# Automate DocumentDB account creation using Azure Resource Manager templates and Azure CLI

[AZURE.SELECTOR] - [Azure Portal](/documentation/articles/documentdb-create-account) - [Azure CLI and ARM](/documentation/articles/documentdb-automation-resource-manager-cli)

This article shows you how to create a DocumentDB account by using Azure Resource Manager templates or the Azure Command-Line Interface (CLI). To create a DocumentDB account using the Azure portal, see [Create a DocumentDB database account using the Azure portal](/documentation/articles/documentdb-create-account).

* [Create a DocumentDB account using CLI](#quick-create-documentdb-account)
* [Create a DocumentDB account using an ARM template](#deploy-documentdb-from-a-template)

DocumentDB database accounts are currently the only DocumentDB resource that can be created using ARM templates and the Azure CLI.

## Getting ready

Before you can use the Azure CLI with Azure resource groups, you need to have the right Azure CLI version and an Azure account. If you don’t have the Azure CLI, [install it](/documentation/articles/xplat-cli-install).

### Update your Azure CLI version

At the command prompt, type azure --version to see whether you have already installed version 0.9.11 or later.

azure --version  
0.9.11 (node: 0.12.7)

If your version is not 0.9.11 or later, you need to either [install the Azure CLI](/documentation/articles/xplat-cli-install) or update by using one of the native installers, or through **npm** by typing npm update -g azure-cli to update or npm install -g azure-cli to install.

### Set your Azure account and subscription

If you don’t already have an Azure subscription but you do have a Visual Studio subscription, you can activate your [Visual Studio subscriber benefits](/pricing/member-offers/msdn-benefits-details/). Or you can sign up for a [free trial](/pricing/free-trial/).

You need to have a work or school account or a Microsoft account identity to use Azure resource management templates. If you have one of these accounts, type the following command.

azure login

Which produces the following output:

info: Executing command login  
|info: To sign in, use a web browser to open the page https://aka.ms/devicelogin.   
Enter the code E1A2B3C4D to authenticate. If you're signing in as an Azure  
AD application, use the --username and --password parameters.

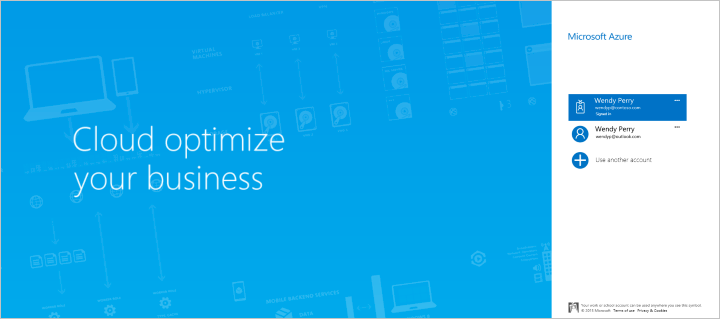
[AZURE.NOTE] If you don’t have an Azure account, you’ll see an error message indicating that you need a different type of account. To create one from your current Azure account, see [Creating a work or school identity in Azure Active Directory](/documentation/articles/virtual-machines-windows-create-aad-work-id).

Open <https://aka.ms/devicelogin> in a browser and enter the code provided in the command output.



Screenshot showing the device login screen for Microsoft Azure CLI

Once you’ve entered the code, select the identity you want to use in the browser and provide your user name and password if needed.



Screenshot showing where to select the Microsoft identity account associated with the Azure subscription you want to use

You’ll receive the following confirmation screen when you’re successfully logged in, and you can then close the browser window.



Screenshot showing confirmation of login to the Microsoft Azure Cross-platform Command Line Interface

The command shell also provides the following output.

/info: Added subscription Visual Studio Ultimate with MSDN  
info: Setting subscription "Visual Studio Ultimate with MSDN" as default  
+  
info: login command OKK

In addition to the interactive login method described here, there are additional Azure CLI login methods available. For more information about the other methods and information about handling multiple subscriptions, see [Connect to an Azure subscription from the Azure Command-Line Interface (Azure CLI)](/documentation/articles/xplat-cli-connect).

### Switch to the Azure CLI resource group mode

By default, the Azure CLI starts in the service management mode (**asm** mode). Type the following to switch to resource group mode.

azure config mode arm

Which provides the following output:

info: Executing command config mode  
info: New mode is arm  
info: config mode command OK

You can switch back to the default set of commands by typing azure config mode asm.

## Task: Create a DocumentDB account using Azure CLI

Use the instructions in this section to create a DocumentDB account with Azure CLI.

