



March 28, 2019 — 8:30 AM - 5:00 PM | Detroit, Michigan

Azure Logic Apps

Learn, architect, and
develop solutions on Azure

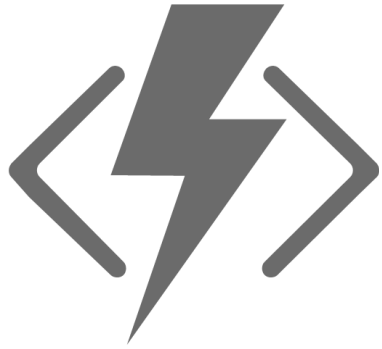
#AzureDevDays
for developers, by developers



Learn.
Connect.
Explore.

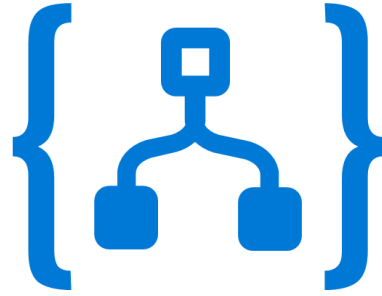
Overview

Serverless in Azure



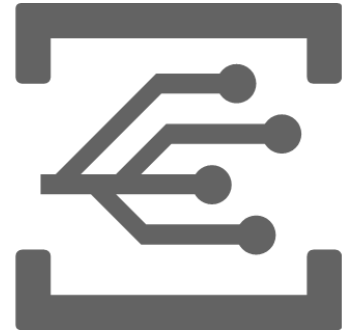
Functions

Serverless compute



Logic Apps

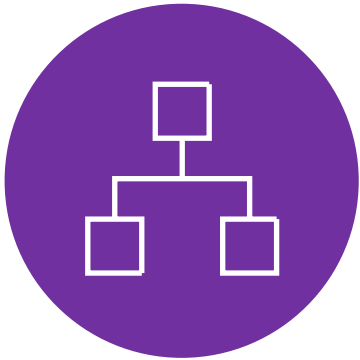
Serverless workflow



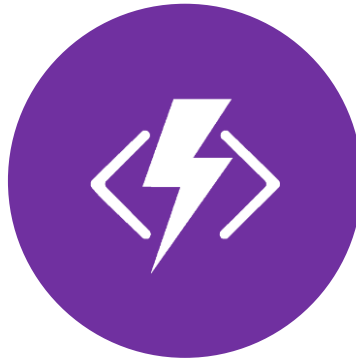
Event Grid

Serverless events

Logic App capabilities



Visual workflow
designer



Workflows are a
series of
conditions and
actions*

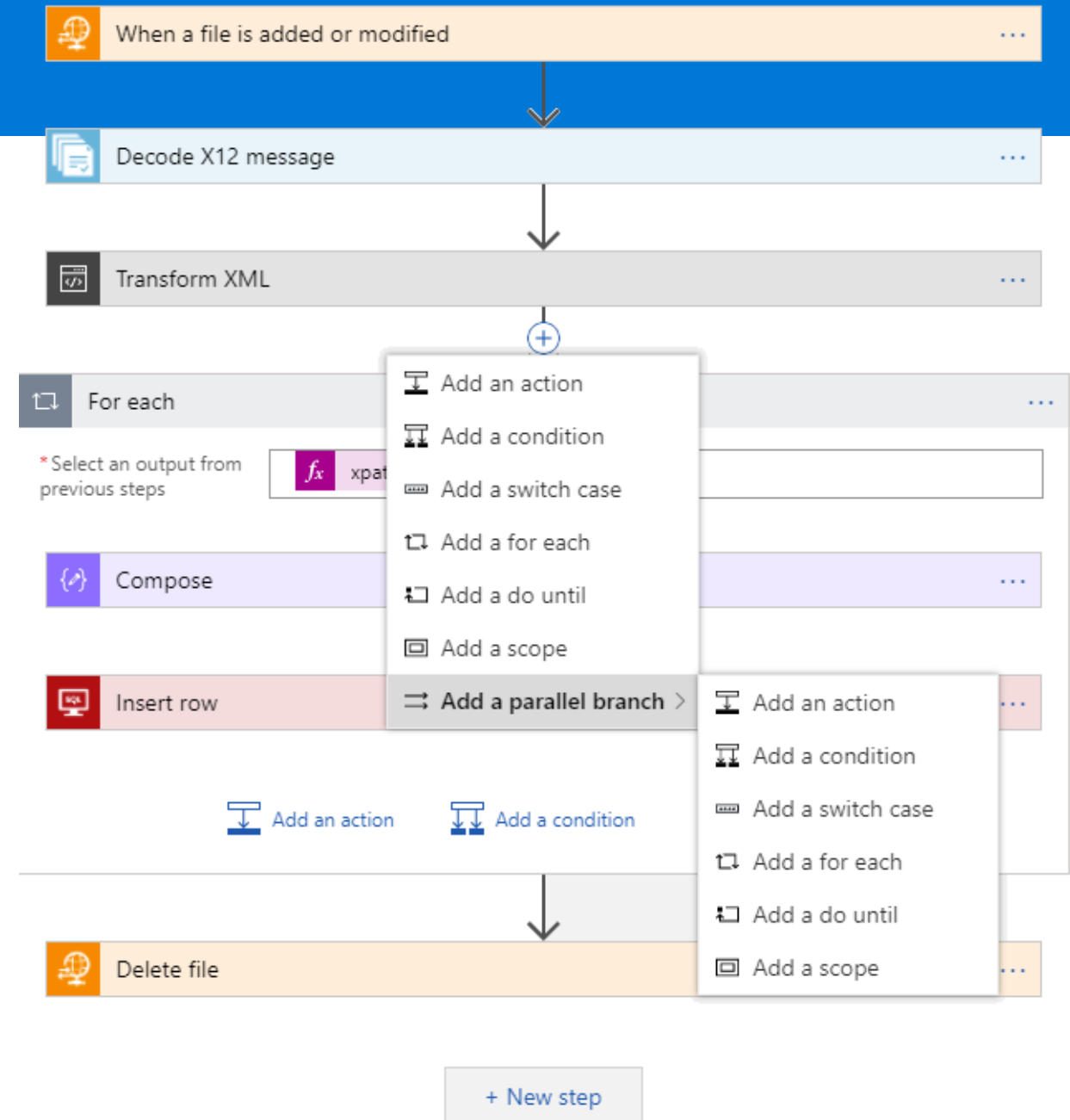


170+ Connectors
to apps and data

*Actions can include Functions

Azure Logic Apps

- Fast integration using innovative Visual Designer
- Easy workflow creation with triggers and actions
- More than 200+ connectors to mashup applications, data and services
- Built for mission critical 24x7 Enterprise Integration
- Devops built-in: Create, deploy, manage and monitor



