

Enterprise Application Lifecycle Management with **Azure Logic Apps**

Building robust automations with Azure Logic Apps

Wherever business takes you

MNPdigital.ca



Agenda for today

- | | |
|-------------------------|---------------|
| 1. Introduction & Goals | 3 min |
| 2. Baseline ALM | 5 min |
| 3. Planning | 8 min |
| 4. Implementation | 16 min |
| 5. Deployment | 8 min |
| 6. Maintenance | 8 min |
| 7. Final Thoughts | 2 min |
| 8. Questions | 5 min |

Introduction



Cameron McKay

Manager, Azure Architect

Cameron.McKay@mnp.ca



Cameron McKay is a Microsoft certified Cloud Developer/Architect with a background in web technology and business analysis; he has blended his expertise in these areas to deliver a variety of enterprise web and cloud applications using the Microsoft technology stack.

Cameron has proven experience leading teams and delivering web and cloud solutions throughout the full software development lifecycle.

Cameron is an advocate for defensive programming, Microsoft Azure development and integrating security into the software development lifecycle.

Cameron spends his off-hours as a tabletop board game enthusiast, cooking up delicious meals, watching anime and training his husky dogs. Cameron is always up for any tabletop game; he especially enjoys strategy games such as Settlers of Catan and Go (Baduk).

Goals for today

Leave you with two ideas and impressions...

01

Understand

Options and decisions at each stage of the
SDLC for Azure Logic Apps

02

Implement

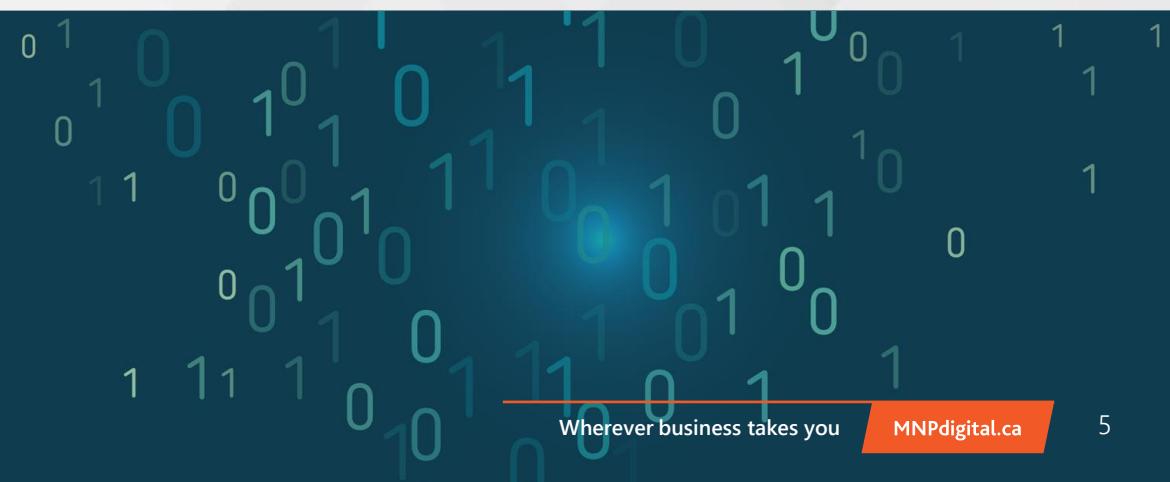
Provide you with the resources to implement
this within your organization.

What do these have in common?