### Step 1: Create or retrieve your resource group

In order to create a DocumentDB account, you first need a resource group. If you already know the name of the resource group that you’d like to use, then skip to [Step 2](#create-documentdb-account-cli).

To review a list of all of your current resource groups, run the following command and take note of the resource group name you’d like to use:

azure group list

To create a new resource group, run the following command, specify the name of the new resource group to create, and the region in which to create the resource group:

azure group create <resourcegroupname> <resourcegrouplocation>

* <resourcegroupname> can only use alphanumeric characters, periods, underscores, the ‘-’ character, and parenthesis and cannot end in a period.
* <resourcegrouplocation> must be one of the regions in which DocumentDB is generally available. The current list of regions is provided on the [Azure Regions page](https://azure.microsoft.com/regions/#services).

Example input:

azure group create new\_res\_group westus

Which produces the following output:

info: Executing command group create  
+ Getting resource group new\_res\_group  
+ Creating resource group new\_res\_group  
info: Created resource group new\_res\_group  
data: Id: /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/new\_res\_group  
data: Name: new\_res\_group  
data: Location: westus  
data: Provisioning State: Succeeded  
data: Tags: null  
data:  
info: group create command OK

If you encounter errors, see [Troubleshooting](#troubleshooting).

### Step 2: Create a DocumentDB account using CLI

Create a DocumentDB account in the new or existing resource group by entering the following command at the command prompt:

[AZURE.TIP] If you run this command in Azure PowerShell or Windows PowerShell you will receive an error about an unexpected token. Instead, run this command at the Windows Command Prompt.

azure resource create -g <resourcegroupname> -n <databaseaccountname> -r "Microsoft.DocumentDB/databaseAccounts" -o "2015-04-08" -l <databaseaccountlocation> -p "{\"databaseAccountOfferType\":\"Standard\"}"

* <resourcegroupname> can only use alphanumeric characters, periods, underscores, the ‘-’ character, and parenthesis and cannot end in a period.
* <databaseaccountname> can only use lowercase letters, numbers, the ‘-’ character, and must be between 3 and 50 characters.
* <databaseaccountlocation> must be one of the regions in which DocumentDB is generally available. The current list of regions is provided on the [Azure Regions page](/regions/#services).

Example input:

azure resource create -g new\_res\_group -n samplecliacct -r "Microsoft.DocumentDB/databaseAccounts" -o 2015-04-08 -l westus -p "{\"databaseAccountOfferType\":\"Standard\"}"

Which produces the following output as your new account is provisioned:

info: Executing command resource create  
+ Getting resource samplecliacct  
+ Creating resource samplecliacct  
info: Resource samplecliacct is updated  
data:  
data: Id: /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/new\_res\_group/providers/Microsoft.DocumentDB/databaseAccounts/samplecliacct  
data: Name: samplecliacct  
data: Type: Microsoft.DocumentDB/databaseAccounts  
data: Parent:  
data: Location: West US  
data: Tags:  
data:  
info: resource create command OK

If you encounter errors, see [Troubleshooting](#troubleshooting).

After the command returns, the account will be in the **Creating** state for a few minutes, before it changes to the **Online** state in which it is ready for use. You can check on the status of the account in the [Azure portal](https://portal.azure.com), on the **DocumentDB Accounts** blade.

## Task: Create a DocumentDB account using an ARM template

Use the instructions in this section to create a DocumentDB account with an Azure resource manager (ARM) template and an optional parameters file, both of which are JSON files. Using a template enables you to describe exactly what you want and repeat it without errors.

### Understanding ARM templates and resource groups

Most applications are built from a combination of different resource types (such as one or more DocumentDB account, storage accounts, a virtual network, or a content delivery network). The default Azure service management API and the Azure portal represented these items by using a service-by-service approach. This approach requires you to deploy and manage the individual services individually (or find other tools that do so), and not as a single logical unit of deployment.

*Azure Resource Manager templates* make it possible for you to deploy and manage these different resources as one logical deployment unit in a declarative fashion. Instead of imperatively telling Azure what to deploy one command after another, you describe your entire deployment in a JSON file – all of the resources and associated configuration and deployment parameters – and tell Azure to deploy those resources as one group.

You can learn lots more about Azure resource groups and what they can do for you in the [Azure Resource Manager overview](/documentation/articles/resource-group-overview). If you’re interested in authoring templates, see [Authoring Azure Resource Manager templates](/documentation/articles/resource-group-authoring-templates).