Connected with over 200 connectors



Cloud APIs and platform functionality

Over 200 out of box connectors

SaaS, on-prem, protocols, B2B and message manipulation

Hybrid connectivity

Hosted and managed within the platform

Scales to meet your needs

First class designer experience



Custom Connectors

Access any REST/SOAP API

Cloud or on-premises

Simple creation wizard

Connections and managed secrets

First class designer experience



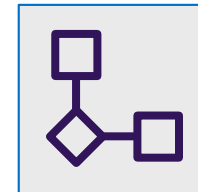
API connections

Authenticate once and reuse

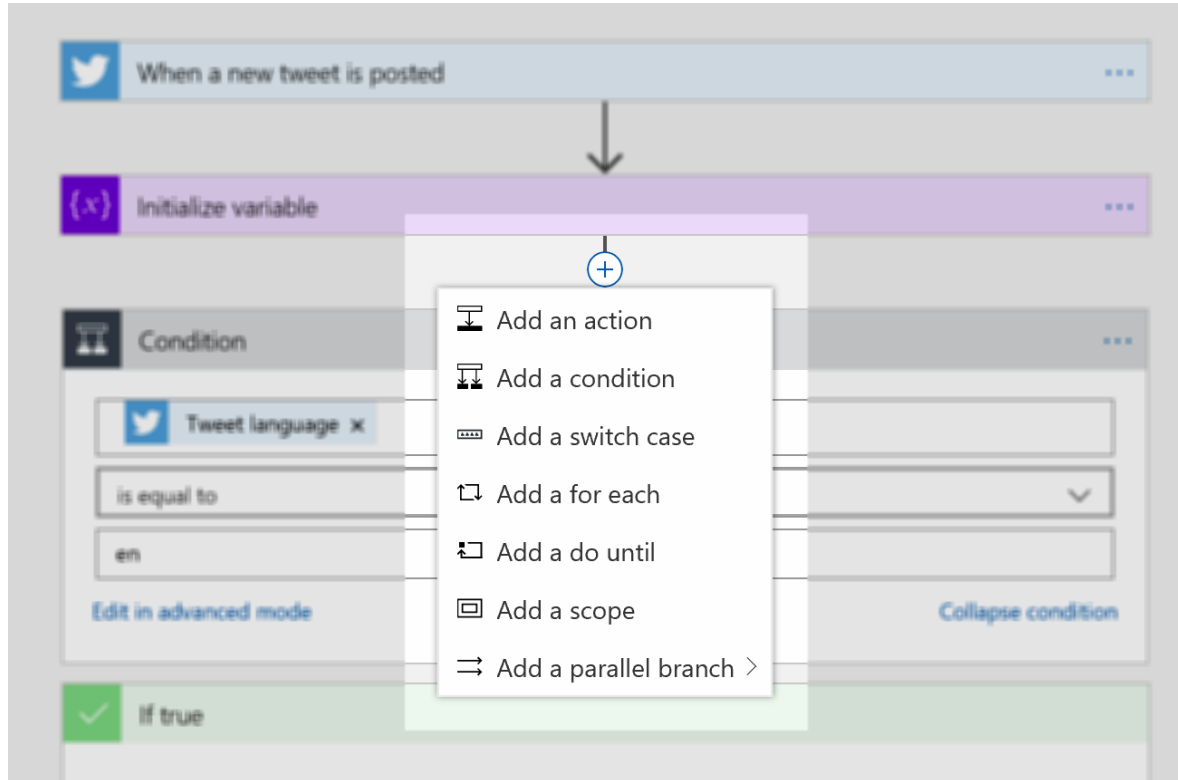
Differentiate connection configuration

Simple to deploy

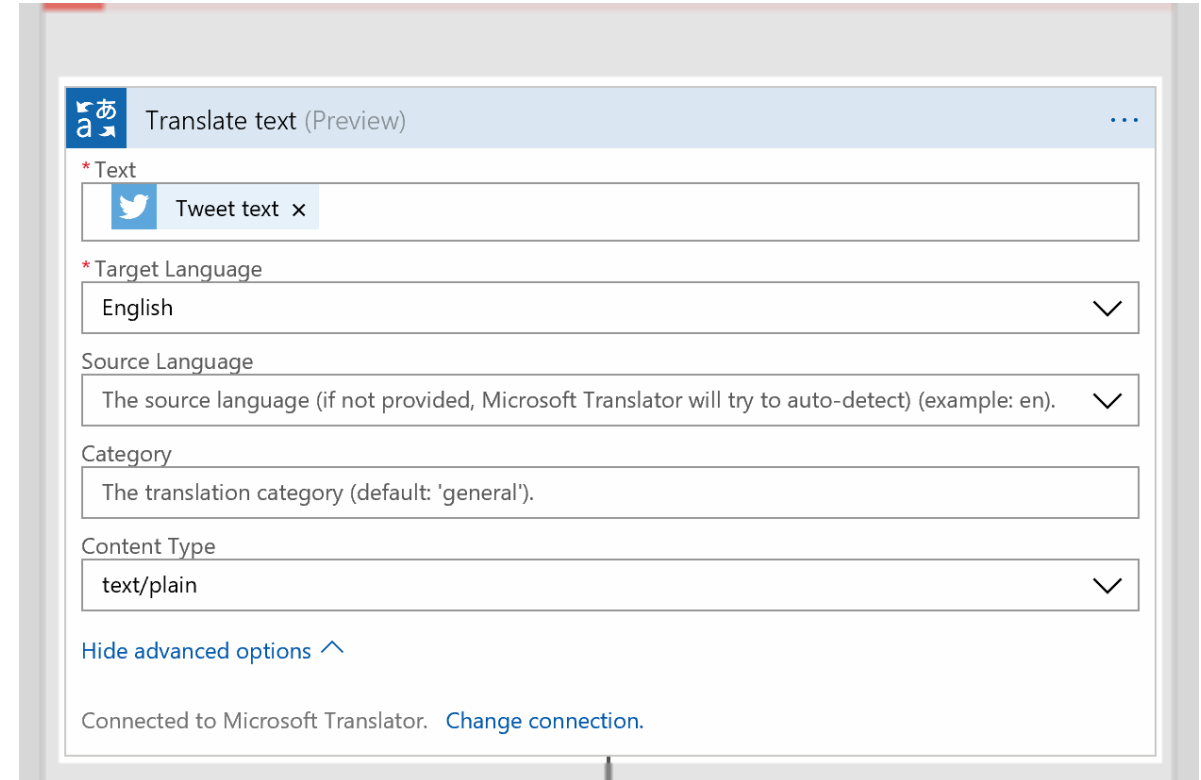
Portal experience for managing API
Connections



