



Automating DevOps Releases using .NET and Azure Durable Functions

JONAH ANDERSSON

JONAH ANDERSSON



- IT Consultant
- DevOps Engineer Lead
- Developer, Cloud & Infrastructure

- Microsoft MVP (Azure)
- Microsoft Certified Trainer
- Book Author
- Mentor in Tech
- Inclusive Community Leader



forefront
consulting



WHAT I DO = CONSULTANT+++





MY MISSION TODAY!

How I solved my weekly dilemma of
manual repetitive deployment task!



Microservices and distributed applications

Overview New All releases

Releases Deployments Analytics

Releases Created Stages

Release	Created	Stages
Release-0.1.07 20221024.1 master	10/24/2022, 7:37:12 AM	Dev QA UAT Prod
Release-0.1.06 20221018.1 master	10/18/2022, 3:41:16 PM	Dev QA UAT Prod
Release-0.1.05 20221017.3 master	10/18/2022, 1:56:04 AM	Dev QA UAT Prod
Release-0.1.04 20221017.2 master	10/17/2022, 8:52:41 PM	Dev QA UAT Prod
Release-0.1.03 20221017.1 master	10/17/2022, 4:24:11 PM	Dev QA UAT Prod
Release-0.1.02 20221014.1 master	10/14/2022, 3:19:45 PM	Dev QA UAT Prod
Release-0.1.01 20221011.1 master	10/14/2022, 2:00:01 AM	Dev QA UAT Prod
Release-0.1.100 20221011.1 master	10/11/2022, 9:57:50 AM	Dev QA UAT Prod
Release-0.1.99 20221007.1 master	10/7/2022, 11:15:28 AM	Dev QA UAT Prod
Release-0.1.98 20221005.1 master	10/5/2022, 1:56:31 PM	Dev QA UAT Prod
Release-0.1.97 20221002.1 master	10/3/2022, 10:33:27 AM	Dev QA UAT Prod

Repos Pipelines Pipelines Environments Releases Library Task groups Deployment groups Artifacts

Project settings <> <> <>



Code INFRASTRUCTURE AS CODE
(IaC) + Deployments

THE BIG PROJECT

A new corporate enterprise web application and services that will be used by internal users – technicians, business, sales, distributors, etc



THE BIG PICTURE



RELEASE MANAGEMENT

- 2x a week QA Deployments
- UAT Deploy End of Sprints

GOAL:

Go PRODUCTION – Q1 2023



THE PROBLEM #1

The tools we use are not in all-in-one place.

It is everywhere!

Why?

Decided by different teams, different people ☺



EXPERIENCED SELF-ORGANIZING DEVELOPERS

DEVOPS / CLOUD INFRASTRUCTURE ENGINEERS

CONSULTANTS

DATABASE ADMINISTRATORS

SCRUM MASTERS

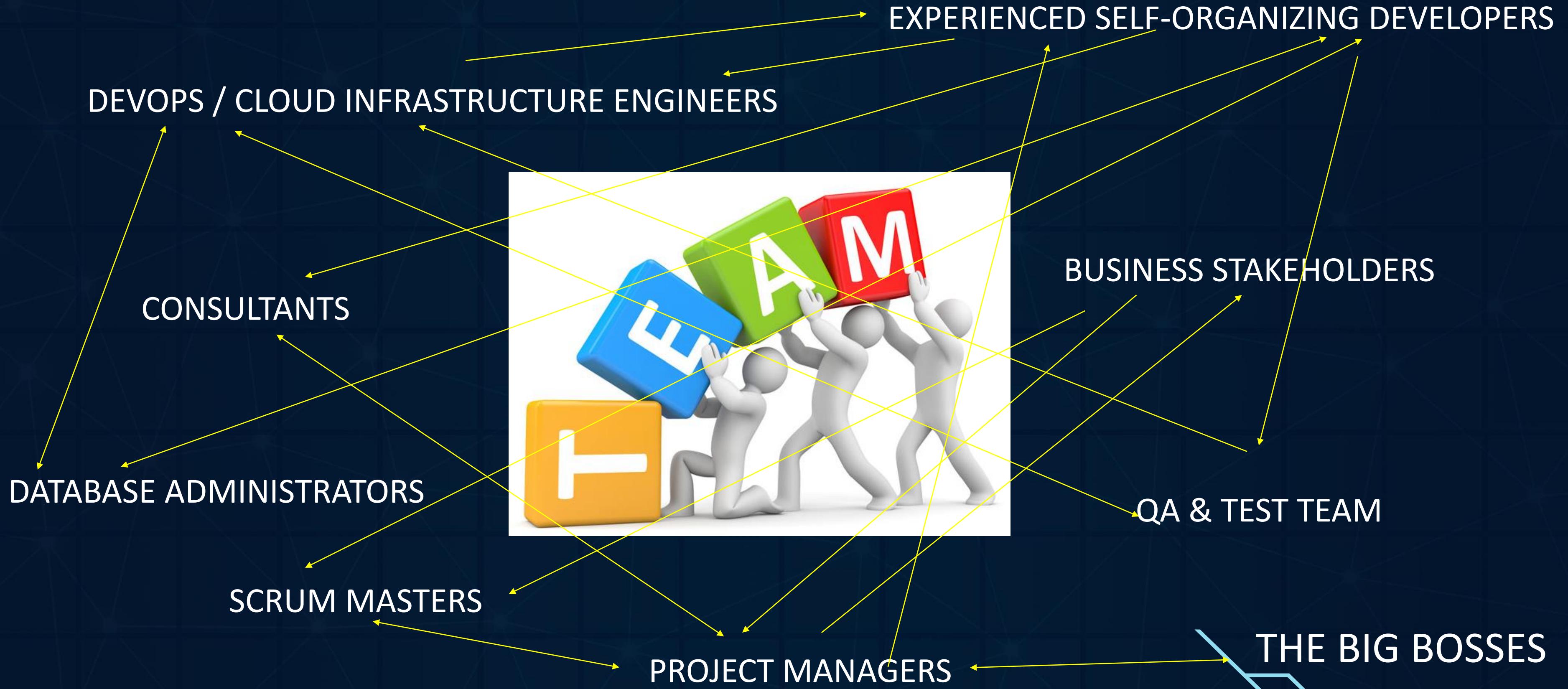
PROJECT MANAGERS

BUSINESS STAKEHOLDERS

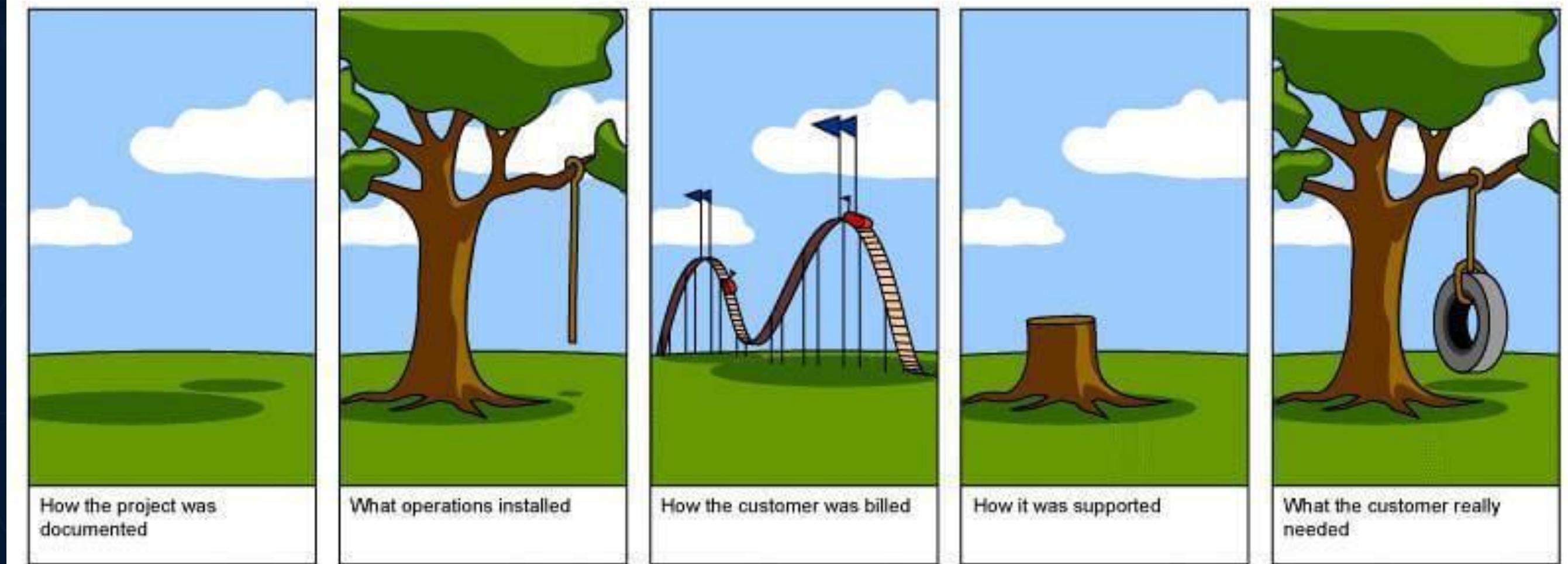
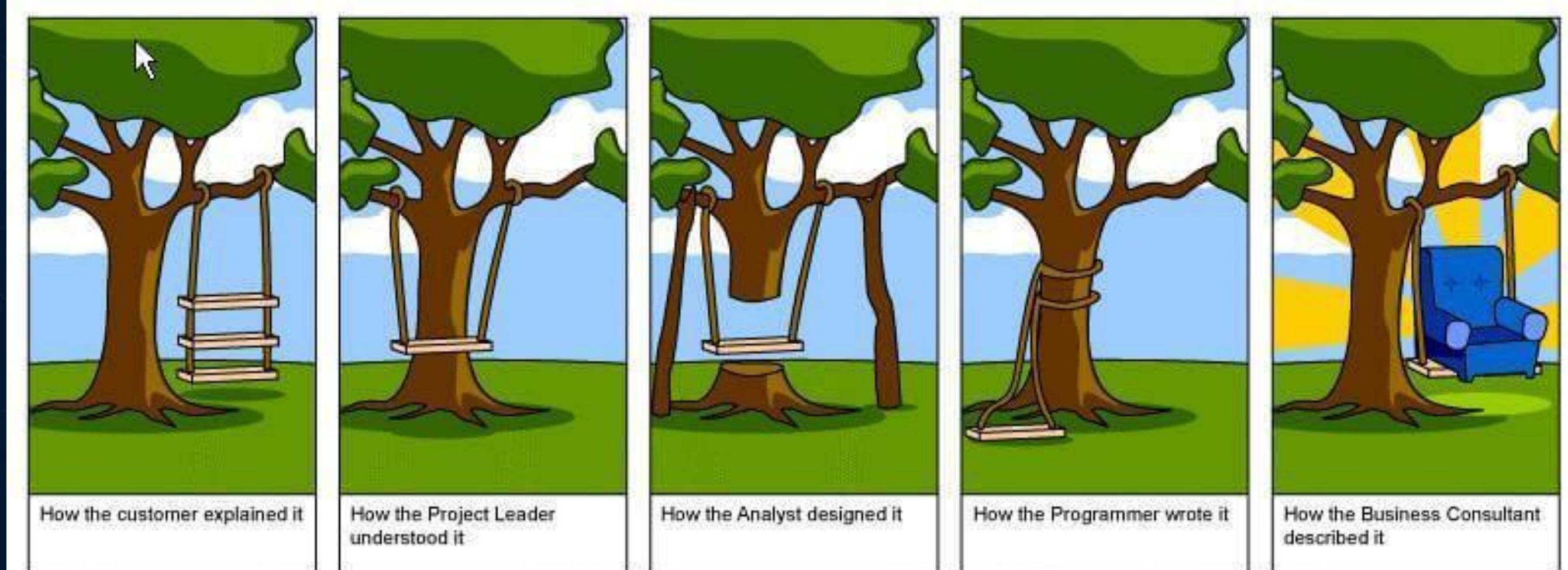
QA & TEST TEAM



