

Final Project
Azure Batch

Martin Bertrand

Deep Azure
Dr. Zoran B. Djordjević

The problem...

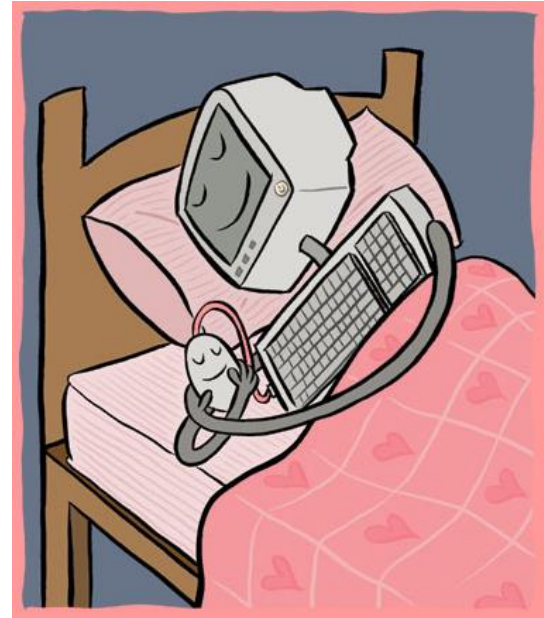
- You are receiving a steady flow of X12 EDI files that need to be transformed to XML format for further processing.
- Once a week a client sends you a LARGE number of X12 files (40 000), which overwhelms your X12 to XML conversion infrastructure.
- How would you solve such problem?

The traditional solution

- Your infrastructure needs to be large enough to handle the highest workload possible.
 - Analysis and architecture;
 - Hardware purchase (RFP, PO, ...);
 - Computer room infrastructure (floor space, power, UPS, ...)
 - Network infrastructure;
 - Systems installation and configuration;
 - Software installation and configuration;
 - Support all this!

The traditional solution - 2

- Most of the time, that infrastructure would sit there doing nothing, since the large bursts of files are infrequent.
- This implies costs and waste of resources.