Visual designer

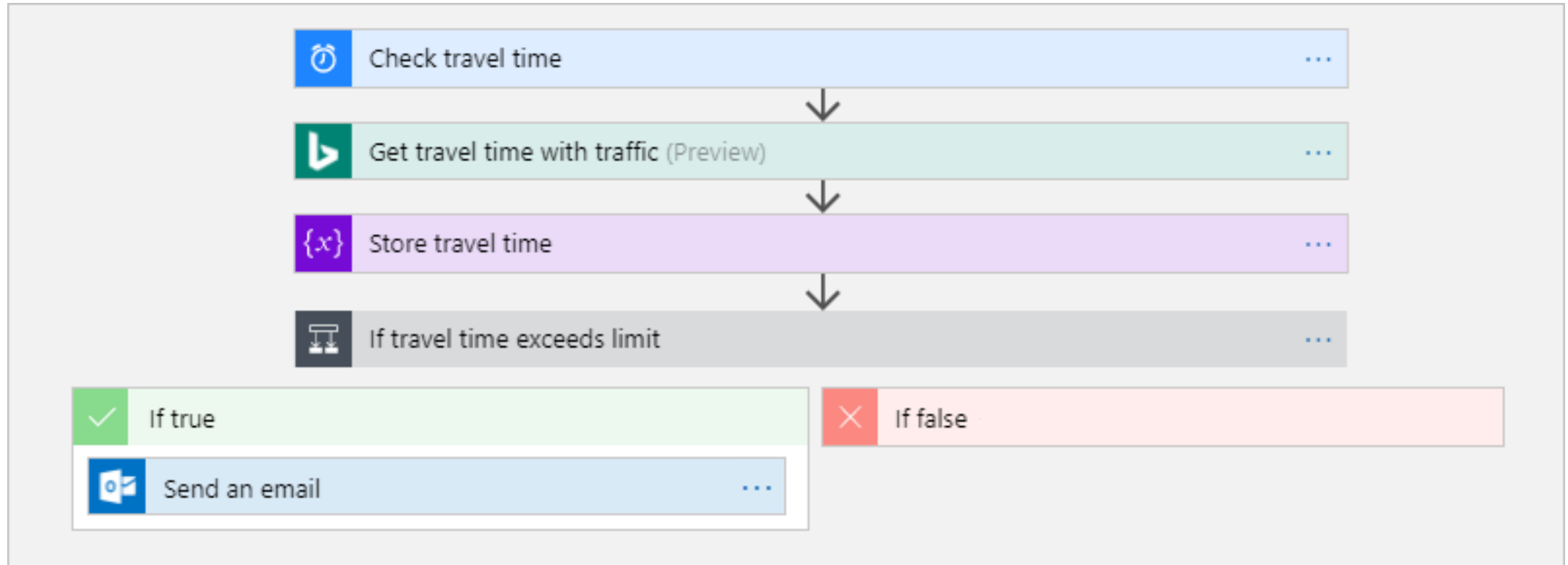


Add triggers, actions, conditions and loops visually

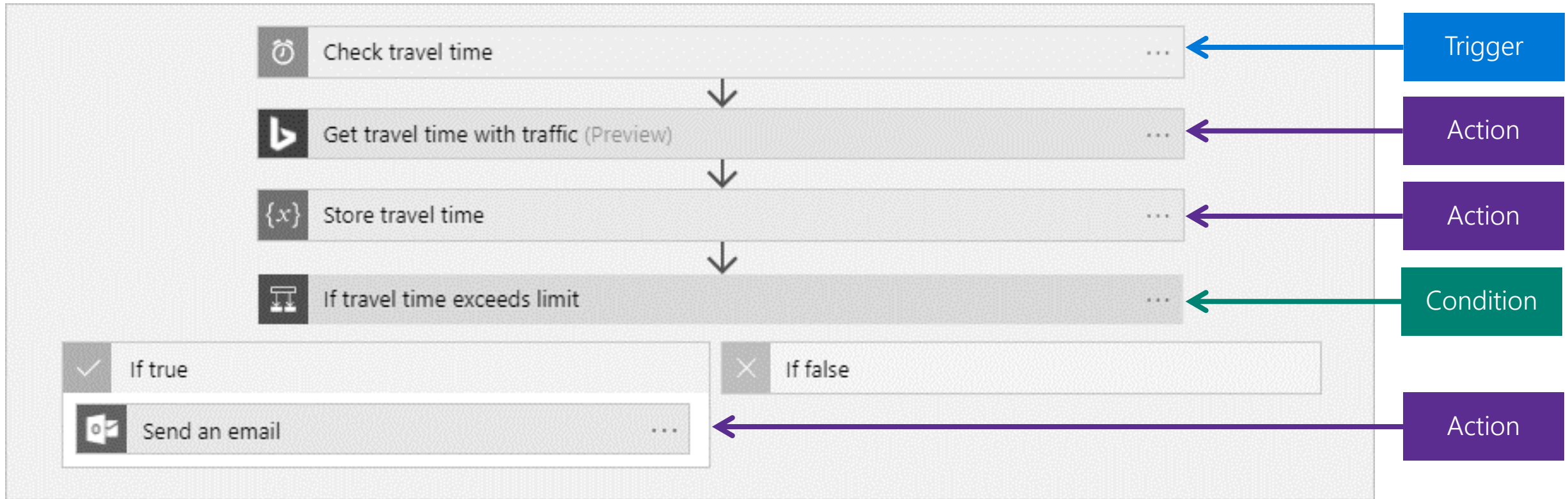


Configure properties with no coding needed

Example logic app in the visual designer

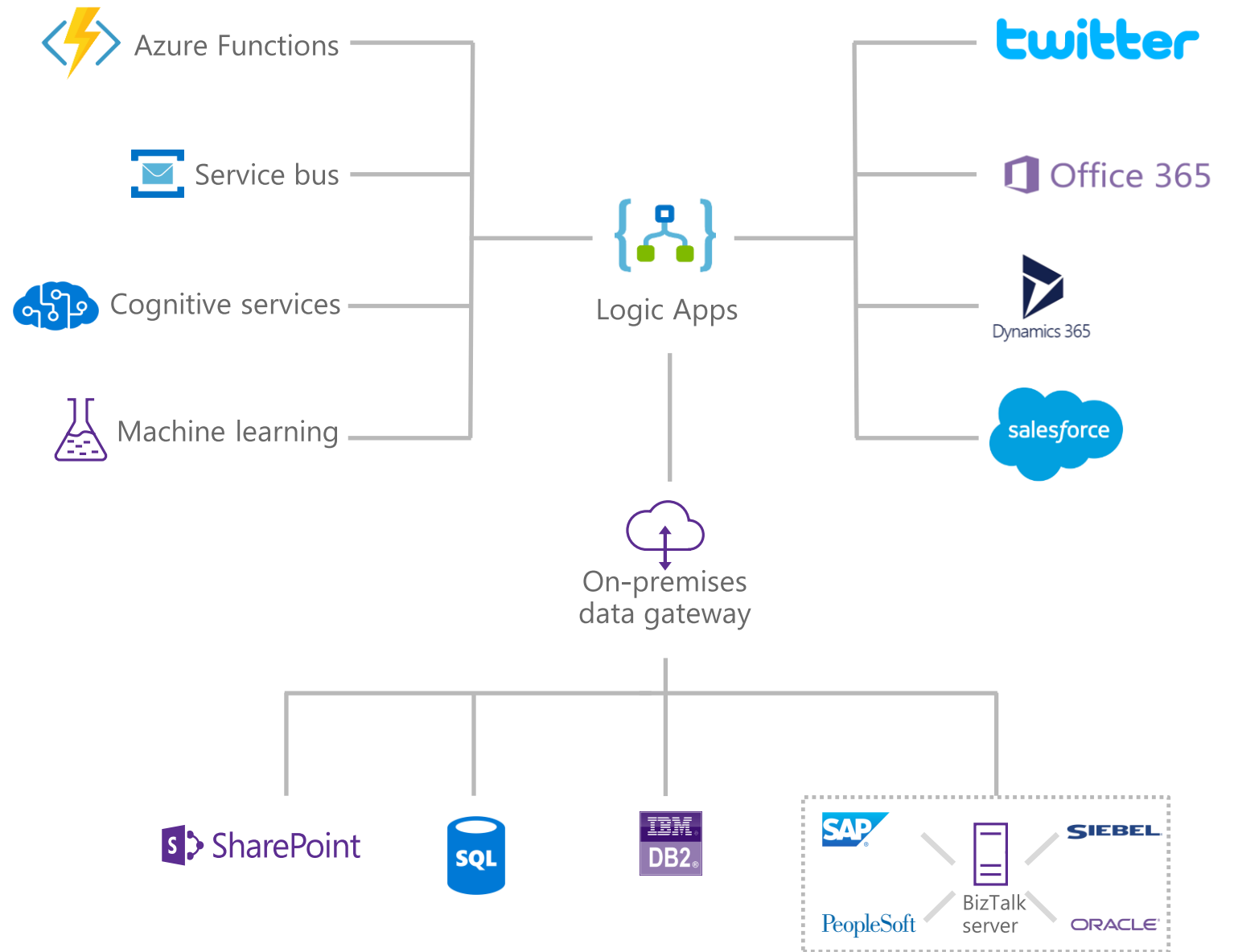


Example logic app in the visual designer



Logic Apps connect everything

(200+ connectors)



Flow Control

Triggers

- Schedule
- HTTP Request
- Azure Queue
- Via Connectors
 - SharePoint
 - Email Received
 - FTP
 - OneDrive
 - JIRA
 - BOX
 - SAP
 - Salesforce
 - Twitter
 - Adobe

Scheduling executions

- Example workloads
 - Cleanup jobs, data synchronization, running reports, data compression
- Simple Recurrence
 - Every 5 minutes, every 2 days
- Recurrence with deterministic time
 - Use start time
 - Every hour on the hour, every day at midnight
- Complex schedules
 - Every Sunday at noon, every 15 minutes during work hours
- Run once jobs
 - Request trigger + Start time input
 - Delay Until (start time)

The screenshot displays a workflow editor interface with three main steps connected by downward arrows:

- Step 1: When a HTTP request is received**
 - Trigger icon: Globe with a plus sign.
 - Header: "When a HTTP request is received" with a menu icon (three dots).
 - Message: "You can update request schema to pass in custom values for actions to be executed." with a close icon (X).
