# FETCH CLIMATE 2 DEPLOYMENT GUIDE

## Table of Contents

# Overview

A typical Fetch Climate 2 installation consists of two cloud services: the **FetchClimate service** and the **Storage service**.

**FetchClimate service:**

- Provides a web interface to environmental data.
- Provides a REST API for programmatic data access.
- Hosts computation roles.
- Manages the queue of requests.

**Storage service:**

- Stores raw environmental data as Dmitrov [1] datasets consumed by the FetchClimate service.

There are three types of users who work with a FetchClimate installation.

**Administrator (IT skilled person):**

- Has Microsoft Azure subscription credentials.
- Works with Azure management tools and websites.
- Deploys the FetchClimate and Storage services.
- Manages storage accounts and SQL databases.

**Operator:**

- Has storage account keys and the SQL database password.
- Works with the **fetchconfig.exe** utility.
- Uploads and updates data in the Storage service.
- Configures environmental variables and data sources.

**User:**

- Explores data using the web interface and the **fetchclimate.exe** utility.
- Queries data programmatically using the REST API and the API in the **fetchclimate.exe** assembly.

**The Fetch Climate 2 deployment process has six steps:**

1. Creating Azure Blob Storage and the Azure SQL database (performed by the Administrator).
2. Deploying the FetchClimate service (performed by the Administrator).
3. Deploying the Storage service (performed by the Administrator).
4. Configuring FetchClimate variables and data sources (performed by the Operator).
5. Installing FetchClimate (performed by the Operator).
6. Verifying the FetchClimate deployment (performed by the Operator).

The following sections describe the steps in more detail.

## Prerequisites

Before you start, make sure you have the following.

1) **Distribution packages**:
   - FetchClimate_ServicePacks.zip

     **Download URL:** http://research.microsoft.com/en-us/downloads/dd32af78-27e0-412d-8122-d62c059f5e18/default.aspx
   - FetchClient_ClientTools_vX.X.X.X.zip (X.X.X.X is actual version)
     **Download URL:** http://research.microsoft.com/en-us/downloads/8eea0db4-05f0-4760-ac6c-312840ac1dcf/
2) **Azure resources:**
   - Microsoft Azure subscription with at least two storage accounts, two SQL databases, and seven CPU cores available.
3) **Additional software to install:**
   - Dmitrov: Scientific Data-Set library and tools

     **Download URL:** http://research.microsoft.com/en-us/um/cambridge/groups/science/tools/dmitrov/default.htm

## Step 1. Create the Azure Storage and Database

Both the FetchClimate service and the Storage service require blob storage and a SQL database.

FetchClimate stores active requests and server-side cache in blob storage. A current list of environmental variables and their bindings to data sources are stored in a SQL database.

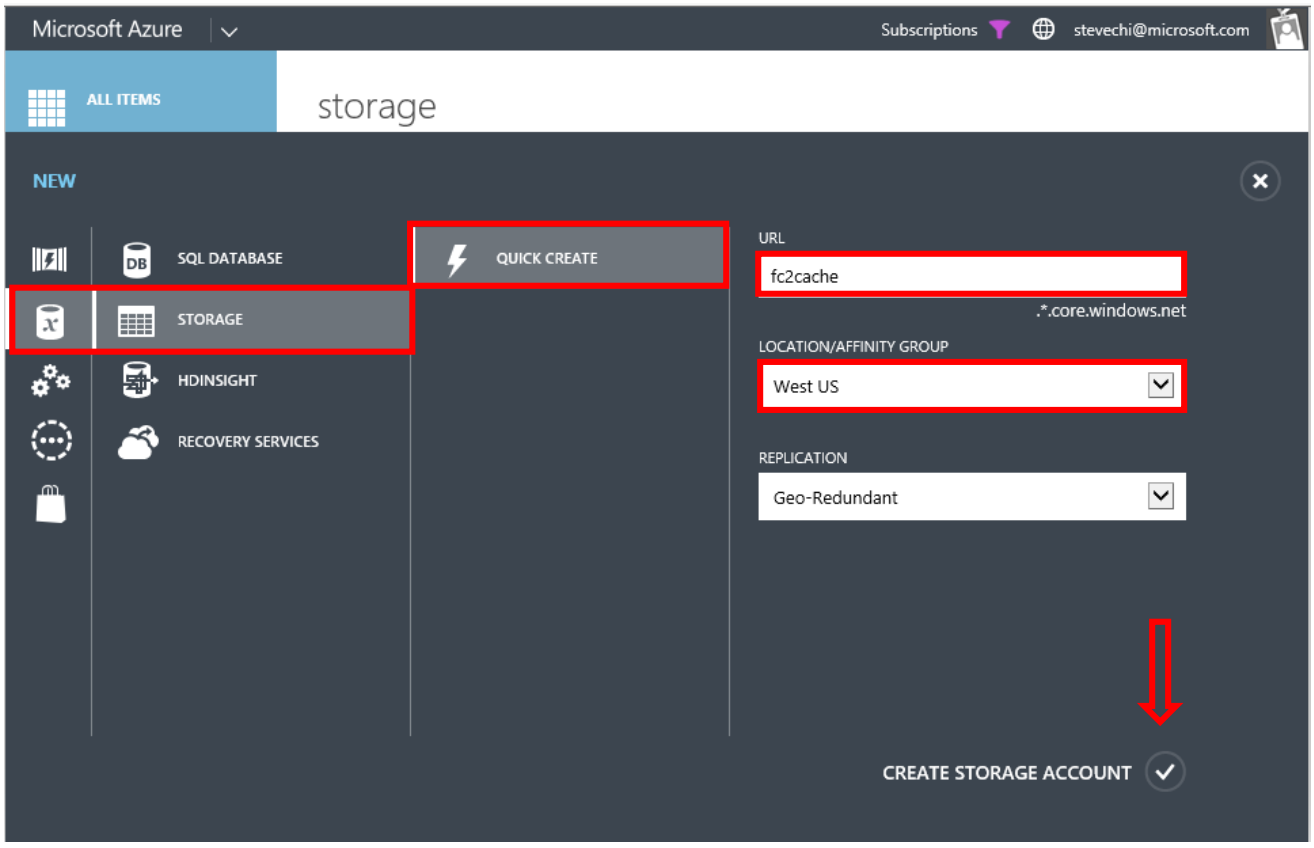
The Storage service uses blob storage for chunks of the raw data arrays, and the SQL database stores the scheme.

Conflicts will occur if you are sharing the same storage account or SQL database for creating multiple instances of the FetchClimate and Storage services. For maintenance, it is better to have separate accounts.

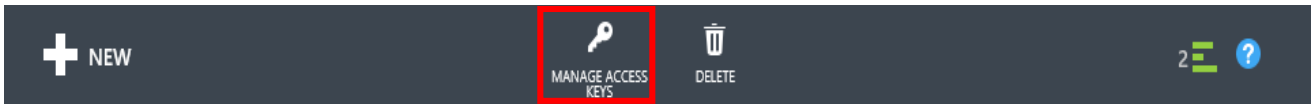
### 1.1. Create the Azure storage account

1. Sign in to the Microsoft Azure Management Portal: **http://manage.windowsazure.com**
2. Click **NEW** in the lower-left corner.
3. Choose **DATA SERVICES** > **STORAGE** > **QUICK CREATE**.

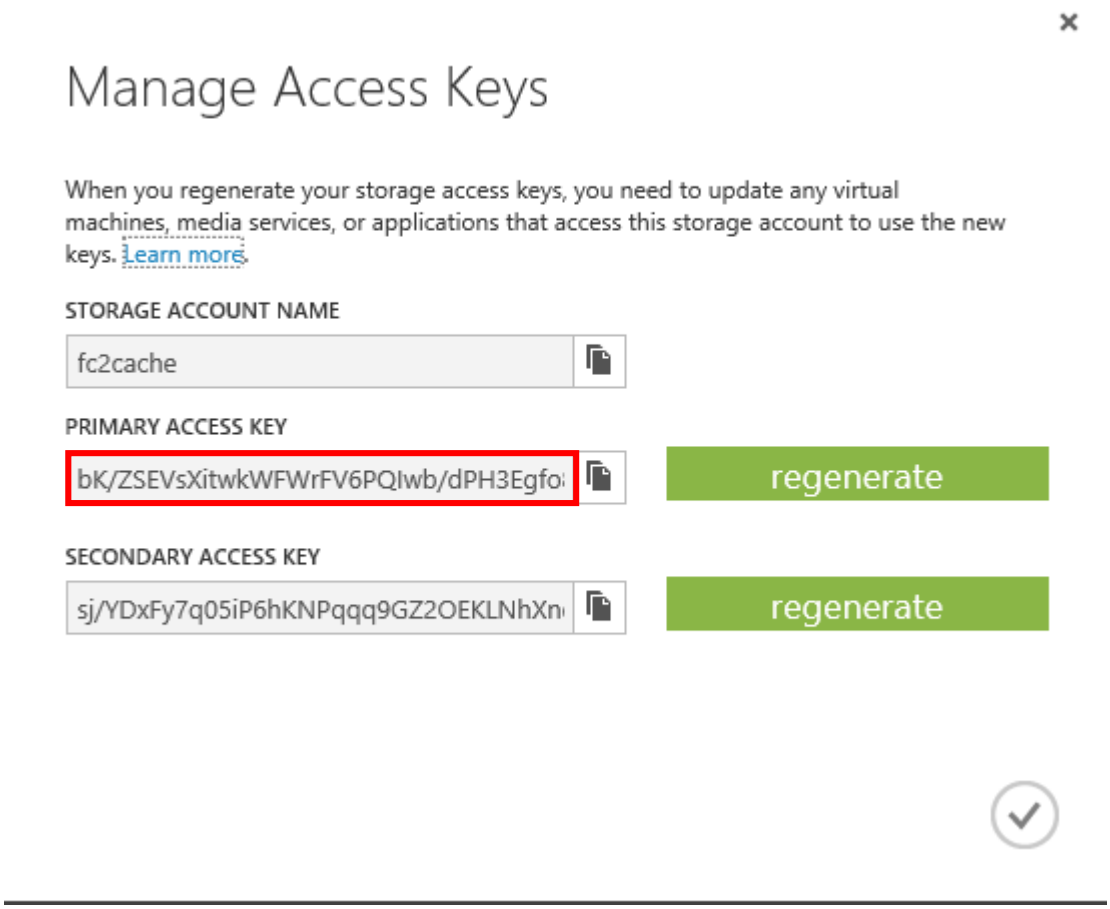
4. Provide a URL for the storage, and set the storage region. The optimal storage region should be geographically close to your location.




5. After the storage account is created select the storage account you just created, in this case, *f2cache*; next click **MANAGE ACCESS KEYS**.



6. Copy the storage account name and primary access key (in Notepad or in a secure place for use during cloud service configuration).




7. Repeat steps 2–6 to create a second storage account.

**Note:** In this document we will refer to the created storage account names as:

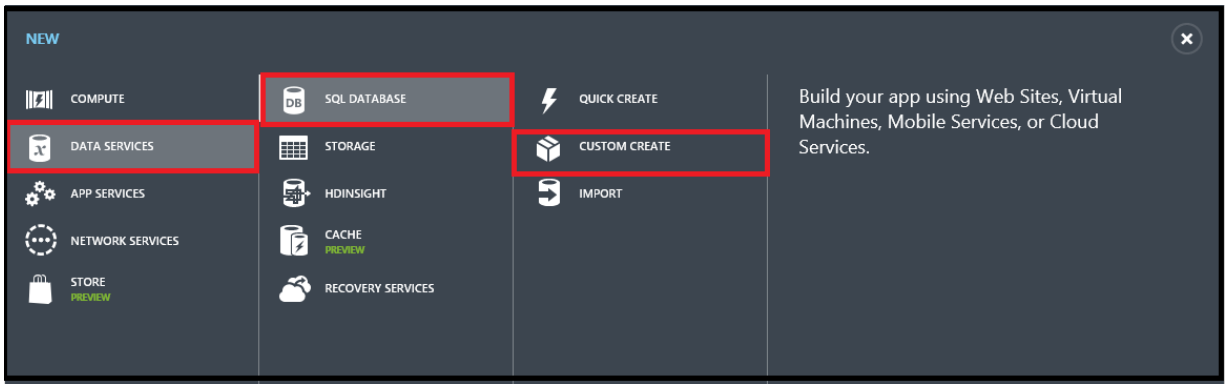
***<fc2cache>*** with an access key of ***<CACHEKEY>***
***<fc2data>*** with an access key of ***<DATAKEY>***
You may use any names you would like. Please use the storage accounts and the access keys that you create.

