



Azure Configuration

Table of contents

Table of contents	2
Install Azure services	3
Azure CosmosDB	3
App Services	5
SQL Database	6
Deploy applications	7
KissLogBackend	7
KissLogFrontend	8
Troubleshooting	10

Install Azure services

Azure CosmosDB

Create new Azure CosmosDB Account, and specify the **API = Core (SQL)**

[Home](#) > [Azure Cosmos DB](#) >

Create Azure Cosmos DB Account

Basics Networking Encryption Tags **Review + create**

Basics

Subscription	RnD MPN Catalin Gavan
Resource Group	ResourceGroup-1
Location	West Europe
Account Name	(new) kisslog-cosmosdb
API	Core (SQL)
Account Type	Non-Production
Geo-Redundancy	Disable
Multi-region Writes	Disable
Availability Zones	Disable

Networking

Connectivity method	All networks
---------------------	--------------

Create

Previous

Next

[Download a template for automation](#)


After the CosmosDB Account has been created, create a new Database.

Please select:

Provision throughput = true

Throughput = 1000

New Database ×

 Start at \$24/mo per database, multiple containers included
[More details](#)

* Database id ⓘ

Provision throughput ⓘ

* Throughput (400 - 100,000 RU/s) ⓘ

Autoscale Manual

Estimated cost (USD): **\$0.080 hourly / \$1.92 daily / \$58.40 monthly**
(1 region, 1000RU/s, \$0.00008/RU)

App Services

Create two App Services, one for **KissLogBackend** application, the second for **KissLogFrontend** application.

Both of these applications are built on .NET Core 3.1.

Please select

Publish = Code

Runtime stack = .NET Core 3.1

Operating System = Windows

SQL Database

Create one SQL Database.

Please select

Compute + storage = Basic 2 GB

KissLog does not use SQL intensively, and you can start with a small server configuration. If necessary, you can upgrade it later.

Deploy applications

Once the Azure services have been installed, you can deploy the KissLog applications (App Services).

KissLog.Backend AppService must be deployed first.

KissLogBackend

1. Before deploying the KissLogBackend app, please update the Configuration\KissLog.json file.

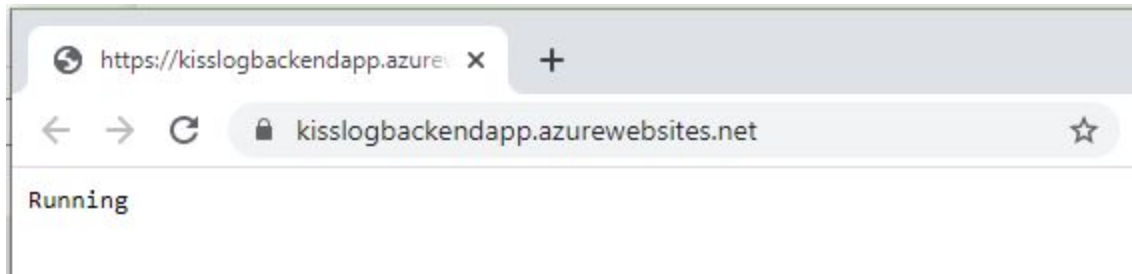
Replace the values highlighted with appropriate keys from your Azure services.

```
{
  "IsReadOnlyMode": false,
  "KissLogBackendUrl": "https://kisslogBackend.azurewebsites.net",
  "KissLogFrontendUrl": "https://kisslogFrontend.azurewebsites.net",
  "Database": {
    "Provider": "AzureCosmosDb",
    "AzureCosmosDb": {
      "AccountEndpoint": "https://kisslog-cosmosdb.documents.azure.com:443/",
      "AccountKey": "1Lv9A9qp2HxCaZPhyY20CimHPj1bpMsow==",
      "DatabaseName": "KissLog"
    }
  },
  "CreateRequestLog": ...,
}
```

2. After the Configuration\KissLog.json file has been updated, upload the application contents to KissLogBackend AppService application.

3. After the application has been uploaded, load it (by accessing root url “/”).

If everything went successfully, you will receive a “200 OK Running” message.