 - Field: "HTTP POST URL" with a placeholder "URL will be generated after save" and a document icon.
 - Field: "Request Body JSON Schema" containing a JSON schema:

```
{  "properties": {    "startTime": {      "type": "string"    }  },  "type": "object"}
```

 - Link: "Use sample payload to generate schema"
 - Link: "Show advanced options" with a dropdown arrow.
- Step 2: Delay until**
 - Trigger icon: Clock.
 - Header: "Delay until" with a menu icon (three dots).
 - Field: "* Timestamp" with a dropdown menu showing "startTime" and a close icon (x).
- Step 3: Scheduled actions**
 - Trigger icon: Document with a checkmark.
 - Header: "Scheduled actions" with a menu icon (three dots).

Flow control

Until Condition is met

- Do-until loop
- Evaluated condition, total time, or count

When a Timeout has occurred

- Configure timeout action setting

While other work is being done

- Parallel actions

Concurrency Control

Runs

- Instances are created concurrently
- Singleton trigger executions
- Degrees of parallel execution
 - Only for non split-on polling triggers

For Each Loops

- Collections handled concurrently
- Sequential For Each
- For each degrees of parallelism

Parallel actions

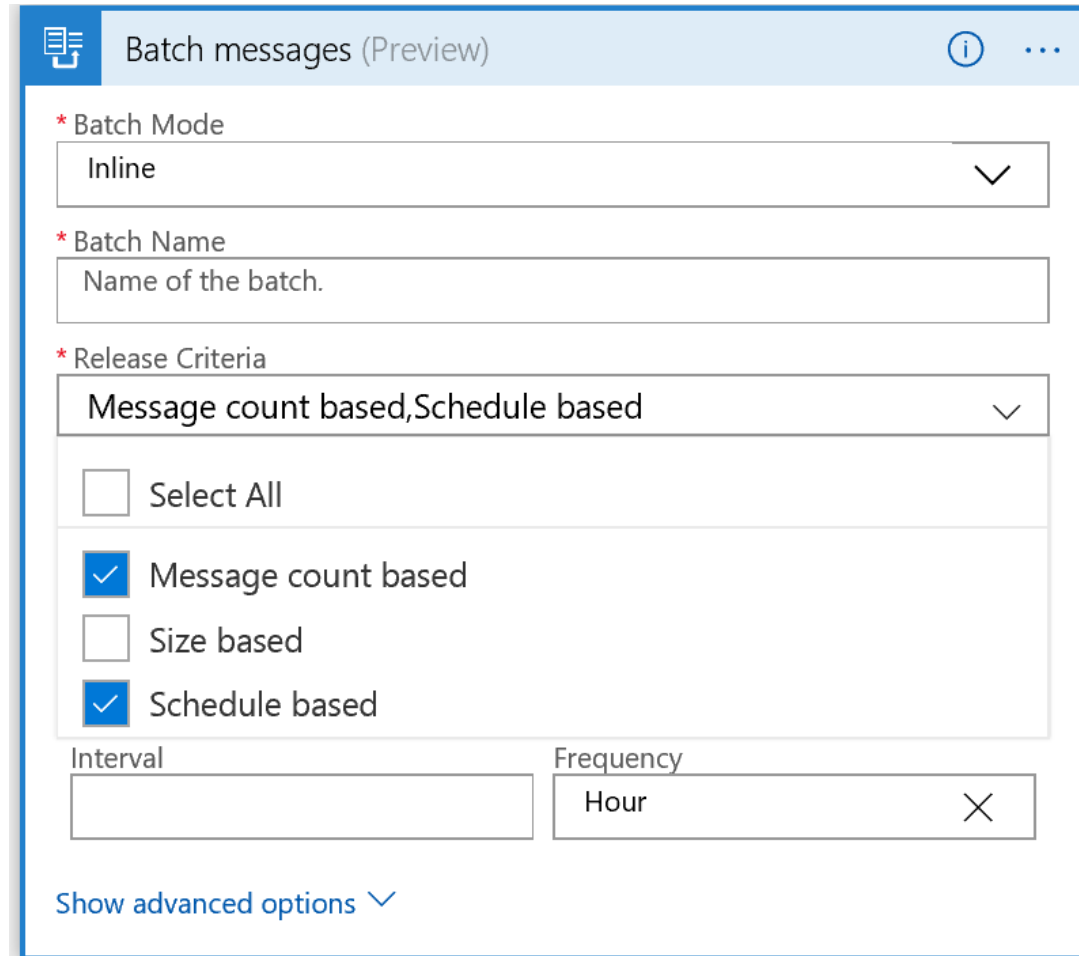
- Implicit parallelization
- Join with Run After