Jonah's Deep Mind Map Thinking in Action! 😊



The Ugly TRUTH of IT Projects!



THE PROBLEM#2

The business team and stakeholders wants to track what was **done** and released to QA & UAT.

As a Release Manager, Jonah needs to send the release notes with Jira Ticket IDs every Tuesday, and Thursday every week ☺

As a Release Manager, Jonah needs to send the release notes for QA and UAT every sprint (3-week sprint) ☺

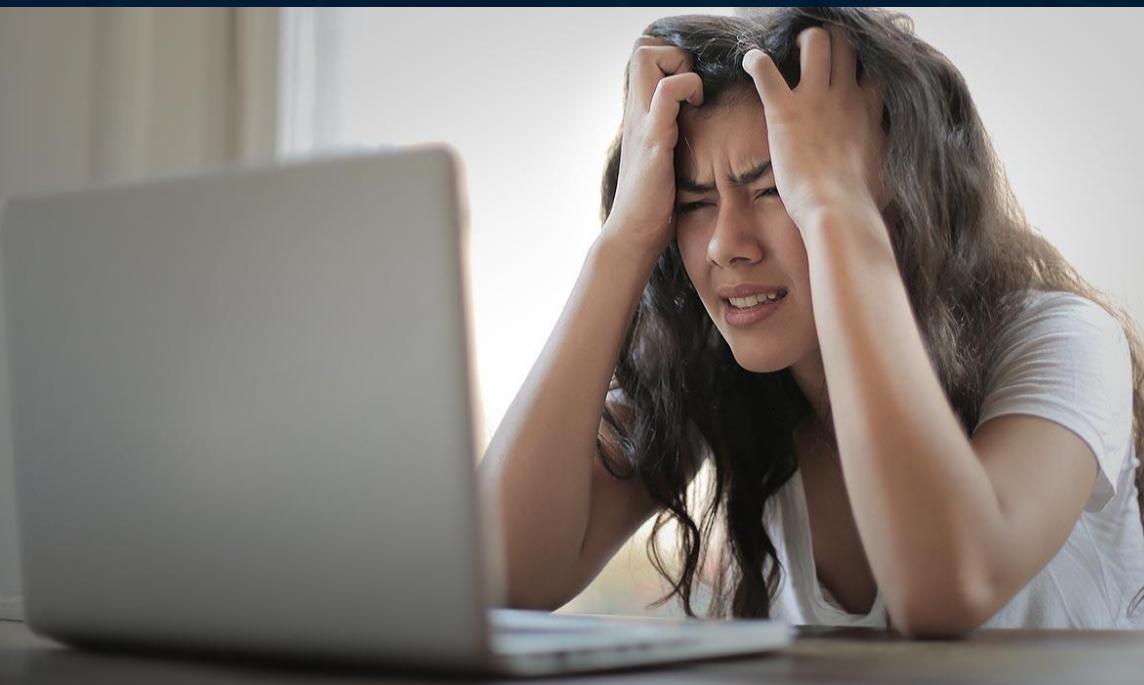


THE DILEMMA



Source Code (CLOUD)

?



Project Tasks Tracking Management
ON-PREMISE

?

?



CI/CD (Cloud)



ARE WE REALLY IN THE CLOUD?







“First, solve the problem.
Then, **write the code**”.

- John Johnson





PROBLEM

DRY – DON'T
REPEAT
YOURSELF!