Bubble Wrap

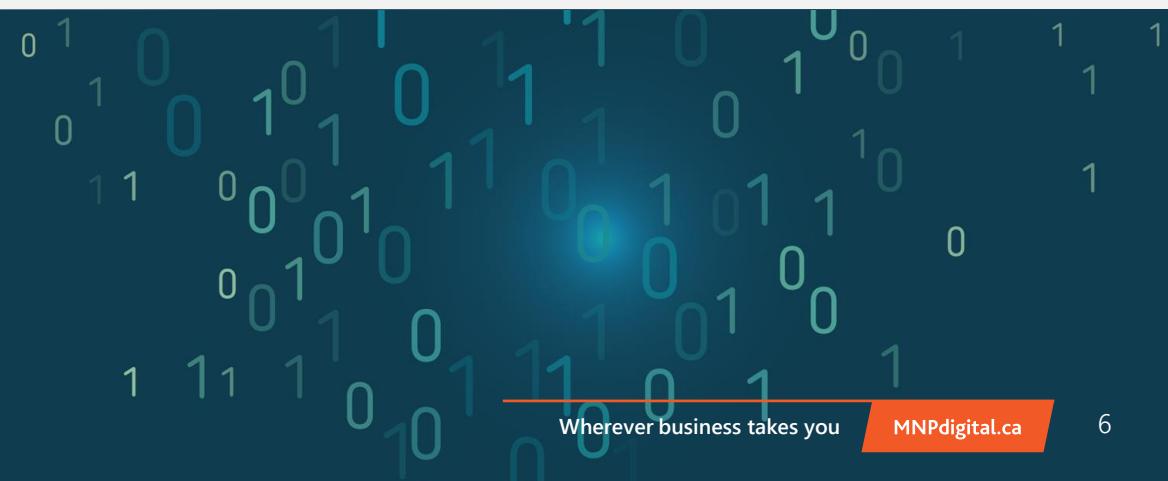
Play-Doh

Pacemaker



What is Application Lifecycle Management?

*Application lifecycle management is the product lifecycle management of computer programs. It encompasses requirements management, **software architecture**, computer programming, **software testing**, **software maintenance**, change management, **continuous integration**, project management, and **release management**. - [Wikipedia](#)*