## 1.2. Create the SQL Azure database

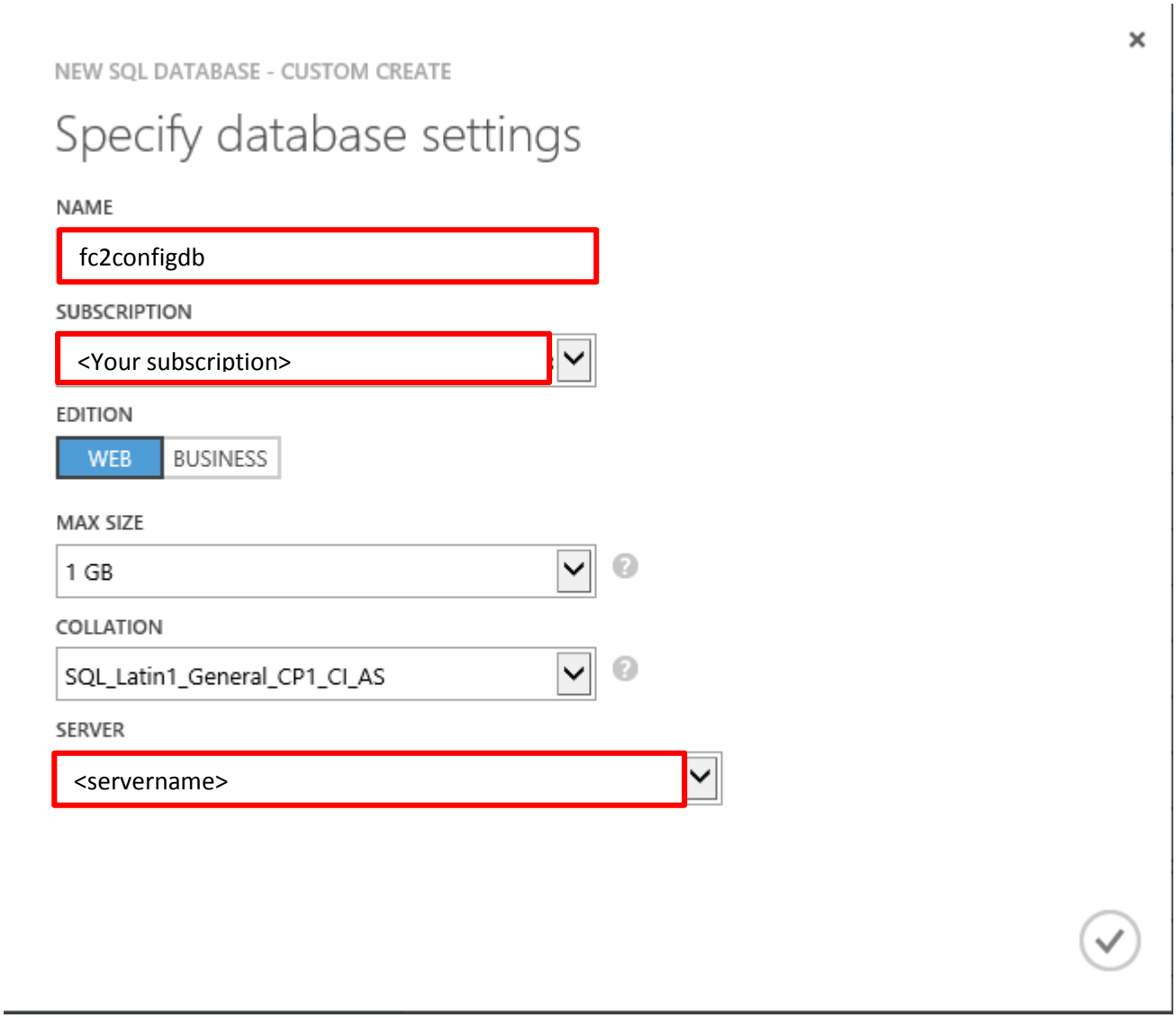
If you do not have an Azure SQL server please see Appendix A for guidance in creating one.

1. In the Azure Management Portal, click **NEW** in the lower-left corner.

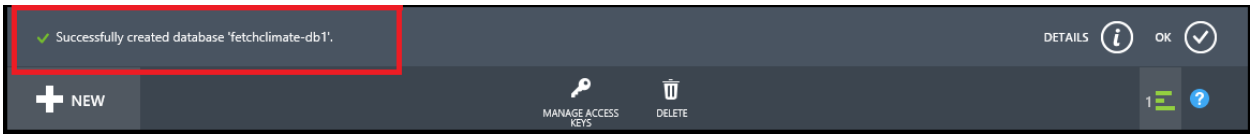
2. Choose **DATA SERVICES** > **SQL DATABASE** > **CUSTOM CREATE**.




3. Provide the name of the database, select the server name, and specify other options if needed.

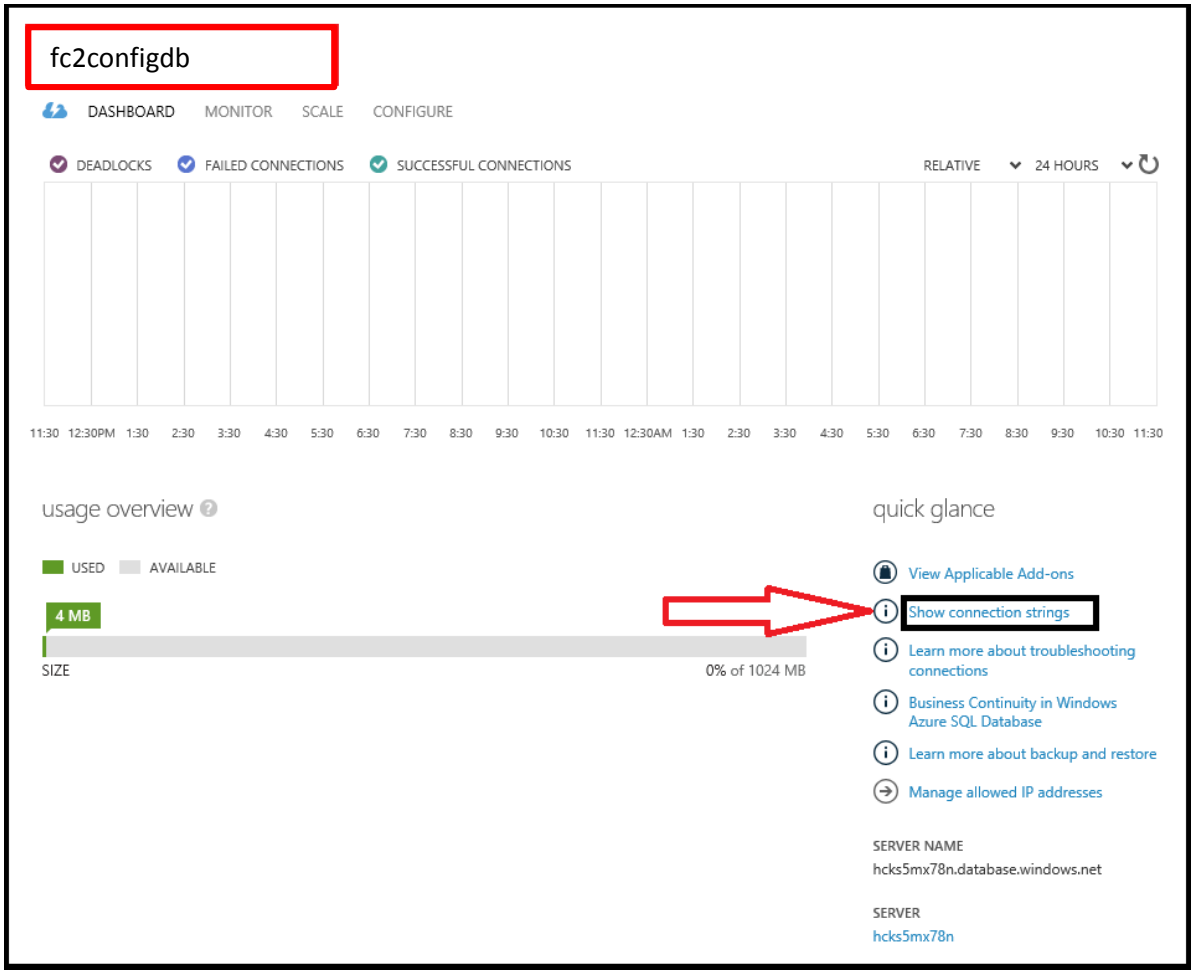


4. Click **OK** to confirm the creation of the new database.

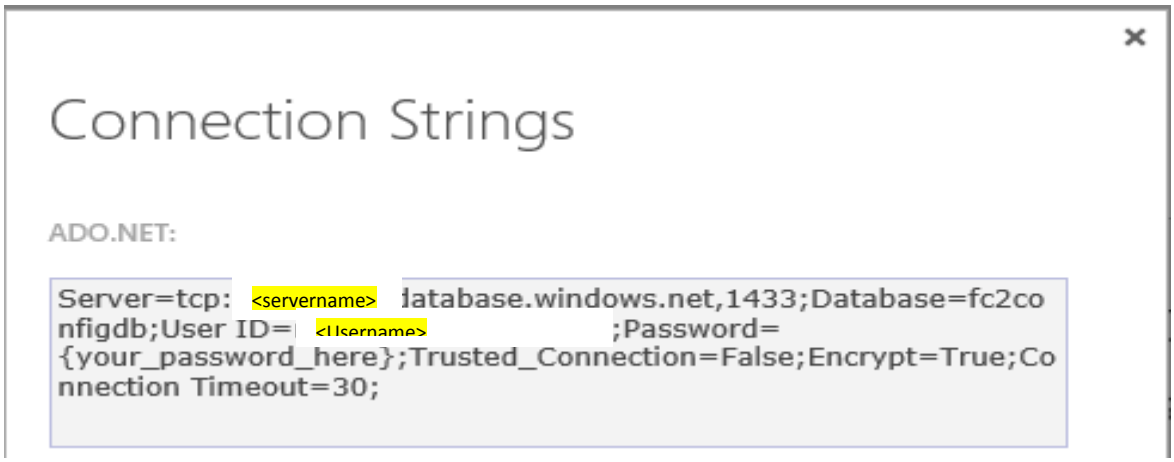



5. After the database is created, click the SQL databases in the left panel. Double-click the newly created database, and then click **DASHBOARD**.

6. Click **Show connection strings**.



7. Copy the ADO.NET connection string to Notepad or in a secure place for use during cloud service configuration.




**Note:** In this example, replace *{your_password_here}* with the password you use for accessing SQL. The password will be according to which server you chose, which you should know already.

**For example:**

```
Server=tcp:<servername>.database.windows.net,1433;Database=fc2_db1;User
ID=<UserName>;Password=<Password>;Trusted_Connection=False;Encrypt=True;Co
nnection Timeout=30;
```

8. Repeat steps 1–7 to create a second SQL database.

**Note:** In this document, we refer to the connection string for the created databases as ***<fc2configdb>*** and ***<fc2storagedb>*.** However, while following the steps, replace them with your two created connection strings.

## Step 2. Deploy the FetchClimate Service

1. Unzip FetchClimate_ServicePacks.zip; then, open the **FetchClimateServicePackage_v2.0.19885.0** folder.

   Two files are in the folder: **ServiceConfiguration.Cloud.cscfg** and **AzureDeployment.cspkg**.

2. In the ServiceConfiguration.Cloud.cscfg file, replace the placeholders (marked in yellow) with your created account name, account key, and connection string, and save the file.

   **Note:** We recommend setting <Instances count="1"> for the initial deployment. If the new site becomes busy you can redeploy with more instances or increase the instances via the Azure Management portal.

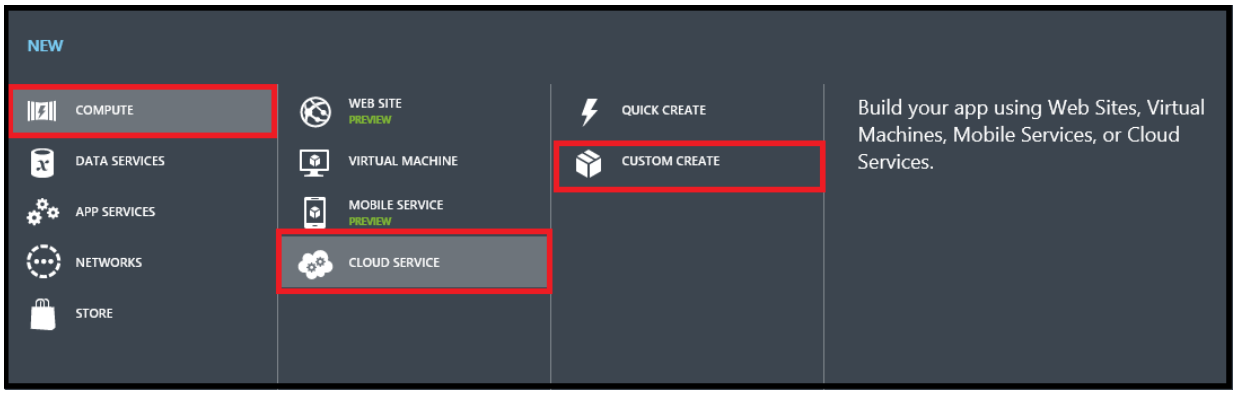
   Examples:
   **CACHEKEY**
   bK/ZSEVsXitwkWFWrFV6PQIwb/dPH3EgfoMoK5pz3Tj8aMo2UTVCE3J1k/XfxWgS/thGI+eL+w==

   fc2configdb
   Server=tcp:<your server>.database.windows.net,1433;Database=fc2configdb;User ID=<Your username>;Password=<Your Password>;Trusted_Connection=False;Encrypt=True;Connection Timeout=30;