Do Until Loops

- Runs sequentially

Messaging Patterns

- Request/Reply (sync and async)
- Splitter
- Pub-Sub
- Claim-Check
- Batch
 - Sender and receiver logic apps: 1 or more senders and 1 receiver that uses the new Batch trigger
 - Batch release criteria: count, time, size



The screenshot shows a configuration window titled "Batch messages (Preview)". It contains several sections for setting up batch processing:

- * Batch Mode:** A dropdown menu currently set to "Inline".
- * Batch Name:** A text input field with the placeholder text "Name of the batch."
- * Release Criteria:** A dropdown menu currently set to "Message count based, Schedule based". Below this dropdown are three checkboxes:
 - ☐ Select All
 - ☒ Message count based
 - ☐ Size based
 - ☒ Schedule based
- Interval:** A text input field for specifying the time interval.
- Frequency:** A dropdown menu currently set to "Hour".
- Show advanced options:** A link with a downward arrow to expand the configuration options.

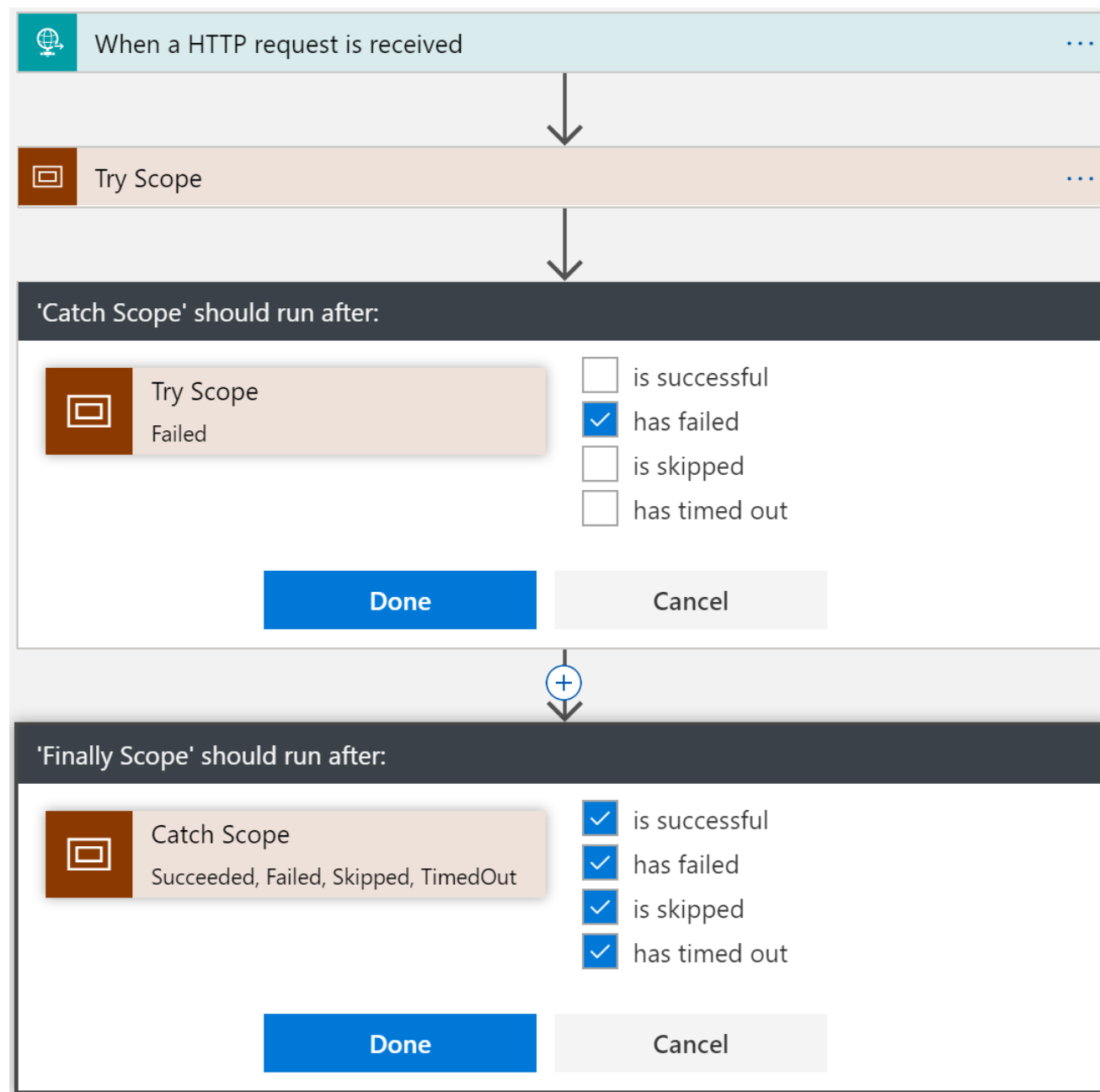
Error Handling

Error Handling

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-exception-handling>

- Retry policy
 - Default, Exponential Interval, Fixed Interval, and disabled
- Run After
 - Conditional dependency control
 - Status: Succeeded, Failed, TimedOut, Skipped
- Terminate
 - Early termination
 - Failed or Successful status
- Scopes
 - Encapsulate a set of actions
 - Status: Determined by the status of the leaf nodes within the Scope
 - Succeeded: all must have succeeded
 - Skipped: all must have been skipped
 - Failed: any not succeeded or skipped

```
try
{
    ...
}
catch
{
    ...
}
finally
{
    ...
}
```