Our goals with ALM



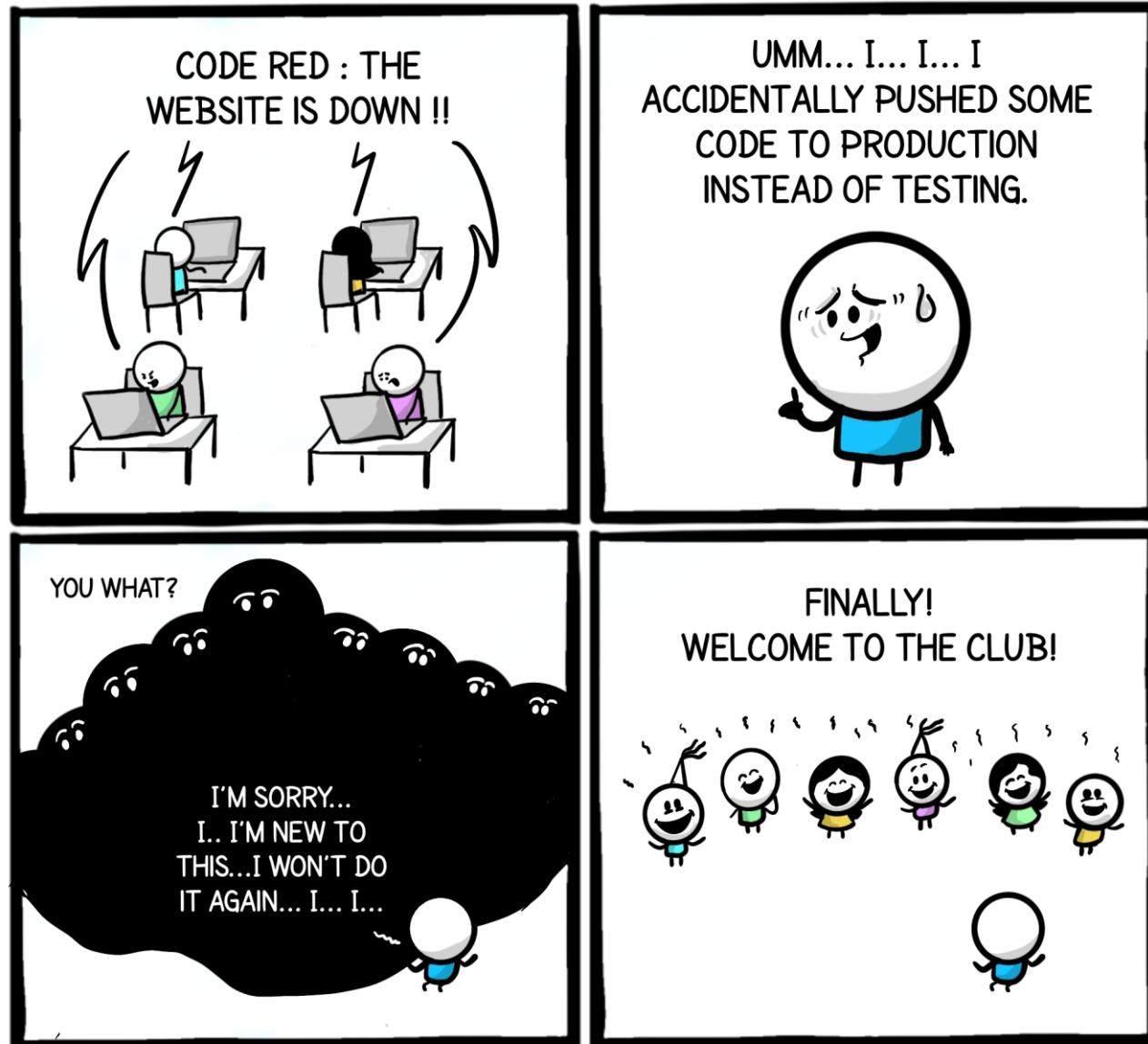
Smooth-running App Development Process



Continuous delivery of software



Workflow visibility





Planning, Analysis, Design

Calm before the storm



ALM Team



Azure Architect



Azure Developer

Azure Engineer

DevOps Engineer



Business Analyst

Project Manager

Change Management

Stakeholders

Tenancy

Different hosting options available for Logic Apps



Single-Tenant

Dedicated resources



Multi-Tenant

Logic apps across Microsoft Entra tenants share the same processing (compute), storage, network, etc.



ISE

Retiring in August 2024

Connectors

Connectors in Logic Apps are categorized based on price for use



Built-in

The basics required to build an automation (for loops, if statements, variable declaration, etc.)



Managed - Standard

Pre-build connector for various SaaS solutions such as Business Central, Azure Storage, etc.



Managed - Enterprise

For well-defined industry protocols like EDI and sophisticated SaaS solutions like SAP

Resource Types



Consumption



Standard

Consumption

What do we get with consumption logic apps?



Multi-tenant

Logic apps across Microsoft Entra tenants share the same processing (compute), storage, network, etc.



Pricing

First 4,000 built-in executions are free. Connectors have a cost per execution.



One Workflow

One workflow per logic app.

Standard

What do we get with standard logic apps?



Single-tenant

Logic apps across Microsoft Entra tenants share the same processing (compute), storage, network, etc.

Pricing

Pay a set monthly cost for the underlying infrastructure to run built-in connectors.

Many workflows

Many workflows deployed to a single logic app



Why does Standard hit the mark?



Virtual network Integration



Predictable pricing with ASP



Multiple workflows



Hybrid pro and low code



Integration Service Account



Enterprise Integration Pack



Easily work with AS2, X12, EDI



Secure, manageable and scalable container for the integration artifacts that we create