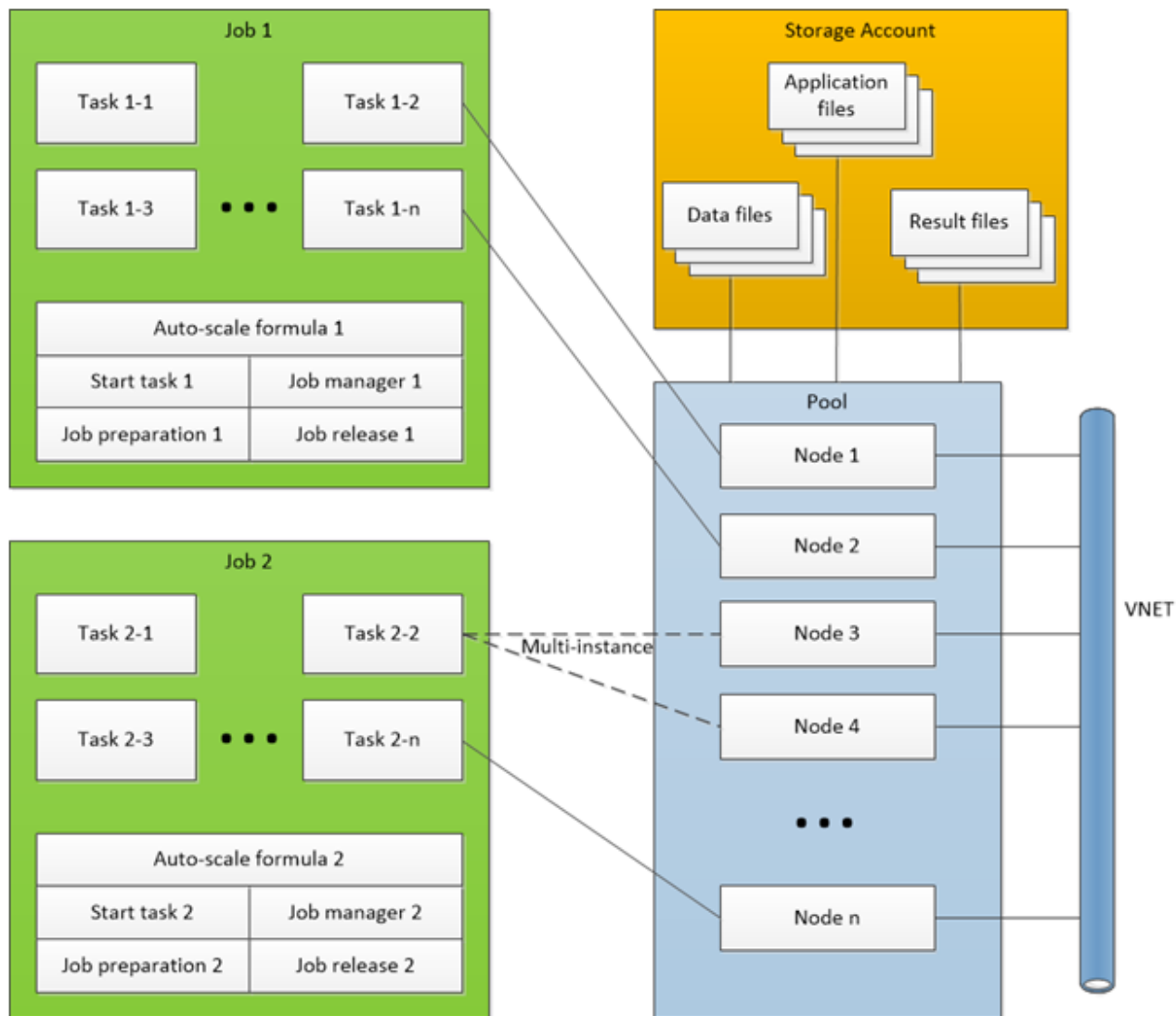
The Azure Batch solution

- Azure Batch is a platform created to run parallel, high-performance computing.
- It also offers auto-scaling of compute resources, to meet varying demands.
- This reduces costs and resources waste.

Azure Batch examples

- Batch processing
 - Invoices, billing
 - Inventory updates
 - Payroll
 - Financial reports
- High-performance computing applications
 - Computational Fluid Dynamic (CFD)
 - Deep Learning
 - Molecular Dynamics
 - Video processing

Azure Batch - Resources



Azure Batch - Jobs

- Application are broken down in to jobs.
- Jobs are collections of tasks and define on which compute node pool(s) they will run.
- Jobs also define
 - Tasks priority and constraints.
 - The auto-scaling formula (based on the number of queue tasks, completion rate, time, resources, other metrics).

Azure Batch - Tasks

- Tasks are units of computation.
- Each task can be executed on one or more compute nodes.
- They define:
 - What command to execute.
 - What files are required (application and data).
 - Environment variables.
 - Constraints.
 - Application packages or container images to use.
- Tasks can have dependencies between one another.
- The output of a task can be the input of another.

Azure Batch – Compute nodes

- Compute nodes are virtual machines (Windows or Linux) or cloud service VMs (Windows only).
- Provide CPU, memory and disks resources.
- Are all identical within a unique pool. Create other pools is different nodes are required.
- Can be :
 - accessed like a regular VM (RDP or SSH).
 - based on standard or custom images.
 - dedicated (more expensive but never pre-empted) or low-priority (less expensive, uses surplus capacity).
 - added to the pool (scaling up), or removed from the pool (scaling down), depending on the auto-scaling formula, defined at the job level.
 - created for each job, and deleted as soon as it is complete, or be created ahead of time, thus reducing the start time, but increasing costs.

Azure Batch - Files

- Types: application, data files (input) and result files (output).
- They are stored in a storage account blob.
- Application files are downloaded on compute nodes and executed.
- Data files are downloaded and processed by the application.
- Result files are uploaded back to the storage account.
- Files associated to a compute node are lost when the node is destroyed.

Azure Batch - Applications

- They can be managed via packages.
- Can have many versions of an application used at the same time.
- Can be defined at the job, or task level.
- Job level: deployed to all nodes in the pool.
- Task level: only on nodes that are defined to run that particular package.