MANUAL
TEDIOUS
TASKS

TIME WASTERS

BORING TASKS



.NET



Proposed Solution



Azure
Durable Functions

 Microsoft Azure



Azure DevOps Services REST API

<https://learn.microsoft.com/en-us/rest/api/azure/devops>



Bitbucket (Cloud) REST API

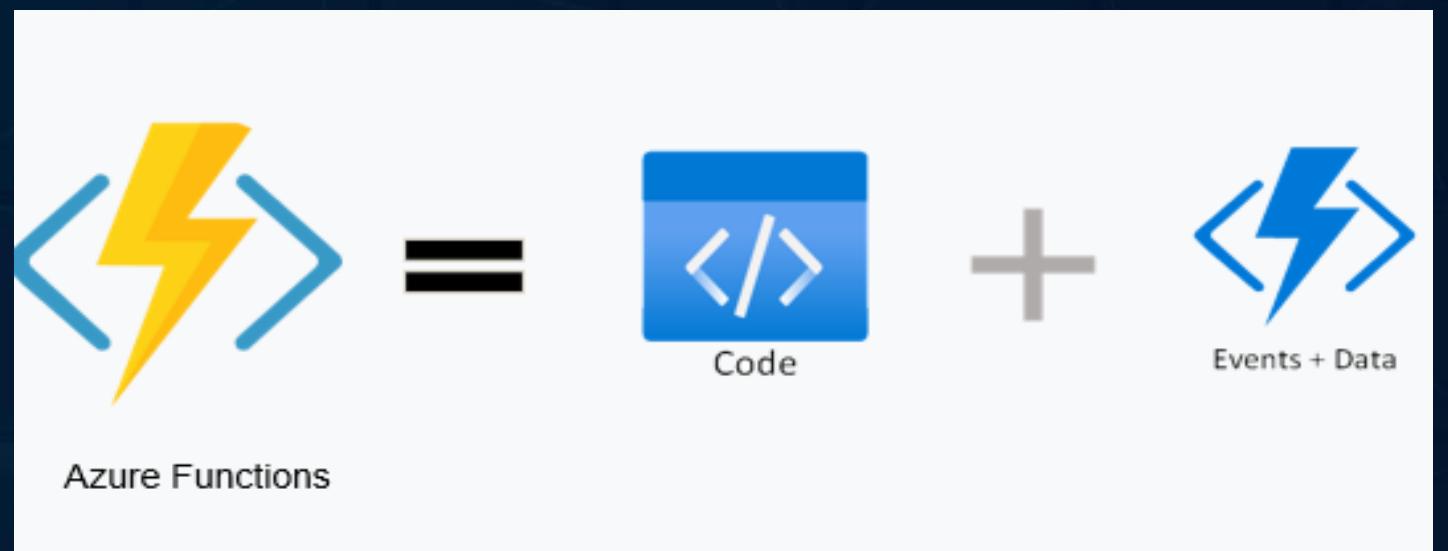
<https://developer.atlassian.com/>



Azure Service Bus



Azure Durable Functions

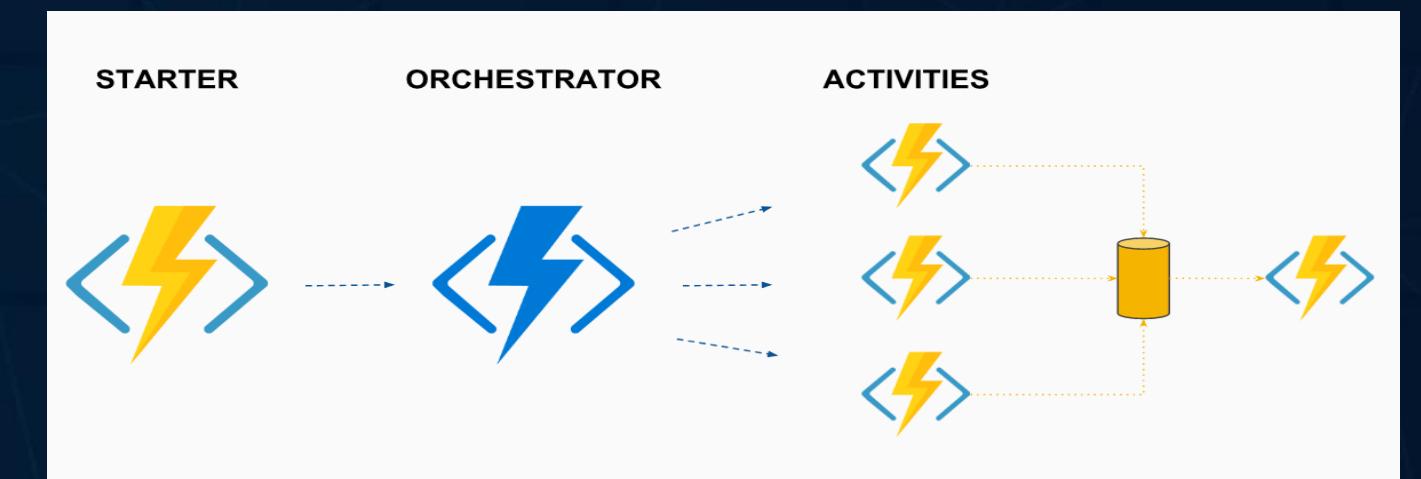


Extension of Azure Functions (Serverless Compute)

Stateful Workflows in Serverless Architecture

Manages, Checkpoints and Restarts State for you

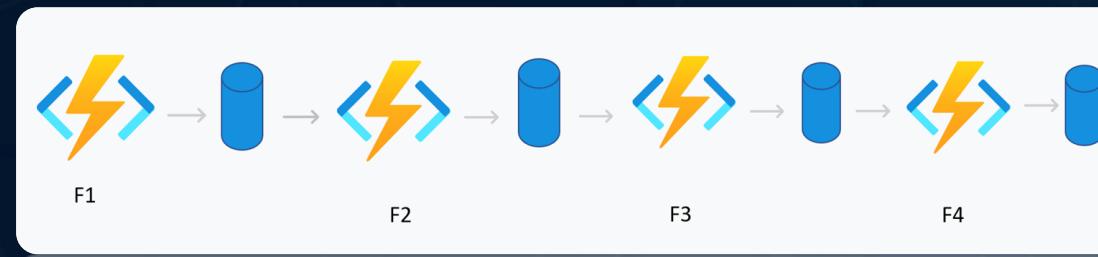
C#, Javascript, Python, F#, Powershell, Java



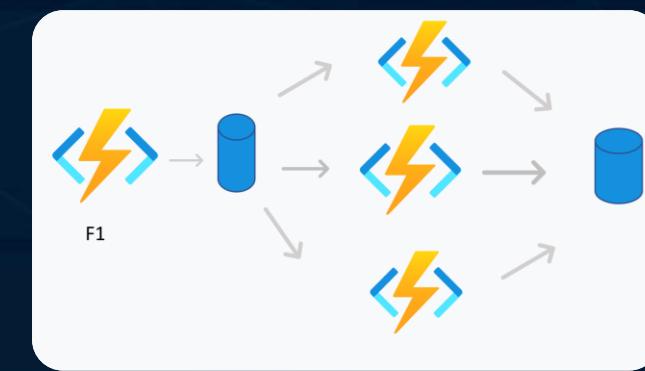
Learn more about Azure Functions



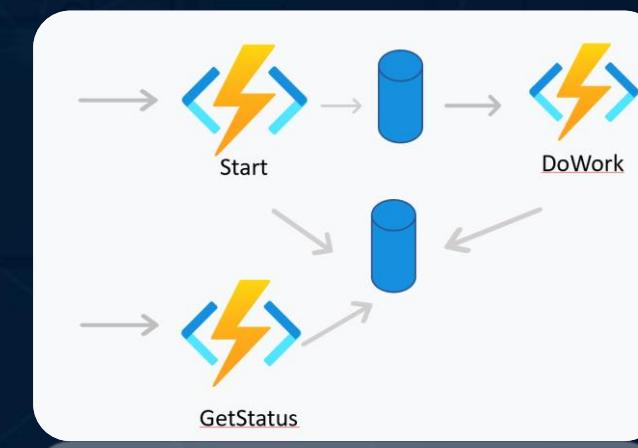
Application Patterns of Azure Durable Functions



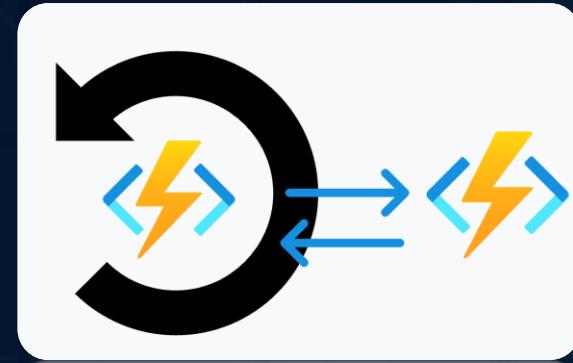
Function Chaining



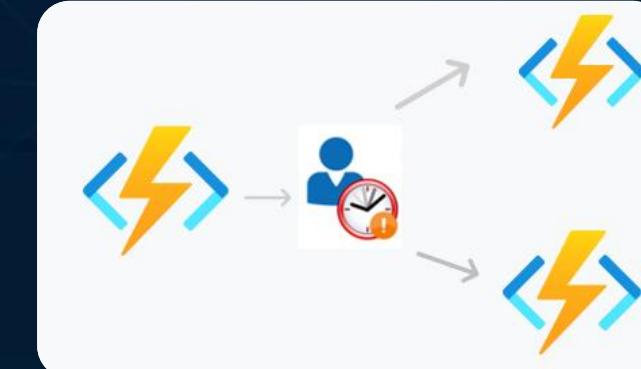
Fan out / Fan in



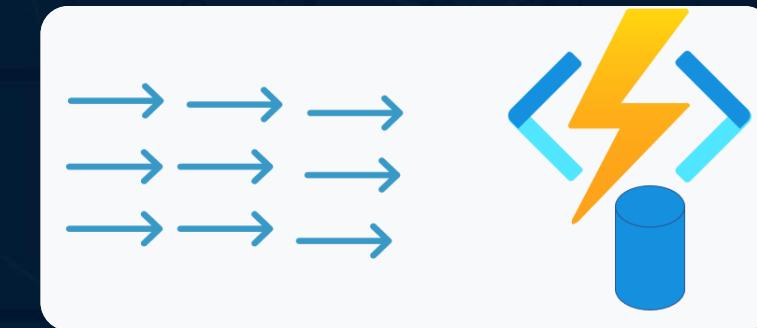
Async HTTP API



Monitor

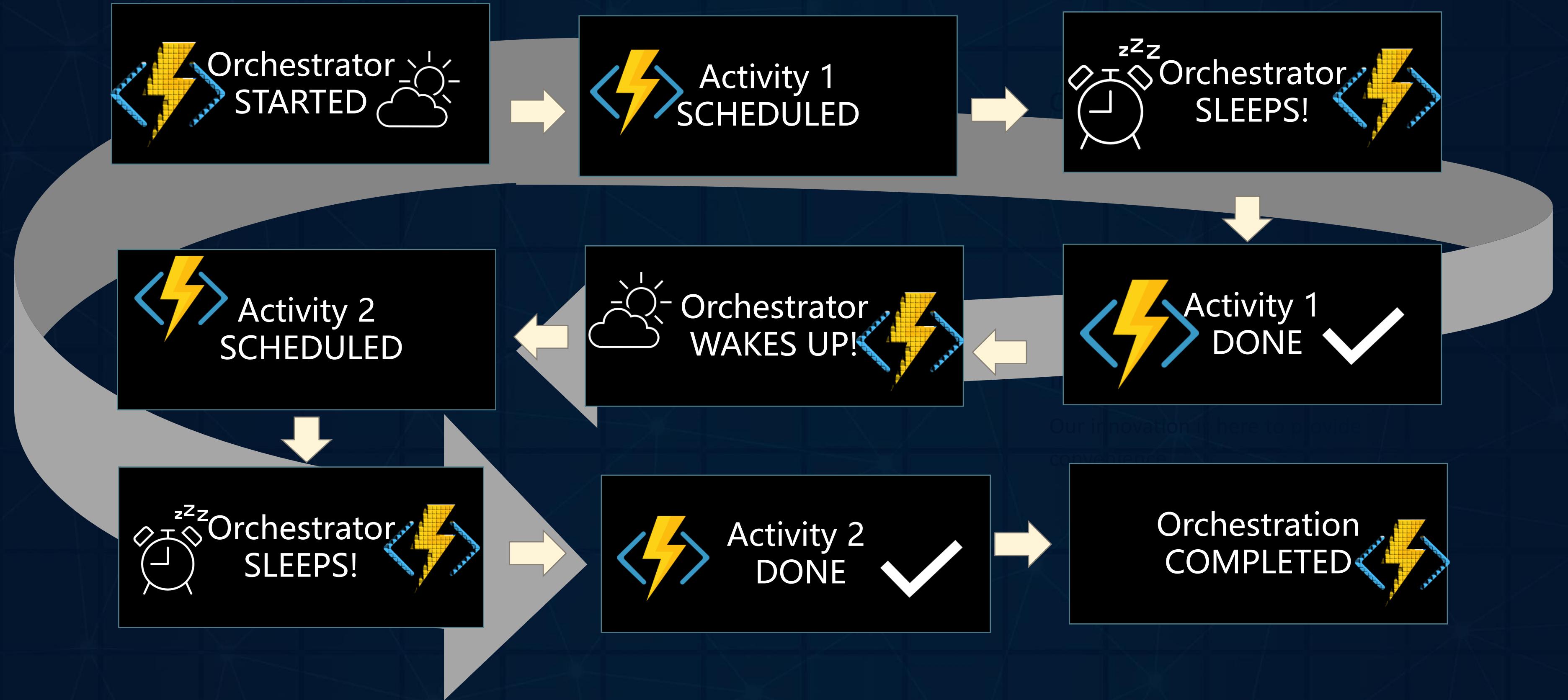


Human Interaction



Aggregator

Function Chaining in Action



The Orchestrator
can be a
sleepyhead
but it is GOOD!