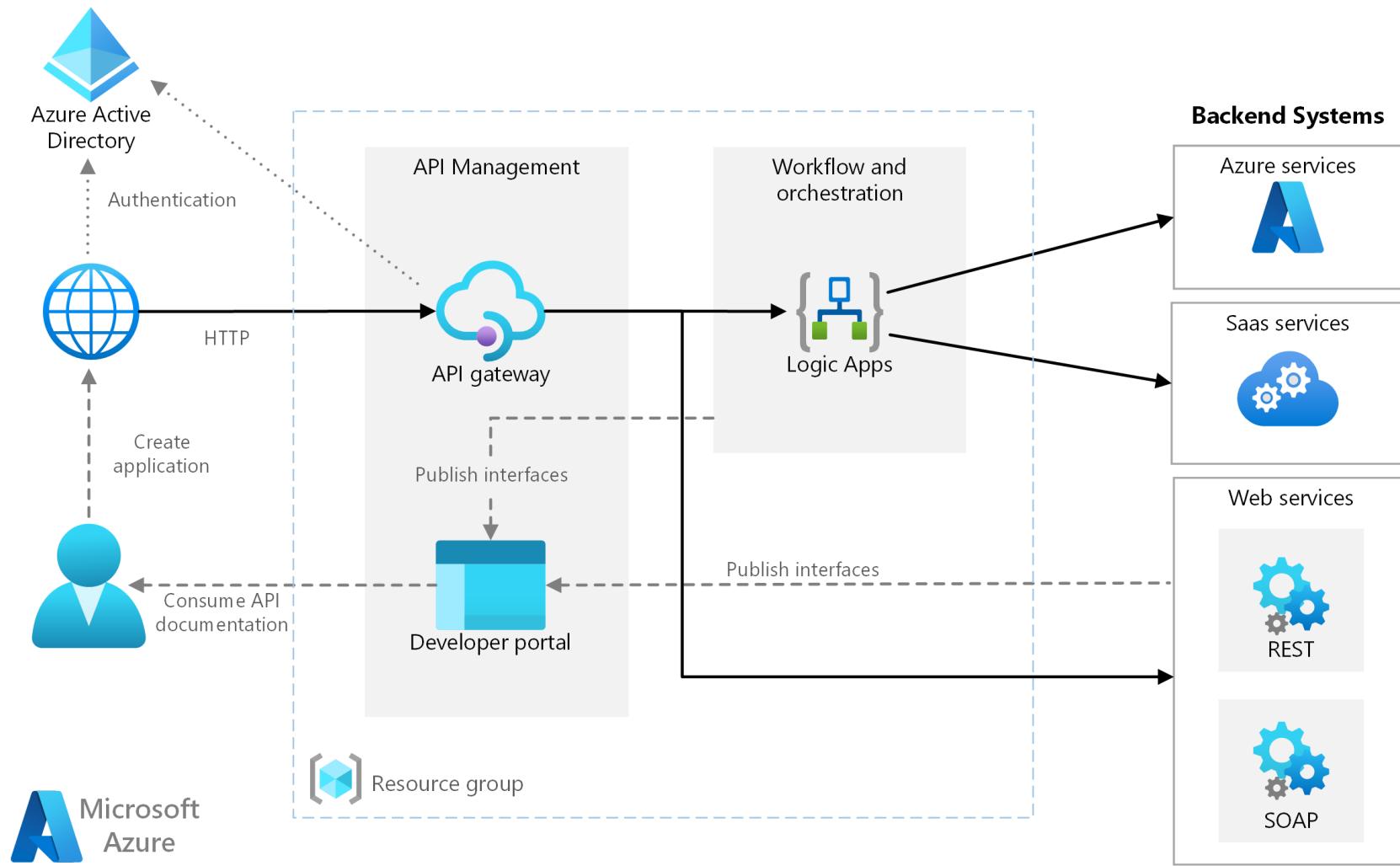
Stateful vs Stateless Workflows

Stateful workflows provide high resiliency if outages happen since they maintain a run history.

Stateless workflows are ideal for shorter runs, faster performance with quicker response times, and higher throughput.



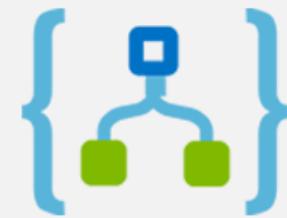
Architecture



Implementation

Into the eye of the storm

Development Environment



Azure Logic Apps



Development Practices

A key component of ALM



Reusability

Reusing code and workflows



Readability

Can someone easily else understand the code



Maintainability

Ease of making bug fixes and enhancing software



Readability

- Leverage in-line code to do more complex processing (e.g. regex)
- Comments
- Naming conventions
- Group actions using scopes

Maintainability – Scopes & Reuse

How can we increase the maintainability of our logic app



Child workflows

Reuse code by calling child workflows



Organize your code

Group your actions into scopes to organize items.
Think of #regions in C#.



SRP

Treat scopes like functions.
Apply single responsibility principle to the actions within them.



Input Validation

- Json Schema!
- Protect your logic app against
“OWASP 3: Injection”



OWASP 3: Injection

What does the attack look like?

- Any mechanism by which a user can send data as inputs to our application.
 - Input form on a page (i.e. login page or search page)
 - Sending input to an API
 - Setting values in a cookie that we consume
- Cross site scripting (XSS)
- SQL injection

Testing

How can we test our logic app



Automated Unit Tests

Write unit tests that can be used for regression testing



Manual Testing w/ Mocks

Group your actions into scopes to organize items.
Think of #regions in C#.



Infrastructure as Code

- Bicep or Terraform
- Protect your logic app against “OWASP 5: Security Misconfiguration”
- Spin up and tear down azure environments for DEV and UAT



PR Best Practices



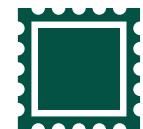
Can't push directly to main



Second set of eyes that aren't your own.



Link to a work item



Validation pipeline

OWASP Code Review Checklist

CATEGORY	DESCRIPTION
Authorization	Are security checks placed before processing inputs?
Business Logic and Design	If request parameters are used to identify business logic methods, is there a proper mapping of user privileges and methods/actions allowed to them?
Cryptography	Are database credentials stored in an encrypted format
Input Validation	Are all the untrusted inputs validated? Input data is constrained and validated for type, length, format, and range.
Logging and Auditing	Do audit logs log connection attempts (both successful and failures)?
Session Management	Session cookies expire in a reasonable short time
Web Services	Web service protocols that are unnecessary are disable (HTTP GET and HTTP POST)

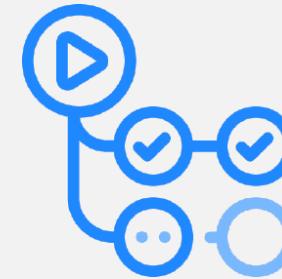
Deployment

Getting out that release!

Deployment Pipeline



+



+



Agents

Where will our deployment scripts run



Azure Public Agents

Shared agents that are hosted
by Azure



Private Agents

On-premise virtual machine
or dedicated virtual machine
in Azure

Securing Our Pipeline

Best practices for securing out pipeline



Approvers

Have an approver for each stage prior to deploying to an environment



Service Principal

Authenticate to azure resources with a service principal. One or one for each environment.



Artifacts

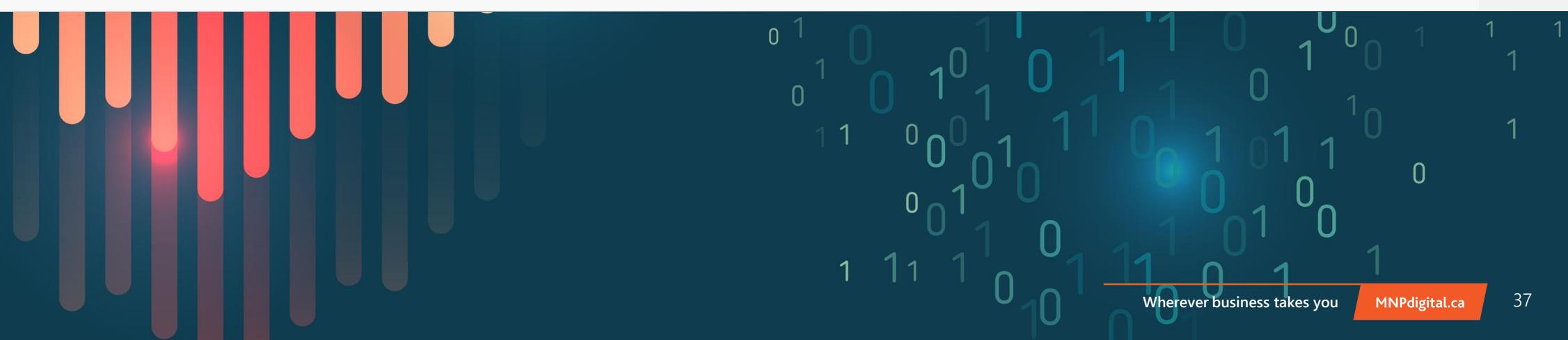
Use safe artifacts! Don't get hit by "OWASP #8 Software and data integrity failures".

Deployment Auditing

Environments

New environment ⚙️ ⋮

Environment	Status	Last activity
Development	✓ #20230823.5 on deploy-terraform-state	Aug 18
Production	Never deployed	Aug 18

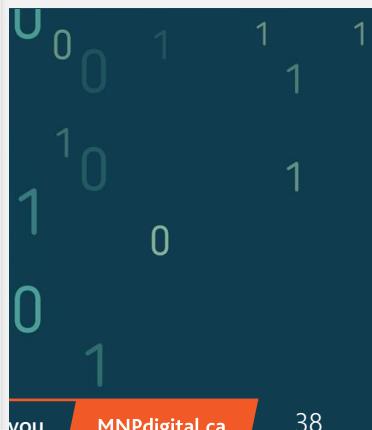


Deployment Auditing

← Development Add resource :