Azure Batch - Network

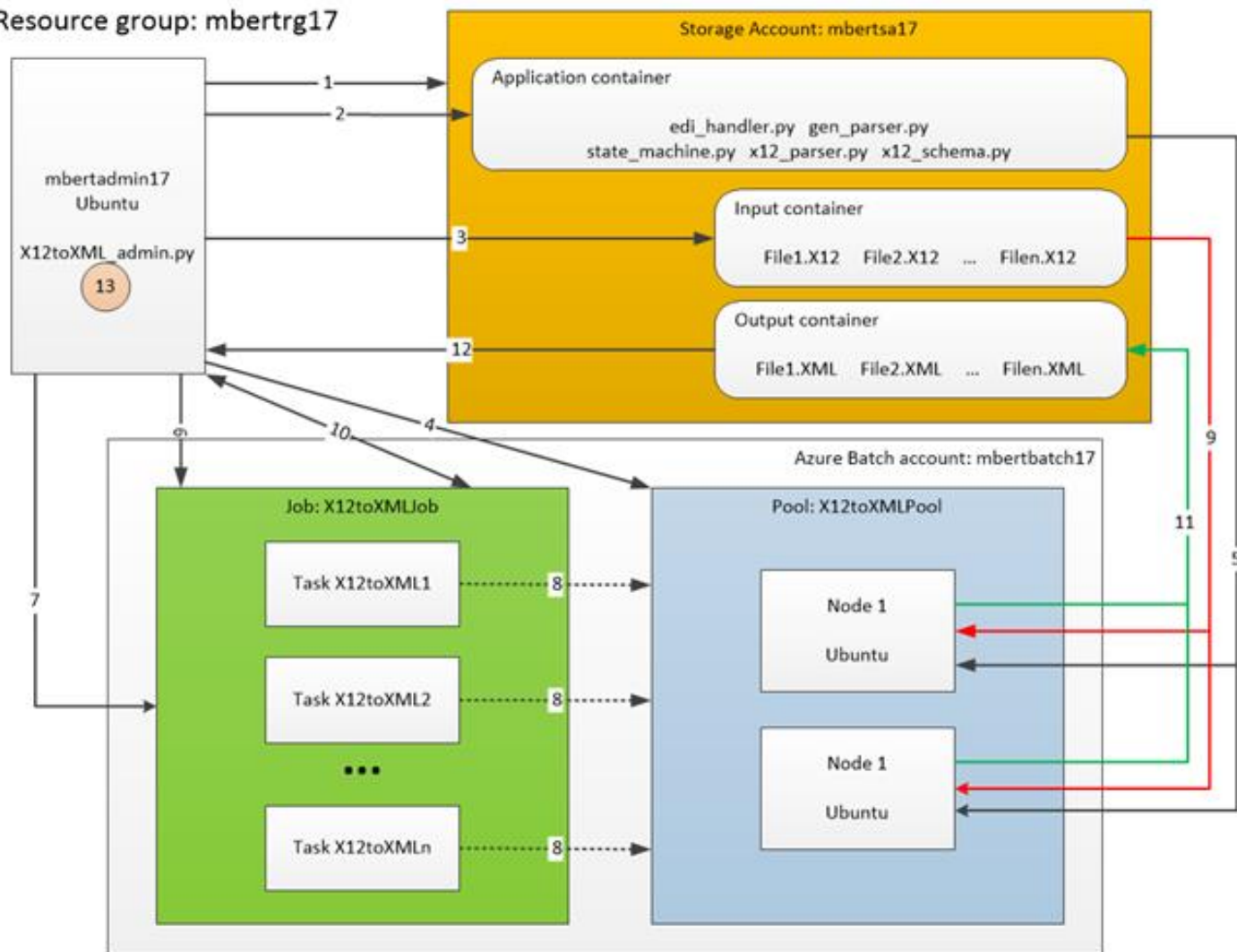
- All nodes must be in the same region, in the same batch account, under the same subscription.
- Network Security Groups must allow communications between batch services and the nodes.

Azure Batch - API

- Communicate with Azure Batch services.
- Create and manage nodes.
- Schedule jobs and tasks.
- Can be used via CLI, REST, .NET, Python, Node.js or Java.

X12 to XML conversion - Infrastructure

Resource group: mbertrg17



X12 to XML conversion – Infrastructure - 2

- An administration VM (Linux based) runs a Python script to administer Batch.
- This Python script uses the Azure Batch API to communicate with Batch and the storage account.
- The administration script creates required components and prepares files for processing using Batch.
- It also cleans up when the processing is done, to keep costs to a minimum.

X12 to XML conversion - Executing

- A Linux bash script creates the required components on Azure, using Azure CLI commands.
- It creates:
 - The resource group.
 - A storage account.
 - A batch account.
 - The administration VM, with associated external IP and network components.
 - It configures the administration VM to run the conversion script.
- It uploads the application and data files to the administration VM.