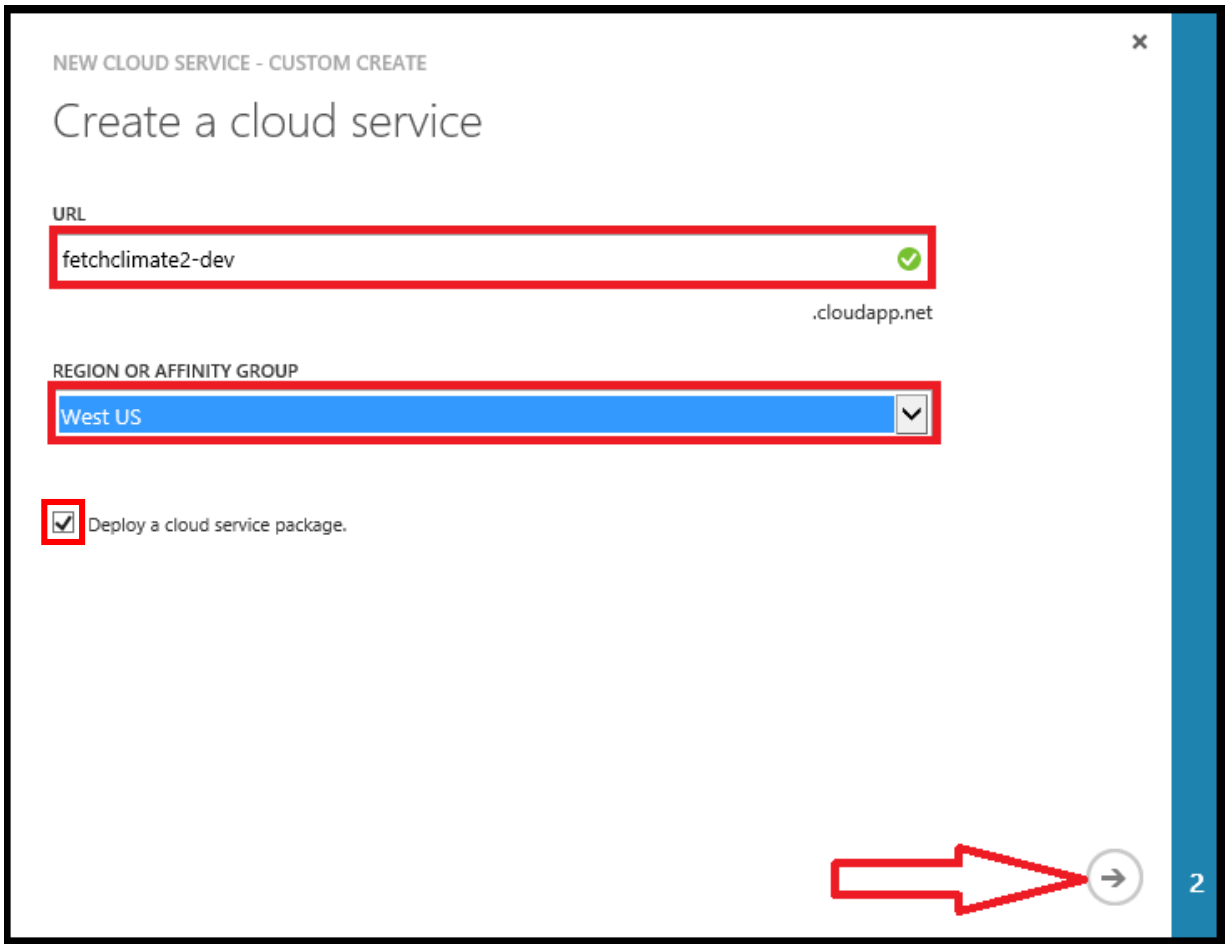
```
<?xml version="1.0" encoding="utf-8"?>
<ServiceConfiguration serviceName="WebService.Azure"
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration"
osFamily="3" osVersion="*" schemaVersion="2012-10.1.8">
  <Role name="Frontend">
    <Instances count="1" />
    <ConfigurationSettings>
      <Setting name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString"
value="DefaultEndpointsProtocol=https;AccountName=fc2cache;AccountKey=CACHEKEY"
/>
      <Setting name="DatabaseConnectionString" value="fc2configdb" />
      <Setting name="BlobConnectionAccountName" value="fc2cache" />
      <Setting name="BlobConnectionAccountKey" value="CACHEKEY" />
      <Setting name="AllowedJobRegistrationSpan" value="60" />
      <Setting name="WaitingFastResultPeriodSec" value="50" />
      <Setting name="MinPtsPerPartition" value="2500" />
      <Setting name="MaxPtsPerPartition" value="1024000" />
      <Setting name="JobTouchTimeTreshold" value="120" />
      <Setting name="FrontendTraceLevel" value="Verbose" />
      <Setting name="JobManagerTraceLevel" value="Verbose" />
      <Setting name="JobStatusCheckIntervalMilisec" value="100" />
    </ConfigurationSettings>
    <Certificates></Certificates>
  </Role>
  <Role name="FetchWorker">
    <Instances count="16" />
    <ConfigurationSettings>
      <Setting name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString"
value="DefaultEndpointsProtocol=https;AccountName=fc2cache;AccountKey=CACHEKEY"
/>
      <Setting name="DatabaseConnectionString" value="fc2configdb" />
      <Setting name="BlobConnectionAccountName" value="fc2cache" />
      <Setting name="BlobConnectionAccountKey" value="CACHEKEY" />
      <Setting name="JobTouchPeriod" value="10" />
      <Setting name="JobQueuePollingMilisec" value="100" />
      <Setting name="FetchWorkerTraceLevel" value="Verbose" />
      <Setting name="JobManagerTraceLevel" value="Verbose" />
      <Setting name="DataHandlerTraceLevel" value="Verbose" />
      <Setting name="FetchEngineTraceLevel" value="Verbose" />
      <Setting name="HeavyJobsPermitedCount" value="7" />
      <Setting name="LightJobExecutionTimeLimitSec" value="180" />
      <Setting name="DaysBeforeJobDeletion" value="60" />
      <Setting name="HoursBetweenCleanup" value="23" />
    </ConfigurationSettings>
    <Certificates></Certificates>
  </Role>
</ServiceConfiguration>
```

3. Open the Azure Management Portal.
4. Click **NEW** in the lower-left corner.

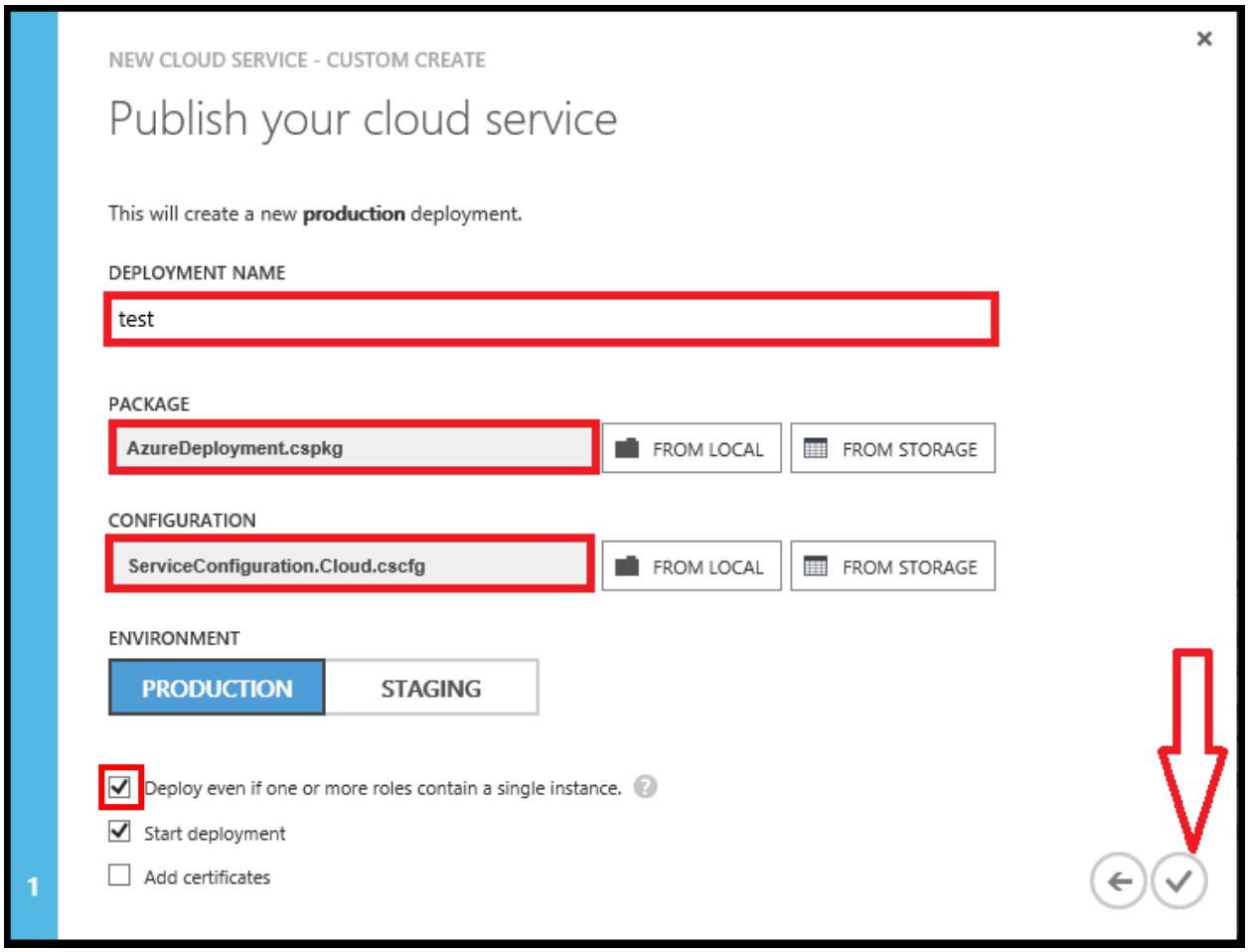
5. Choose **COMPUTE** > **CLOUD SERVICE** > **CUSTOM CREATE**.



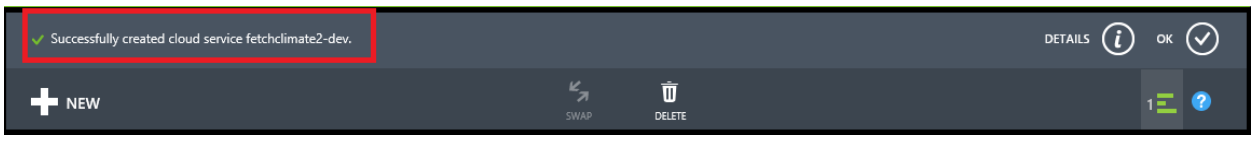

6. Enter the URL of the service and the region of hosting. Select the **Deploy a cloud service package** check box, and click the **Next** button.



7. Specify a name for your service deployment. Enter the path to the package and configuration file prepared in the previous step by clicking **FROM LOCAL**.




8. Click the **OK** button, and wait until the service is started.




## Step 3. Deploy the Storage Service

1. After unzipping FetchClimate_ServicePacks.zip to a folder, open the **AzureStorageServicePackage_v2.0.20156.0** folder.

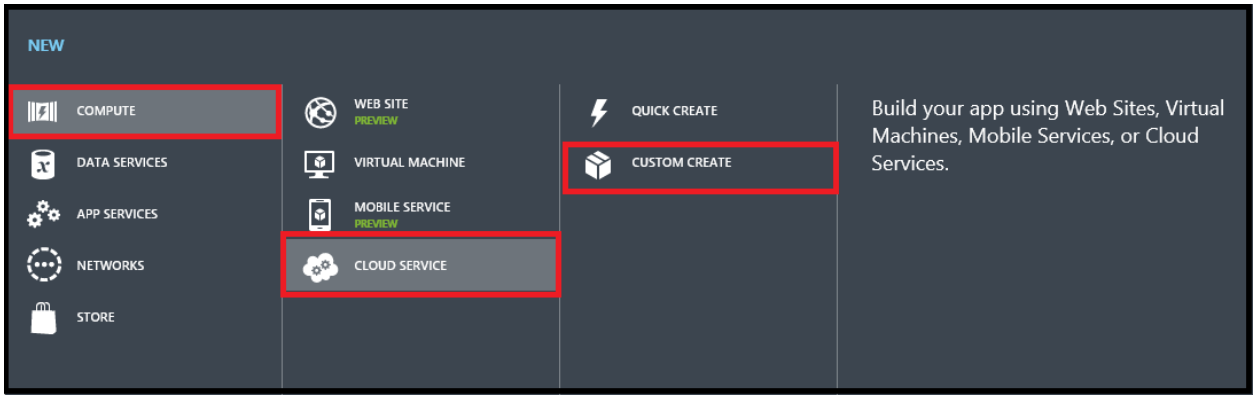
   Two files are in the folder: **ServiceConfiguration.Cloud.cscfg** and **ChunkedStorageCloudService.cspkg**.

2. In the ServiceConfiguration.Cloud.cscfg file, replace the placeholders (marked in yellow) with your created account name, account key, and connection string.

