

Installing the Microsoft Azure BLAST Example

Introduction

This paper explains how to install and run the BLAST for .NET demo, which demonstrates how to use Microsoft Azure Cloud Services to perform a Basic Local Alignment Search Tool ([BLAST](#)) query. Initially, the application runs locally and uses a single compute node. In this lab, you will deploy the application to Microsoft Azure. From there, you can take advantage of Microsoft Azure scalability to increase the number of compute nodes, and use the cloud-based Microsoft Azure Blob and Table services for storage.

This paper has four main sections.

- **Using the Management Portal.** In this section, you use the Microsoft Azure Management Portal to create a cloud service, a Microsoft Azure Service Bus namespace, a Service Bus Queue, and a storage account.
- **Configuring the Environment.** In this section, you download the source code, edit some settings so that you can deploy the BLAST demo to Microsoft Azure, and configure the Microsoft Azure environment.
- **Publishing the BLAST demo.** In this section, you deploy the BLAST demo to Microsoft Azure. You also learn how to change your Microsoft Azure environment to suit your needs. For example, you learn how to add more compute nodes.
- **Testing the BLAST demo.** In this section, you run a BLAST job to ensure that the demo is running correctly.

You can download the code for the demo at https://github.com/MSRConnections/Azure4Research-TechnicalPapers/tree/master/Scaling_a_Windows_Azure_Cloud_Service_BLAST. See the code documentation at https://github.com/MSRConnections/Azure4Research-TechnicalPapers/blob/master/Scaling_a_Windows_Azure_Cloud_Service_BLAST/BLAST.docx?raw=true.

Prerequisites

Make sure you meet the following requirements before you try to install and run the demo.

Microsoft Azure prerequisites

In order to use Microsoft Azure you need a Microsoft Azure subscription. If you don't have one, go to [Get Started with Microsoft Azure](#) and follow the links to create one.

Development machine prerequisites

You must have the following components installed on your development machine.

- [Microsoft Visual Studio 2012](#)
- [Microsoft Azure SDK and Azure Tools for Microsoft Visual Studio \(June 2013\)](#)

Installing the Microsoft Azure BLAST Example

- [Microsoft Azure Explorer Freeware](#) application from CloudBerry Lab. This application makes it easy to upload, download, and browse files stored in blobs. An alternative application is [Azure Storage Explorer](#), which is available on CodePlex.

Note: The steps to follow are similar for Visual Studio 2013.

Using the Management Portal

In this section, you use the management portal to create a cloud service, a Microsoft Azure Service Bus namespace, a Service Bus Queue, and a storage account. You will save the names and credentials in text files because you will need them to configure the web and worker roles.

Task 1: Creating a cloud service

Follow these steps to create a cloud service to host the application runtimes.

1. Log into the [Microsoft Azure portal](#).
2. Click **NEW** at the bottom left of the page.
3. A wizard appears that allows you to select the type of resource you would like to create. In this case, select **COMPUTE**, select **CLOUD SERVICE**, and then select **QUICK CREATE**. In the **URL** field, enter the name of your service. Make a note of this name.
4. In the **REGION OR AFFINITY GROUP** field, select from one of the available global data centers. Remember which data center you selected and use the same one throughout all the applicable tasks in this document. Figure 1 shows an example of a completed page.