Deployments Approvals and checks

Run	Jobs	
#20230823.5 on deploy-terraform-state	✓ Deployment	Aug 23 2m 17s
Merged PR 156: Added development deployment stage bac... #20230822.2 on deploy-terraform-state	✓ Deployment	Aug 22 2m 23s
#20230818.9 on deploy-terraform-state	✓ Deployment	Aug 18 1m 6s
#20230818.8 on deploy-terraform-state	✓ Deployment	Aug 18 1m 45s
#20230818.7 on deploy-terraform-state	✓ Deployment	Aug 18 1m 21s



Deployment Auditing

✓ #20230822.2 • Merged PR 156: Added development stage to pipeline [Run new](#) :

This run is being retained as one of 3 recent runs by main (Branch). [View retention leases](#)

[Summary](#) [Environments](#) [Code Coverage](#) [Associated pipelines](#)

Manually run by  Cameron.McKay [View 2 changes](#)

Repository and version	Time started and elapsed	Related	Tests and coverage
❖ terraform_state	🕒 Aug 22 at 4:21 PM	⌚ 1 work items	⚠ Get started
➢ main ❖ 33a27efc	⌚ 6m 15s	⌚ 1 published	

[Stages](#) [Jobs](#)



```
graph LR; Build[Build] --> Lint[Lint]; Lint --> DevValidate[DEV Validate]; DevValidate --> DevDeploy[DEV Deploy];
```

The pipeline consists of four stages: Build, Lint, DEV Validate, and DEV Deploy. The Build stage completed 1 job and produced 1 artifact, taking 13 seconds. The Lint stage completed 1 job in 20 seconds. The DEV Validate stage completed 1 job in 1m 12s. The DEV Deploy stage completed 2 jobs in 2m 18s, with 1 check passed.

Infrastructure Deployment Pipeline



Bicep Lint

SAST

Pre-flight

What-If

Deploy

Build your bicep into ARM templates. Check for syntax errors and enforce coding standards.

Static analysis scanning to ensure we did not introduce any security vulnerabilities

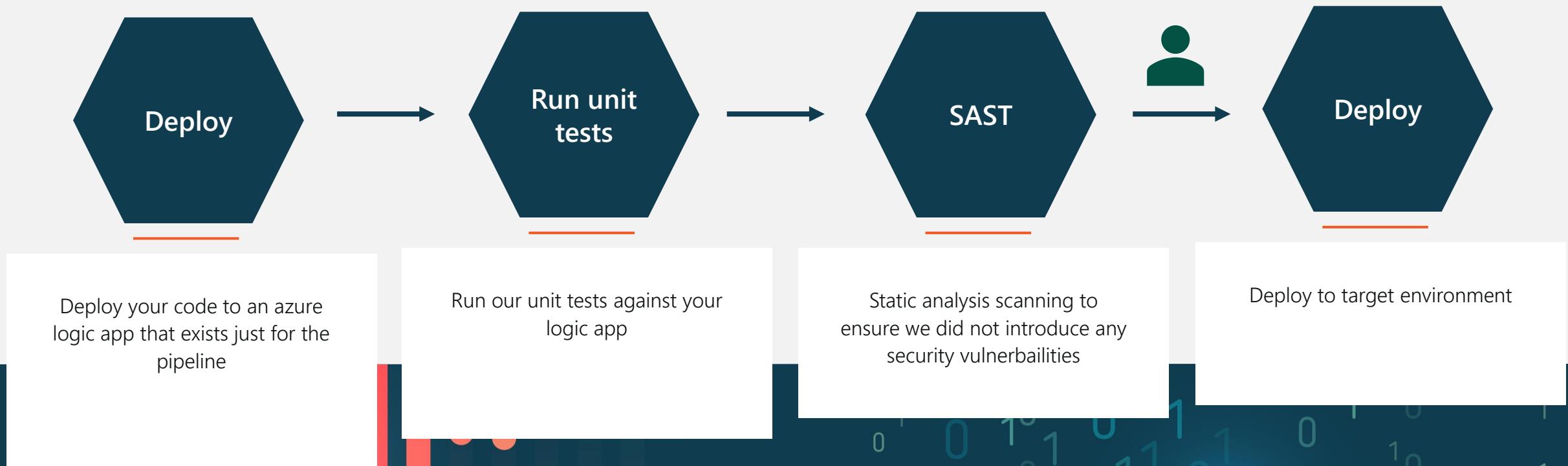
Performs a deployment without deploying. Flags issues for resolution.

