi2 | Pass | At the time of this report this Web App was configured to use 3 workers |

Solution

No changes needed as application is already using multiple instances.

To add more instances:

- Azure Portal:
 - Open the **Azure Portal**
 - Click on **App Service plans**
 - Click on the App Service Plan hosting the Web App(s) you want to scale out.
 - Under **Settings** click on **Scale out (App Service plan)**
 - Click **Manual scale**
 - Increase the number of instances to the desired value (at least 2)
- PowerShell:
 - Use the **Set-AzAppServicePlan** command.
For more information see: <https://docs.microsoft.com/en-us/azure/app-service/scripts/powershell-scale-manual>
- Azure CLI:
 - Use the **az appservice plan update** command.
For more information see: <https://docs.microsoft.com/en-us/azure/app-service/scripts/cli-scale-manual>

More information

- The Ultimate Guide to Running Healthy Apps in the Cloud - Use Multiple Instances
<https://azure.github.io/AppService/2020/05/15/Robust-Apps-for-the-cloud.html#use-multiple-instances>

Health Check

Description

App Service makes it easy to automatically scale your apps to multiple instances when traffic increases. This increases your app's throughput, but what if there is an uncaught exception on one of the instances? App Service allows you to specify a health check path on your apps. The platform pings this path to determine if your application is healthy and responding to requests. If an instance fails to respond to the ping, the system determines it is unhealthy and removes it from the load balancer rotation. This increases your application's average availability and resiliency. When your site is scaled out to multiple instances, App Service will exclude any unhealthy instance(s) from serving requests, improving your overall availability. Your app's health check path should poll the critical components of your application, such as your database, cache, or messaging service and return a 5xx error if any of them fail. This ensures that the status returned by the health check path is an accurate picture of the overall health of your application.

Status of verified Web Apps

| Site name | Grade | Comments |
|-----------|-------|--|
| flowi2 | Pass | Web App is running on 3 workers and Health Check feature is currently configured for it. |

Solution

No changes are needed as Health Check is being used.

If you need to enable Health check on another Web App, follow these instructions:

- Using the **Azure Portal**:
 - Open the **Azure Portal**
 - Click on an **App Services**
 - Click on the Web App for which you want to enable Health Check.
 - Under **Monitoring** click in **Health check**.
 - Click **Enable**.
 - Under **Path** add the path to a page that will only return 200 once your app and its dependencies are responsive.
 - Configure the time in **Load Balancing** and click **Save**.

More information

- Health Check is now Generally Available
<https://azure.github.io/AppService/2020/08/24/healthcheck-on-app-service.html>
- The Ultimate Guide to Running Healthy Apps in the Cloud – Set your Health Check path
<https://azure.github.io/AppService/2020/05/15/Robust-Apps-for-the-cloud.html#set-your-health-check-path>

Auto-Heal

Description

Sometimes your app may run into issues, resulting in downtimes, slowness, or other unexpected behaviors. We've built App Service Diagnostics to help you diagnose and solve issues with your web app with recommended troubleshooting and next steps. However, these unexpected behaviors may be temporarily resolved with some simple mitigation steps, such as restarting the process or starting another executable, or require additional data collection, so that you can better troubleshoot the ongoing issue later. With Auto Healing, you can set up custom mitigation actions to run when certain conditions (that you define as unexpected or a sign of unhealthy behavior) are met:

- Request Duration: examines slow requests
- Memory Limit: examines process memory in private bytes
- Request Count: examines number of requests
- Status Codes: examines number of requests and their HTTP status code

Status of verified Web Apps

| Site name | Grade | Comments |
|-----------|-------|------------------------------|
| flowi2 | Fail | Auto-Heal is not implemented |

Solution

Auto-Heal is highly recommended for production applications that need to ensure high availability and resilience. Although Auto-Heal is not a fix for unexpected issues that your application may encounter (whether platform or application related), it does allow for quick recovery.