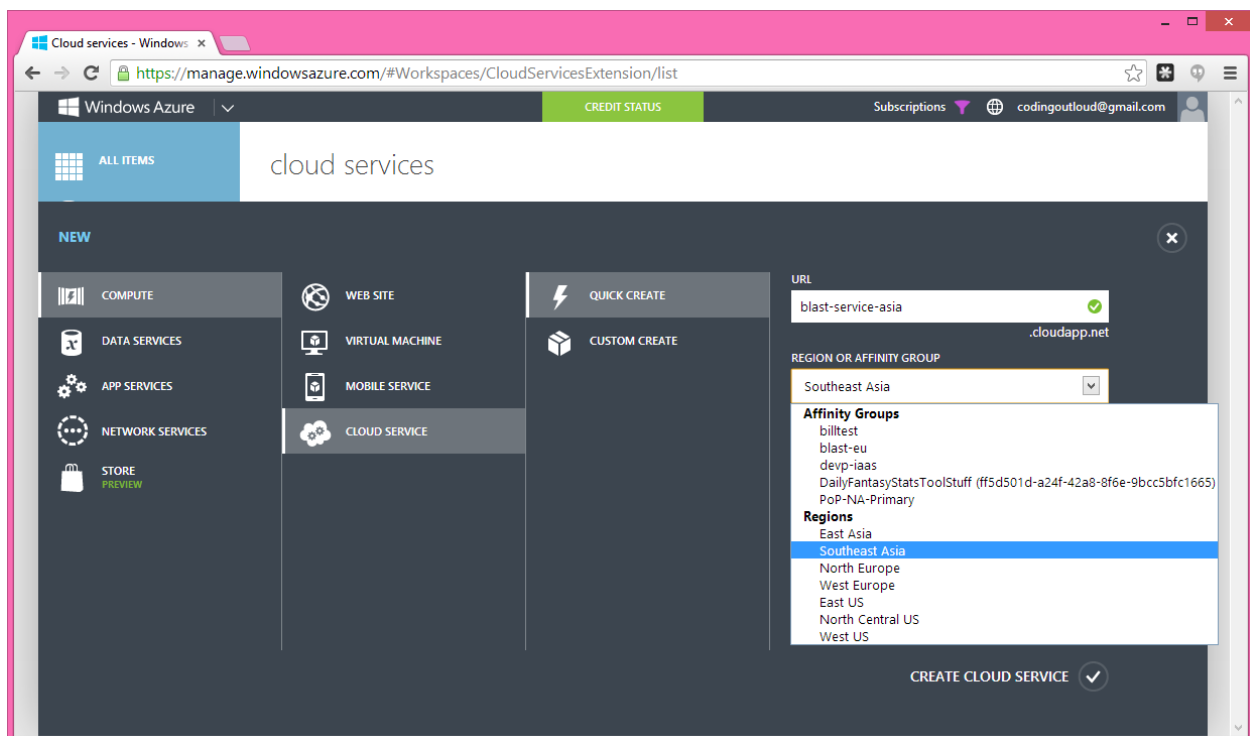


Figure 1: Screenshot of a completed cloud service wizard

Installing the Microsoft Azure BLAST Example

Task 2: Creating the Service Bus namespace

The Service Bus namespace is a globally unique identifier for message bus services. Creating a namespace generates a connection string that consists of an issuer and a default key. Follow these steps to create a Service Bus namespace.

1. Log on to the Microsoft Azure Management Portal.
2. In the left navigation pane, select **Service Bus**.
3. In the lower pane of the Management Portal, select **Create**.
4. In the **Add a New Namespace** dialog, enter a namespace name. The system immediately checks to see if the name is available.
5. Choose the country or region where your namespace will be hosted. This should be the same region as you selected for your cloud service.
6. Select the **check mark**. The system will create the namespace. This may take a few minutes.
7. After the namespace is created, you can retrieve the connection string. In the left navigation pane, select **Service Bus**.
8. Select the Service Bus that you just created. At the bottom of the page, select **Connection Information**. The Access Information page appears.
9. Copy the information in the **CONNECTION STRING** field and paste it into a text editor such as Notepad. Save this information for later.

Task 3: Creating a Service Bus Queue

The web role places messages that are BLAST jobs in the Service Bus Queue and the worker role retrieves those messages to perform the job. Follow these steps to create a Service Bus Queue.

1. Go to the **Microsoft Azure Management Portal**.
2. Select your **Service Bus namespace**.
3. Click **Queues**.
4. Click **Create A New Queue**.
5. Click **Quick Create**.
6. In the **Queue Name** field, enter the name **JobQueue**.
7. Select the same region you have been using.
8. Click the **check mark** toward the bottom of the page. After a few moments, you should see that the new queue was created.