Security

Secure access to your logic apps

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-securing-a-logic-app>

Constrains on who can initiate the workflow

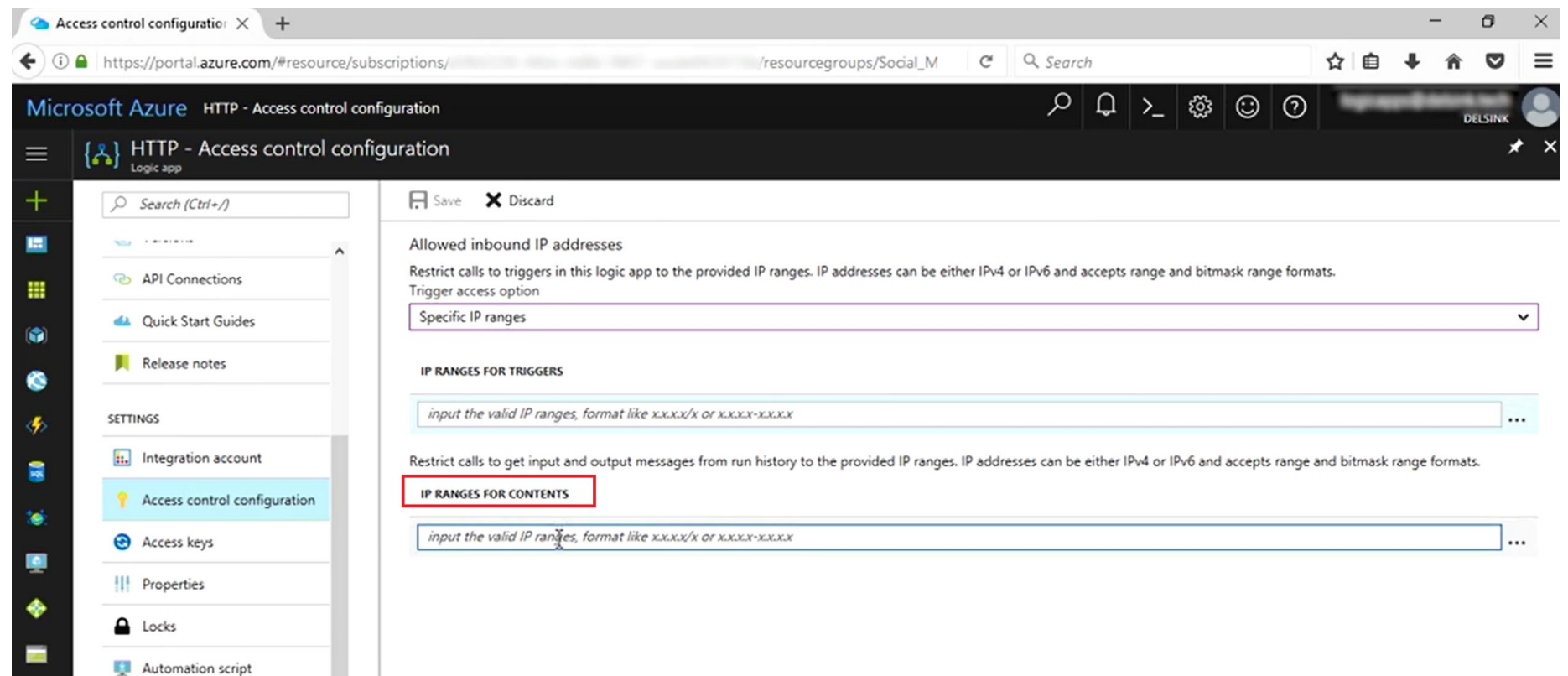
- API Management
- HTTP or WebHook triggered workflows
 - Firewall
 - Shared access keys (part of URL) - <https://docs.microsoft.com/en-us/azure/storage/common/storage-dotnet-shared-access-signature-part-1>

Secure access to resources

- Credentials used in connectors
 - Individual vs. shared accounts
 - <http://mikaelsand.se/securing-passwords-in-logic-apps/> and <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-manager-keyvault-parameter>
 - Shared between workflows in same resource group

Secure operational access

- RBAC Roles: Logic App Contributor and Logic App Operator
 - Azure Resource Lock
- Restrict access to Run History and diagnostic logs



Monitoring

Monitoring and Diagnostics

- Trigger and run history
- Runtime details – monitoring view
- Diagnostic logging and Alerts
 - Tracked properties
 - Tracking API
 - Operations Management Suite

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-monitor-your-logic-apps>

Trigger and run history

The screenshot displays the Microsoft Azure portal interface for a Logic App named 'Twitter_Monitor'. The left sidebar contains navigation options such as Overview, Activity log, Access control (IAM), Tags, and various development tools. The main content area is divided into two sections: 'Essentials' and 'Runs history'.

Essentials

- Resource group (change): Social_Media_LogicsApps
- Location: West US
- Subscription (change): Pay-As-You-Go
- Subscription ID: [redacted]
- Definition: 1 trigger, 1 action
- Status: Disabled
- Runs last 24 hours: 1 successful, 0 failed
- Integration Account: -- --
- Plan: Consumption

Runs history

Filters: All, Start time e..., Pick a date, Pick a time

Specify the run identifier to open monitor view directly: [input field]

STATUS	START TL...	IDENTIFIER	DURATION
✓ Succeeded	7/18/20...	08587012460412042920664197873	535 Millisec...

Trigger History

Filters: All, Start time e..., Pick a date, Pick a time

Trigger: When_a_new_tweet_appears

STATUS	START TL...	FIRE
● Skipped	7/18/20...	
✓ Succeeded	7/18/20...	Fired
✓ Succeeded	7/18/20...	
● Skipped	7/18/20...	
● Skipped	7/18/20...	

LinkedIn

Runtime details – monitoring view

The screenshot displays the Microsoft Azure portal interface. The left sidebar contains navigation links for various services: Dashboard, Resource groups, All resources, Recent, App Services, SQL databases, Virtual machines (classic), Virtual machines, Cloud services (classic), Subscriptions, Logic apps, App Service plans, Azure Active Directory, Monitor, Security Center, and Cost Management. The main content area is titled 'Runs history' for a 'TrafficTrigger' logic app. It includes a 'Refresh' button and filters for 'All' runs, 'Start time earlier than', and a search bar. A table lists the run history with columns for 'START TIME' and 'DURATION'. The right pane shows the 'Logic app run' details for a specific run, including 'Run Details', 'Resubmit', and 'Cancel Run' options. The details view shows the output of the logic app, including the 'brandLogoUri', 'copyright', 'resourceSets', 'statusCode', 'statusDescription', and 'traceId'.

START TIME	DURATION
9/28/2017, 2:38 ...	--
9/28/2017, 2:05 ...	15.57 Seconds
9/28/2017, 2:03 ...	17.26 Seconds
9/28/2017, 1:57 ...	18.25 Seconds
9/28/2017, 1:56 ...	18.21 Seconds
9/28/2017, 1:52 ...	20.87 Seconds
9/28/2017, 1:47 ...	18.8 Seconds
9/28/2017, 1:45 ...	19.17 Seconds
9/28/2017, 1:42 ...	10.26 Seconds
9/28/2017, 1:36 ...	10.47 Seconds
9/28/2017, 1:35 ...	10.21 Seconds
9/28/2017, 1:32 ...	1.35 Minutes
9/28/2017, 1:30 ...	1.19 Minutes
9/28/2017, 1:27 ...	36.01 Seconds

Logic app run details:

- brandLogoUri:** http://dev.virtualearth.net/Branding/logo_powered_by.png
- copyright:** Copyright © 2017 Microsoft and its suppliers. All rights reserved. This API cannot be accessed and the content and any results may not be used, reproduced or transmitted in any manner without express written permission from Microsoft Corporation.
- resourceSets:**
 - coordinates:** 25.89823, -80.234642
 - description:** "Between 19th Ave and 135th St - Construction work".
 - end:** "/Date(1506643303000)/".
 - incidentId:** 2713741756690097700.
 - lastModified:** "/Date(1506623405575)/".
- statusCode:** 200
- statusDescription:** OK
- traceId:** c5939ca3a10f4bca9d7c73b6348037af7CH110450308[7.7.0.0]EAP8192168

Diagnostic Logging and Alerts

The screenshot displays the Microsoft Azure portal interface for configuring an alert rule. The left-hand navigation pane is visible, with the 'Alert rules' option highlighted in blue and enclosed in a red rectangular box. The main content area is titled 'Twitter_Monitor - Alert rules' and shows the configuration for a specific alert rule named 'Twitter_Monitor_Alert1'.