X12 to XML conversion – Executing - 2

- The Python administration script performs:
 - Create 3 containers in the storage account:
 - “application”: contains all the scripts that will perform the transformation.
 - “input”: will contain all the X12 format data files.
 - “output”: will contain the XML transformed files.
 - Upload the application and input files in the appropriate containers.
 - Create a pool that will contain compute nodes.
 - Create a job which will contain the tasks.
 - Create tasks inside the job. One task is created per input file. Each task uploads the converted file to the “output” storage container.
 - Waits for all tasks to complete.
 - Download the output files from the output container.
 - Delete the storage containers.
 - Delete the job.
 - Delete the pool.

X12 to XML conversion – Portal

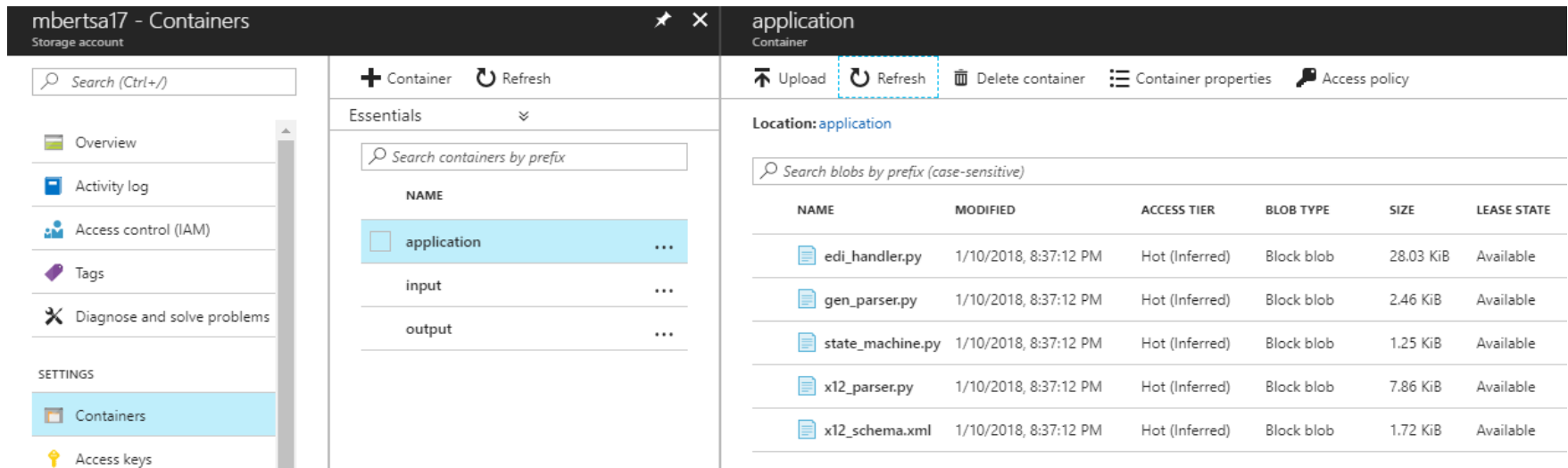
- While files are converted, it is possible to view the status of Batch resources via the Azure Portal. This view shows the Storage Account containers:

The screenshot shows the Azure Portal interface for the storage account 'mbertsa17'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and a SETTINGS section with Containers, Access keys, Configuration, and Custom domain. The main content area shows the 'Containers' view with a '+ Container' button and a 'Refresh' button. Below this, the storage account details are listed: Storage account 'mbertsa17', Status 'Primary: Available', Location 'Canada East', Subscription 'McKesson Deep Dive Training (7)', and Subscription ID '6f5d1e5e-5295-4b19-9069-76aa53bdb9c'. The Blob service endpoint is 'https://mbertsa17.blob.core.windows.net/'. A table below lists the containers:

NAME	LAST MODIFIED	PUBLIC ACCESS L...	LEASE STATE
application	1/10/2018, 8:25:43 PM	Private	Available
input	1/10/2018, 8:25:43 PM	Private	Available
output	1/10/2018, 8:25:43 PM	Private	Available

X12 to XML conversion – Portal - 2

- The application container contains the scripts:



The screenshot displays the Azure Storage portal interface for a storage account named 'mbertsa17 - Containers'. The left sidebar shows navigation options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and SETTINGS (Containers, Access keys). The main area is divided into two panes. The left pane shows the 'Essentials' section with a search bar and a list of containers: 'application', 'input', and 'output'. The 'application' container is selected. The right pane shows the 'application' container details, including a search bar and a table of blobs.