Task 4: Creating a storage account

In this task, you create a storage account that you will use for blob storage of the BLAST databases and input files. Later in this procedure, you upload the BLAST files to blob storage where they become the master copy. Each worker role retrieves a copy of the BLAST files from the master copy and caches it locally.

You will also use a table store. In order for users to monitor BLAST job requests, the application keeps the status of all received job requests in the table.

1. Go to the **Microsoft Azure Management Portal**.
2. Click **New**.
3. In the wizard, select **DATA SERVICES**, select **STORAGE**, and then select **QUICK CREATE**.

Installing the Microsoft Azure BLAST Example

4. In the **URL** field, enter the name of your storage account. Wait for the system to approve it by showing a green check mark.
5. In the **LOCATION/AFFINITY GROUP** field, select the same region you have used before.
6. Click **CREATE STORAGE ACCOUNT**. It will take a few moments to create the account.
7. After the account is created, click **MANAGE ACCESS KEYS**.
8. Copy and save the storage account name and the primary access key in a text file. You will use it later.

Configuring the Environments

In this section, you perform a set of tasks that prepare the environments both for Microsoft Azure and for your development machine.

Task 1: Downloading and inspecting the BLAST solution

Follow these steps to download the code for the demo to your local machine and then use Visual Studio to inspect the various projects.

1. On your machine, create the **Blast\Source\Blast** folder.
2. [Download](https://github.com/MSRConnections/Azure4Research-TechnicalPapers/archive/master.zip) a copy of the source code from <https://github.com/MSRConnections/Azure4Research-TechnicalPapers/archive/master.zip> to the BLAST folder.
3. Open **Visual Studio** as an administrator.
4. In Visual Studio, open the **BLAST.Cloud** solution file, which is located in the Blast folder.
5. In **Solution Explorer**, examine the solution tree. It includes several Microsoft Azure projects and some class libraries, as shown in Figure 2.

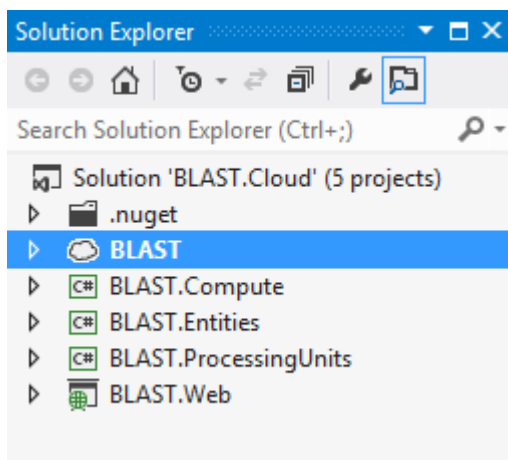


Figure 2: Contents of the Solution Explorer

The different projects are summarized below.

BLAST

The BLAST project is the Microsoft Azure Cloud Services project. It defines a single front-end web role that is named **BLAST.Web** and a single back-end worker role that is named **BLAST.Compute**. The cloud

Installing the Microsoft Azure BLAST Example

service also includes a Microsoft Azure service queue that allows the web role and the worker role to communicate. The web role sends messages, which are requests to run BLAST jobs, to the queue and the worker role takes the messages from the queue and runs the jobs.

BLAST.Compute

The **BLAST.Compute** project is the back-end worker role that performs the requests made by **BLAST.Web**. Initially, there is a single worker role, but you can create as many as you need once you deploy the application to Microsoft Azure.

BLAST.Entities

The **BLAST.Entities** project is a class library that defines the entities used in the demo. In this case, there is only one, named **SearchTask**.

BLAST.ProcessingUnits

The **BLAST.ProcessingUnits** project is a class library that implements the logic for the **SearchTask** entity.

BLAST.Web

The **BLAST.Web** project is the front-end web role.