At the top of the configuration area, there are filters for Subscription (Pay-As-You-Go), Source (All sources), Resource group (Social_Media_LogicsApps), Resource type (Logic apps), and Resource (Twitter_Monitor). Below these filters, a breadcrumb path reads 'Pay-As-You-Go > Social_Media_LogicsApps > Twitter_Monitor'.

The 'Diagnostics settings' section includes a search bar labeled 'Filter alerts...'. Below this is a table listing the alert rules. The table has columns for NAME, STATUS, CONDITION, RESOURCE GROUP, RESOURCE, and LAST FIRED. The row for 'Twitter_Monitor_Alert1' shows it is 'Active' with the condition 'Triggers Failed > 3 Count'.

NAME	STATUS	CONDITION	RESOURCE GROUP	RESOURCE	LAST FIRED
Twitter_Monitor_Alert1	Active	Triggers Failed > 3 Count	Social_Media_LogicsApps	Twitter_Monitor	Never

In the bottom right corner of the screenshot, there is a 'LinkedIn' logo.

Service Description

Usage Costs

	PRICE PER EXECUTION
Actions	\$0.000025
Standard Connector	\$0.000125
Enterprise Connector	\$0.001

+

Per hour pricing
(Standard or Basic integration
accounts – see resource limits)

+

Per GB per hour pricing
(data retention, e.g. diagnostic logs)

<https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-pricing>
<https://azure.microsoft.com/en-us/pricing/details/logic-apps/>
<https://azure.microsoft.com/en-us/pricing/calculator/?service=logic-apps>

Demo

Wrap

Logic apps benefits

- Serverless – no server sizing/capacity planning, auto-scaling based on demand, high availability (no need to plan anything for fault tolerance), use-based billing (charge for consumption of resources only)
- Drag and drop experience (no extensive coding experience required)
- Lots of pre-built connectors (200+ and counting)
 - Receiving an e-mail, monitoring a twitter feed, sending a text message, ServiceBus, Hybrid connectors, IBM MQ, SAP, B2B support e.g. or AS2, X12, EDIFACT, etc.
 - Connectors are just API apps, so, if one is not available, you can create your own
- 300+ actions
- Flow-control (If-then, switch, for-each, do-until, scope)
- BizTalk 2016 has Logic App adapter for connecting to logic apps
- OOB monitoring and logging

Microsoft Enterprise Integration Platform



Microsoft Flow

Enable the Citizen Developer to Do More

API Management



Document and catalog LOB APIs

Discover, manage and secure connectivity

Gain insights, measure and monetize

Logic Apps



Connect SaaS applications

Serverless compute, auto scaling, fully managed

Leverage Azure Services

BizTalk Server



Automate mission critical processes

Integrate on-premises LOB to SaaS apps

Support for latest Microsoft Platforms

Service Bus



Cloud-based messaging service

Connect to the cloud

Secure and predictable message delivery



 Azure

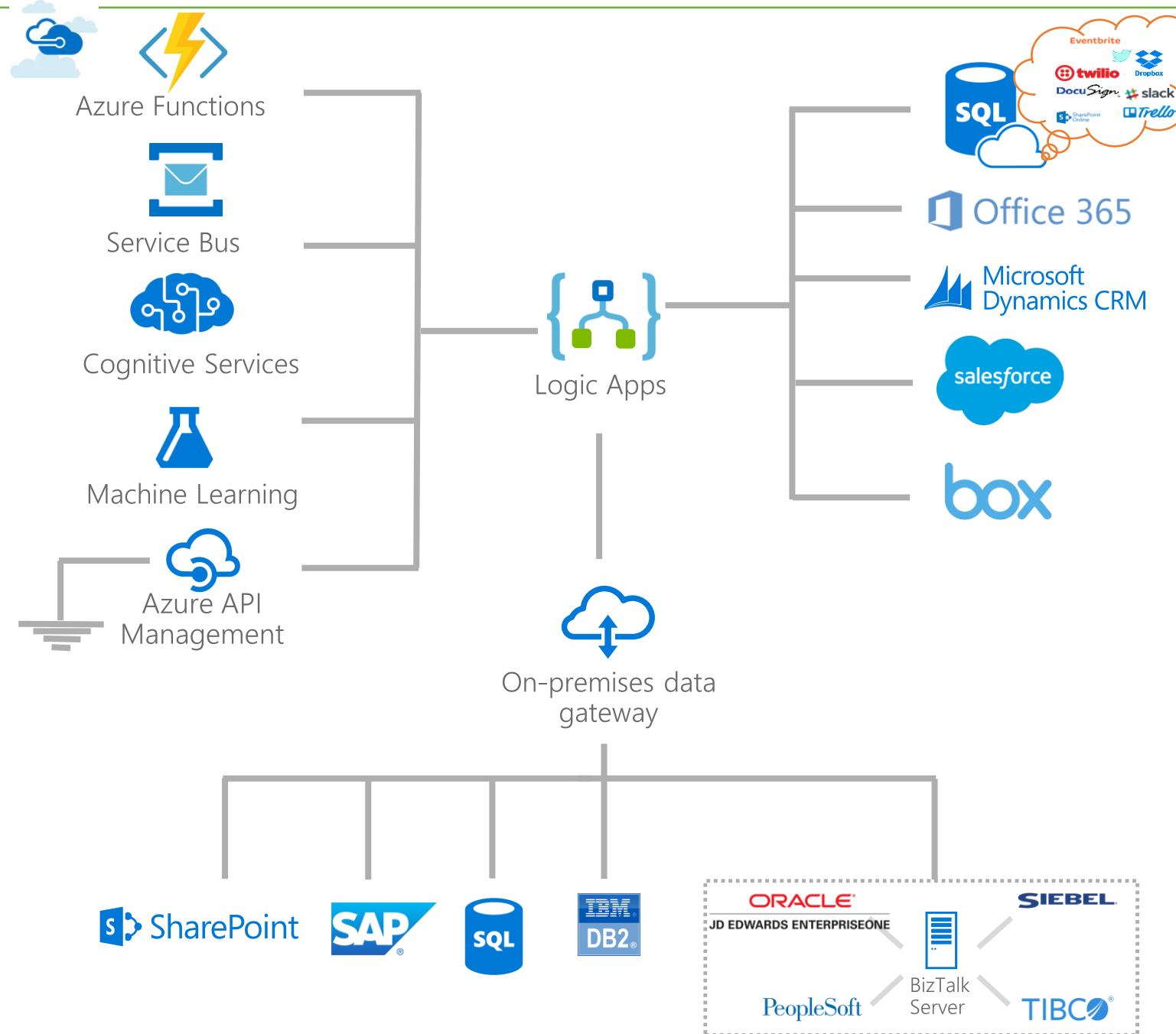
Management Portal, Azure Resource Manager