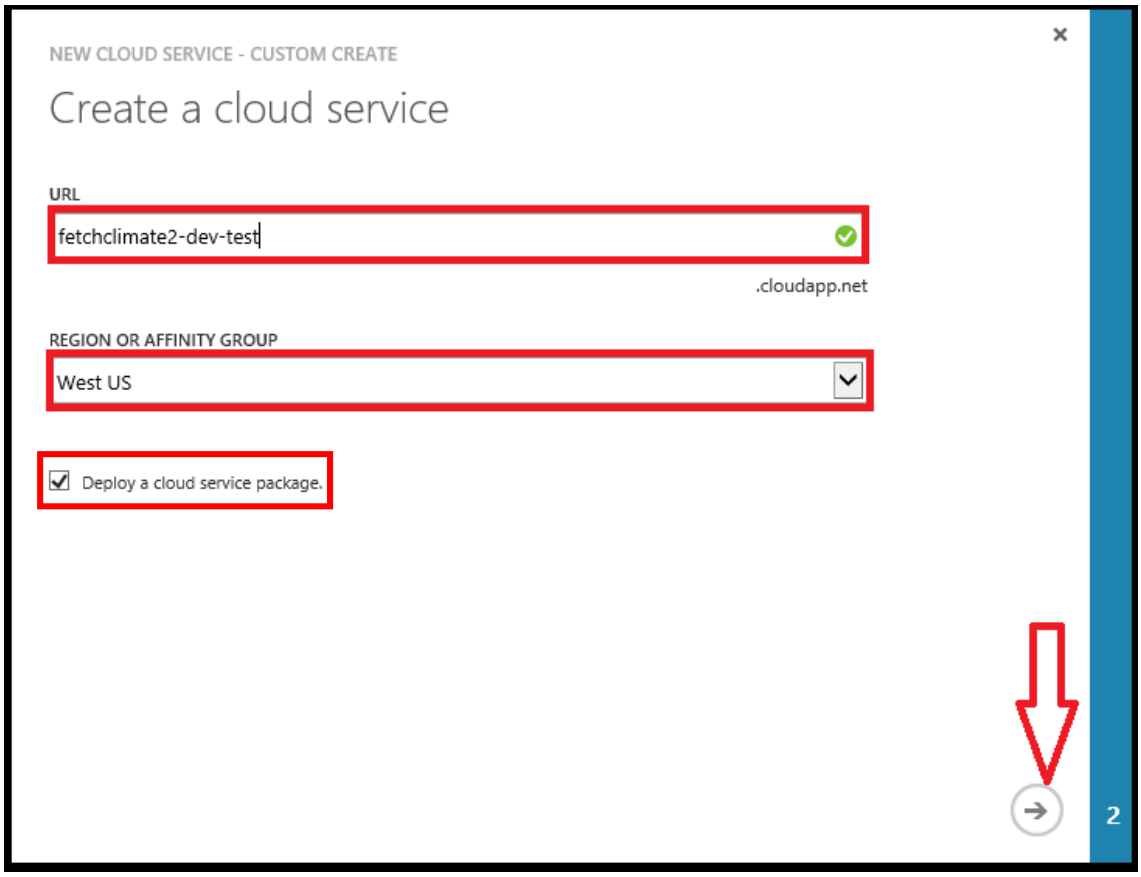
   Example:
   **DATAKEY**:
   QNbAR9vo7fM617NZEDX6bMKr5+IRBCghlgL7oQfEHYkX/V6JbluUYWD8BDxTwSxTmkoAIhbFrwyQgfQ==

```
<?xml version="1.0" encoding="utf-8"?>
<ServiceConfiguration serviceName="ChunkedStorageCloudService"
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfigura
tion" osFamily="3" osVersion="*" schemaVersion="2012-10.1.8">
  <Role name="ChunkedStorageWorker">
    <Instances count="1" />
    <ConfigurationSettings>
      <Setting name="DataConnectionString"
value="DefaultEndpointsProtocol=http;AccountName=fc2data;AccountKey=DATAKEY
" />
      <Setting
name="Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString"
value="DefaultEndpointsProtocol=https;AccountName=fc2data;AccountKey=DATAKE
Y" />
    </ConfigurationSettings>
  </Role>
</ServiceConfiguration>
```

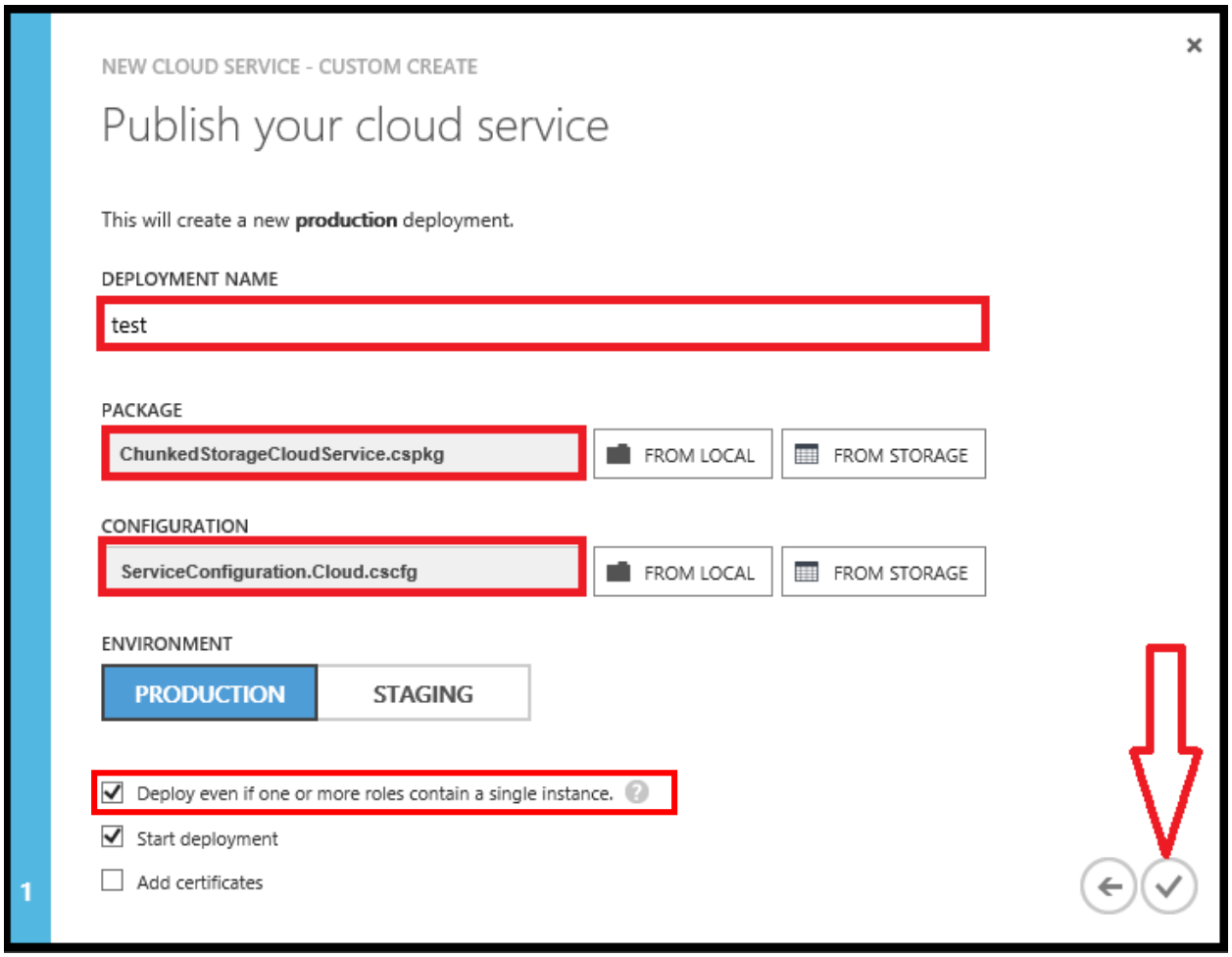
3. Open the Azure Management Portal.
4. Click **NEW** in the lower-left corner.
5. Choose **COMPUTE** > **CLOUD SERVICE** > **CUSTOM CREATE**.



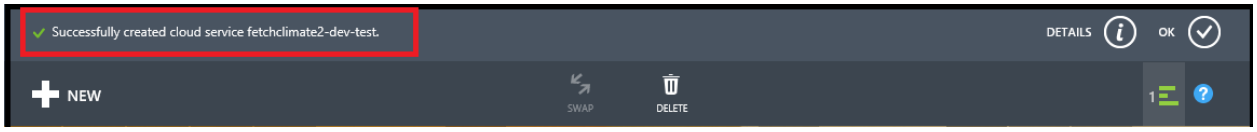
6. Enter the URL of the service and the region of hosting. Select the **Deploy a cloud service package** check box, and click the **Next** button.

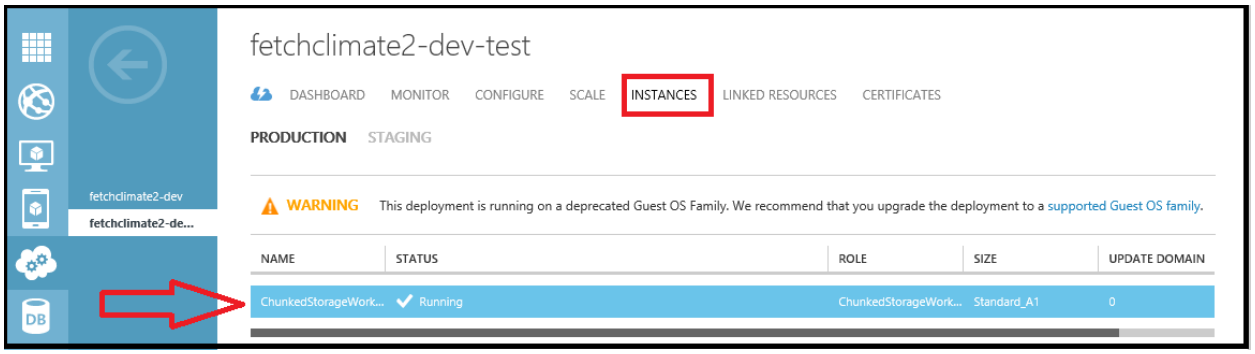


7. Specify a name for your service deployment. Enter the path to the package and configuration file prepared in the previous step by clicking **FROM LOCAL**.



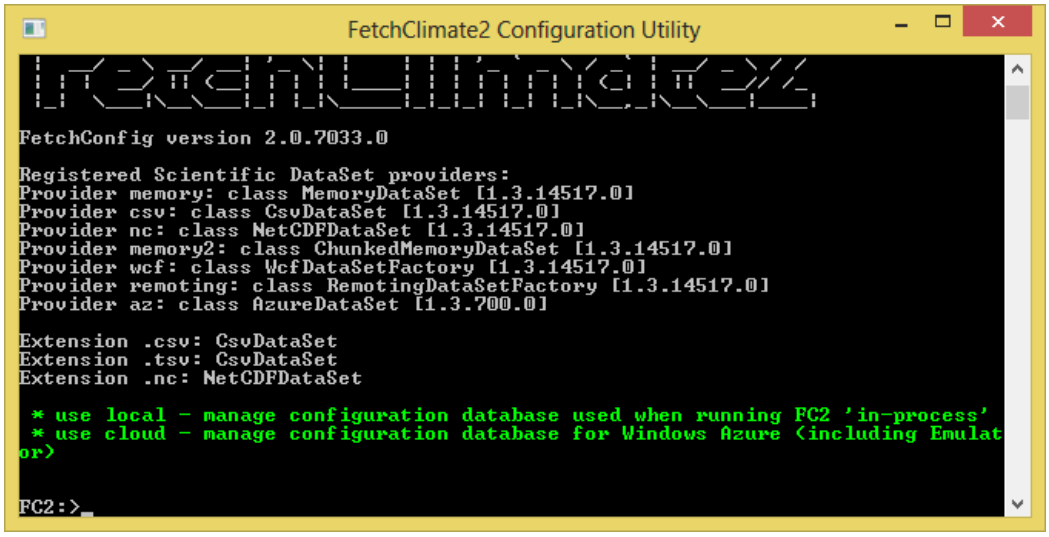

8. Click the **OK** button, and wait until the service is started.



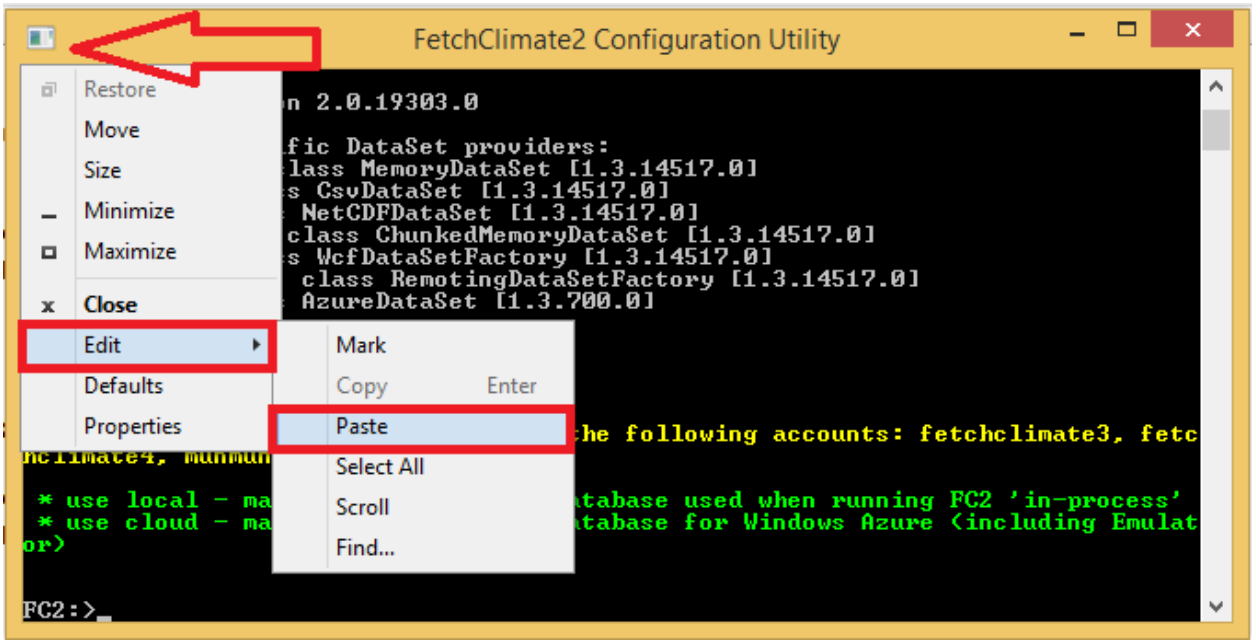
## Step 4. Configure the FetchClimate Service

1. Unzip the FetchClient_ClientTools.zip file.
2. Run FetchConfig.exe.




The FetchConfig utility allows you to store account keys for specific account names locally. Please beware that account keys give unlimited control over your Azure storage, so use this feature only on trusted systems.