Task 2: Preparing the solution for deployment

Follow these steps to edit the projects' configuration files so that you can deploy them to Microsoft Azure nodes.

1. In Visual Studio, in **Solution Explorer**, expand the **BLAST** node. Expand the **Roles** node. Right-click on **BLAST.Compute**. Select **Properties**. The **BLAST.Compute** property page opens.
2. Select **Settings** in the left column.
3. At the top of the property page, for the **Service Configuration** field, select **Cloud**.
4. In the **Microsoft.ServiceBus.ConnectionString** row, set the **Type** field to **String**.
5. Paste the connection string that you saved in Using the Management Portal, Task 2 into the **Microsoft.ServiceBus.ConnectionString Value** field.
6. In the **StorageAccount** row, set the **Type** field to **Connection String**.
8. Click the ellipses (...) at the end of the row. The **Create Storage Connection String** dialog box appears. Select **Manually entered credentials**.
9. Paste the name of your storage account in the **Account name** field.
10. Paste the name of the primary access key in the **Account key** field. Click **OK**.
11. In the **BOVWebSite** field, enter `http://bov.bioinfo.cas.unt.edu/cgi-bin/parseBlast.cgi`. This is the location of the BLAST output visualization (BOV) tool.
12. In the **Value** field of **HubAddress** enter the address where you are planning to deploy the cloud service. This address has the following format: `http://nameofyourcloudservice.cloudapp.net`.
13. Close the **BLAST.Compute** property page.
14. Repeat steps 2 through 11 for the **BLAST.Web** role.

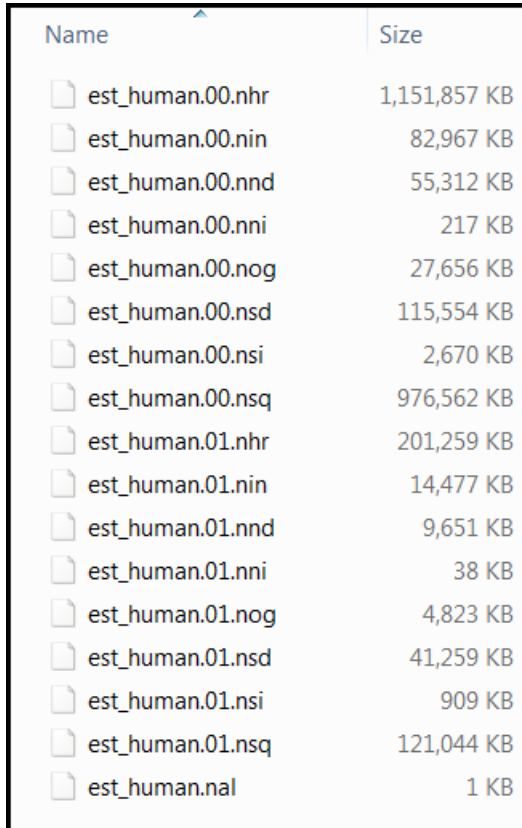
Task 3: Uploading the BLAST database files to blob storage

Follow these steps to upload the two BLAST databases that are required to perform the BLAST jobs to a blob in your Microsoft Azure storage account.

1. If you haven't already done so, download and install the [Microsoft Azure Explorer Freeware](#) application from CloudBerry Lab.

Installing the Microsoft Azure BLAST Example

2. Download and extract the human genome database files from the NCBI FTP server.
 - a. Browse to the NCBI FTP server at <ftp://ftp.ncbi.nih.gov/blast/db>.
 - b. Download the files named **est_human.00.tar.gz** and **est_human.01.tar.gz**.
 - c. Extract both files to the same folder. After extracting the files, you should have 17 files, as shown in Figure 3.