NAME	MODIFIED	ACCESS TIER	BLOB TYPE	SIZE	LEASE STATE
edi_handler.py	1/10/2018, 8:37:12 PM	Hot (Inferred)	Block blob	28.03 KiB	Available
gen_parser.py	1/10/2018, 8:37:12 PM	Hot (Inferred)	Block blob	2.46 KiB	Available
state_machine.py	1/10/2018, 8:37:12 PM	Hot (Inferred)	Block blob	1.25 KiB	Available
x12_parser.py	1/10/2018, 8:37:12 PM	Hot (Inferred)	Block blob	7.86 KiB	Available
x12_schema.xml	1/10/2018, 8:37:12 PM	Hot (Inferred)	Block blob	1.72 KiB	Available

X12 to XML conversion – Portal - 3

- The input container contains the X12 files:

The screenshot displays the Azure Portal interface for a storage account named 'mberts17'. The left sidebar shows the 'Containers' section under 'SETTINGS'. The main area is divided into two panes. The left pane, titled 'input Container', shows a list of containers: 'application', 'input' (selected), and 'output'. The right pane, titled 'input', shows a list of blobs. The location is set to 'input'. The blobs are listed in a table with columns: NAME, MODIFIED, ACCESS TIER, BLOB TYPE, SIZE, and LEASE STATE. The blobs are 'File1.X12' through 'File9.X12', all created on 1/10/2018 at 8:25:43 PM, with an access tier of 'Hot (Inferred)', a blob type of 'Block blob', and a size of 514 B or 852 B. All blobs are in an 'Available' lease state.

NAME	MODIFIED	ACCESS TIER	BLOB TYPE	SIZE	LEASE STATE
File1.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	514 B	Available
File2.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	852 B	Available
File3.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	672 B	Available
File4.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	908 B	Available
File5.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	514 B	Available
File6.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	514 B	Available
File7.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	514 B	Available
File8.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	514 B	Available
File9.X12	1/10/2018, 8:25:43 PM	Hot (Inferred)	Block blob	514 B	Available

X12 to XML conversion – Portal - 4

- A batch pool is created:

The screenshot shows the 'mbertbatch17 - Pools' management page in the Azure portal. On the left is a navigation sidebar with links to Properties, Quotas, Storage account, Keys, and Locks. The main area displays a table of batch pools. At the top of the main area are controls for '+ Add', 'Columns', and 'Refresh'. Below these is a dropdown menu set to 'All pools' and an 'Advanced query' section. A search bar prompts the user to 'Search for pool by ID or display name'. The table below contains one entry, 'X12toXMLPool', with the following details:

POOL ID	DEDICATED NODES	LOW-PRIORITY NODES	CURRENT CORES	VM SIZE	ALLOCATION STATE
X12toXMLPool	0 → 1	0	0	basic_a1	Resizing

X12 to XML conversion – Portal - 5

- A job is created:

The screenshot shows the 'Jobs' section of the Azure portal for the batch account 'mberbatch17'. The interface includes a left-hand navigation pane with options like Properties, Quotas, Storage account, Keys, and Locks. The main area displays a table of jobs. At the top, there are controls for adding, managing columns, and refreshing the data. A dropdown menu shows 'All jobs'. Below this, there is an 'Advanced query' section with a filter input. The table itself has columns for ID, STATE, POOL, and CREATED. One job is listed: 'X12toXMLJob' with state 'Active', pool 'X12toXMLPool', and creation time 'Jan 10, 21:26:48'.

ID	STATE	POOL	CREATED
X12toXMLJob	Active	X12toXMLPool	Jan 10, 21:26:48

X12 to XML conversion – Portal - 6

- Tasks are added to the job:

X12toXMLJob - Tasks

Search (Ctrl+/)

Overview

GENERAL

Properties

Environment settings

Metadata

✓ Tasks

Preparation tasks

Release tasks

SETTINGS

Priority

Constraints

Pool information

Auto complete settings

Add

Columns

Refresh

Task counts: Active: 9, Running: 0, Completed: 0, Succeeded: 0, Failed: 0

All tasks

▼

Advanced query ▼

Filter by task ID

TASK	STATE	CREATED	EXIT CODE
topNtask0	Active	Jan 12, 19:47:48	
topNtask1	Active	Jan 12, 19:47:48	
topNtask2	Active	Jan 12, 19:47:48	
topNtask3	Active	Jan 12, 19:47:48	
topNtask4	Active	Jan 12, 19:47:48	
topNtask5	Active	Jan 12, 19:47:48	
topNtask6	Active	Jan 12, 19:47:48	
topNtask7	Active	Jan 12, 19:47:48	
topNtask8	Active	Jan 12, 19:47:48	