COSTS

Pay as you go

Pay for compute capacity by the second, with no long-term commitments or upfront payments. Increase or decrease consumption on demand.

Azure savings plan for compute

Save money across select compute services globally by committing to spend a fixed hourly amount for 1 or 3 years, unlocking lower prices until you reach your hourly commitment. Suited for dynamic workloads while accommodating for planned or unplanned changes.

Region:

Central US

Currency:

United States – Dollar (\$) USD

Display pricing by:

Hour

Consumption

Azure Functions consumption plan is billed based on per-second resource consumption and executions. Consumption plan pricing includes a monthly free grant of 1 million requests and 400,000 GB-s of resource consumption per month per subscription in pay-as-you-go pricing across all function apps in that subscription. Azure Functions Premium plan provides enhanced performance and is billed on a per second basis based on the number of vCPU-s and GB-s your Premium Functions consume. Customers can also run Functions within their App Service plan at regular App Service plan [rates](#).

Meter	Free Grant (Per Month)	Pay as you go
Execution Time*	400,000 GB-s	\$0.000016/GB-s
Total Executions*	1 million executions	\$0.20 per million executions

Free grants apply to paid, consumption subscriptions only.

Note—A storage account is created by default with each Functions app. The storage account is not included in the free grant. Standard [storage rates](#) and [networking rates](#) charged separately as applicable.

[View details on regional availability](#)

Premium plan

Azure Functions Premium plan provides the same features and scaling mechanism used on the Consumption plan (based on number of events) with no cold start, enhanced performance and VNET access. Billing for the Premium plan is based on the number of core seconds and memory allocated across instances. There is no execution charge with the Premium plan. At least one instance must be allocated at all times per plan. To learn more, see [Azure Premium Plan details](#).

When creating or scaling your plan, you can choose between three instance sizes. You will be billed for the total number of cores and memory provisioned, per second that each instance is allocated to you. Your app can automatically scale out to multiple instances as needed.



AZURE SERVICE BUS

Messaging service on cloud

Connect any applications,
devices, and services

A messaging broker



FEATURES

- Scheduling
- Dead-lettering
- Transactional processing
- Message ordering
- Expiration (TTL)
- Duplicate detection
- Deferring
- Pub/Sub
- Filtering

Version

Azure DevOps Services REST API 7.1

Filter by title

Get started with Azure DevOps & REST

- > Accounts
- > Approvals And Checks
- > Artifacts
- > Artifacts Package Types
- > Audit
- > Build
- > Core
- > Dashboard
- > Distributed Task
- > Extension Management
- > Git
- > Graph
- > Identities
- > Member Entitlement Management
- > Notification
- > Operations
- > Permissions Report

Learn /

Azure DevOps Services REST API Reference

Article • 07/21/2022 • 7 minutes to read • 7 contributors

[Feedback](#)

Welcome to the Azure DevOps Services/Azure DevOps Server REST API Reference.

Representational State Transfer (REST) APIs are service endpoints that support sets of HTTP operations (methods), which provide create, retrieve, update, or delete access to the service's resources. This article walks you through:

- The basic components of a REST API request/response pair.
- Overviews of creating and sending a REST request, and handling the response.

Most REST APIs are accessible through our [client libraries](#), which can be used to greatly simplify your client code.

Components of a REST API request/response pair

A REST API request/response pair can be separated into five components:

1. The **request URI**, in the following form: `VERB https://{{instance}}[/{team}-`



[Latest updates](#)

INTRODUCTION AND BASICS

[Installing the Atlassian SDK](#)[Explore the SDK and atlas commands](#)[Beginner guide to Bitbucket Server plugin development](#)[Experienced guide to Bitbucket Server plugin development](#)[Updating your Stash app](#)[Building Bitbucket Server from source](#)

LEARNING

[Adding a column to the branch listing](#)[Adding commit metadata](#)[Adding code insights as part of your CI pipeline](#)[Adding to the audit log](#)[Bitbucket Server Plugin Tutorials](#)

• Bitbucket Server REST APIs

Last updated Jun 14, 2022

Bitbucket Server REST APIs

Bitbucket Server's REST APIs are for developers who want to build functionality on top of Bitbucket Server.

Quick links: [Reference](#)

Overview

Whatever you want to build, our intention is to supply the platform for it. New web or desktop applications, mobile apps, integrations, browser-based extensions or apps, Atlassian plugins, and whatever you can dream up. The Atlassian REST API provide a standard interface for interacting with Bitbucket Server and our other applications.

REST APIs provide access to resources (data entities) via URI paths. To use a REST API, your application will make an HTTP request and parse the response. Your methods will be the standard HTTP methods like GET, PUT, POST and DELETE. REST APIs operate over HTTP(s) making it easy to use with any programming language or framework.

The input and output format for the Bitbucket Server REST APIs is [JSON](#).

Bitbucket Server uses the [Atlassian REST plugin](#) to implement the Bitbucket Server APIs. The REST plugin is bundled with Bitbucket Server. You can add your own REST APIs to Bitbucket Server by creating a Bitbucket Server plugin that includes the REST plugin module. See the basic principles we employ in our [REST API design](#).

The examples below use [curl](#).

Simple Example of Retrieving Repository Commits

Want to retrieve a list of commits for a repository in Bitbucket Server via REST? Here's how easy it is.

Just GET the JSON from your Bitbucket Server server. In this example, the server is: `http://localhost:7990`. This example also uses basic authentication, with a username 'fred' and password 'fred'.



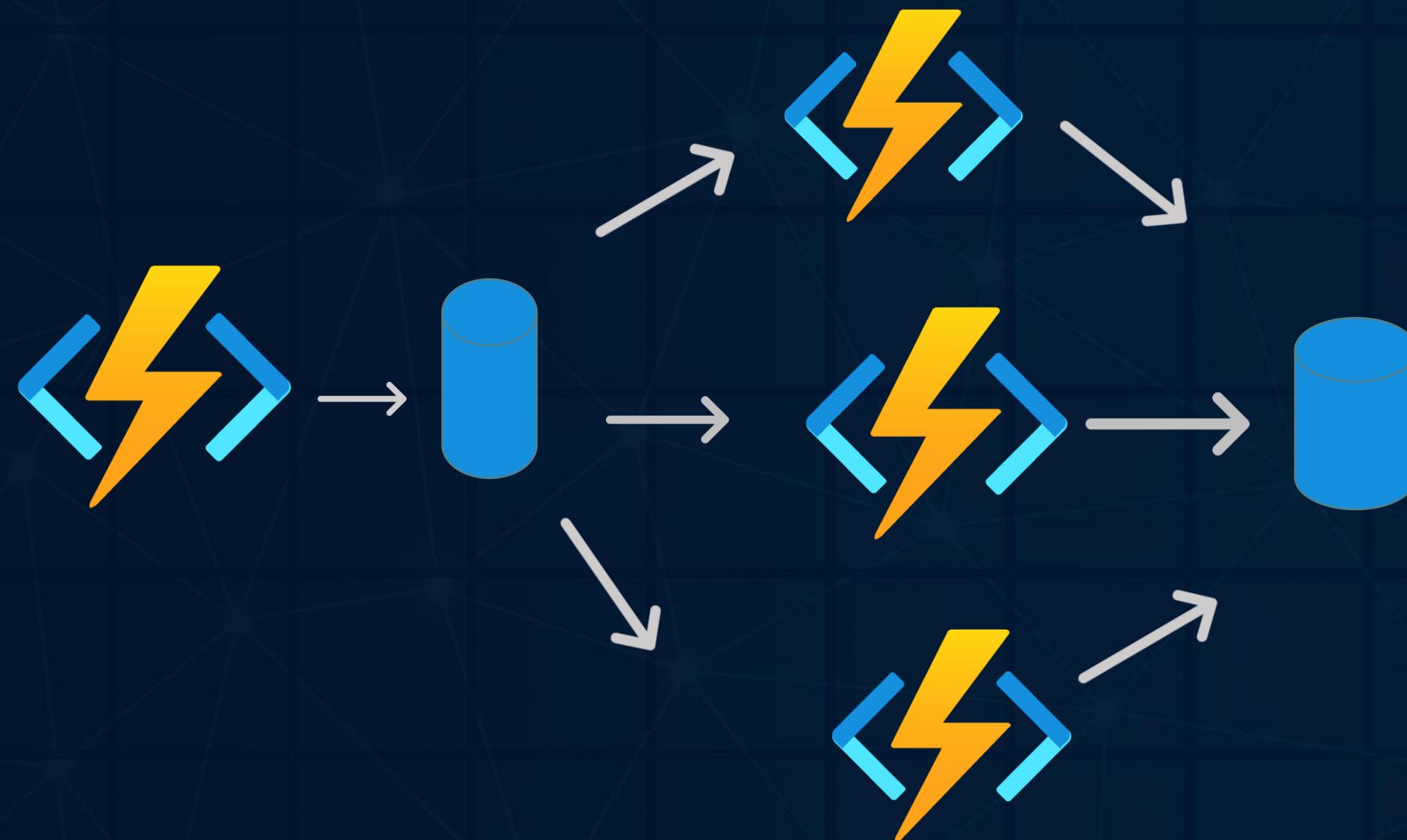
YES
LET'S DO IT!