Tells you what will change if you do this deployment.

Get that release out!



Logic App Deployment Pipeline



Maintenance

Debugging production issues

Azure Debugging

There are errors, how the heck do we debug them?!



Run History

Let's look at the Azure Portal
and how we can use the run
history



Tracking Properties

Querying logs can be useful.
It's even more helpful to
include tracked properties.



OWASP 9: Security Logging & Monitoring Failures

What is it?

- Software and systems have monitoring abilities in order to view logins, traffic, and more
- Attackers rely on the lack of monitoring to exploit vulnerabilities before they are detected



OWASP 9: Security Logging & Monitoring Failures

How do we mitigate the risk of an attack?

- Setup alerts to check for suspicious activity, such as failed logins, to potentially see and stop it.
- All layers of your application stack need a logging mechanism that details who is performing the action, what action is being performed and when.

Alerts and Monitoring

We definitely want to take advantage of this.



Who

Who should receive the alerts?

What

What should we be alerted to?

How

How should those alerts be received?



Recovery



Rerun the flow



Dynamic input



Rollback

Questions

Wherever business takes you

MNPdigital.ca



Gold
Microsoft Partner

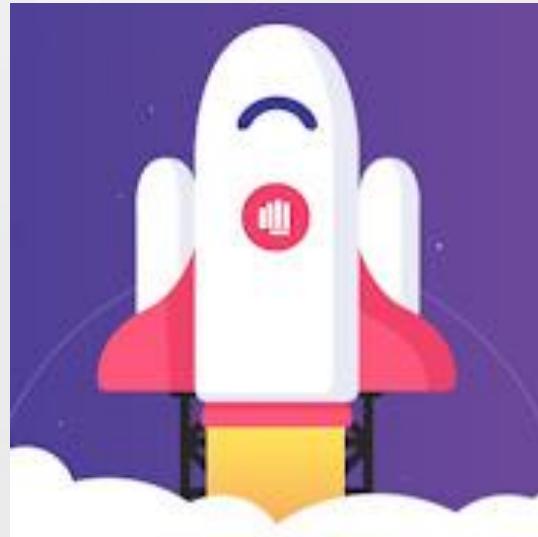


Logic Apps, Aviators Community



Newsletter - Azure Integration Services Blog

Aviators Community – Michael Stephenson



Azure Integration Services - When to use what

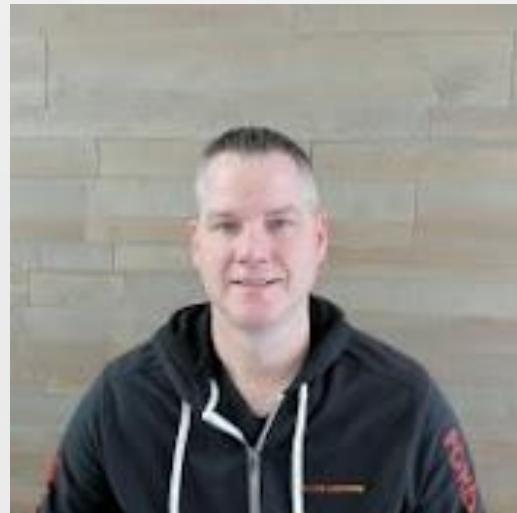
YouTube Channel

Logic Apps Testing YouTube Playlist

Logic App Testing Documentation



Aviators Community – Kent Weare



[Azure Logic Apps Tips and Tricks](#)

[YouTube Channel](#)

[Getting Started with Azure Logic Apps Standard](#)

[Automated Testing for Azure Logic Apps](#)



Aviators Community – Sandro Perreira



[Logic Apps Best Practices](#)

[Professional Blog](#)

[YouTube Channel](#)

[Logic Apps Data Mapper](#)



Final Thoughts

Leave you with two ideas and impressions...

01

Understand

Options and decisions at each stage of the SDLC for Azure Logic Apps

02

Implement

Provide you with the resources to implement this within your organization.

Contact

Cameron McKay

Manager, Azure Architect
Cameron.McKay@mnp.ca

References

- [Basic Enterprise Architecture](#)
- [Integration Service Environments](#)
- [Logic Apps Community Day 2023](#)
- [Logic Apps Pricing](#)
- [Logic Apps Infrastructure As Code](#)
- [Single-Tenant vs Multi-Tenant](#)