Martin Bertrand

24

X12 to XML conversion – Portal - 7

- As tasks complete the conversion, result files are uploaded to the Storage Account, in the “output” container:

The screenshot shows the Azure Storage Portal interface for a container named 'output'. The left sidebar shows a list of containers: 'application', 'input', and 'output' (which is selected and highlighted in blue). The main area displays the 'output' container's contents, including a search bar and a table of blobs. The table has columns for NAME, MODIFIED, ACCESS TIER, BLOB TYPE, SIZE, and LEASE STATE. There are nine rows of data, each representing a converted XML file (File1_1.xml through File9_1.xml). The 'Refresh' button in the top toolbar is highlighted with a dashed blue box.

output
Container

+ Container Refresh Upload Refresh Delete container Container properties Access policy

Location: output

Search containers by prefix

NAME	MODIFIED	ACCESS TIER	BLOB TYPE	SIZE	LEASE STATE
File1_1.xml	1/20/2018, 3:22:55 PM	Hot (Inferred)	Block blob	3.34 KiB	Available
File2_1.xml	1/20/2018, 3:23:01 PM	Hot (Inferred)	Block blob	6.02 KiB	Available
File3_1.xml	1/20/2018, 3:22:57 PM	Hot (Inferred)	Block blob	4.75 KiB	Available
File4_1.xml	1/20/2018, 3:22:47 PM	Hot (Inferred)	Block blob	6.28 KiB	Available
File5_1.xml	1/20/2018, 3:22:32 PM	Hot (Inferred)	Block blob	3.34 KiB	Available
File6_1.xml	1/20/2018, 3:22:51 PM	Hot (Inferred)	Block blob	3.34 KiB	Available
File7_1.xml	1/20/2018, 3:22:43 PM	Hot (Inferred)	Block blob	3.34 KiB	Available
File8_1.xml	1/20/2018, 3:22:39 PM	Hot (Inferred)	Block blob	3.34 KiB	Available
File9_1.xml	1/20/2018, 3:22:35 PM	Hot (Inferred)	Block blob	3.34 KiB	Available

X12 to XML conversion – Results

- Result files are downloaded from the “output” container of the Storage Account by the administration VM.
- If all goes well, these will now be XML representations of X12 files.

```
<?xml version="1.0" ?>
<Interchange>
  <AuthorizationInformation id="" qualifier=""/>
  <SecurityInformation id="" qualifier=""/>
  <Sender id="SENDER" qualifier="ZZ"/>
  <Receiver id="RECEIVER" qualifier="ZZ"/>
  <DateTime date="041201" time="1200"/>
  <EdiControlInformation number="000000101" standards_id="U" version_number="00305"/>
  <AcknowledgementRequested id="1"/>
  <TestIndicator id="P"/>
  <FunctionalGroup>
    <FunctionalIdentifier code="PO" name="Purchase Order"/>
    <Sender id="SENDER"/>
    <Receiver id="RECEIVER"/>
    <DateTime date="041201" time="1200"/>
    <Control number="101"/>
    <EdiIndustryIdentifier code="X" id="003050"/>
    <TransactionSet>
      <Id code="850" name="Purchase Order"/>
      <ControlNumber value="000000101"/>
      <PoInfo>
        <Purpose code="22" name="Information Conv"/>
      </PoInfo>
    </TransactionSet>
  </FunctionalGroup>
</Interchange>
```

Troubleshooting Batch

- The administration script only shows the tasks are completed, but does not show the exit code. While it is running, the status and exit codes can be seen in the console:

X12toXMLJob - Tasks

Search (Ctrl+J)

Overview

GENERAL

Properties

Environment settings

Metadata

Tasks

Preparation tasks

Release tasks

SETTINGS

Priority

Constraints

Pool information

Auto complete settings

+ Add

Columns

Refresh

Task counts: Active: 7, Running: 0, Completed: 2, Succeeded: 0, Failed: 2

All tasks

Advanced query

Filter by task ID

TASK	STATE	CREATED	EXIT CODE
topNtask0	Completed	Jan 12, 20:36:19	1
topNtask1	Active	Jan 12, 20:36:19	
topNtask2	Completed	Jan 12, 20:36:19	1
topNtask3	Completed	Jan 12, 20:36:19	1
topNtask4	Active	Jan 12, 20:36:19	
topNtask5	Active	Jan 12, 20:36:19	
topNtask6	Active	Jan 12, 20:36:19	
topNtask7	Completed	Jan 12, 20:36:19	1
topNtask8	Active	Jan 12, 20:36:19	

Martin Bertrand

27

Troubleshooting Batch - 2

- You can verify the properties of a task, even after it is completed.
- The most crucial property is the command line.