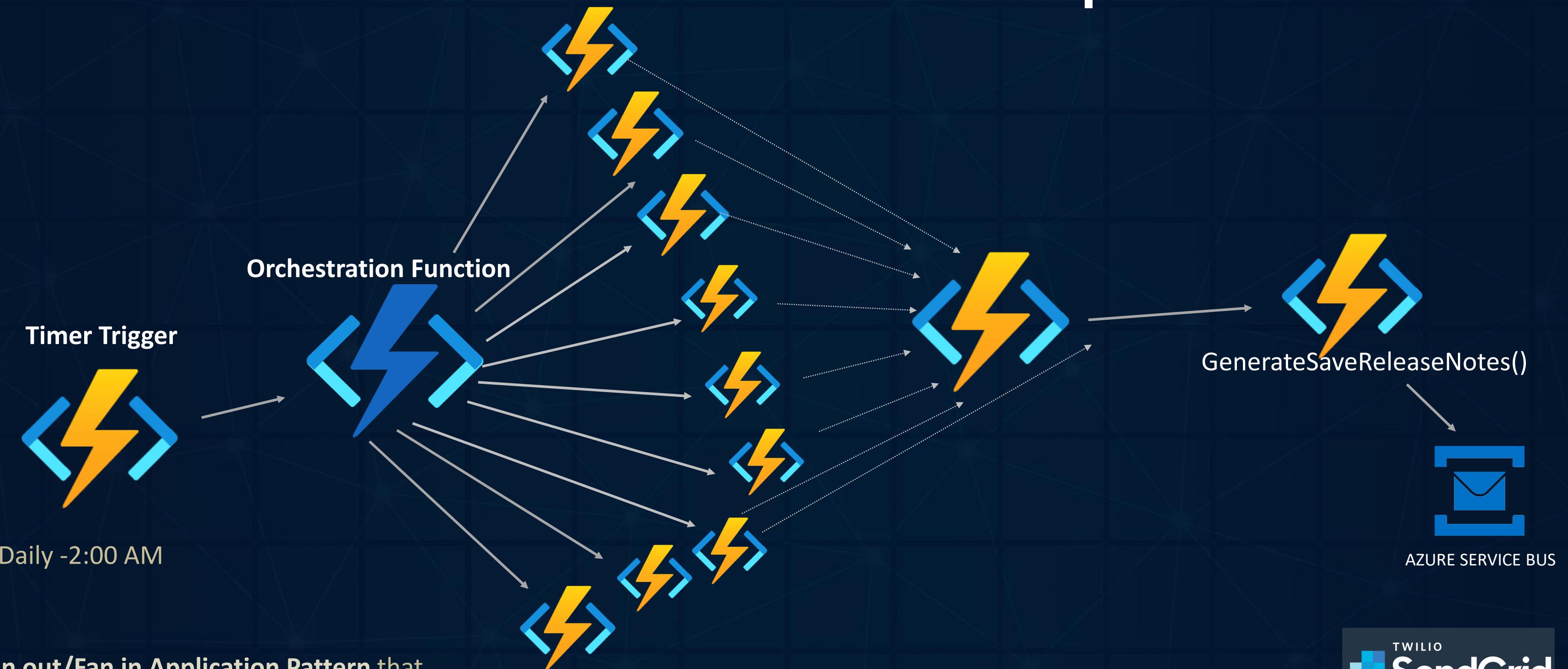


Fan-Out/Fan In Pattern



Execute multiple functions in parallel and then wait for all functions to finish and aggregate the results when done.

#1 Getting the commits from Bitbucket Cloud via Azure DevOps Rest API



Fan out/Fan in Application Pattern that
gets all Commits from Bitbucket Cloud Repositories
from 10-15 applications/web services.
Checks commit difference from since last deploy date.

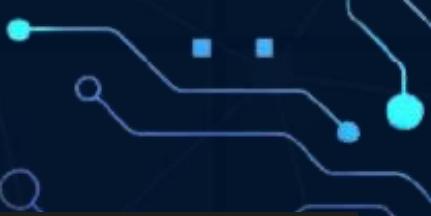


SHOW US SOME....

<code>



Getting Commits #1 - Azure DevOps REST API and C# .NET



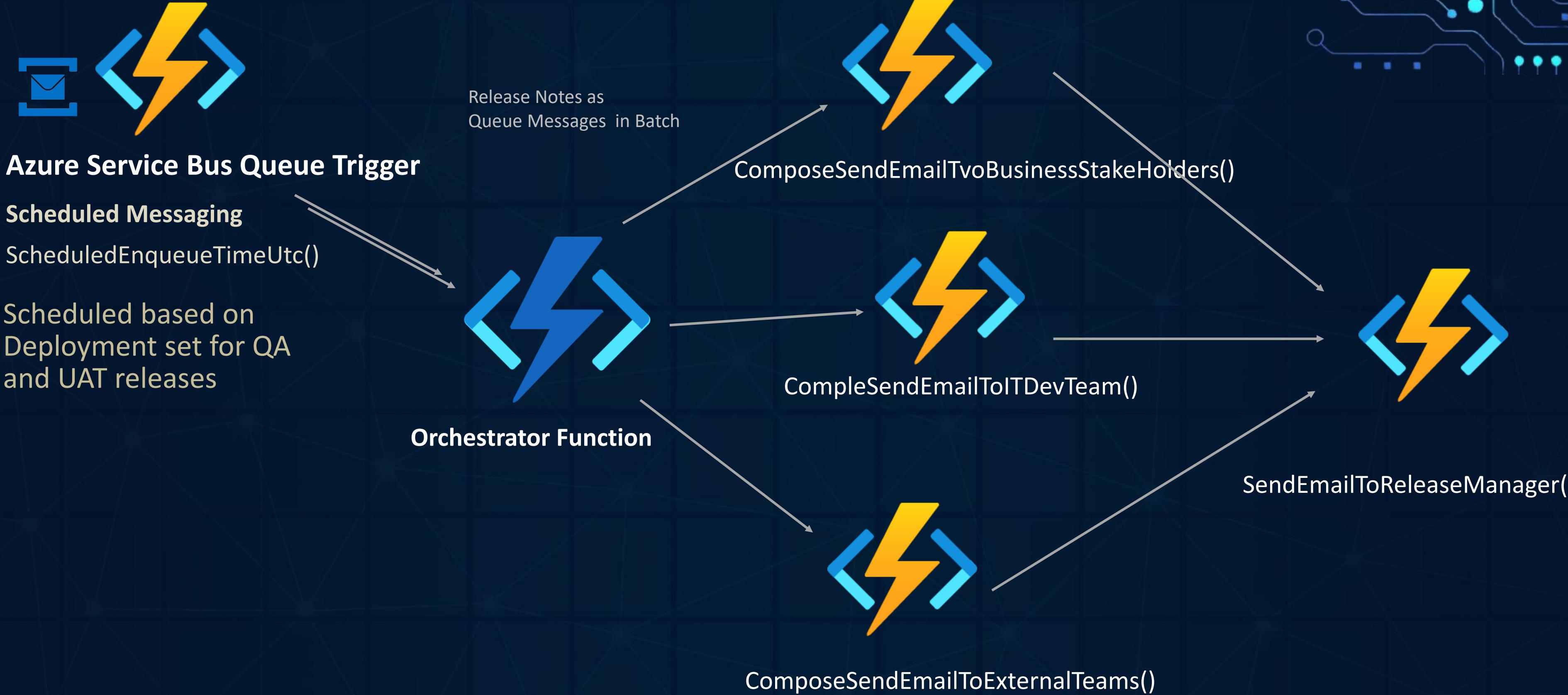
```
55     {
56         var credentials = Convert.ToBase64String(Encoding.ASCII.GetBytes($"{appSettings.BitBucketUserName}:{appSettings.Bi
57         client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Basic", credentials);
58         var isLastPage = false;
59         var nextPage = 0;
60         do
61         {
62             if (build.Repository != null)
63             {
64                 var slug = build.Repository.Url.Substring(build.Repository.Url.LastIndexOf("/") + 1,
65                     build.Repository.Url.LastIndexOf(".git") - build.Repository.Url.LastIndexOf("/") - 1);
66
67                 var url = $"https://bitbucket. .... .com/rest/api/1.0/projects/ .... /repos/{slug}/commits/" +
68                     $"?since={build.SourceVersion}&start={nextPage}";
69
70                 var response = await client.GetFromJsonAsync<CommitsResponse>(url);
71
72                 if (response != null)
73                 {
74                     isLastPage = response.IsLastPage;
75                     nextPage = response.NextPageStart ?? -1;
76                     output.AddRange(response.Values);
77                 }
78             }
79         } while (isLastPage == false);
80     }
81
82     return output;
83 }
```

Getting Commits #2 – Azure DevOps REST API and C# .NET



```
30 public async Task<IEnumerable<Commit>> GetCommitMessagesBetweenLatestReleaseAndHeadCommit(Application application, string environment)
31 {
32     using (var client = new HttpClient())
33     {
34         var credentials = Convert.ToBase64String(Encoding.ASCII.GetBytes($"{appSettings.AzureDevOpsUsername}:{appSettings.AzureDevOpsPAT}"));
35         client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Basic", credentials);
36
37         var url = $"https://vsrm.dev.azure.com/{application}/_apis/release/definitions/?definitionId={(int)application}";
38         var releaseDefinition = await client.GetFromJsonAsync<ReleaseDefinition>(url);
39         var versionInEnvironment = releaseDefinition?.Environments
40             .Single(k => k.Name.Equals(environment, StringComparison.InvariantCultureIgnoreCase));
41
42         url = $"https://vsrm.dev.azure.com/{application}/_apis/Release/releases/{versionInEnvironment?.CurrentRelease?.Id}";
43         var release = await client.GetFromJsonAsync<Release>(url);
44
45         url = $"https://dev.azure.com/{application}/_apis/build/builds/{release?.Artifacts?.Single()?.DefinitionReference?.Version?.Id}";
46         var build = await client.GetFromJsonAsync<Build>(url);
47
48         return await GetCommitsCommentsSince(build);
49     }
50 }
51 }
```

Cont. #2 Auto Scheduling of Release Notes Email



DEVSECOPS

AZURE KEY VAULT
MANAGED IDENTITIES
ZERO TRUST
MFA
CODE SCANS
RBAC AND POLICIES
etc.

OPPORTUNITIES

AUTOMATION
OF MANUAL
CODABLE REPEATABLE
PROCESSES IF POSSIBLE



TODO!

DEVOPS

DEPLOYMENT GATES CONTROL
UNIFIED PLACE FOR
INFRASTRUCTURE AND APPLICATION SOURCE CODE
ON-PREMISE LINUX AGENTS to SERVERS AZURE

CONTINUOUS IMPROVEMENT

PAY TECH DEBT
.NET 3.0 to .NET 7

LOOK INTO CONTAINERS / AKS

AZURE MONITOR / APP INSIGHTS

Do more with less

- 1 Be data-driven and optimize with Azure
- 2 Deliver efficiency with automation and AI
- 3 Innovate with a cloud developer platform
- 4 Re-energize your workforce with Microsoft 365
- 5 Protect everything, everyone, everywhere

Solve problems
by automating it
if possible



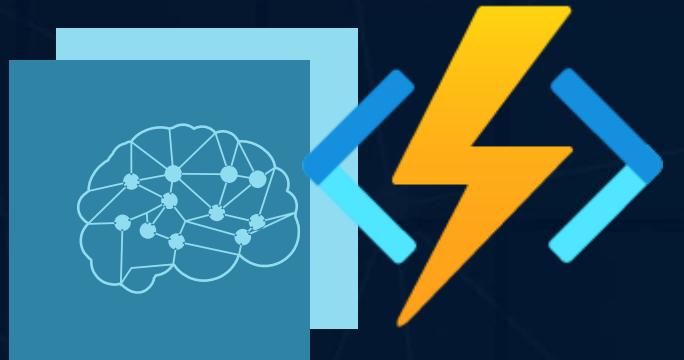
WORK LESS
WITH SERVERLESS



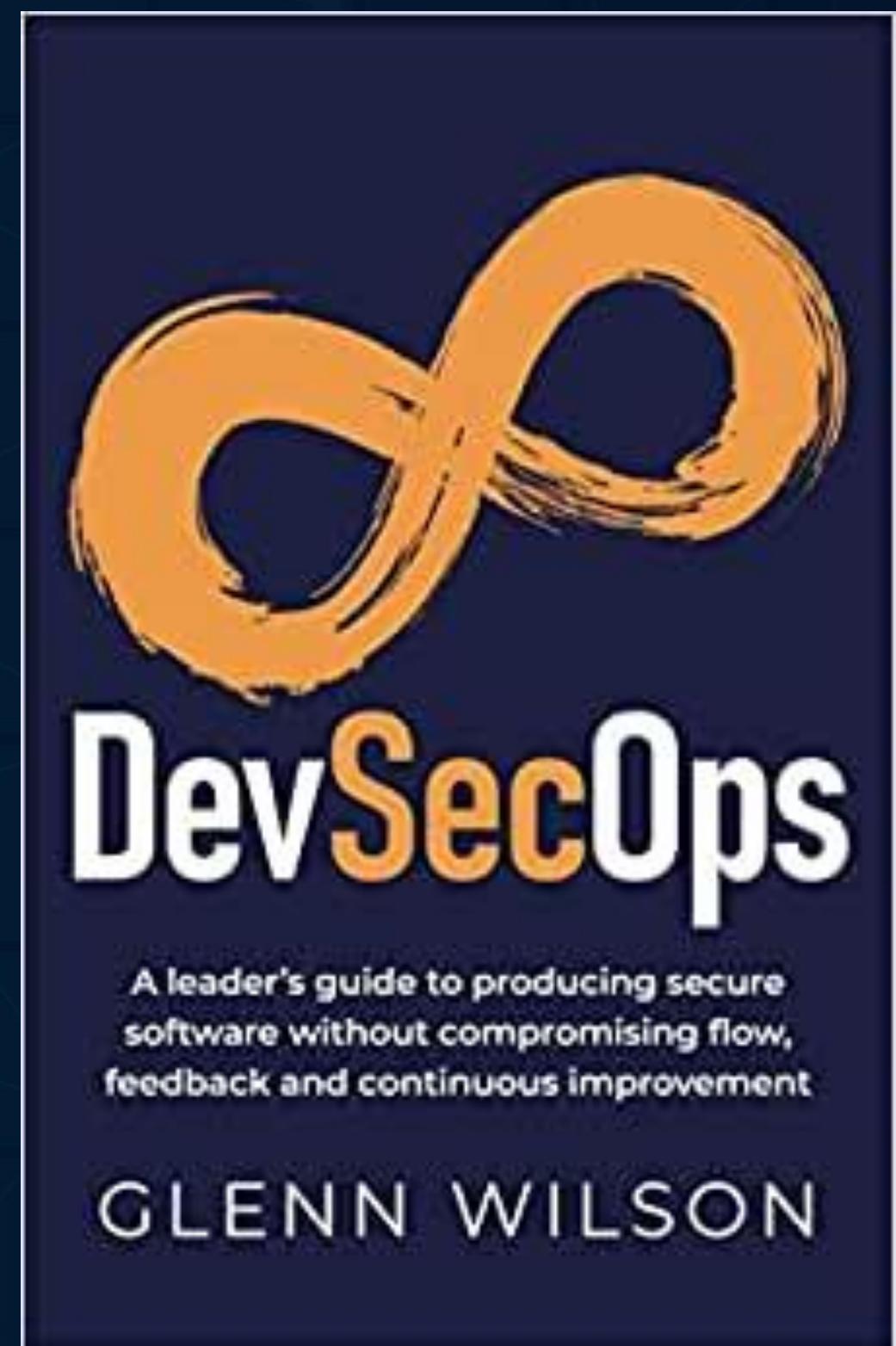
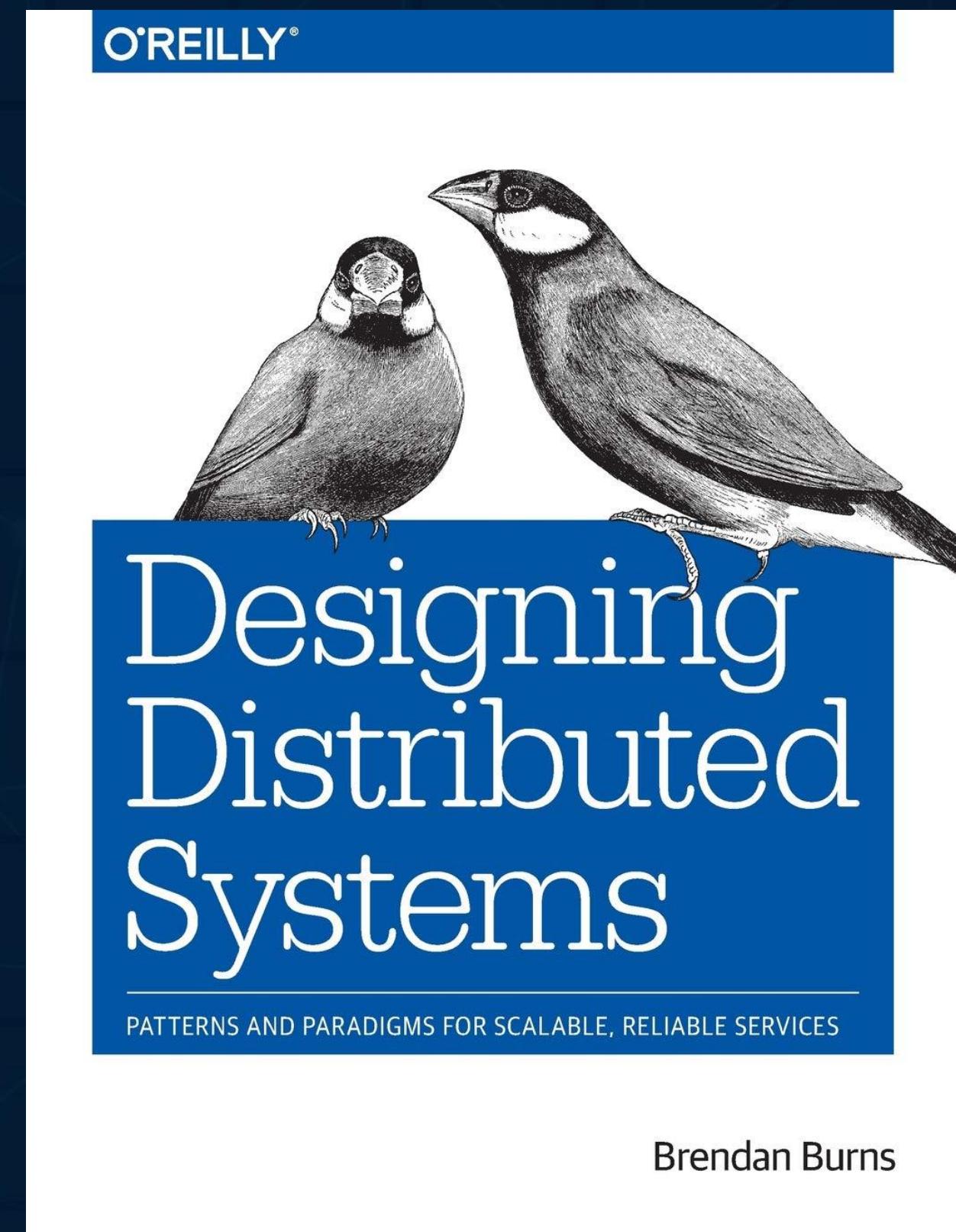
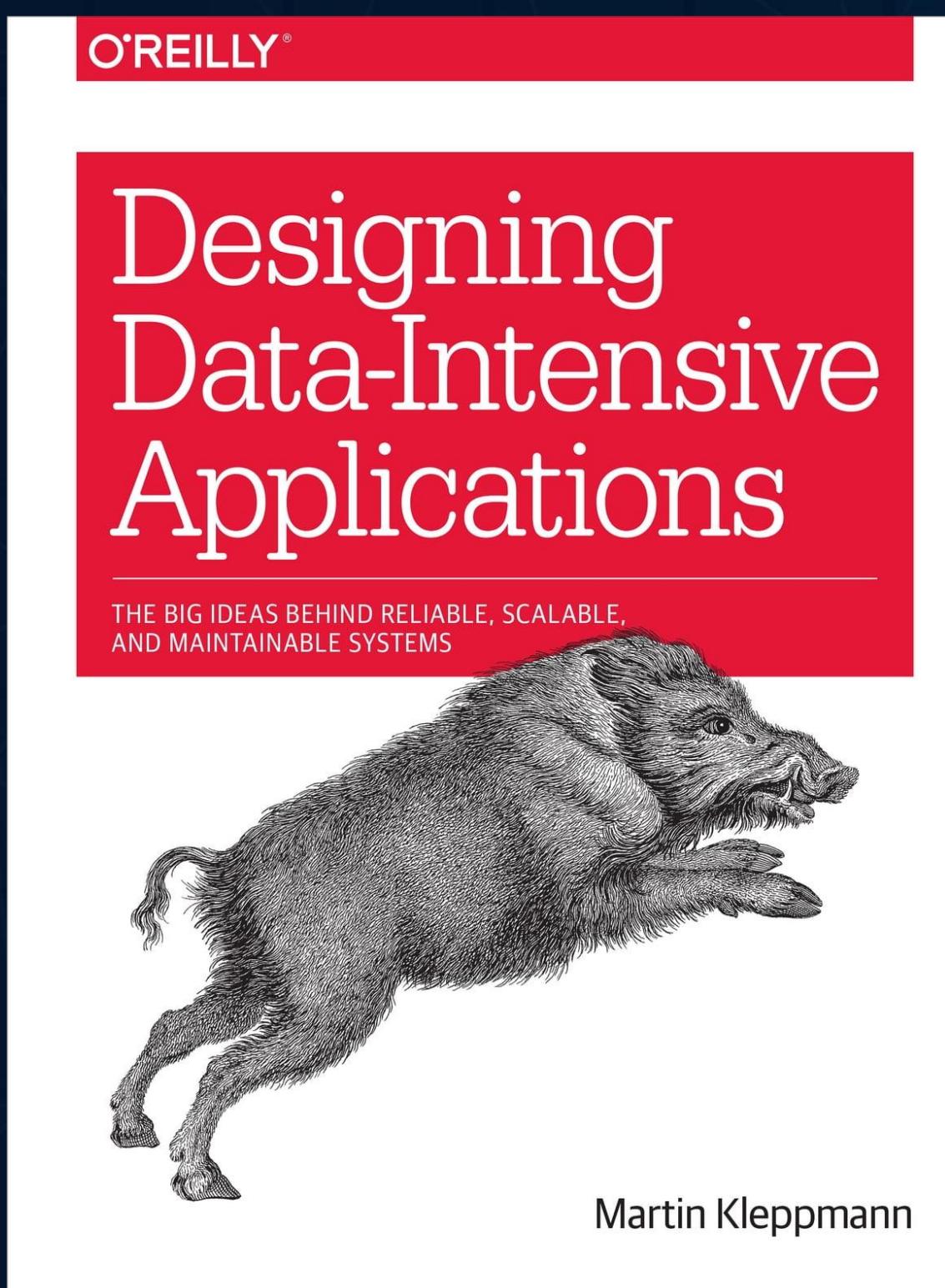


LEARNING RESOURCES for QR Code Scanners

AZURE DURABLE FUNCTIONS



LEARNING RESOURCES for Book Worms!



LEARNING RESOURCES

Azure Cognitive Services is a set of pre-trained, customizable AI models based on Microsoft AI research, enabling access to sophisticated language, vision, decision-making, and speech capabilities through simple API calls. Azure Cognitive Services do not require previous ML experience for integration. In addition, many of these existing models are extensible by training with custom data to fit specific knowledge domains.

In addition to these services, Cognitive Research...

Better together: Visual Studio + GitHub + Azure

Table S.1 shows a series of categories and services that are currently available. Note that this list is growing, and we will cover a few of these services following sections.

Services	Language	Speech
Anomaly Detector	Language Understanding	Speech-to-Text
Content Moderator	QnAMaker	Text-to-Speech
Text Analytics	Text Analytics	Speech Recognition
Translator	Translator	Speaker Identification

* All services have a free tier that you can use to get started.
Table S.1 A quick glance at the Cognitive Services APIs

Visual Studio, GitHub, and Azure provide developers with the tools, platforms, and service workspace needed for modern app development. By combining the AI development, testing, and debugging features of Visual Studio with the open-source knowledge and automation power of GitHub, Microsoft's cloud development teams all the tools they need to work efficiently and collaboratively. GitHub Enterprise, Azure DevOps make it easier to manage progress, measure team metrics, and optimize processes. Visual Studio and GitHub work seamlessly together, providing developers with the most advanced code-to-cloud workflow.

How to build cloud-native applications on Azure

With the combined power of Visual Studio, Azure, and GitHub in Visual Studio subscriptions, you're able to bundle key tools and services to empower your developers to quickly and efficiently deliver modern apps:

- Tools like CI and CD built into GitHub and DevOps allow automating workflows such as running automated tests and deploying to environments, increasing the agility and productivity of development teams.

Depending on the requirements of the software being built, you may have to take different architecture approaches to comply with business needs. Microsoft Azure offers tools and services that can account for any scenario:

- Business-critical applications
- API-first applications
- Real-time data processing
- Geo-distributed applications

Let's take a closer look at each scenario and see if it applies to your organization.