To enable Auto-Heal follow these instructions:

- To enable **Auto-Heal**:
 - Open the **Azure Portal**
 - Click on an **App Services**
 - Click on the Web App where for which you want to enable Auto-Heal
 - Click **Diagnose and solve problems**
 - Type **Auto-Heal** in the "Search for common problems or tools." box and click in **Auto-Heal** under the results
 - For custom rules:
 - Under the **Custom Auto-Heal Rules** tab set **Custom Auto-Heal Rules** to **Enabled**
 - For Proactive Auto-Heal
 - Under the **Proactive Auto-Heal** tab set **Proactive Auto-Heal** to **Enabled**

More information

- Announcing the New Auto Healing Experience in App Service Diagnostics
<https://azure.github.io/AppService/2018/09/10/Announcing-the-New-Auto-Healing-Experience-in-App-Service-Diagnostics.html>

Deploy in Multiple Regions/Zones

Description

You can deploy Azure Front Door or Azure Traffic Manager to intercept traffic before they hit your site. They help in routing & distributing traffic between your instances/regions. If a catastrophic incident happens in one of the Azure Datacenters, you can still guarantee that your app will run and serve requests by investing in one of them.

There are additional benefits to using Front Door or Traffic Manager, such as routing incoming requests based the customers' geography to provide the shortest respond time to customers and distribute the load among your instances in order not to overload one of them with requests.

Status of verified Web Apps

| Site name | Grade | Comments |
|---------------|-------------|---|
| flowi2 | Fail | This site doesn't seem to be using a custom domain, so we couldn't determine if it's deployed to multiple regions |

Solution

In order to create an extra layer of resiliency consider deploying this site to another region and preferably on a Regional pair (See Regional Pairing section ahead).

More information

- The Ultimate Guide to Running Healthy Apps in the Cloud - Deploy in Multiple Regions
<https://azure.github.io/AppService/2020/05/15/Robust-Apps-for-the-cloud.html#deploy-in-multiple-regions>
- Controlling Azure App Service traffic with Azure Traffic Manager
<https://docs.microsoft.com/en-us/azure/app-service/web-sites-traffic-manager>
- Quickstart: Create a Front Door for a highly available global web application
<https://docs.microsoft.com/en-us/azure/frontdoor/quickstart-create-front-door>

Regional Pairing

Description

An Azure region consists of a set of data centers deployed within a latency-defined perimeter and connected through a dedicated low-latency network. This ensures that Azure services within an Azure region offer the best possible performance and security.

An Azure geography defines an area of the world containing at least one Azure region. Geographies define a discrete market, typically containing two or more regions, that preserve data residency and compliance boundaries.

A regional pair consists of two regions within the same geography. Azure serializes platform updates (planned maintenance) across regional pairs, ensuring that only one region in each pair updates at a time. If an outage affects multiple regions, at least one region in each pair will be prioritized for recovery.

In addition, if your Web Apps are running on an App Service Environment (ASE), you can take advantage of the a feature for [Automatically Redirecting Traffic During Platform Upgrades](#).

Status of verified Web Apps

| Site name | Grade | Comments |
|---------------|-------------|--|
| flowi2 | Fail | This Web App is not deployed in a paired region. |

Solution

This Web App is deployed in East US. In order to take advantage of Paired Regions deploy a copy of this site to a Web App in West US and use something like Azure Traffic Manager, Front Door or Application Gateway to load balance the traffic.

More information

- Business continuity and disaster recovery (BCDR): Azure Paired Regions <https://docs.microsoft.com/en-us/azure/best-practices-availability-paired-regions>

Automatically Redirecting Traffic During Platform Upgrades

Description

You can register to receive notifications from the platform before the instances hosting your Azure App Service Web App running on App Service Environment (ASE) will be restarted due to a platform upgrade and again once the upgrade has finished.

With a combination of Azure Front Door and a Logic App, you can configure your environment so that traffic is automatically redirected to your Web App on another region while your Web App is going through a Platform Upgrade by following the steps in the following link:

<https://github.com/Azure-Samples/azure-logic-app-traffic-update-samples>

App density

Description

For production applications, it is recommended that an App Service Plan does not host more than a certain number of sites. The number may be lower depending on how resource intensive the hosted applications are, however as a general guidance, you may refer to the table below:

| App Service Plan SKU | Max sites |
|----------------------|-----------|
| B1, S1, P1v2, I1v1 | 8 |
| B2, S2, P2v2, I2v1 | 16 |
| B3, S3, P3v2, I3v1 | 32 |
| P1v3, I1v2 | 16 |
| P2v3, I2v2 | 32 |
| P3v3, I3v2 | 64 |

Status of verified Web Apps

| Site name | Grade | Comments |
|-----------|-------|---|
| flowi2 | Pass | Total active sites on the App Service Plan are within the recommended value. Your app is running on Premium0V3 Small sized workers. We detected 2 active app(s) within the ASP-openfranceressource-a8d4 App Service Plan. |

Solution

For production applications, it is recommended that an App Service Plan does not host more than a certain number of sites. The number may actually be lower depending on how resource intensive the hosted applications are, however as a general guidance, you may refer to the table above.

Stop non production apps to avoid exhausting system resources.

To stop inactive Web Apps:

1. Navigate to the **App Service Plan** in the **Azure Portal**
2. While on the **Overview** blade, click on the link next to Apps(s) / Slots.
3. Review the apps and slots listed there and stop the ones that are not critical

More information

- Azure App Service plan overview
<https://docs.microsoft.com/en-us/azure/app-service/azure-web-sites-web-hosting-plans-in-depth-overview>

Other Best Practices for Availability & Performance

Always on Check

Description

Websites unload if they sit idle for too long, which helps the system conserve resources. Always On setting (available for Standard tier and above), keeps your site up and running, which translates to higher availability and faster response times across the board.

Keeps the app loaded even when there's no traffic. It's required for continuous WebJobs or for WebJobs that are triggered using a CRON expression.

If Always On is enabled but there's something preventing it from reaching the actual root of the Web App (like redirects due authentication/authorization/HTTPS Only, etc.), it might not be able to keep your application from going idle

Status of verified Web Apps

| Site name | Grade | Comments |
|-----------|-------|--|
| flowi2 | N/A | AlwaysOn Enabled but may not be completely effective |

Solution

AlwaysOn is enabled but is not returning with a 200 OK as a result it may not be actually hitting your application code. Although the process hosting the application will remain active, the actual application may not. Implement an endpoint that listens to '/' endpoint

Enable **Always on**:

- Using the **Azure Portal**:
 1. Browse to the Azure Portal (<https://portal.azure.com>)
 2. Click on the Portal menu on the top left corner
 3. Click on **App Services**
 4. Select the App Service for which you want to enable **Always on**
 5. Click on **Configuration**
 6. Click on **General settings**
 7. Change **Always on** from Off to **On**.
 8. Click on **Save**

- Using the **Azure AzPowerShell**:
 1. Use the **Set-AzWebApp** cmdlet with the **-AlwaysOnset** to true
For example:

```
$app = Get-AzWebApp -ResourceGroupName $ResourceGroupName -Name $ApplicationName
$app.SiteConfig.AlwaysOn = $false
$app | Set-AzWebApp
```

- Using **Azure CLI**:
 1. Use **az web app config set**:
For example:
az webapp config set -g MyResourceGroup -n MyUniqueApp --always-on true

More information

- Configure an App Service app in the Azure portal
<https://docs.microsoft.com/en-us/azure/app-service/configure-common#configure-general-settings>

- Set-AzWebApp
<https://docs.microsoft.com/en-us/powershell/module/az.websites/set-azwebapp?view=azps-5.7.0#parameters>
- az webapp config
<https://docs.microsoft.com/en-us/cli/azure/webapp/config?view=azure-cli-latest#az-webapp-config-set>

App Service Advisor Recommendations

Description

Azure Advisor integrates recommendations for improving your App Service experience and discovering relevant platform capabilities. Examples of App Service recommendations are:

- Detection of instances where memory or CPU resources are exhausted by app runtimes, with mitigation options.
- Detection of instances where co-locating resources like web apps and databases can improve performance and reduce cost

Status of verified Web Apps

| Site name | Grade | Comments |
|---------------|-------------|-------------------------------------|
| flowi2 | Pass | No reports from App Service Advisor |

Solution

No actions needed.

Just keep reviewing periodically **App Service Advisory** recommendations:

- Using the **Azure Portal**:
 - Open the **Azure Portal**
 - Click on an **App Services**
 - Click on the Web App for which you want to review App Service Advisory recommendations
 - Under **Support + troubleshooting**, click on **App Service Advisor**

More information

- Improve the performance of Azure applications by using Azure Advisor - Improve App Service performance and reliability
<https://docs.microsoft.com/en-us/azure/advisor/advisor-performance-recommendations#improve-app-service-performance-and-reliability>
- Best Practices for Azure App Service
<https://docs.microsoft.com/en-us/azure/app-service/app-service-best-practices>

ARR Affinity Disabled Check (Recommendation. Not counted against the score)

Description

ARR Affinity creates sticky sessions so that clients will connect to the same app instance on subsequent requests. However, ARR Affinity can cause unequal distribution of requests between your instances and possibly overload an instance. For production apps that are aiming to be robust, it is recommended to set Always on to On and ARR Affinity to Off. Disabling ARR Affinity assumes that your application is either stateless, or the session state is stored on a remote service such as a cache or database.

Using ARR Affinity for a stateful application is not very reliable as instances could be restarted/replaced at any given time and that will make the client lose its session state.

We are not counting this against the score to account for those customers whose applications rely on ARR Affinity.

Status of verified Web Apps

| Site name | Grade | Comments |
|---------------|-------------|---|
| flowi2 | Fail | ARR Affinity enabled. With ARR Affinity enabled, a client is tied to a specific web worker resulting in unequal distribution of traffic across various worker instances |

Solution

Some applications need the client to be tied to a worker for them to work e.g.. Applications using In-Process session. If this is not the case, disable ARR Affinity to achieve a more even load distribution by following the steps below:

- To disable ARR Affinity:
 1. Open the **Azure Portal**
 2. Click on **App Services**
 3. Click on the Web App for which you want to disable ARR Affinity
 4. Under **Settings** click on **Configuration** then **General settings**.
 5. Set **ARR affinity** to **Off**

More information

- The Ultimate Guide to Running Healthy Apps in the Cloud - Set your Health Check path
<https://azure.github.io/AppService/2020/05/15/Robust-Apps-for-the-cloud.html#set-your-health-check-path>

Production SKU used

Description

Azure App Service brings together everything you need to create websites, mobile backends, and web APIs for any platform or device. Free and Shared (preview) plans provide different options to test your apps within your budget. Basic, Standard and Premium plans are for production workloads and run on dedicated Virtual Machine instances. Each instance can support multiple application and domains. The Isolated plan hosts your apps in a private, dedicated Azure environment and is ideal for apps that require secure connections with your on-premises network, or additional performance and scale. App Service plans are billed on a per second basis.

Status of verified Web Apps

| Site name | Grade | Comments |
|---------------|-------------|------------------------------|
| flowi2 | Pass | Using 3 (Premium0V3) workers |

Solution

No action is required. If you need to scale up other Web Apps use the following instructions:
For Web Apps that are not under a Production SKU, to scale up the App Service Plan using the Azure Portal:

- Using the **Azure Portal**:
 - Open the **Azure Portal**
 - Click on **App Service Plans**
 - Click on the App Service Plan hosting the Web App(s) you want to scale out
 - Under **Settings** click on **Scale up (App Service Plan)**
 - Select any plan under the **Production** tab

More information

- App Service pricing
<https://azure.microsoft.com/en-us/pricing/details/app-service/windows>