Name	Size
est_human.00.nhr	1,151,857 KB
est_human.00.nin	82,967 KB
est_human.00.nnd	55,312 KB
est_human.00.nni	217 KB
est_human.00.nog	27,656 KB
est_human.00.nsd	115,554 KB
est_human.00.nsi	2,670 KB
est_human.00.nsq	976,562 KB
est_human.01.nhr	201,259 KB
est_human.01.nin	14,477 KB
est_human.01.nnd	9,651 KB
est_human.01.nni	38 KB
est_human.01.nog	4,823 KB
est_human.01.nsd	41,259 KB
est_human.01.nsi	909 KB
est_human.01.nsq	121,044 KB
est_human.nal	1 KB

Figure 3: Human genome database files downloaded from the NCBI FTP server

3. Open **CloudBerry Explorer for Azure Blob Storage**.
4. Open the **File** menu and select **Azure Blob Storage Accounts**. The **Account Registration** dialog will appear, as shown in Figure 4.

Installing the Microsoft Azure BLAST Example

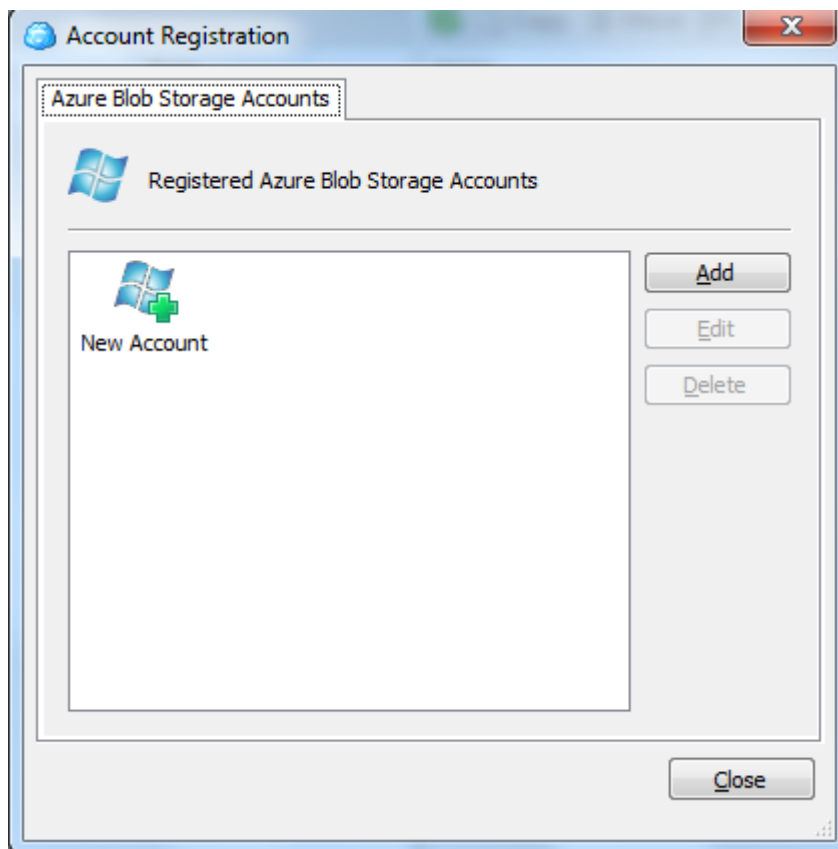


Figure 4: Account Registration dialog box

5. Click **Add** and enter the display name of the storage account, the storage account name, and the primary access key of the account. The display name is any friendly name you choose. The storage account name is the name of your Microsoft Azure storage account. It and your primary access key are available from the Management Portal. See the previous task, step 6, for how to retrieve this information.
6. Click **Test Connection** and wait for the approval message. Close the approval message, click **OK** to add the storage account, and then click **Close**.
7. In the left pane, in the Source field, select the **display name** of your storage account.
8. You should now see your blob storage in the left pane of the application, and your machine ("My Computer") in the right pane.
9. Create a new container in the blob by clicking the **New Container** icon in the left pane, as shown in Figure 5.



Figure 5: The New Container icon

10. In the **Create New Container** dialog, set the container name to **inputncbi**, and select **Full public read access** from the **Access control** options, as shown in Figure 6. Click **OK**.

Installing the Microsoft Azure BLAST Example

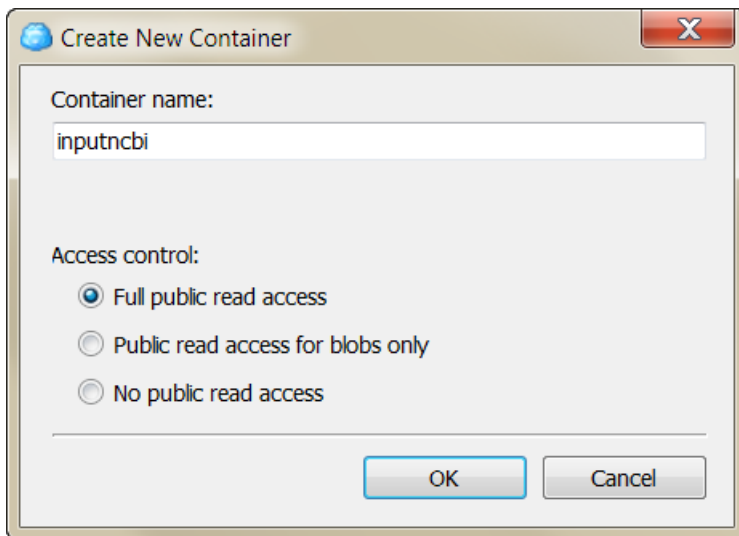


Figure 6: The Create New Container dialog box

11. Right-click the **inputncbi** container and select **Open** to view its contents (it should be empty).
12. In the right pane, click the **Folder** icon. Navigate to the **Source\Input** folder that is in the **BLAST** sample folder.
13. Select all the files (200 files) from the **Input** folder and click **Copy**. Click **Yes** in the confirmation message that appears, and then wait for the copy procedure to complete.
14. Repeat steps 9 and 10 to create another new container and name it **ncbi**.
15. In the left pane, select the **Root** folder in the address path to move to the root of the blob.
16. Locate the **ncbi** container in the list of containers. Right-click it and select **Open** to see its content (it should be empty).
17. In the right pane, navigate to the folder that contains the 17 extracted database files. Click **OK**. The files in the folder appear.
18. Select all the files and click **Copy**. Click **Yes** in the confirmation message that appears, and then wait for the copy procedure to complete. The database is about 2.5 GB, so this operation may take some time, depending on your bandwidth.
19. After the upload completes, close the CloudBerry Explorer application.

Task 5: Creating a table

Follow these steps to create table storage that will hold the statuses of the BLAST jobs that you submit.

1. In Visual Studio, click **View**. Click **Server Explorer**. The **Server Explorer** view appears.
2. Expand **Microsoft Azure**. Expand **Storage**.
3. If your storage account does not appear, right-click **Storage**. The **Add New Storage Account** dialog appears.
4. Enter your storage account name in the **Account name** field.
5. Enter your storage account's primary key in the **Account key** field. Click **OK**.

Installing the Microsoft Azure BLAST Example

6. Your storage account appears in the Server View.
7. Expand the storage account node.
8. Right-click **Table**. Select **Create Table**.
9. The **Create Table** dialog appears. Enter the name **SearchTask**. Click **OK**.
10. The **SearchTask** table appears in Server Explorer.

Publishing the BLAST Demo

In this section, you publish (deploy) the BLAST cloud project to Microsoft Azure. Once the project is published, you can adjust the Microsoft Azure environment to fit your needs.

Task 1: Publishing the BLAST project

Follow these steps to publish BLAST project.

1. In Visual Studio, go to **Solution Explorer**. Right-click the **BLAST** project. **Select Publish**.
 - a. The **Microsoft Azure Publish Sign In** dialog appears.
 - b. Click **Sign in to download credentials**.
 - c. A browser window opens that says your subscription file is being generated and the download will begin shortly.