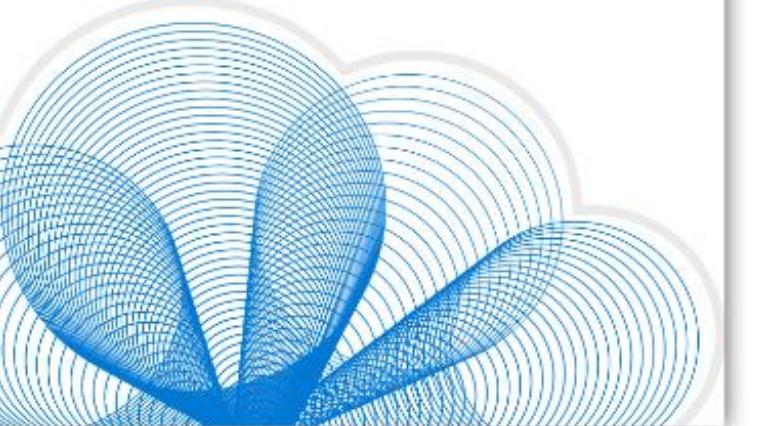
The Developer's Guide to Azure

SECOND EDITION

E-Book series

Microsoft Azure

New content on:
Cloud-native apps, dev tools, DevOps, data & AI, and security



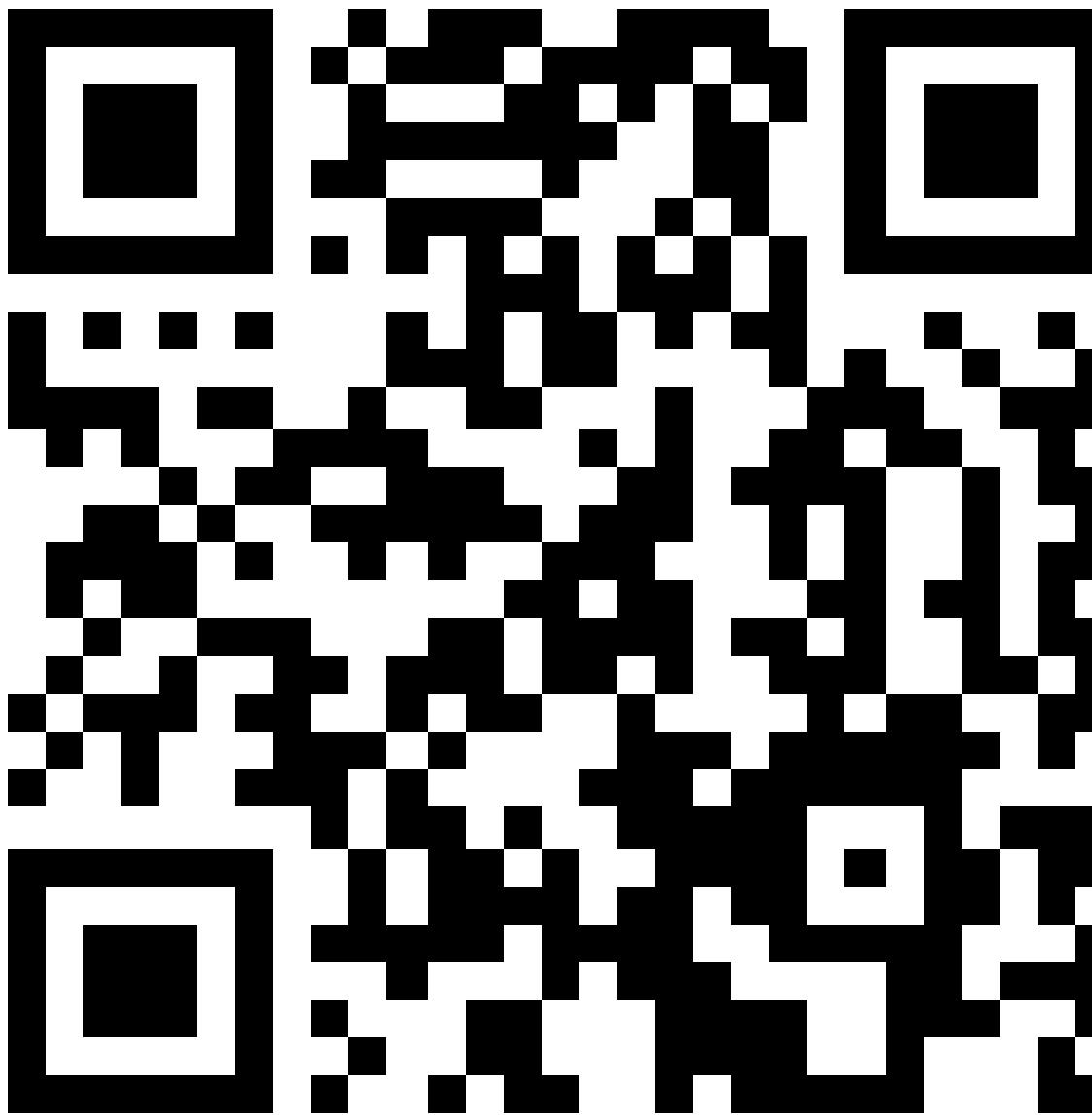
Jonah's Book Project



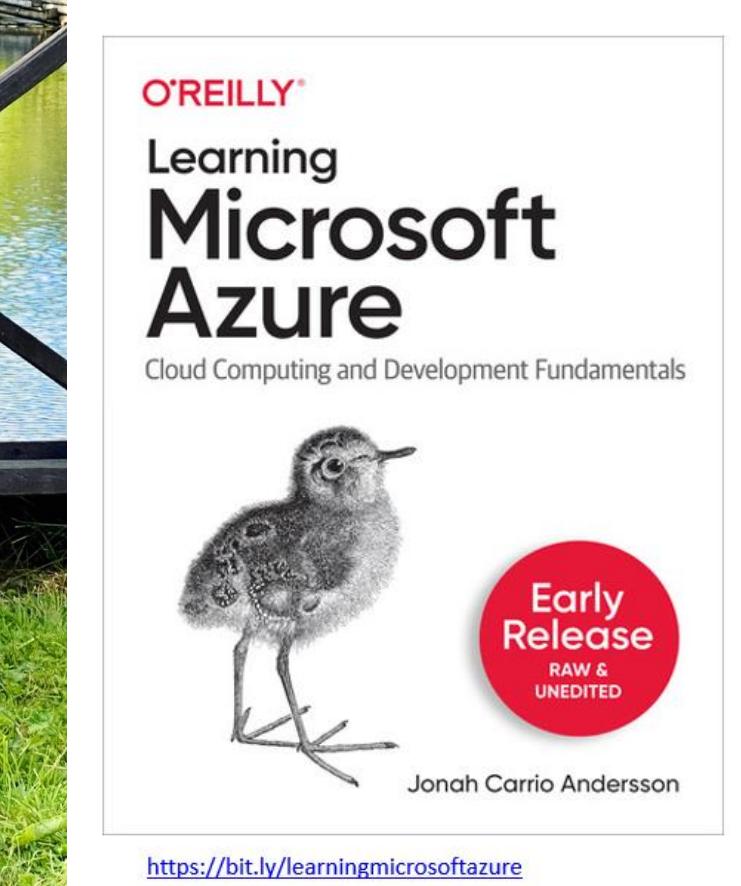
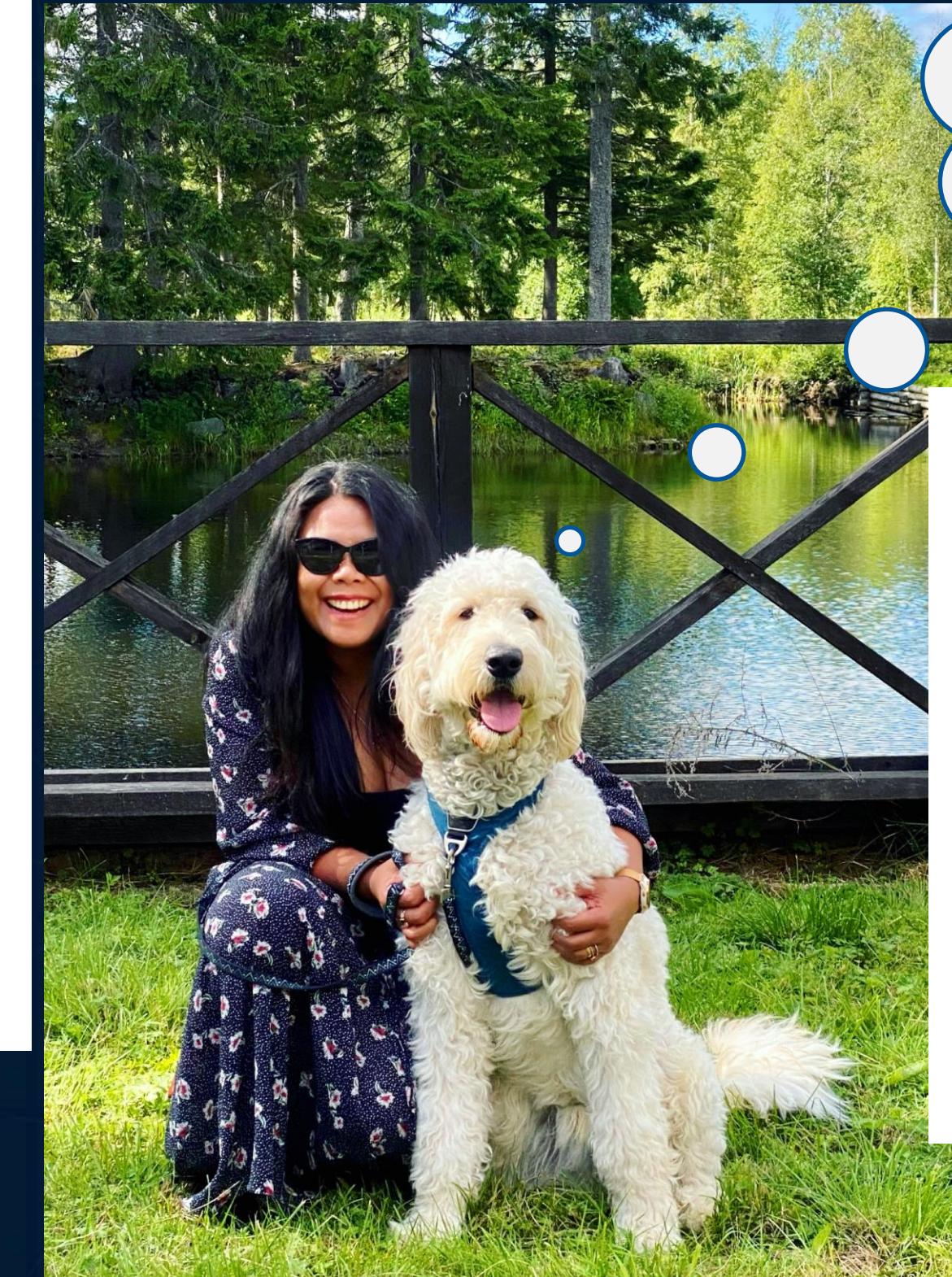
<https://bit.ly/learningmicrosoftazure>

AZURE22

THANK YOU & LET'S CONNECT!



THANK YOU & LET'S CONNECT!



<https://bit.ly/learningmicrosoftazure>



QUESTIONS, SHARING, DISCUSSION