**Note:** Copy all the below commands in a Notepad or Microsoft Word file, and paste sequentially in FetchConfig.exe as instructed below. Please do not close FetchConfig.exe until all commands are executed.

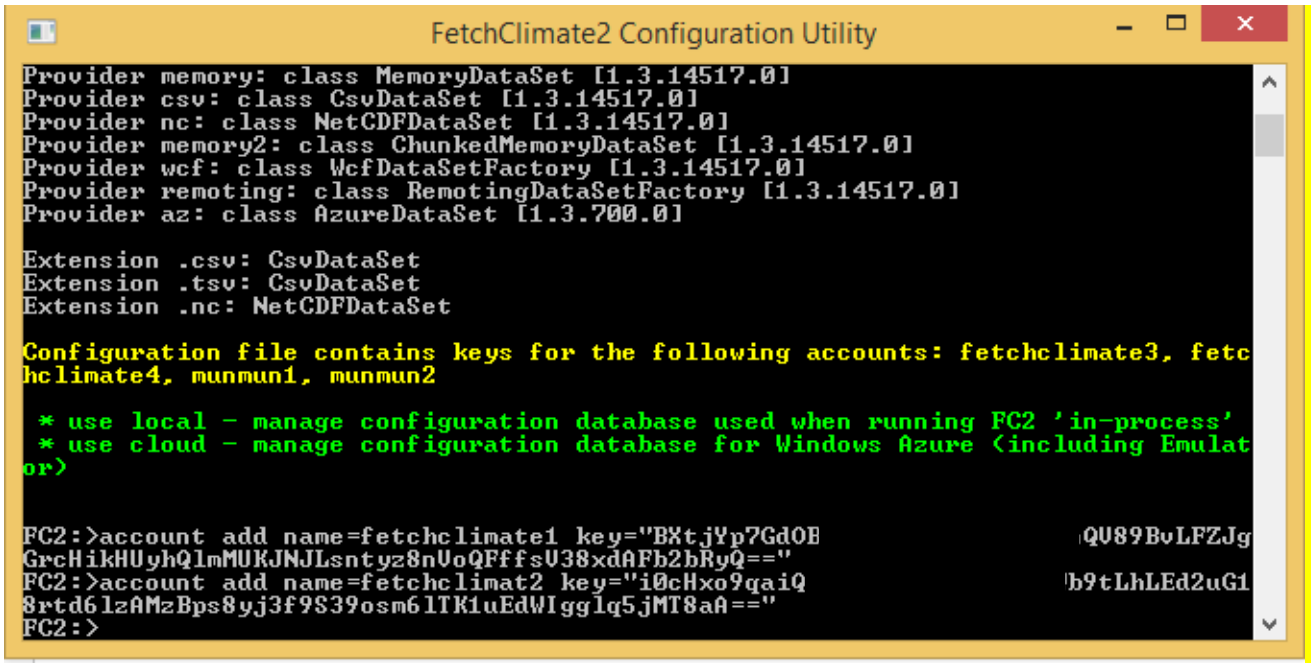


3. Paste following command to store the key for the *fc2cache* account. Replace the highlighted parts with your actual account name and key.

   ```
   account add name=fc2cache key="CACHKEY"
   ```

4. Paste following command to store the key for the fc2data account. Replace the highlighted parts with the actual name and key.

   ```
   account add name=fc2data key="DATAKEY"
   ```

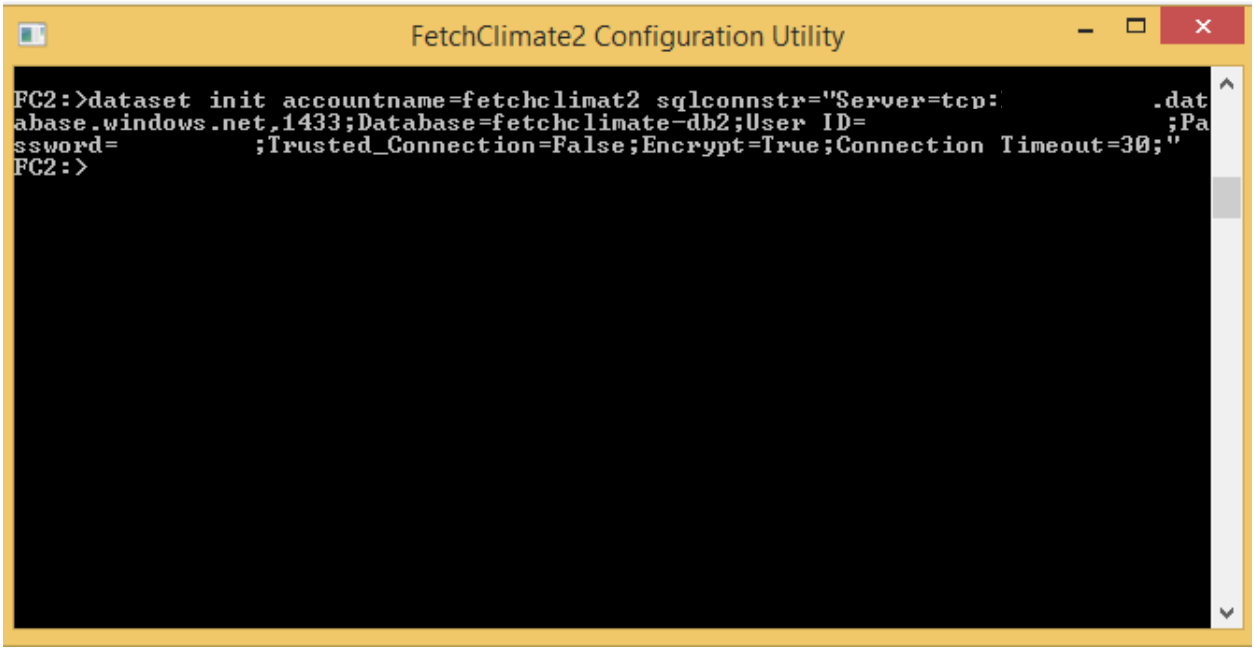



5. The Storage service's database must be initialized. You do this once after creating the new storage. Paste the following command in the FetchConfig utility. Don't forget to replace the highlighted parts with your actual account name and connection string.

   ```
   dataset init accountname=fc2data sqlconnstr="fc2storagedb"
   ```

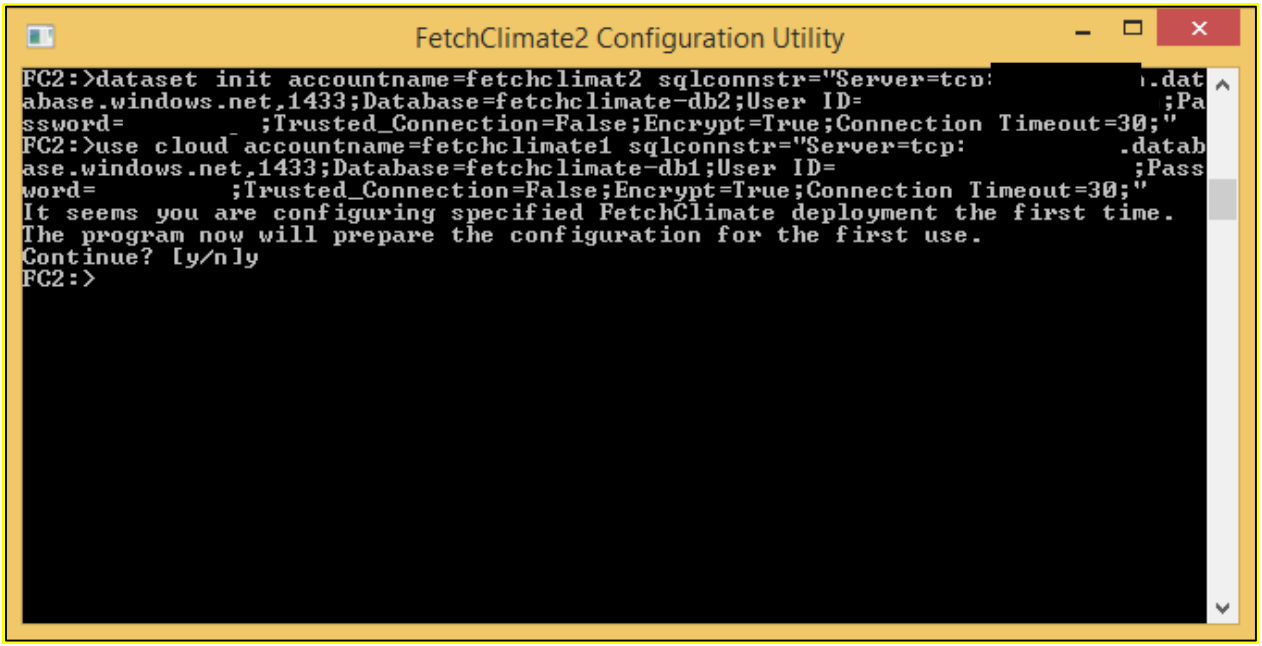
   If you see the following error trying to connect to your database, it may be because you need to configure firewall rules for your Azure SQL server. Please see Appendix B for instructions on configuring the firewall rules.

   ```
   Specified SQL server can't be reached
   (Server=tcp:<Servername>.database.windows.net,1433;Database=fc2storagedb;User
   ID=<Username>;Password=<Password>;Trusted_Connection=False;Encrypt=True;Connection TimeOut=30;)
   ```




6. Now initialize the FetchClimate variables and data source database. To do this, paste the following command and confirm that you want to create the initial database. Don't forget to replace the highlighted parts with the actual account name and connection string.

   `use cloud accountname=`*`fc2cache`* `sqlconnstr="`*`fc2configdb"`*




**Congratulations! The FetchClimate deployment is ready. However, it is empty. The next steps will add one variable and one data source for it.**

## Step 5. Populate the FetchClimate Installation

1. Define the elevation variable.

```
variable add name="elevation" units="meters" description="elevation above sea level"
```

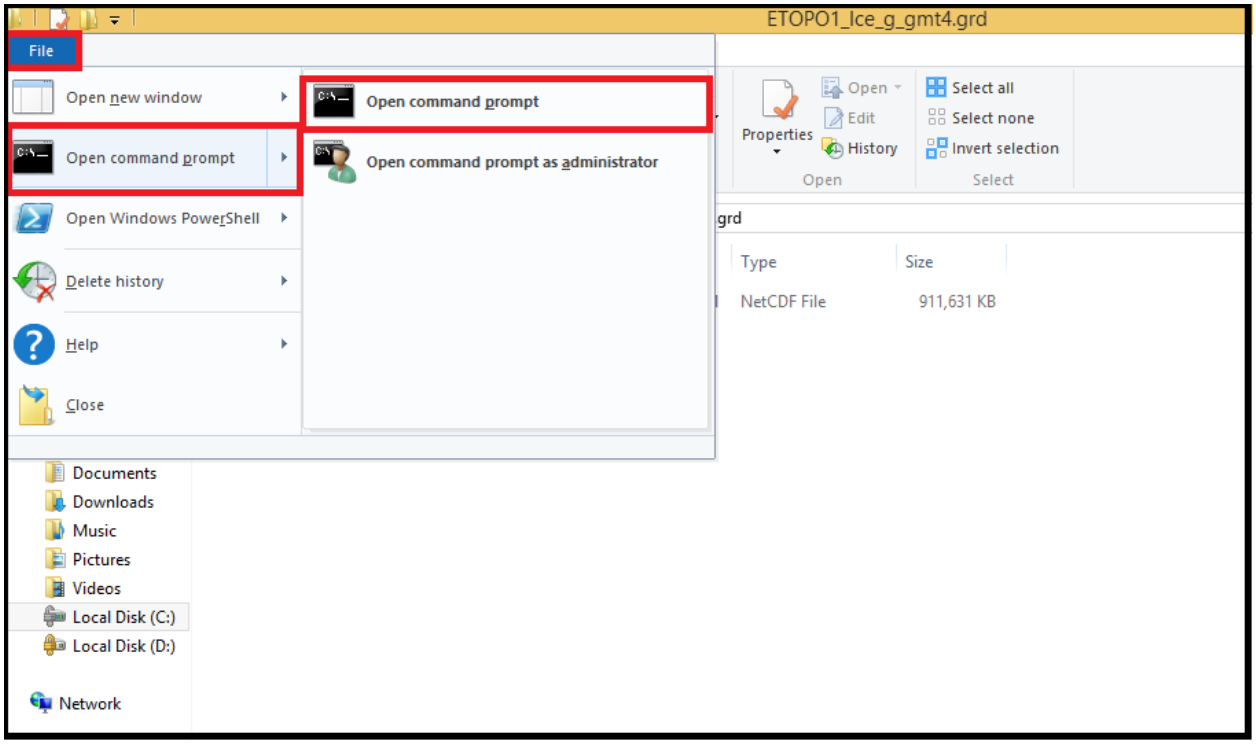
2. Acquire the elevation data in NetCDF file from here: http://www.ngdc.noaa.gov/mgg/global/relief/ETOPO1/data/ice_surface/grid_registered/netcdf/. Download the file ETOPO1_Ice_g_gmt4.grd.gz. Extract the file to a directory (for example, **C:\ FetchClimate2\ ETOPO1_Ice_g_gmt4.grd**). Rename the **ETOPO1_Ice_g_gmt4.grd** file to **ETOPO1_Ice_g_gmt4.grd.nc**.