When the download is complete, you are asked if you want to open or save the file. Click **Save As**. Make a note of where you saved the file because you will import it into Visual Studio. Note that this file contains your private download credentials. Once you have imported it to Visual Studio, you will probably want to delete this sensitive information.

- d. In Visual Studio, in the **Microsoft Azure Publish Sign In** dialog, click **Import**.
- e. Navigate to the location where you saved the credentials file and select it.
- f. In the **Microsoft Azure Publish Sign In** dialog, click **Next**.
- g. The **Microsoft Azure Publish Settings** dialog appears.
- h. Click **Next**. The **Microsoft Azure Publish Summary** appears. Click **Publish**.
- i. Select the **Microsoft Azure Activity Log** to see the progress of the deployment.

After you have published the BLAST demo, you can reconfigure it by using the Microsoft Azure Management Portal. For example, you can modify your storage account, Service Bus namespace, as well as BOV website addresses, as shown in Figure 7.

Installing the Microsoft Azure BLAST Example

blast.web

SETTINGS

StorageAccount	DefaultEndpointsProtocol=https;AccountName=azl
Microsoft.ServiceBus.ConnectionString	Endpoint=sb://mybooks.servicebus.windows.net;St
BOVWebSiteViewAddress	http://bov.bioinfo.cas.unt.edu/cgi-bin/viewhits.cgi?f

CERTIFICATES

None of the certificate settings are configurable.

blast.compute

SETTINGS

Microsoft.ServiceBus.ConnectionString	Endpoint=sb://mybooks.servicebus.windows.net;St
StorageAccount	DefaultEndpointsProtocol=https;AccountName=azl
BOVWebSite	http://bov.bioinfo.cas.unt.edu/cgi-bin/parseBlast.cg
HubAddress	http://blaster.cloudapp.net/

Figure 7: Settings for the BLAST demo that you can modify in the Microsoft Azure Management Portal

You can also increase the number of worker roles if you need to. Because each worker role instance only handles one task at a time, you can perform multiple tasks in parallel if you create multiple instances, as shown in Figure 8.

blast.compute

EDIT SCALE SETTINGS FOR SCHEDULE [set up schedule times](#)

SCALE BY METRIC NONE CPU QUEUE

INSTANCE COUNT SMALL (1 CORE, 1.75 GB MEMORY)

instance(s)

■ CORES USED BY ROLE ■ CORES USED BY ALL ■ AVAILABLE CORES

Figure 8: Creating multiple instances so that you can perform multiple tasks in parallel

Testing the Cloud Service

In this section, you test the BLAST cloud service to make sure that it is working.

Task 1: Create a new search request

Follow these steps to create a new BLAST search request.

Installing the Microsoft Azure BLAST Example

1. Open a browser. Navigate to the address of your cloud service, which is **http://[cloud service name].cloudapp.net**. The NCBI BLAST Microsoft Azure Sample page opens
2. Click the **New Task** link.
3. On the **New Task** page, pick an input file you want to use, or enter a range of input files. The list of input files is made up of files that you uploaded to the *inputncbi* container. The system generates a test name based on the server's current UTC time, but you can change the name so that you can find it more easily. Figure 9 shows an example of the New Task page.
4. Click **Queue**.

NCBI BLAST Windows Azure Sample
Search Tasks New Task About Contact

New Task

Name

Input File or, Input a range of input file indexes:

Example:1-50, 108

Figure 9

5. Once you put the task on the queue, you are redirected to the task list page where you can observe a job's progress, delete a job, retry a failed job, and view task results. An example is shown in Figure 10.