KissLogFrontend

1. Before deploying the KissLogFrontend app, please update the Configuration\KissLog.json file.

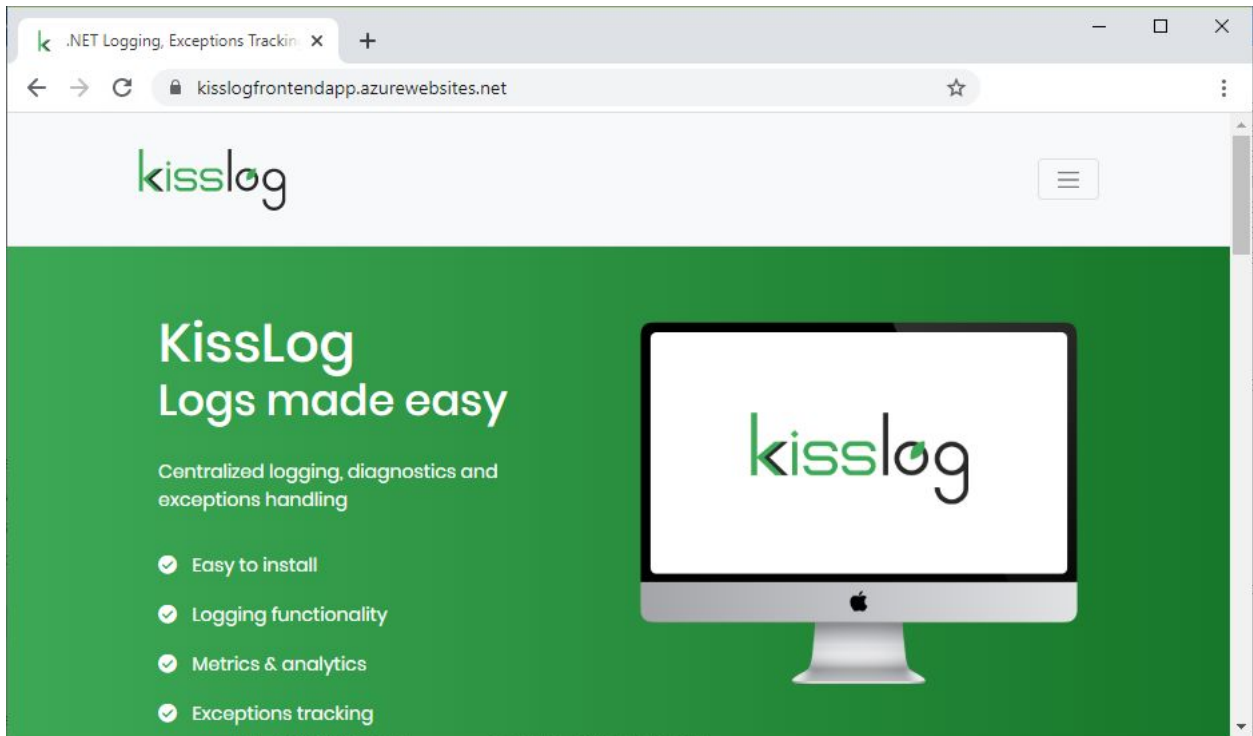
Replace the values highlighted with appropriate keys from your Azure services.

```
{
  "ApplicationName": "KissLog",
  "KissLogBackendUrl": "https://kisslogbackendapp.azurewebsites.net",
  "KissLogFrontendUrl": "https://kisslogfrontendapp.azurewebsites.net",
  "KissLogOrganizationId": "200c5a1b-5efc-48a8-8a20-5223e22a4437",
  "KissLogApplicationId": "a2f9f32b-a580-42fb-85c0-eeed4055e6fb",
  "Database": {
    "Provider": "SqlServer",
    "KissLogDbContext": "Server=tcp:kisslogsql.database.windows.net,1433;Initial Catalog=KissLogDb;Persist Security Info=False;User ID={user};",
  },
  "Authorization": "...",
}
```

2. After the Configuration\KissLog.json file has been updated, upload the application contents to KissLogFrontend AppService application.

3. After the application has been uploaded, load it (by accessing root url "/").

If everything went successfully, you will see the Home page.



Troubleshooting

If any of the applications fail to run, check, for each application, the logs located in the **Logs** folder.

Quick checklist:

1. Make sure you deploy and run KissLogBackend first
2. Running KissLogBackend for the first time will also create the necessary CosmosDB containers
3. On the first run, KissLogFrontend will try to connect to the specified SQL Server, and will try to create the required Tables
4. A common error can be caused by the SQL Server Firewall which prevents the connection from KissLogFrontend AppService
5. KissLogFrontend will also try to connect to KissLogBackend (using an HTTP request)