   **Note:** In the ETOPO1_Ice_g_gmt4.grd.nc file, the variable for the horizontal axis has the name 'x', and the variable for the vertical axis has the name 'y'. FetchClimate doesn't understand these names, so we have to rename 'x' to 'lon' and 'y' to 'lat'.

3. Download the netCDF Operator (NCO) for Windows from here: http://nco.sourceforge.net/src/nco-4.4.2.windows.mvs.exe

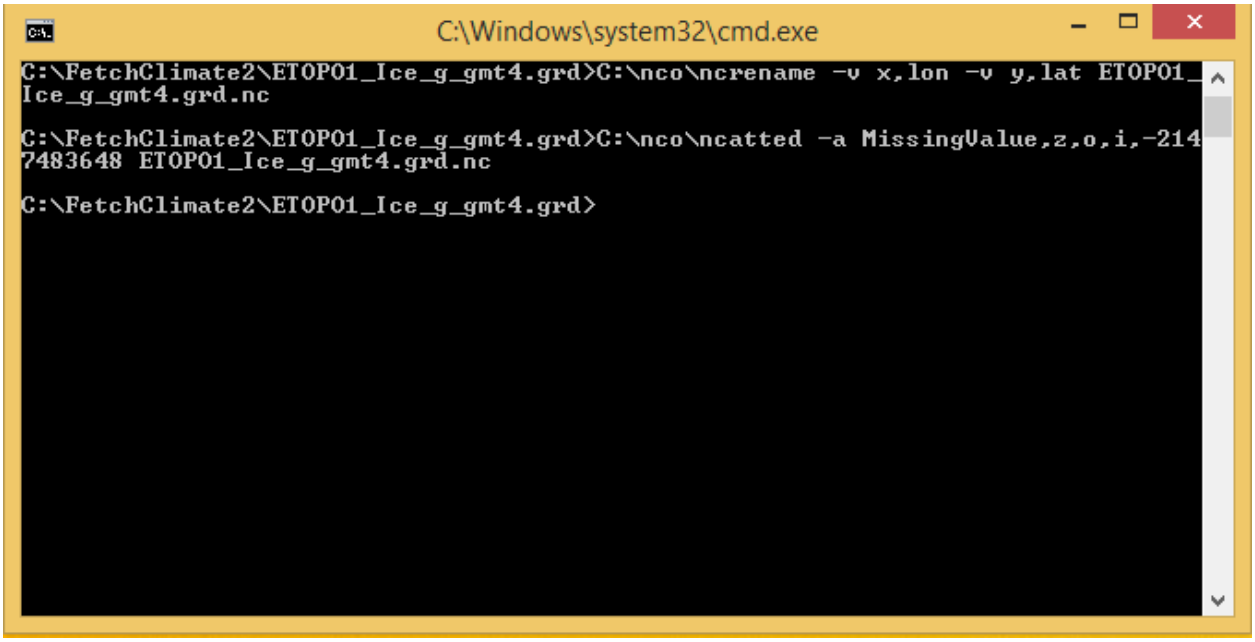
4. Install it to a folder—for example, **C:\nco**.

5. Open a command prompt from the folder with the climate data (C:\FetchClimate2\ETOPO1_Ice_g_gmt4.grd).



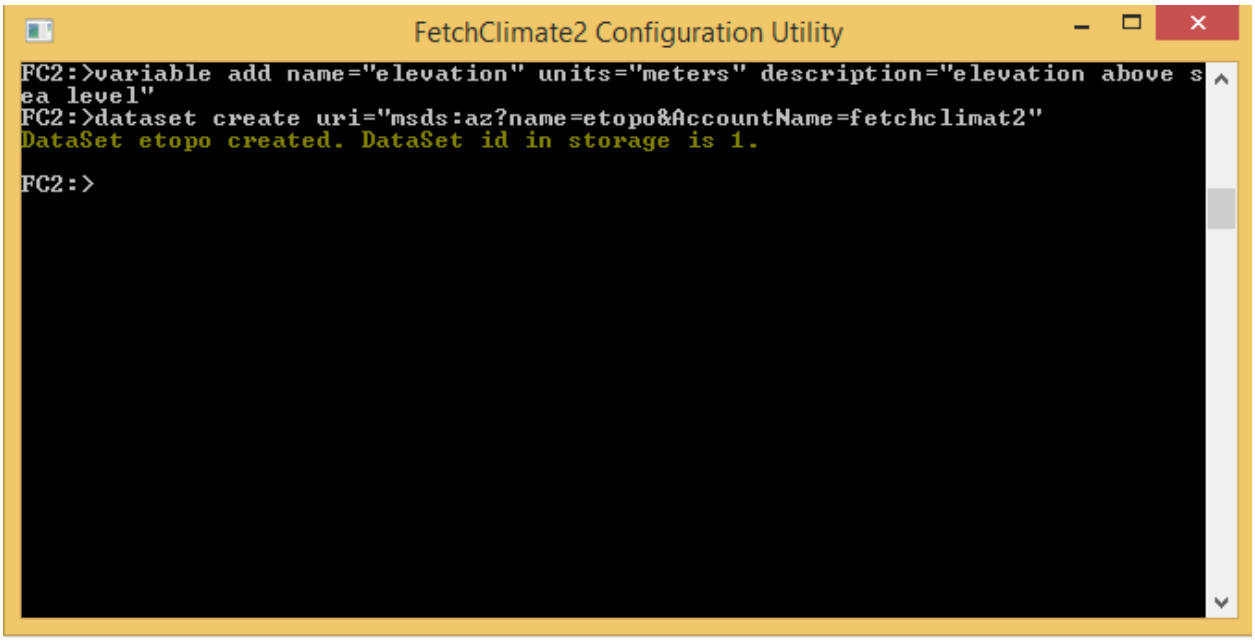

6. Run the following commands. This command has to be executed once when the ETOPO1_Ice_g_gmt4.grd file is downloaded for the first time.

```
C:\nco\ncrename -v x,lon -v y,lat ETOPO1_Ice_g_gmt4.grd.nc
C:\nco\ncatted -a MissingValue,z,o,i,-2147483648 ETOPO1_Ice_g_gmt4.grd.nc
```



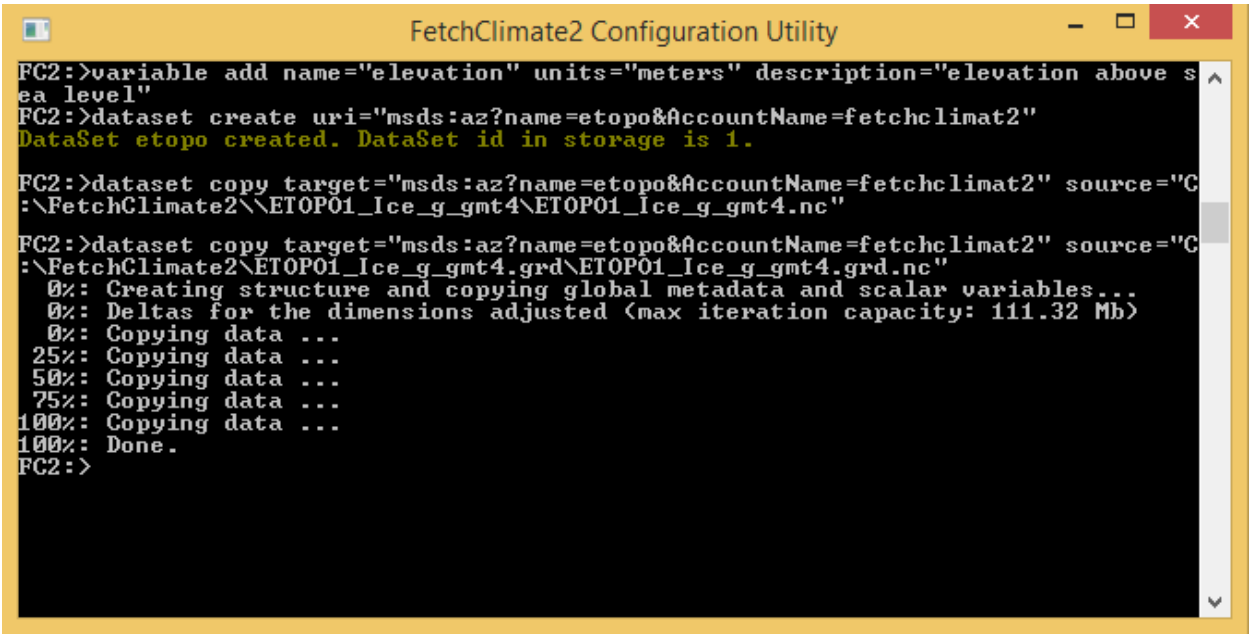

7. Open FetchConfig.exe. Run the following commands, replacing the placeholder for the dataset in the Storage service. Be sure to use the correct account name.

```
dataset create uri="msds:az?name=etopo&AccountName=fc2data"
```



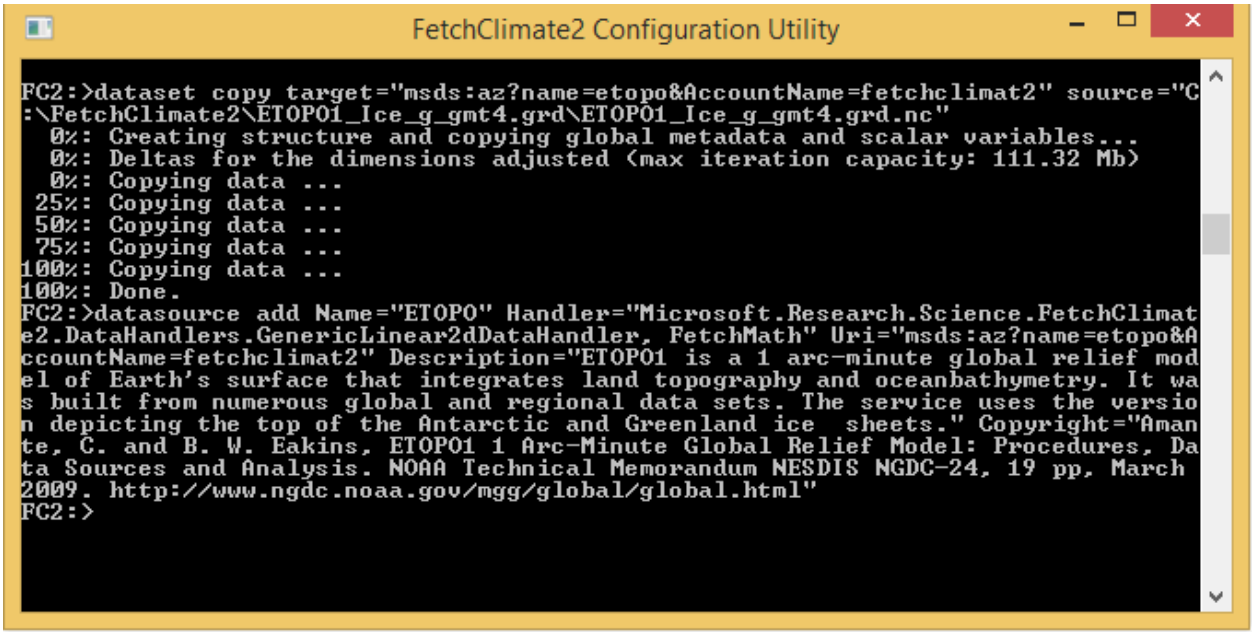

8. Upload the data from NetCDF to the cloud storage. Please wait approximately 15 minutes to finish the data upload.

```
dataset copy target="msds:az?name=etopo&AccountName=fc2data"
source="C:\FetchClimate2\ETOPO1_Ice_g_gmt4.grd\ETOPO1_Ice_g_gmt4.grd.nc"
```