Summary

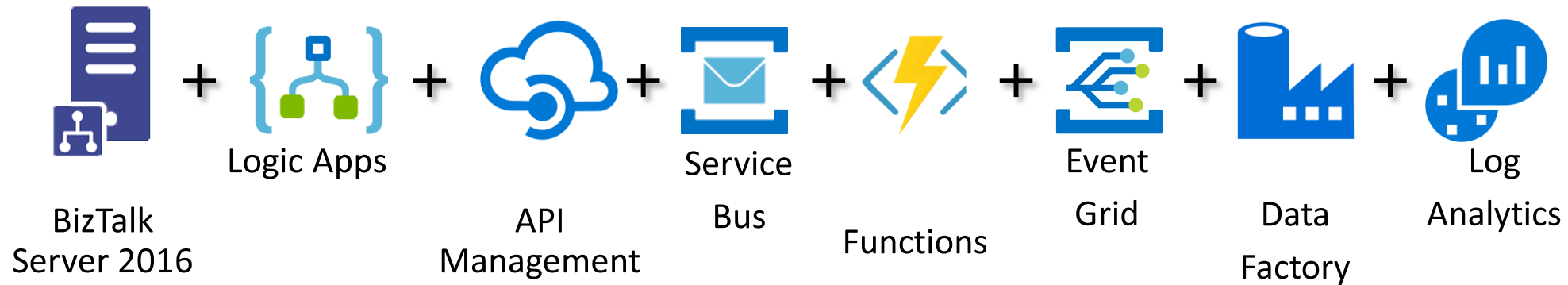
Connect on-premises, hybrid and cloud applications

Run mission-critical, complex integration scenarios with ease

Build "Smart" Integrations leveraging Machine Learning/Cognitive Services



Better Together.



=

Limitless Possibilities

More info

Microsoft Hybrid Integration Platform

- <https://aka.ms/hybrid-integration>
- <https://aka.ms/infopedia-hip>

Logic Apps Monthly Webcasts

- <http://aka.ms/logicappslive>

Logic Apps Documentation

- <http://aka.ms/logicapps-docs>

Logic Apps Blog

- <http://aka.ms/logicappsblog>

Logic Apps Ideas

- <http://aka.ms/logicapps-wish>

Azure Logic Apps

- <http://aka.ms/logicapps-try>
- <http://aka.ms/logicapps-tutorial>
- <http://aka.ms/logicappslive>

Twitter

- APIM - @AzureApiMgmt
- BizTalk Server @BizTalk_Server
- Logic Apps - @Logicappsio

Azure Logic Apps monthly updates (non-MSFT)
<https://blogs.biztalk360.com/tag/azure-logic-apps/>

Cool examples

Taking an order from a scanned image (OCR)

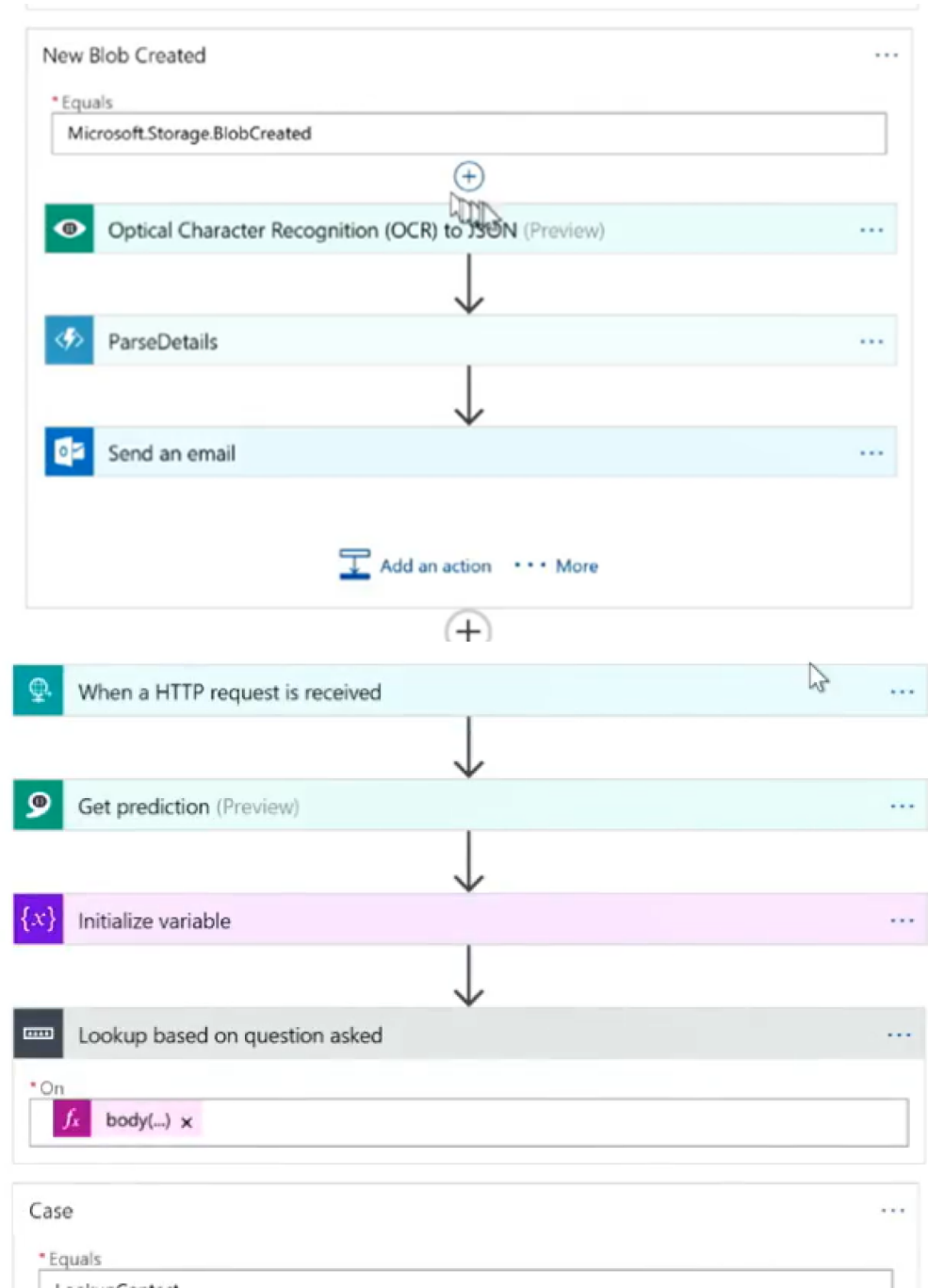
Demo:

<https://youtu.be/uluHbZWxqQY?t=2244>

Bots invoking logic apps that get intent and use connectors to get info by issuing a query to the right backend system (e.g. 'when did I last hear from this person?' -> query to dynamics CRM)

Demo:

<https://youtu.be/uluHbZWxqQY?t=3602>



Thank you!

Questions?