### Step 1: Create a template and parameter file

Create a local template file with the following content. Name the file azuredeploy.json.

{  
 "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",  
 "contentVersion": "1.0.0.0",  
 "parameters": {  
 "databaseAccountName": {  
 "type": "string"  
 }  
 },  
 "variables": { },  
 "resources": [  
 {  
 "apiVersion": "2015-04-08",  
 "type": "Microsoft.DocumentDb/databaseAccounts",  
 "name": "[parameters('databaseAccountName')]",  
 "location": "[resourceGroup().location]",  
 "properties": {  
 "name": "[parameters('databaseAccountName')]",  
 "databaseAccountOfferType": "Standard"  
 }  
 }  
 ]  
}

This template requires only one parameter, the database account name to create. The location of the new database account is set to the same location as the resource group.

Because the template only takes one parameter, you can either enter the value at the command line, or create a parameter file to specify the value.

To create a parameters file, copy the following content into a new file and name the file azuredeploy.parameters.json. If you plan on specifying the database account name at the command prompt, you can continue without creating this file.

{  
 "$schema": "http://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",  
 "contentVersion": "1.0.0.0",  
 "parameters": {  
 "databaseAccountName": {  
 "value": "samplearmacct"  
 }  
 }  
}

In the azuredeploy.parameters.json file, update the value “samplearmacct” to the database name you’d like to use, then save the file. <databaseAccountName> can only use lowercase letters, numbers, the ‘-’ character, and must be between 3 and 50 characters.

### Step 2: Create or retrieve your resource group

In order to create a DocumentDB account, you first need a resource group. If you already know the name of the resource group that you’d like to use, ensure that the location is a [region where DocumentDB is generally available](/regions/#services), then skip to [Step 3](#create-account-from-template). In the template, the location of the account is created in the same region as the resource group, so attempting to create an account in a region where DocumentDB is not available will result in a deployment error.

To review a list of all of your current resource groups, run the following command and take note of the resource group name you’d like to use:

azure group list

To create a new resource group, run the following command, specify the name of the new resource group to create, and the region in which to create the resource group:

azure group create <resourcegroupname> <databaseaccountlocation>

* <resourcegroupname> can only use alphanumeric characters, periods, underscores, the ‘-’ character, and parenthesis and cannot end in a period.
* <databaseaccountlocation> must be one of the regions in which DocumentDB is generally available. The current list of regions is provided on the [Azure Regions page](/regions/#services).

Example input:

azure group create new\_res\_group westus

Which produces the following output:

info: Executing command group create  
+ Getting resource group new\_res\_group  
+ Creating resource group new\_res\_group  
info: Created resource group new\_res\_group  
data: Id: /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/new\_res\_group  
data: Name: new\_res\_group  
data: Location: West US  
data: Provisioning State: Succeeded  
data: Tags: null  
data:  
info: group create command OK

If you encounter errors, see [Troubleshooting](#troubleshooting).

### Step 3: Create the DocumentDB account by using an ARM template

To create a DocumentDB account in your resource group, run the following command and provide the path to the template file, the path to the parameter file or the parameter value, the name of the resource group in which to deploy, and a deployment name (-n is optional).

To use a parameter file:

azure group deployment create -f <PathToTemplate> -e <PathToParameterFile> -g <resourcegroupname> -n <deploymentname>

* <PathToTemplate> is the path to the azuredeploy.json file created in step 1. If your path name has spaces in it, put double quotes around this parameter.
* <PathToParameterFile> is the path to the azuredeploy.parameters.json file created in step 1. If your path name has spaces in it, put double quotes around this parameter.
* <resourcegroupname> is the name of the existing resource group in which to add a DocumentDB database account.
* <deploymentname> is the optional name of the deployment.

Example input:

azure group deployment create -f azuredeploy.json -e azuredeploy.parameters.json -g new\_res\_group -n azuredeploy

OR to specify the database account name parameter without a parameter file, and instead get prompted for the value, run the following command:

azure group deployment create -f <PathToTemplate> -g <resourcegroupname> -n <deploymentname>

Example input which shows the prompt and entry for a database account named new\_db\_acct:

azure group deployment create -f azuredeploy.json -g new\_res\_group -n azuredeploy  
info: Executing command group deployment create  
info: Supply values for the following parameters  
databaseAccountName: samplearmacct