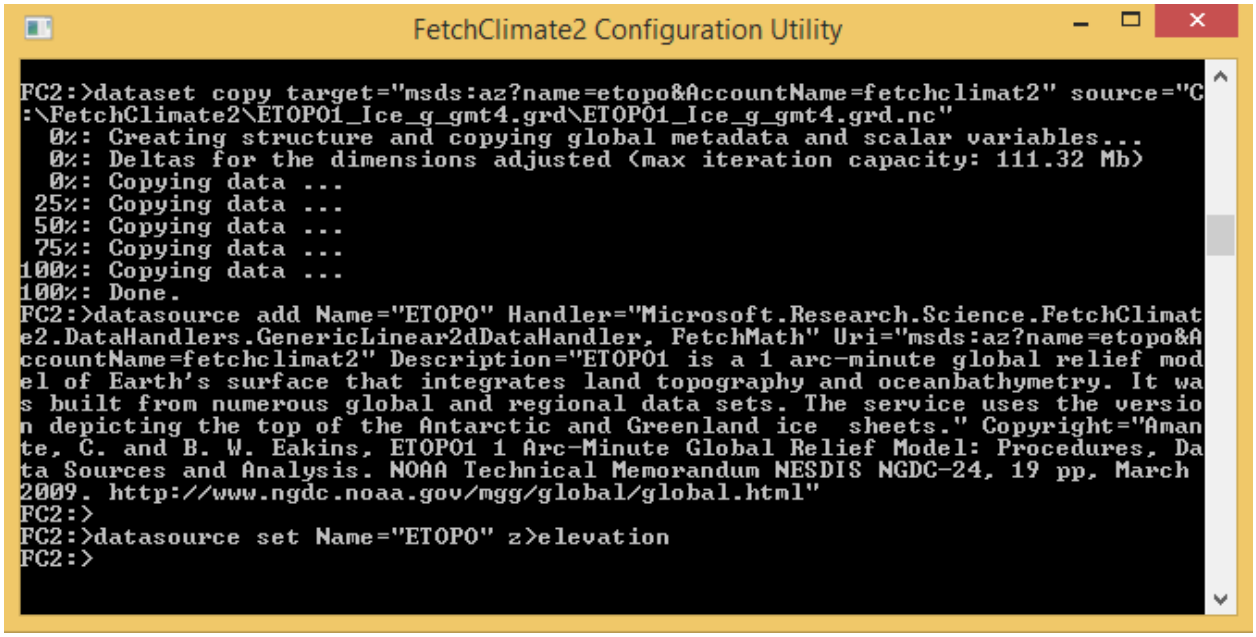

9. Add the data source definition.

```
datasource add Name="ETOPO"
Handler="Microsoft.Research.Science.FetchClimate2.DataHandlers.GenericLinea
r2dDataHandler, FetchMath" Uri="msds:az?name=etopo&AccountName=fc2data"
Description="ETOPO1 is a 1 arc-minute global relief model of Earth's
surface that integrates land topography and oceanbathymetry. It was built
from numerous global and regional data sets. The service uses the version
depicting the top of the Antarctic and Greenland ice sheets."
Copyright="Amante, C. and B. W. Eakins, ETOPO1 1 Arc-Minute Global Relief
Model: Procedures, Data Sources and Analysis. NOAA Technical Memorandum
NESDIS NGDC-24, 19 pp, March 2009.
http://www.ngdc.noaa.gov/mgg/global/global.html"
```



10. Set the variable name mapping.

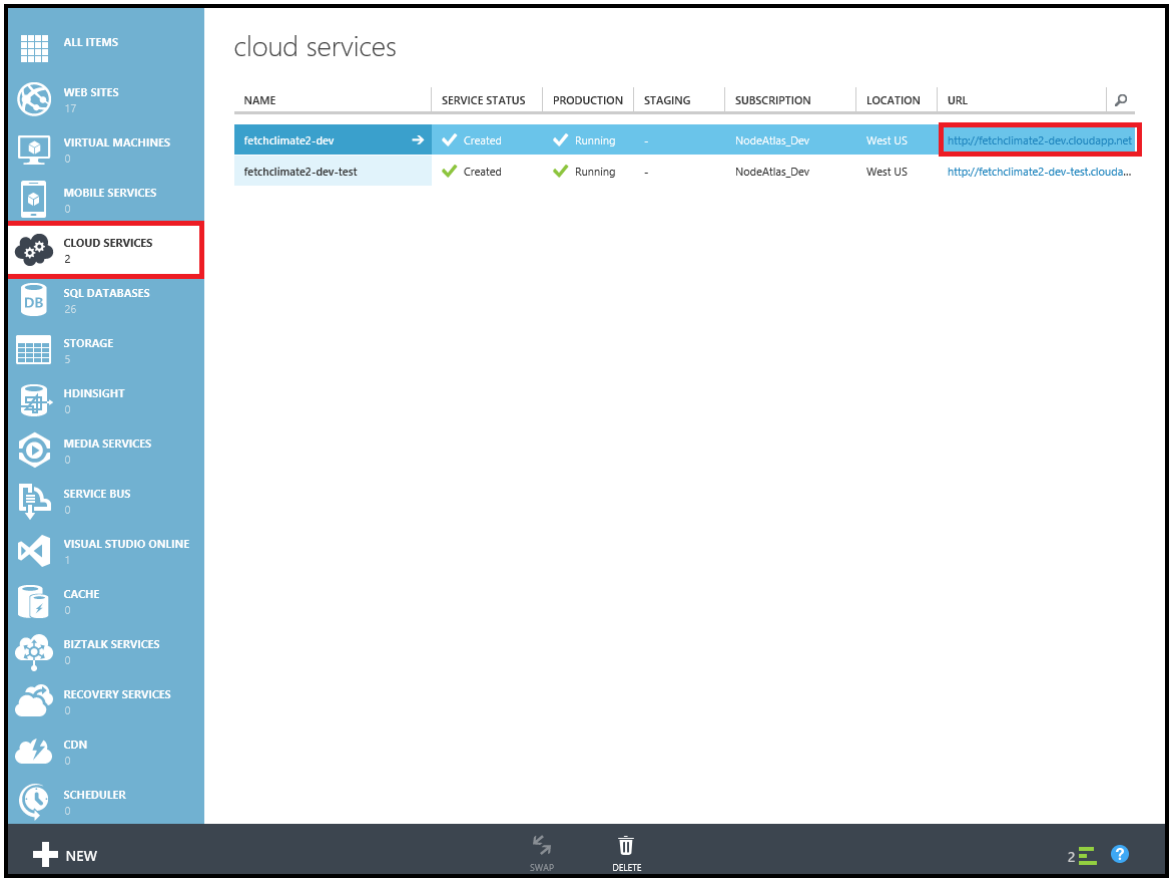
```
datasource set Name="ETOPO" z>elevation
```



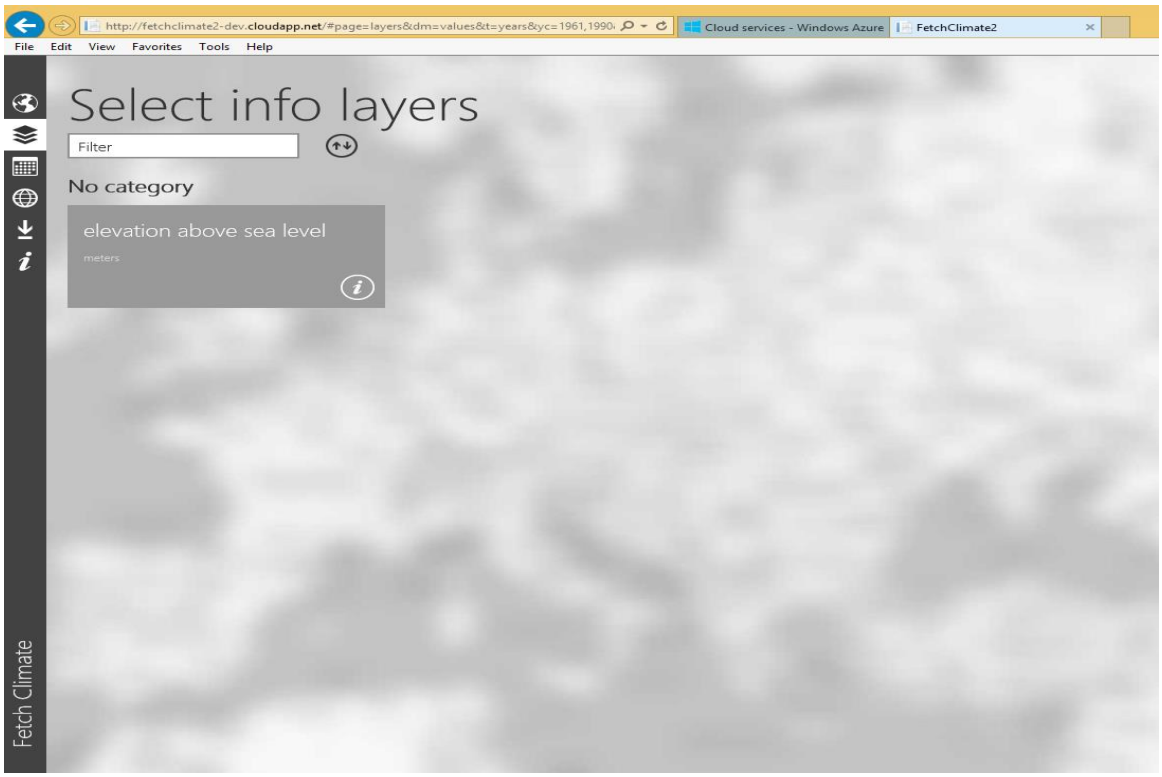

**Congratulations again! The FetchClimate deployment now has one variable and one data source.**

## Step 6. Verify Your FetchClimate Deployment

1. Open the Azure Management Portal: **http://manage.windowsazure.com**
2. In the left panel, click **CLOUD SERVICES**.



3. You can see two cloud services running: one is the FetchClimate service, and the other is the Storage service. Select the URL of the FetchClimate service. Your FetchClimate instance should be live now.



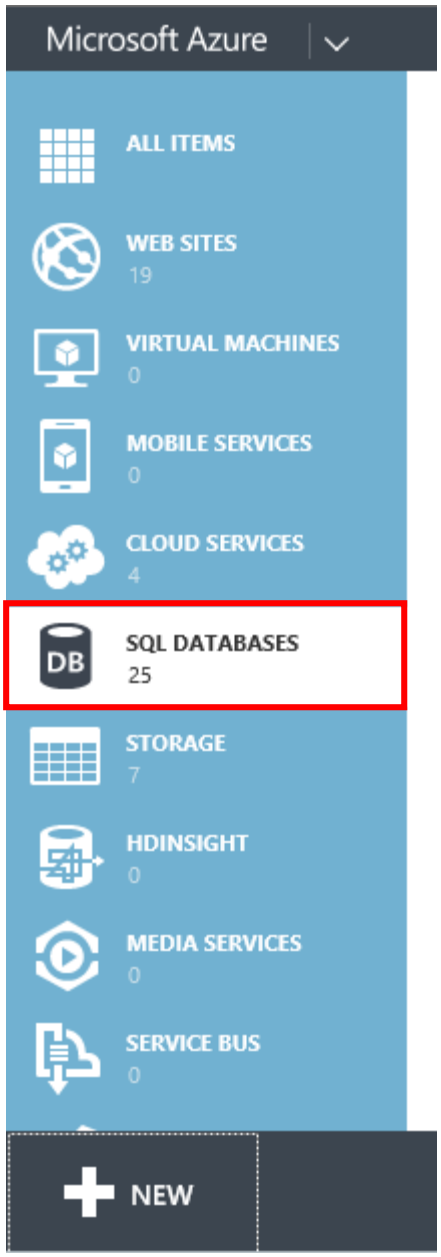

## References

[1] Microsoft Research Dmitrov package: http://research.microsoft.com/en-us/um/cambridge/groups/science/tools/dmitrov/default.htm

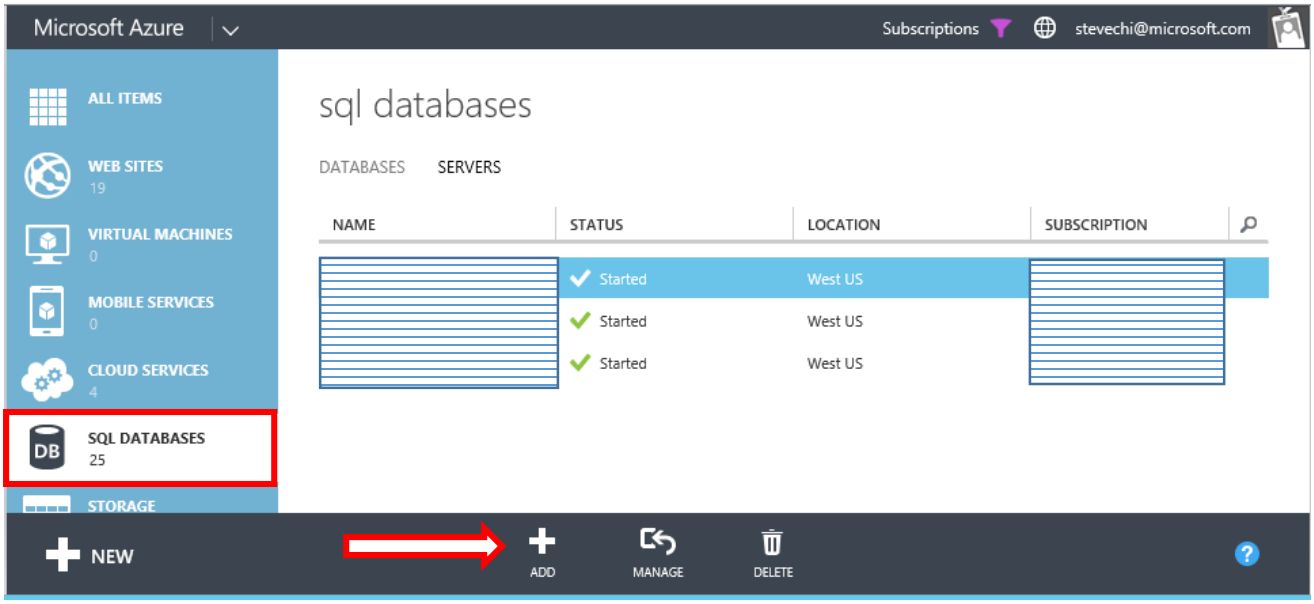
## Appendix A – Create a New Azure SQL Database Server

If you do not have an Azure SQL database server configured, follow the instructions in this section.

1. Select **SQL DATABASES** in the Azure Management Portal.



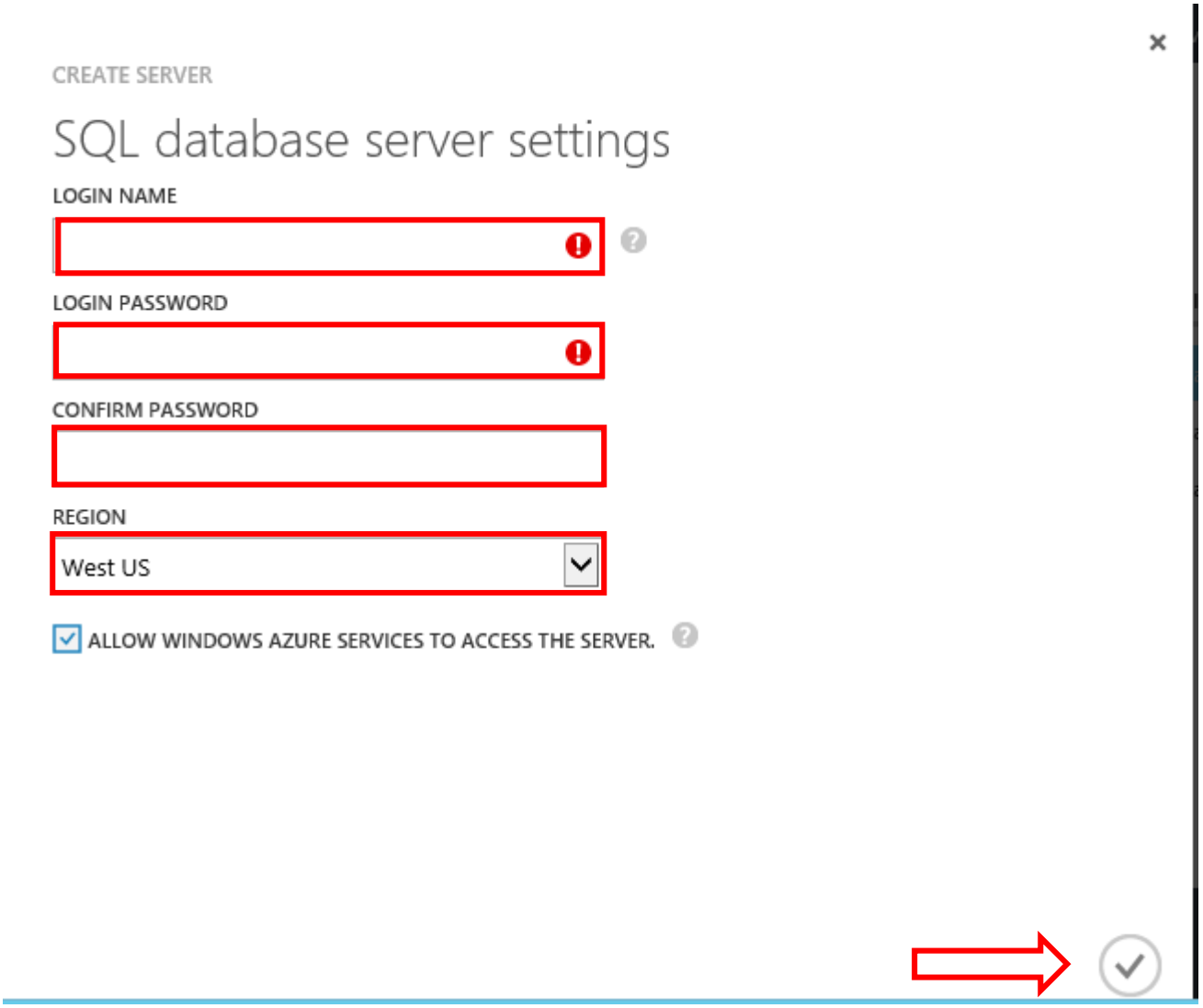
2. Select Servers in the sql databases view, and click **ADD**.



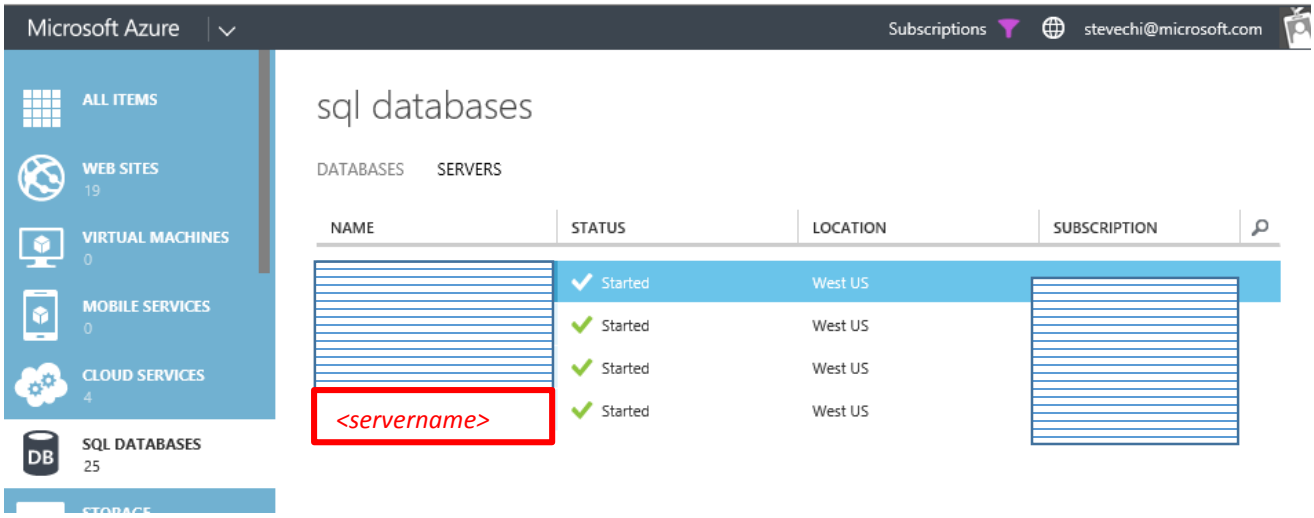
3. Create a User Name and Password for this SQL server.

   This user name and password will be required to create and manage databases that are added to this server. Choose the same region as the one chosen for the storage accounts. Finally, click the check mark in the lower-right corner of the window.

   The user name will be denoted as *<UserName>* and password as *<Password>.*



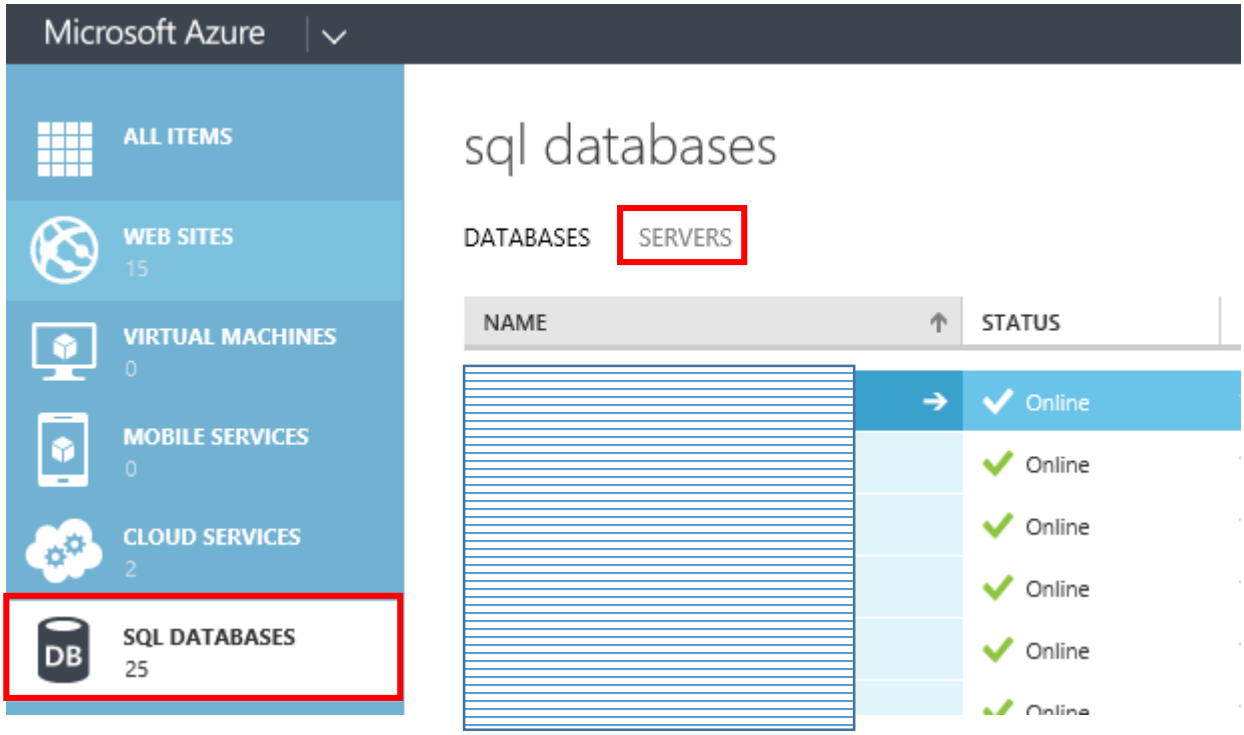
4. A new Azure SQL server will be created. Its name will be added to the bottom of the server list. Please note this name so that the databases created below will be added to the correct server.



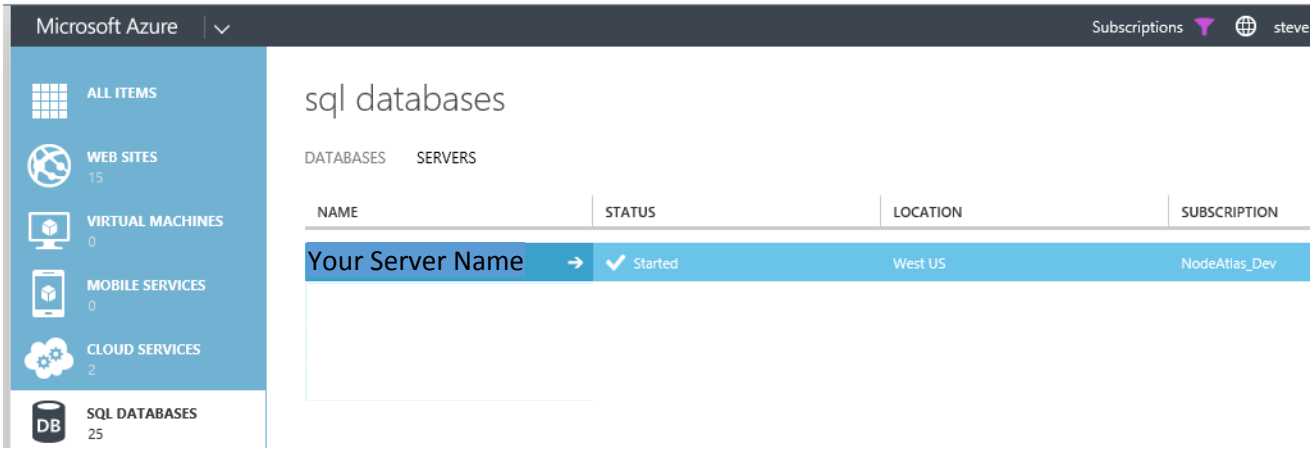

X

## Appendix B – Configuring the Firewall on Azure SQL Server

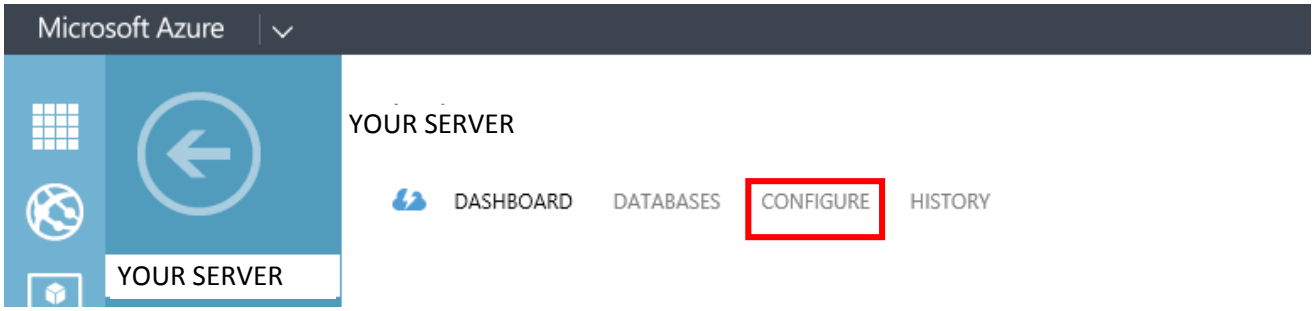
1. In the Azure management portal, select **SQL DATABASES**, and then select the **SERVERS** tab.



2. Select your server.




3. Select **CONFIGURE**.



4. Add your client IP address to the firewall, and click **SAVE**.

   **Note**: It may take 5 or 10 minutes for this change to take effect.