Search Tasks

[Refresh](#) [Delete All Completed Tasks](#) [Delete All Queued Tasks](#)

TEST 2013-7-23-16-56-11-333 (input_50)	
<i>Worker Role [Compute_IN_0]: Search Completed.</i>	
TEST 2013-7-23-16-56-11-333 (input_49)	View Result Delete
<i>Completed</i>	
TEST 2013-7-23-16-56-11-333 (input_48)	View Result Delete
<i>Completed</i>	

Figure 10: Example of Search Task view

Note: Because the system needs to download the database files, the first task can take a long time. However, because the system caches the database files, subsequent jobs run much

- After a task completes, you can click **View Result** to view the search result. The link takes you to bov.bioinfo.cas.unt.edu to view a summary of the results. Figure 11 shows a sample summary page.



Summary of BLAST output file

The table below shows all query sequences and the number of hits for each query. Click on the query sequence you are interested in to display the list of hits in a new table.

Number of Queries found in the BLAST output file with Hits: 1

Name of the Query sequence	No. of Hits
gil26452 emb Z19704.1 	3

Figure 11: Sample search result summary page

- On the summary page, click the link to expand the hit sequences. Figure 12 shows an example page.

Installing the Microsoft Azure BLAST Example

BLAST OUTPUT VISUALIZATION TOOL

HOME | TUTORIAL | SOFTWARE | CONTACT

Summary of BLAST output file

The table below shows all query sequences and the number of hits for each query. Click on the query sequence you are interested in to display the list of hits in a new table.

Number of Queries found in the BLAST output file with Hits: 1

Name of the Query sequence	No. of Hits
gi 26452 emb Z19704.1 	3

List of Hits for query sequence 'gi|26452|emb|Z19704.1|'. The table below provides the Score and E-value for each hit. Click on the hit sequence of interest to display details.

Hit Sequences	Score	E Value
emb Z19704.1 	424	2e-116
emb AL534569.3 	418	9e-115
emb AL534568.3 	351	9e-095

[TOP](#)

Figure 12: Expanded search result summary page showing hit sequences

8. Click any of the hit sequences to see its corresponding high-scoring sequence pair (HSP) image. Figure 13 shows an example.

BLAST OUTPUT VISUALIZATION TOOL

HOME | TUTORIAL | SOFTWARE | CONTACT

gi|26452|emb|Z19704.1| :: emb|AL534569.3| :: HSP Image

Query : gi|26452|emb|Z19704.1| (total length = 229)

Hit : emb|AL534569.3| (total length = 773)

HSP Select | Zoom | Help

High-Scoring Pair (HSP) View

filter by E-value: Filter

Refresh the Image

CheckAll	Highlight	Alignment	E-Value	Query		Hit	
<input type="checkbox"/>				Start	End	Start	End
<input checked="" type="checkbox"/>	Off	View	9e-115	1	226	747	522

Refresh the Image

[Download Image](#)

Figure 13: Example of hit sequence HSP image

Task 2: Task management (optional)

In this task, you learn some ways to manage BLAST jobs. This task is optional because it is specifically about BLAST, and not about Microsoft Azure. By using the Search Task view (Figure 10), you can observe job statuses and perform simple management operations.

1. Click **Delete** for a particular job to remove it from the queue. Note you can't delete a job that is being executed.
2. If you have a task that failed, click **Retry** to resubmit the job.
3. Click **View Result** to view the results of a completed job.
4. Click **Refresh** at the top of the page to refresh the page.
5. Click **Delete All Completed Tasks** to delete all completed tasks.
6. Click **Delete All Queued Tasks** to delete all queued tasks. These are tasks that haven't started yet.

Installing the Microsoft Azure BLAST Example

Summary

After running the BLAST sample, you should have learned how to perform the following operations.

- Use the Microsoft Azure Management Portal to create a cloud service, a storage account, a Service Bus namespace, and a queue
- Use the Microsoft Azure Management Portal to find primary keys
- Populate Microsoft Azure Blob storage
- Create Microsoft Azure Table storage in Visual Studio
- Publish an application to Microsoft Azure
- Submit a BLAST job from the web
- Check a running BLAST job's status

© 2014 Microsoft Corporation. All rights reserved. Except where otherwise noted, content on this site is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.