As the account is provisioned, you will receive the following information:

info: Executing command group deployment create  
+ Initializing template configurations and parameters  
+ Creating a deployment  
info: Created template deployment "azuredeploy"  
+ Waiting for deployment to complete  
data: DeploymentName : azuredeploy  
data: ResourceGroupName : new\_res\_group  
data: ProvisioningState : Succeeded  
data: Timestamp : 2015-11-30T18:50:23.6300288Z  
data: Mode : Incremental  
data: Name Type Value  
data: ------------------- ------ ------------------  
data: databaseAccountName String samplearmacct  
data: location String West US  
info: group deployment create command OK

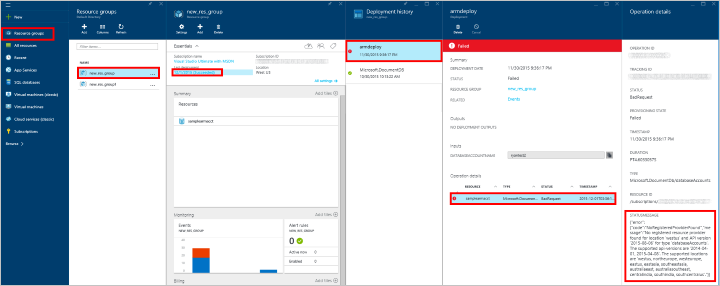
If you encounter errors, see [Troubleshooting](#troubleshooting).

After the command returns, the account will be in the **Creating** state for a few minutes, before it changes to the **Online** state in which it is ready for use. You can check on the status of the account in the [Azure portal](https://portal.azure.cn), on the **DocumentDB Accounts** blade.

## Troubleshooting

If you receive errors like Deployment provisioning state was not successful while creating your resource group or database account, you have a few troubleshooting options.

[AZURE.NOTE] Providing incorrect characters in the database account name or providing a location in which DocumentDB is not available will cause deployment errors. Database account names can only use lowercase letters, numbers, the ‘-’ character, and must be between 3 and 50 characters. All valid database account locations are listed on the [Azure Regions page](/regions/#services).

* If your output contains the following Error information has been recorded to C:\Users\wendy\.azure\azure.err, then review the error info in the azure.err file.
* You may find useful info in the log file for the resource group. To view the log file, run the following command:
* azure group log show <resourcegroupname> --last-deployment
* Example input:
* azure group log show new\_res\_group --last-deployment
* Then see [Troubleshooting resource group deployments in Azure](../resource-manager-troubleshoot-deployments-cli.md) for additional information.
* Error information is also available in the Azure Portal as shown in the following screenshot. To navigate to the error info: click Resource Groups in the Jumpbar, select the Resource Group that had the error, then in the Essentials area of the Resource group blade click the date of the Last Deployment, then in the Deployment history blade select the failed deployment, then in the Deployment blade click the Operation detail with the red exclamation mark. The Status Message for the failed deployment is displayed in the Operation details blade.
* 
* Screenshot of the Azure portal showing how to navigate to the deployment error message

## Next steps

Now that you have a DocumentDB account, the next step is to create a DocumentDB database. You can create a database by using one of the following:

* The Azure portal, as described in [Create a DocumentDB database using the Azure portal](/documentation/articles/documentdb-create-database).
* The C# .NET samples in the [DatabaseManagement](https://github.com/Azure/azure-documentdb-net/tree/master/samples/code-samples/DatabaseManagement) project of the [azure-documentdb-dotnet](https://github.com/Azure/azure-documentdb-net/tree/master/samples/code-samples) repository on GitHub.
* The [DocumentDB SDKs](https://msdn.microsoft.com/library/azure/dn781482.aspx). DocumentDB has .NET, Java, Python, Node.js, and JavaScript API SDKs.

After creating your database, you need to [add one or more collections](/documentation/articles/documentdb-create-collection) to the database, then [add documents](/documentation/articles/documentdb-view-json-document-explorer) to the collections.

After you have documents in a collection, you can use [DocumentDB SQL](/documentation/articles/documentdb-sql-query) to [execute queries](documentdb-sql-query.md#executing-queries) against your documents by using the [Query Explorer](/documentation/articles/documentdb-query-collections-query-explorer) in the portal, the [REST API](https://msdn.microsoft.com/library/azure/dn781481.aspx), or one of the [SDKs](https://msdn.microsoft.com/library/azure/dn781482.aspx).

To learn more about DocumentDB, explore these resources:

* [Learning path for DocumentDB](https://azure.microsoft.com/documentation/learning-paths/documentdb/)
* [DocumentDB resource model and concepts](/documentation/articles/documentdb-resources)

For more templates you can use, see [Azure Quickstart templates](/documentation/templates/).