

DB Creation process breakdown via Azure Portal A-Z steps:

Last updated by | Vitor Tomaz | Aug 5, 2020 at 12:43 PM PDT

Friday, February 23, 2018

12:49 AM

What happen when create/restore Azure SQL DB through Portal, to understand what is going behind the scene, below detailed steps explain each layer how this process works:

1. When creating a database through the portal there are multiple systems and layers involved.

1. The **Portal**,
 - a. This is the website portal.azure.com that you interact with
2. The **ARM** front door (or ARM for short)
 - a. This is a layer between the client (portal) and the backend services (Azure SQL). The purpose of this layer is to do caching of resources, authentication, authorization, and routing of requests.
 - b. **Deployment Engine**
 - i. This is a subservice in ARM that orchestrates creating resources
3. The **Azure SQL** service.
 - a. This is the backend service that actually creates the database.

2. When a user issues a request to create a database in the portal the following happens:

1. Portal creates a deployment template with specified parameters to create the database and sends it to the deployment engine (a part of ARM).
2. ARM receives the request from the client (portal) and does a bunch of checks:
 - a. Is the user Authenticated (logged-in properly)
 - b. Is the user Authorized to perform the action. This is where RBAC comes in. ARM will validate that the client has the permissions Microsoft.Sql/servers/databases/write. If the client has these permissions the operation will continue, otherwise, it gets rejected.
 - c. The Deployment Engine in ARM queues up the work and submits the create request to the Azure SQL service.
3. Azure SQL receives the request and starts processing it.
 - a. Once the operation has been successfully queued the service responds back to ARM with an "Operation Accepted" message indicating that the operation has passed validation and is queued to be completed.
 - b. The Service additionally adds a HTTP Header "AzureAsyncReponse" which contains a URL for the client to use to get the status of the operation. In the case the client becomes the Deployment Engine in ARM.
4. ARM receives the "Operation Accepted" response from the Azure SQL service and starts polling.
 - a. The Deployment Engine uses the url in the "AzureAsyncOperation" header to poll for the status of the operation.
5. ARM Polls until the operation has finished.

- a. Polling for the operation status is a new request and so ARM goes through the process of authorizing the request for the user that initiated the operation. In this case the operation is Microsoft.Sql/servers/databases/azureAsyncOperation/read. If the user has this permission then the polling continues until the operation completes. If the user does not have access to this operation then polling will fail with an Unauthorized error.
 - b. If polling fails then ARM does not know the outcome of the operation and will keep retrying until it reaches its polling limit of 2 days.
6. ARM gets notified when Operation has completed
- a. When the polling returns a status indicating that the operation has completed ARM will add the resource to its Cache.
 - b. The deployment engine will signal to the portal that the operation has completed.

3. When a user requests to see a database in the portal the following happens:

1. Portal sends the GET request to ARM
2. ARM checks the users RBAC permissions to ensure that the account is allowed to see the resource
3. ARM checks its cache to see if the resource exists.
 - a. If the resource does not exist in the ARM cache then ARM will respond to the portal saying the resource doesn't exist.
4. ARM sends a request to the Azure SQL service
5. Azure SQL responds with the details of the database.

How good have you found this content?