The screenshot shows a web-based interface for viewing task properties. The title bar reads 'topNtask0 - Properties'. On the left, there is a sidebar with a search bar labeled 'Search (Ctrl+)' and two menu items: 'Overview' and 'Properties'. The 'Properties' item is selected and highlighted in light blue. The main content area is titled 'General' and contains three rows of properties: 'ID' with the value 'topNtask0', 'Display name' with an empty field, and 'Command line' with the value '/bin/bash -c 'set -e; set -o pipefail; python \$AZ_BATCH_NODE_SHARED_DIR/gen_parser.py --filepath File4.X12 --stora...'. Each property value is displayed in a light gray box with a blue document icon to its right. A 'Refresh' button with a circular arrow icon is located at the top left of the main content area.

- Ensure the command line is correct and the number of arguments to the application is correct.

Troubleshooting Batch - 3

- It is also possible to view certain files on the node that ran a task.

topNtask0 - Files on node

Columns

Refresh

Filter by file name ...

FILE NAME	SIZE	CONTENT TYPE	LAST MODIFIED
stderr.txt	0 Bytes	text/plain	Jan 12, 19:47:58
wd/File4.X12	908 Bytes	application/octet-stream	Jan 12, 19:47:58
stdout.txt	74 Bytes	text/plain	Jan 12, 19:47:58

Overview

GENERAL


Properties

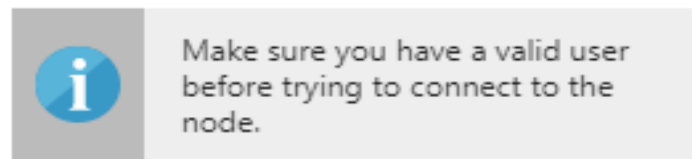
Environment settings

Application packages

- The most useful for debugging are “stderr.txt” and “stdout.txt”, which contain the output of the conversion script.

Troubleshooting Batch - 4

- It is also possible to SSH connect directly to a node for further troubleshooting and tests.
- Select “Connect” in the sub-menu displayed when clicking on the  icon.
- You will have the option of adding a user.
- Azure will then display connect information.



Username




IP



Port



SSH command line



Batch Labs

- Azure Portal is the primary tool for administering Azure Batch.
- Microsoft also provides Batch Labs (<https://azure.github.io/BatchLabs>).
- It can be used to create, debug and monitor Batch applications.
- Batch Labs dash view while running the demo application:

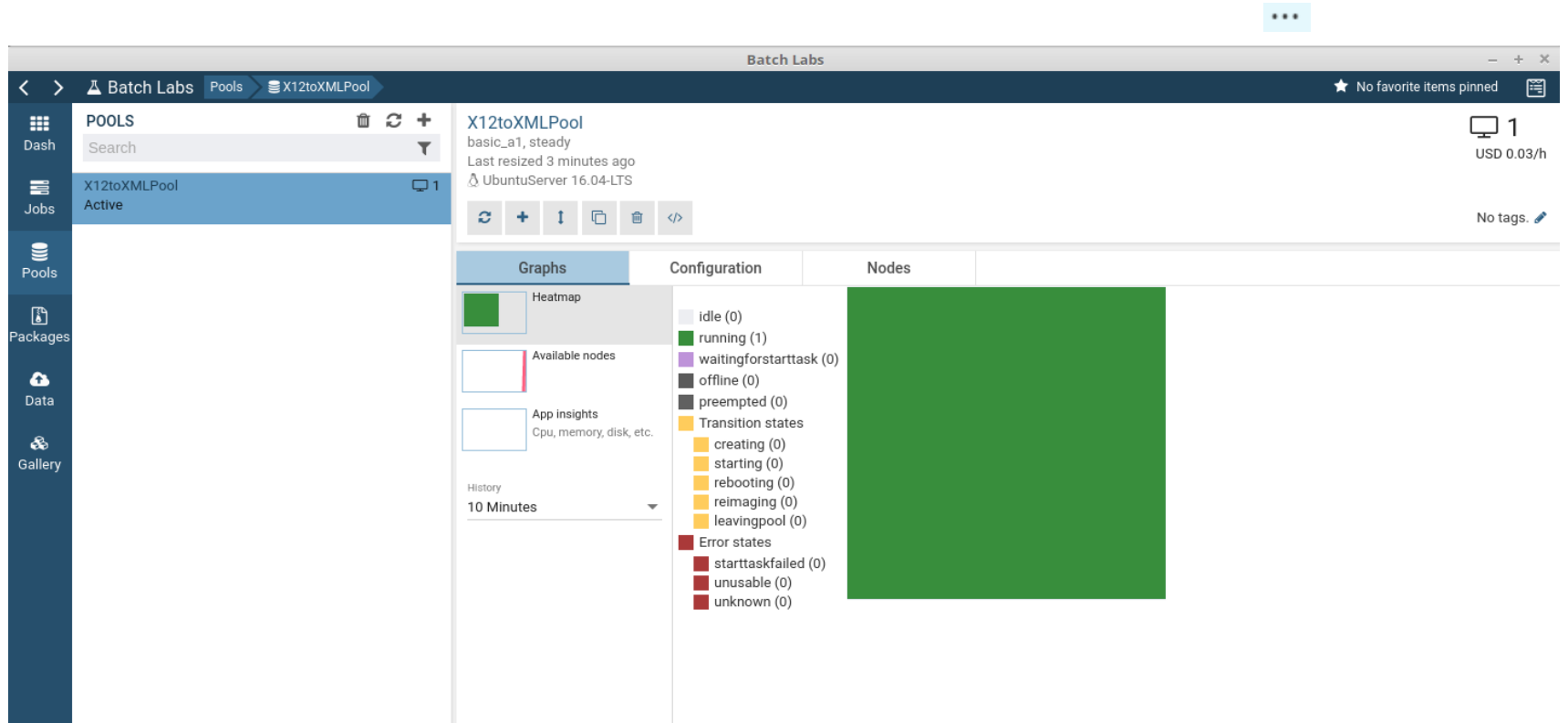
The screenshot shows the Batch Labs dashboard for the account **mbertbatch17**. The interface includes a left sidebar with navigation options: Dash, Jobs, Pools, Packages, Data, and Gallery. The main content area is divided into several sections:

- BATCH ACCOUNTS**: A section with a search bar and a list of accounts. The selected account is **mbertbatch17** (canadaeast).
- Account Details**: A summary of the account's status and usage.

Pool usage	Job quota
1/20 (5%)	20
Dedicated core usage	Low-pri core usage
1/100 (1%)	0/100 (0%)
- Storage account**: A section showing the storage account **mbertsai17** with an **Edit** button.
- Job status**: A section showing the status of jobs. The current job is **X12toXMLJob** with a status of **active**. A link to **View all jobs** is provided.
- Pool status**: A section showing the status of pools. The current pool is **X12toXMLPool** with a status of **1**. A link to **View all pools** is provided.
- App packages**: A section showing the status of application packages. A message states: "There are no application packages in the account. Click [here](#) to add one." A link to **View all packages** is provided.

Batch Labs - 2

- Batch Labs Pools view while running the demo application:



Batch Labs - 3

- Batch Labs Jobs view while running the demo application:

The screenshot shows the Batch Labs interface for the X12toXMLJob. The left sidebar contains navigation links: Dash, Jobs, Pools, Packages, Data, and Gallery. The main view is titled 'X12toXMLJob' and shows it is 'Active' with a pool of 'X12toXMLPool'. A 'Job statistics' section displays a gauge chart with '9 Queued', '0 Running tasks', and '0 Failed' tasks, resulting in a '0.00%' completion rate. Below this is a table of tasks.

Id	State	Created	Started	Completed	Exit code
topNtask0	active	4 minutes ago			
topNtask1	active	4 minutes ago			
topNtask2	active	4 minutes ago			
topNtask3	active	4 minutes ago			
topNtask4	active	4 minutes ago			
topNtask5	active	4 minutes ago			
topNtask6	active	4 minutes ago			
topNtask7	active	4 minutes ago			
topNtask8	active	4 minutes ago			

YouTube URLs, GitHub URL

- Two minute (short):
- 15 minutes (long):
- GitHub Repository with all artifacts: