Microsoft

Azure DevOps

Christian Dennig
Andreas Mock

Agenda

- Theorie
  - What is DevOps?
  - DevOps Building blocks
  - DevOps on Azure
- Praxis / HandsOn
  - Let’s Hack ☺
  - https://github.com/CSA-OCP-GER/DevOpsHackEssentials

Who are we?

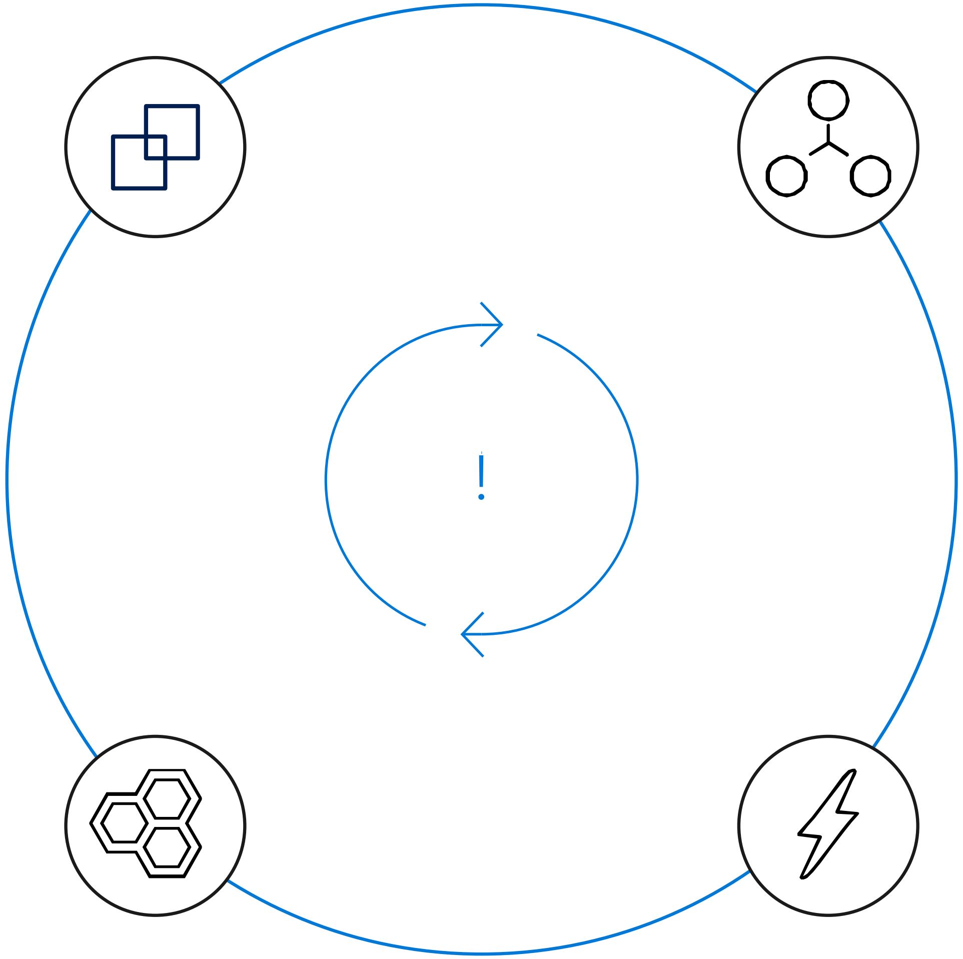
Christian Dennig
Microsoft
Cloud Solution Architect
@chrisdennig

Andreas Mock
Microsoft
Cloud Solutions Architect
@andreasM009

Feedback erwünscht...

https://aka.ms/icbdevops

Common software delivery challenges



Low Deployment Frequency

High Lead Time For Changes

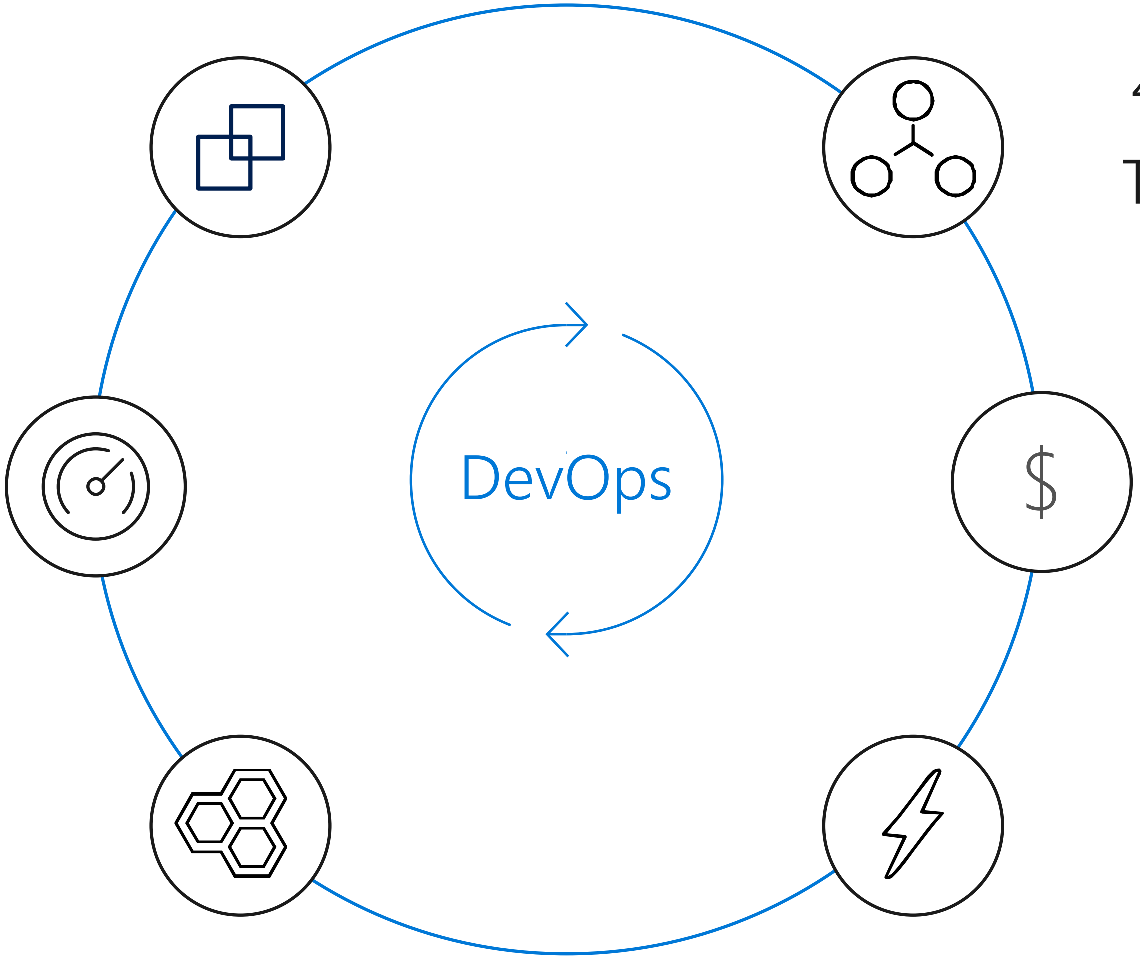
High Change Failure Rate

Long Mean Time to Recover

High Performance DevOps Companies Achieve...

46x Deployment Frequency

440x Faster Lead Time For Changes



20% Faster Time to Market

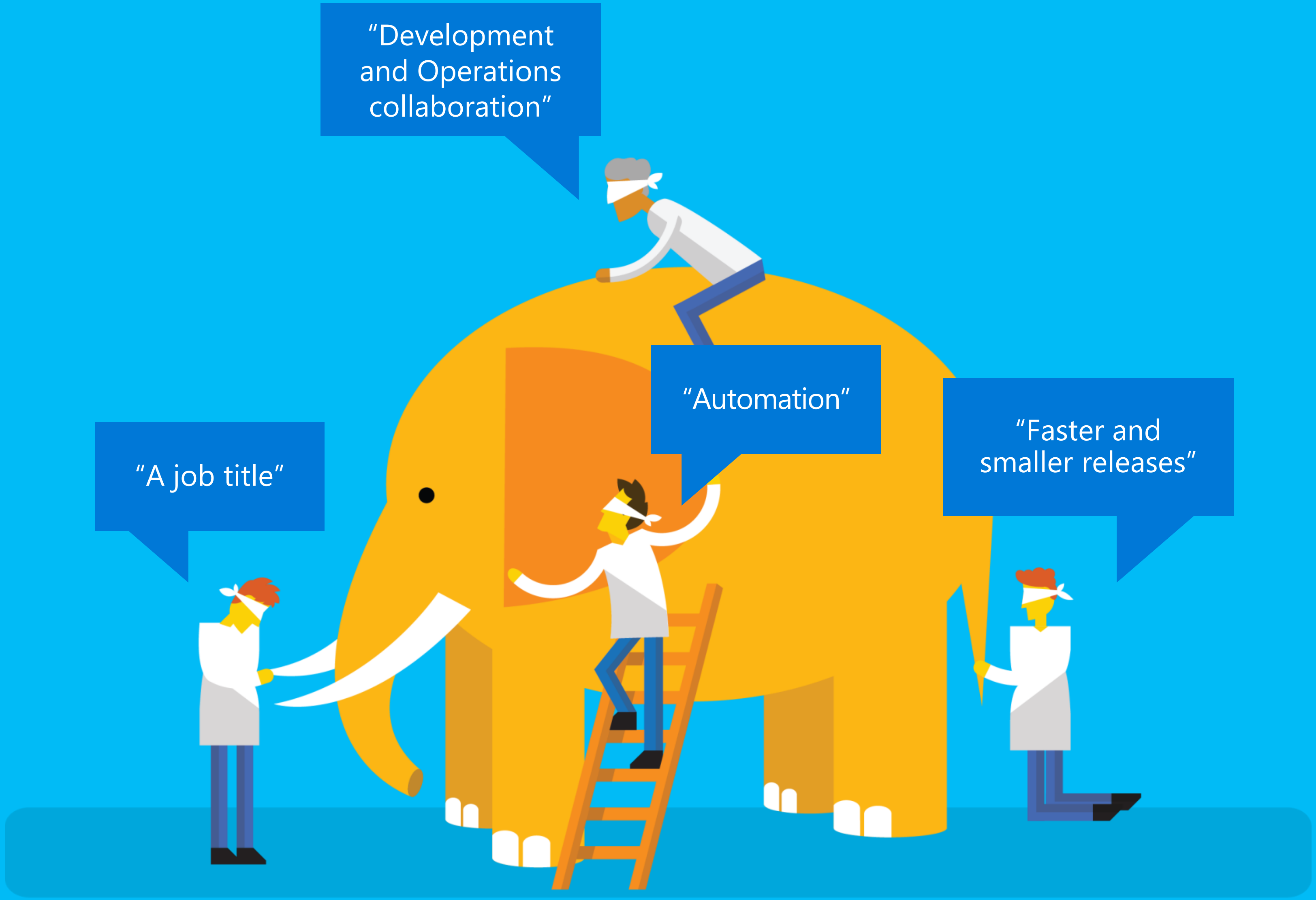
20% Increase Revenue

5x Lower Change Failure Rate

96x Faster Mean Time to Recover

Source: 2017 State of DevOps Report: Presented by Puppet and DORA

What is DevOps?


“Development and Operations collaboration”
“A job title”
“Automation”
“Faster and smaller releases”

What is DevOps?

People. Process. Technology.

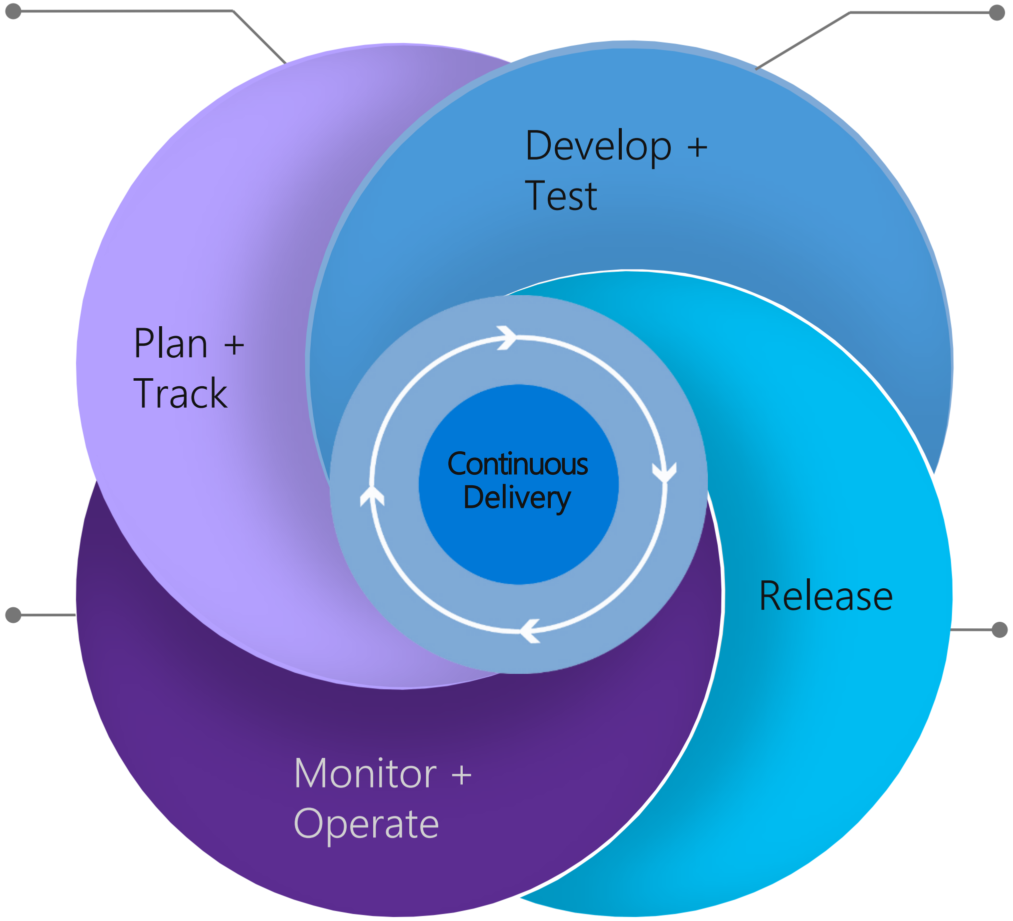
DevOps is the union of people, process, and technology to enable continuous delivery of value to your end users.

Plan & Track

Agile Planning
Dashboards
Kanban Boards
Taskboards

Build & Test

Git Source Control
Modern Code Workflow
Continuous Integration
Continuous Testing
Package Management
Open Source Compliance



Monitoring & Operations

Telemetry
Diagnostics
Analysis
Compliance

Deployment

Deployment of app and infrastructure
PaaS, IaaS and Containers

Deliver faster and more reliably.

Simple

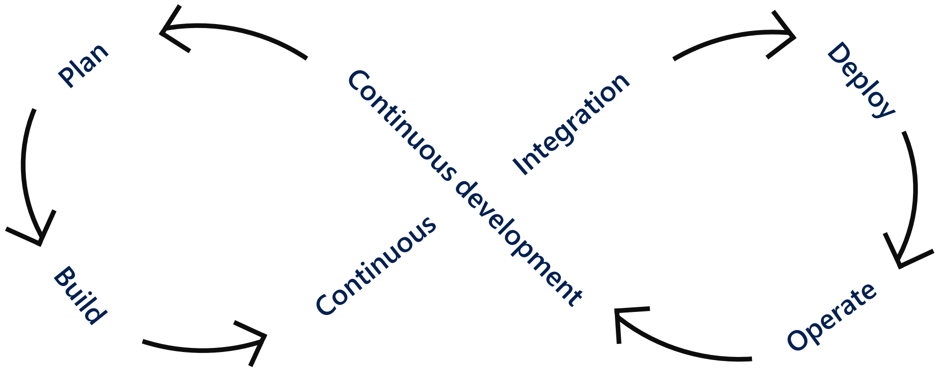
End-to-end DevOps Solutions

Reliable

Enterprise-grade platforms & services

Open

Use the tools & languages you know



Plan

Build

Continuous Integration

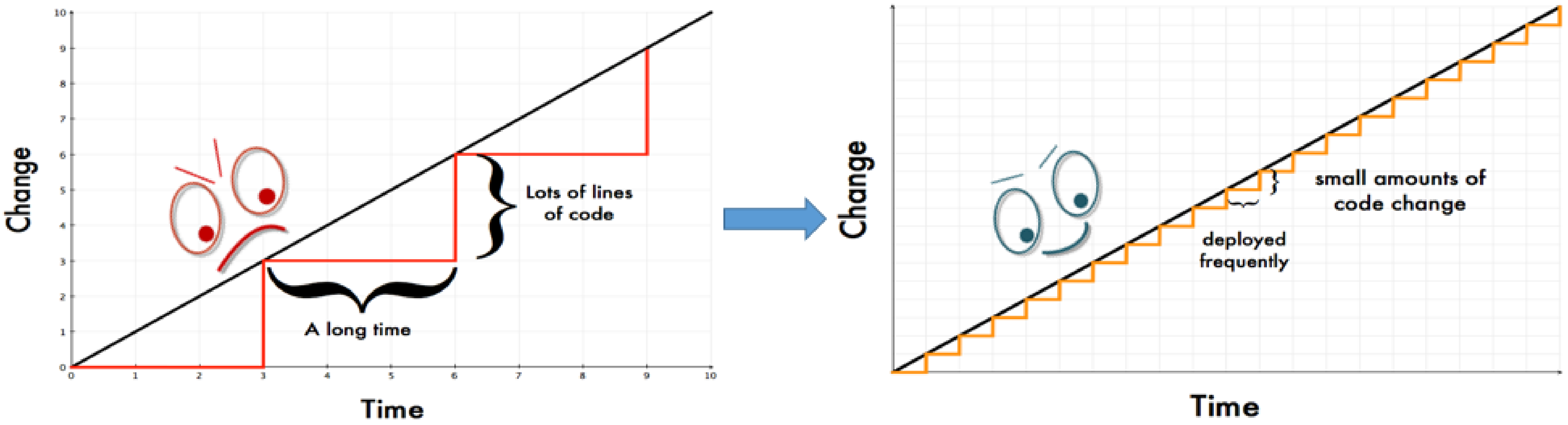
Deploy

Operate

Continuous development

From slow delivery cycles to fast delivery cycles

John Allspaw's visual – From slow delivery cycles to fast delivery cycles



Azure DevOps

Azure Boards

Agile Management

Azure Repos

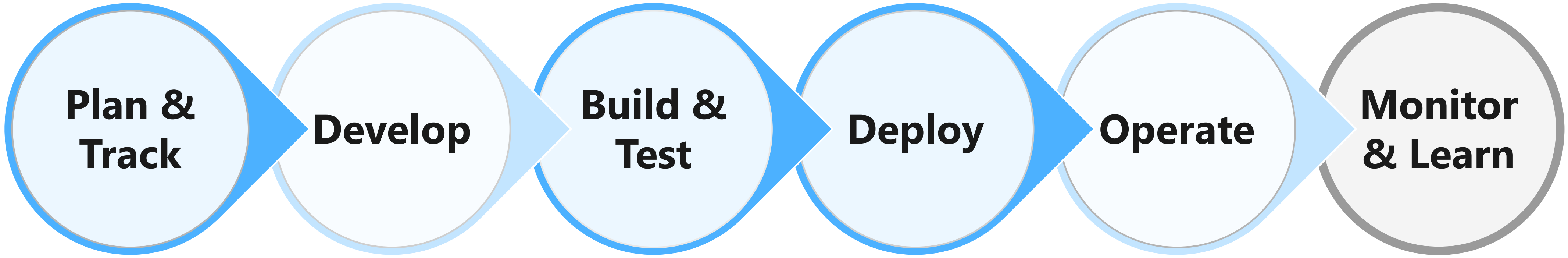
Source Control

Azure Pipelines

CI/CD

Azure Test Plans

Testing



Plan & Track

Develop

Build & Test

Deploy

Operate

Monitor & Learn

Azure Boards

Agile Management

Introduce an Agile Projectmanagement

Introduce a well-defined agile process like Scrum, or

define your own agile process and visualize and control the value flow through Kanban.

But always use the basic ingredients of an agile Process

- Product vision statement
- Product Roadmap
- Release Plan
- Product Backlog
- Sprint Backlog
- Increment

Planing: Minimum Viable Product

A minimum viable product has just those core features sufficient to deploy the product, and no more. Developers typically deploy the product to a subset of possible customers - such as early adopters thought to be more forgiving, more likely to give feedback, and able to grasp a product vision from an early prototype or marketing information.

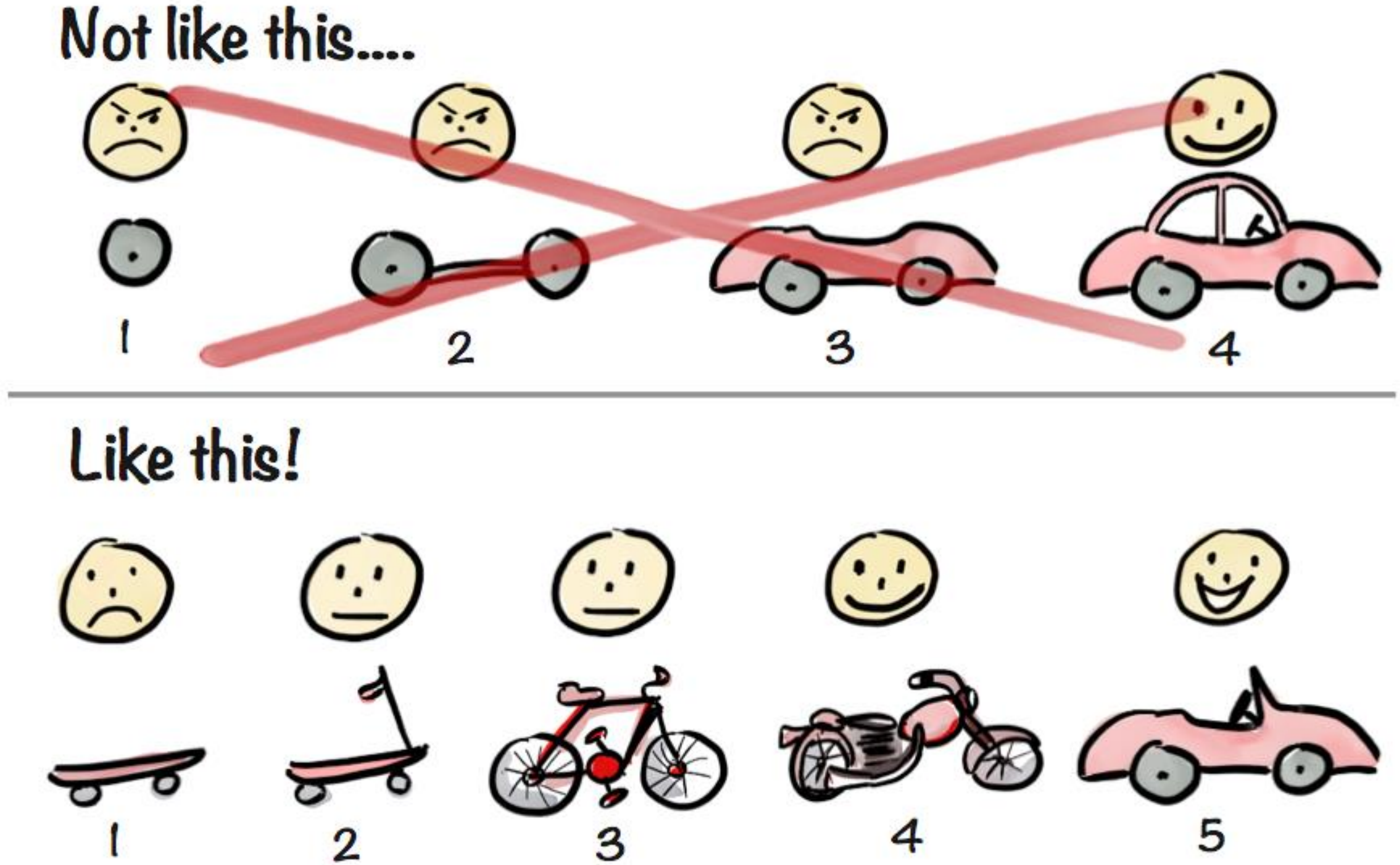
Steve Blank typically refers to minimum viable product as minimum feature set.

Purpose:

- Be able to test a product hypothesis with minimal resources
- Accelerate learning
- Reduce wasted engineering hours
- Get the product to early customers as soon as possible

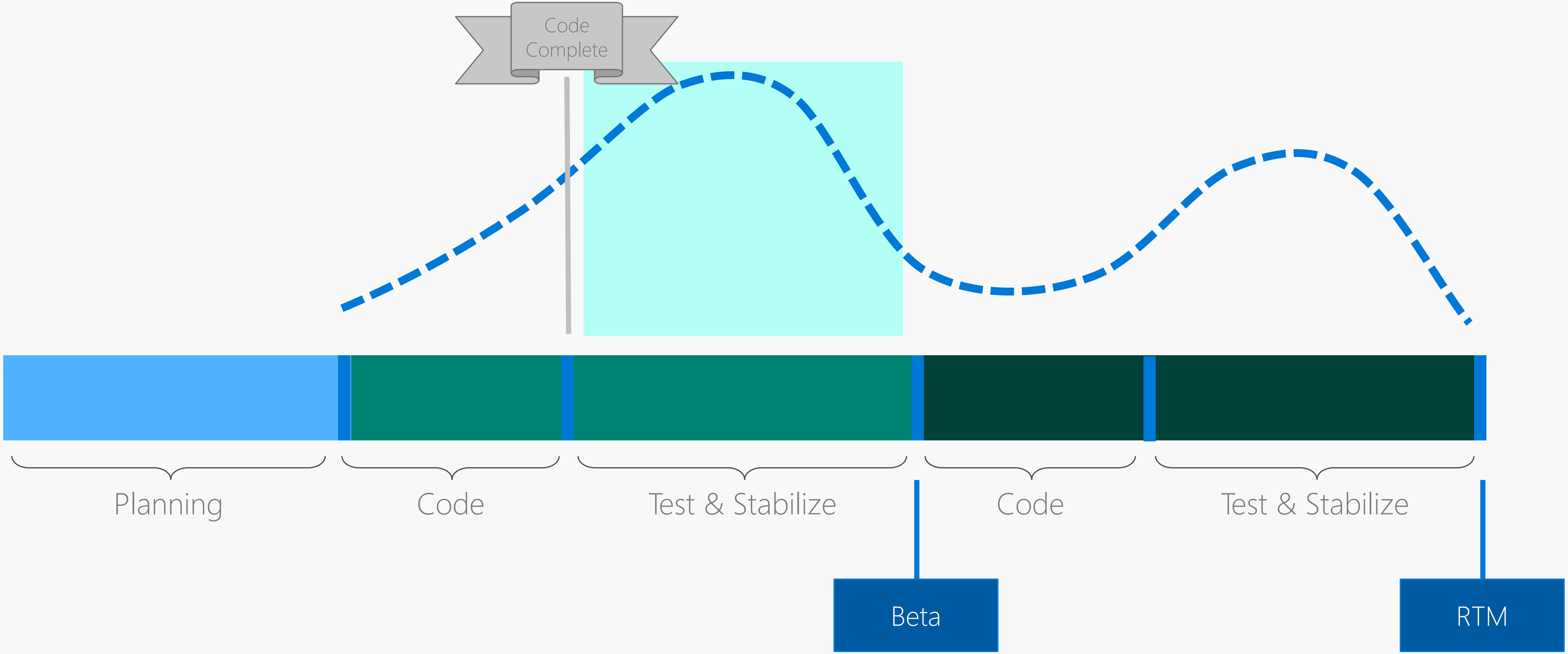
https://en.wikipedia.org/wiki/Minimum_viable_product

Planing: Minimum Viable Product

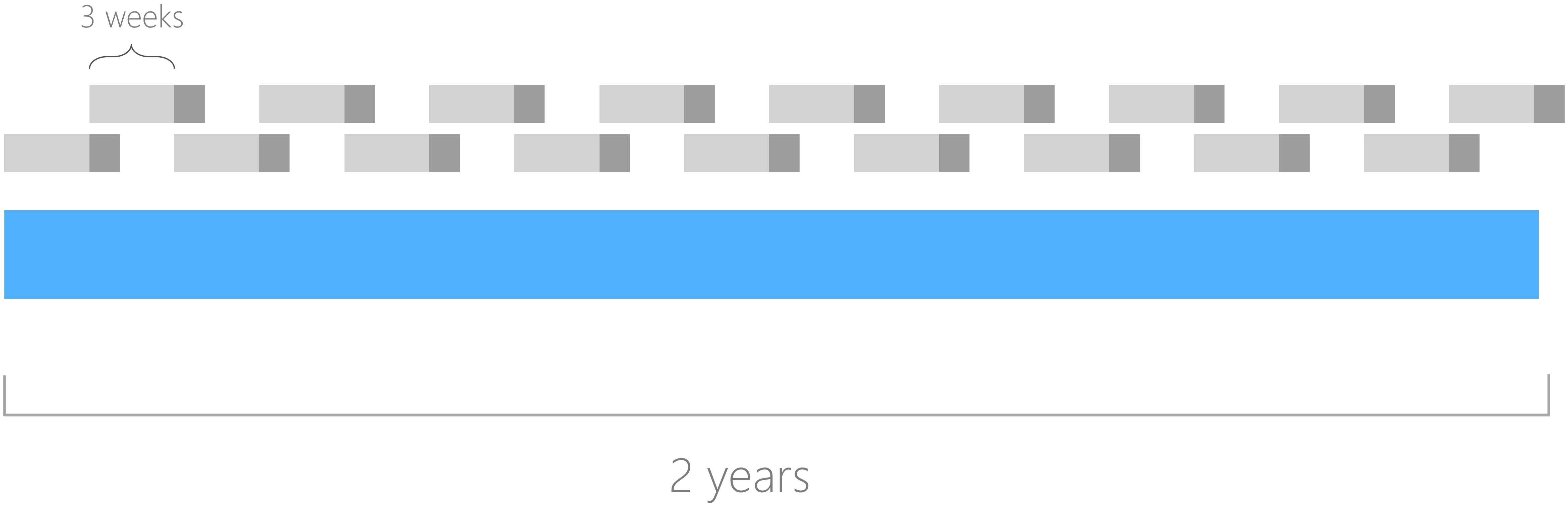


by Henrik Kniberg

The way before Agile

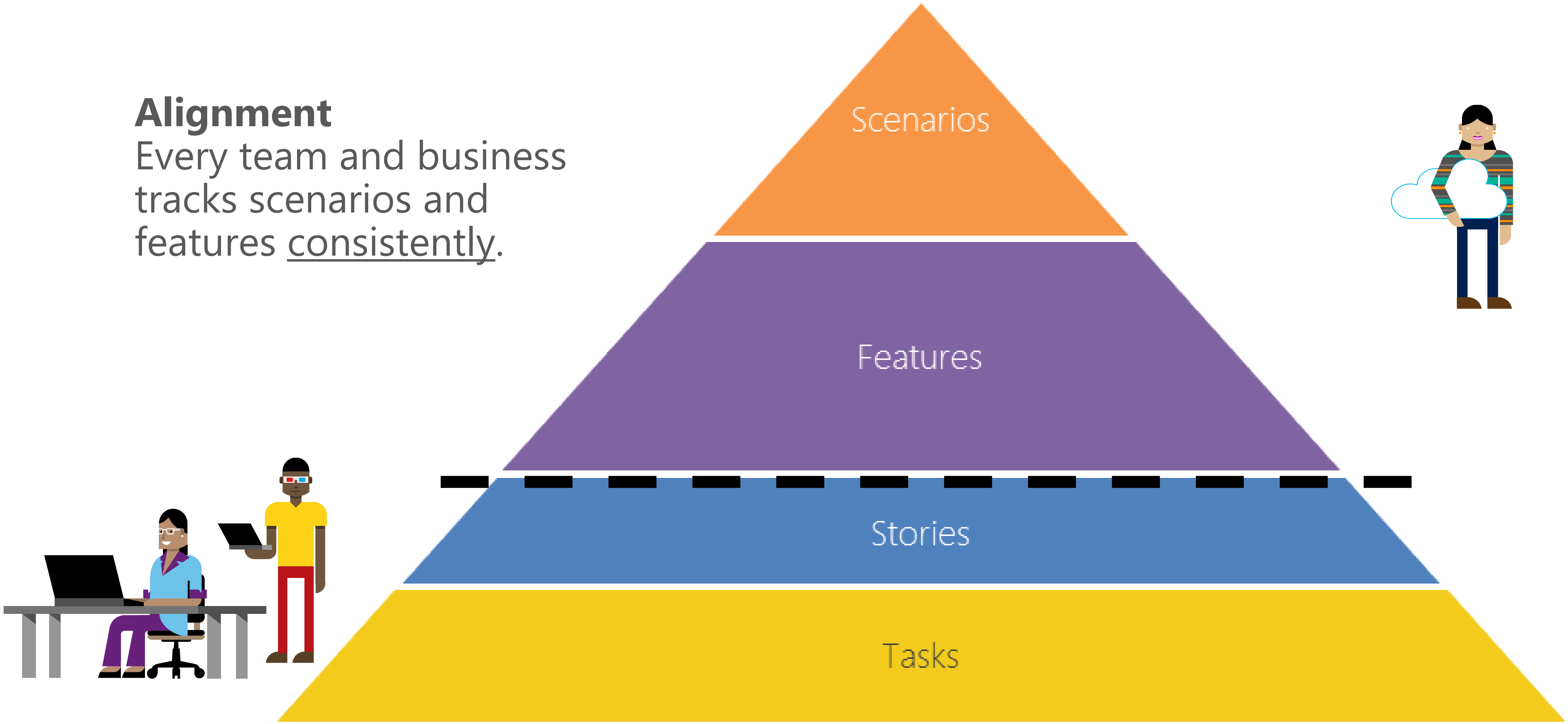


Now



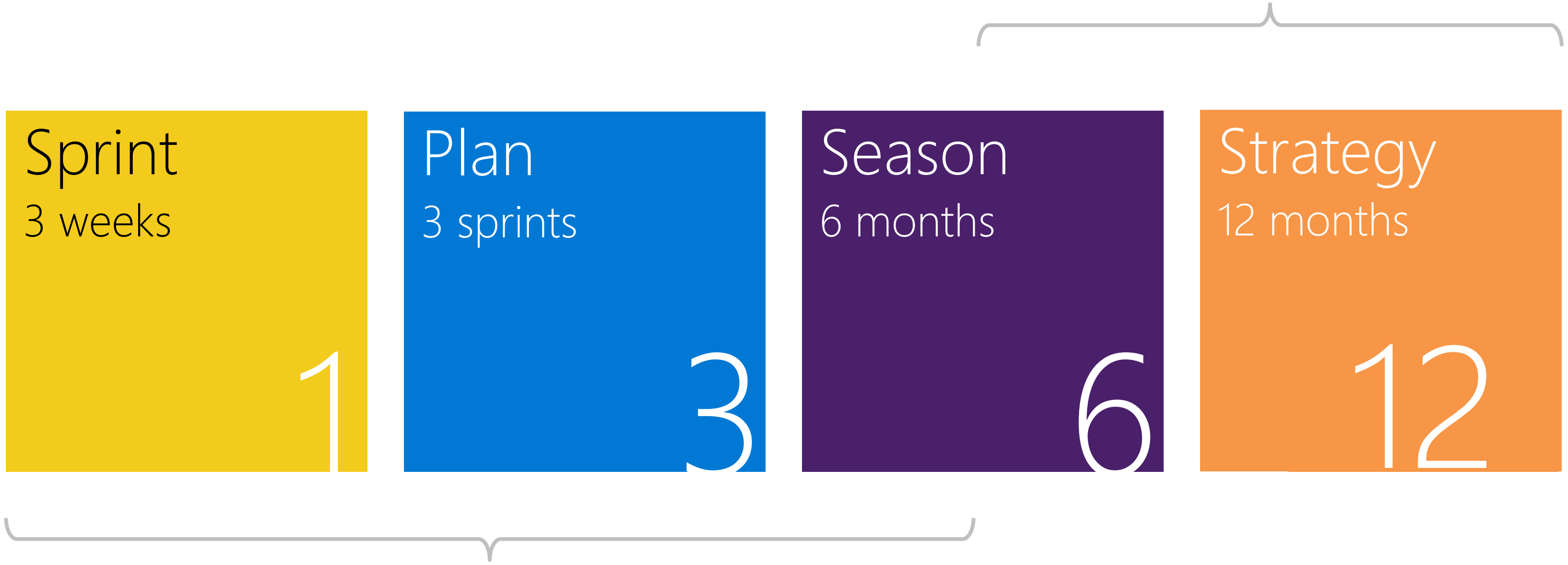
Planning: Staying Aligned

Alignment
Every team and business tracks scenarios and features consistently.



Team (Dev, Ops)
Team chooses how to manage stories and/or tasks

Planning



Leadership is responsible for driving the big picture

Sprint
3 weeks
1

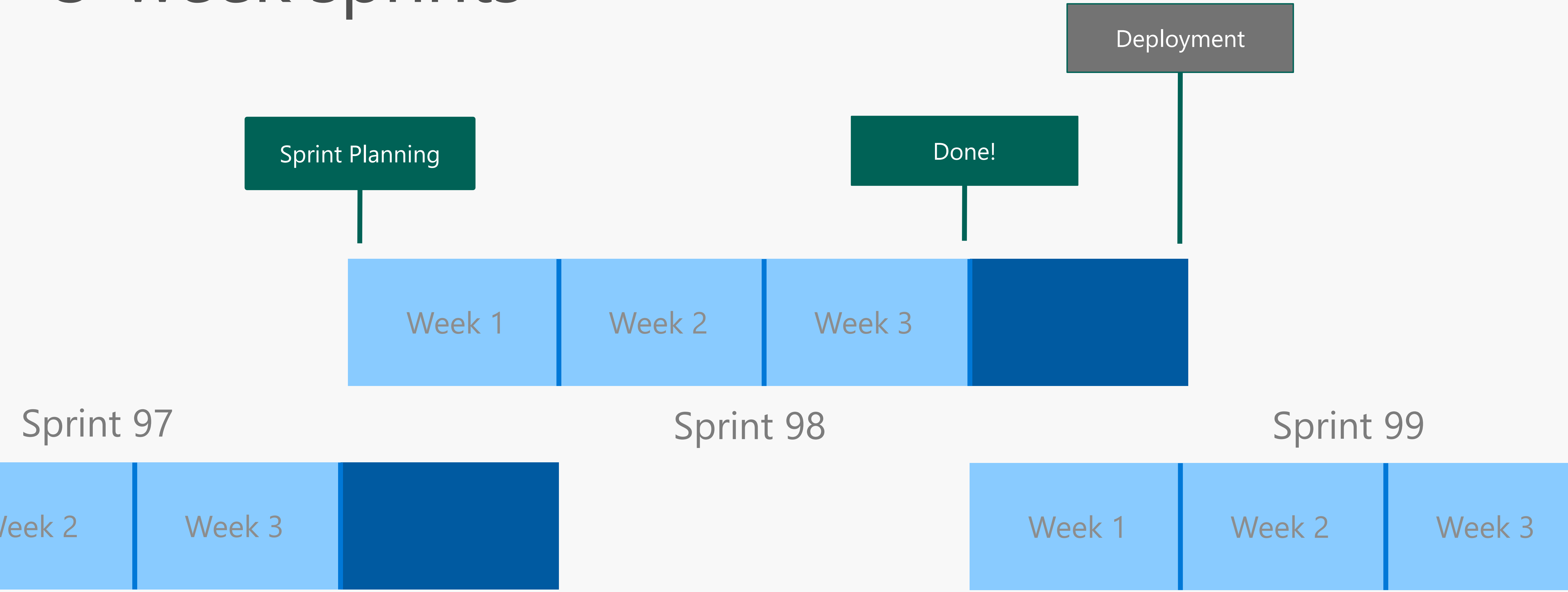
Plan
3 sprints
3

Season
6 months
6

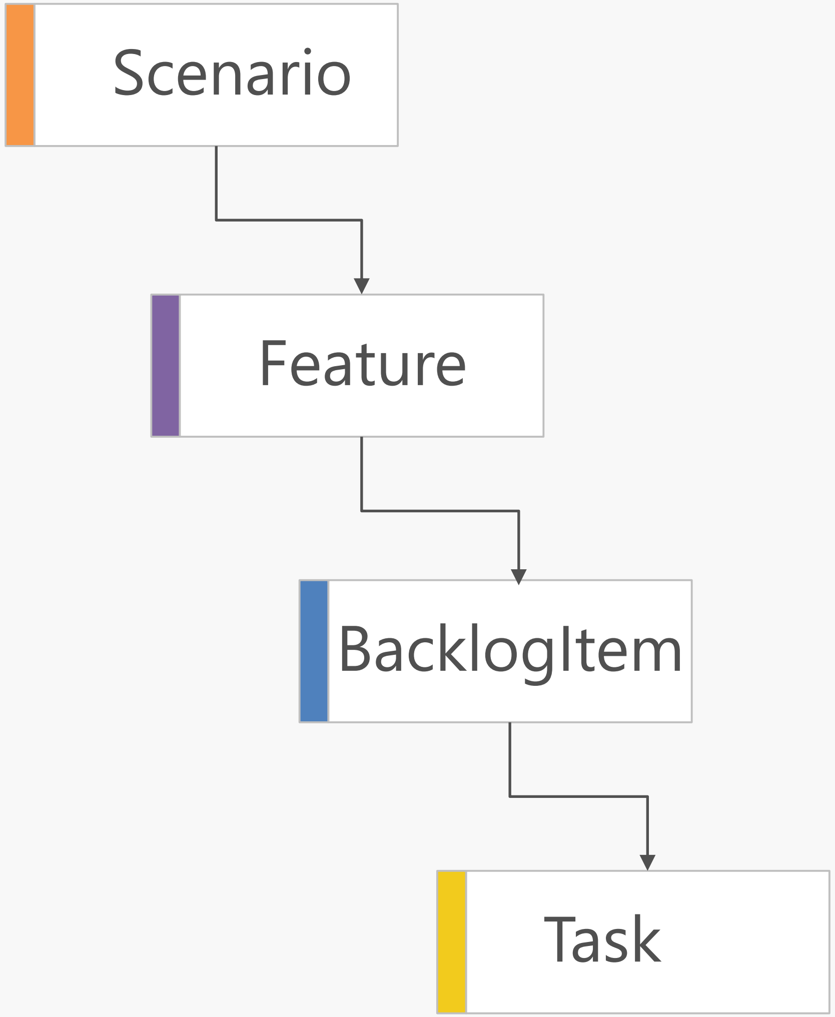
Strategy
12 months
12

Teams are responsible for the detail

3-week sprints



Azure DevOps Workitems



		Iteration Path
Scenario	Used to define your Scenarios	*Scenario1*
Feature	Used to define your Seasons	*Scenario1/Season1*
BacklogItem	Used to define your Sprint	*Scenario1/Season1/Sprint1*
Task	Used to define your work	*Scenario1/Season1/Sprint1*

ORG CHART

PROGRAM MANAGEMENT

DEVELOPMENT

TESTING

ORG CHART

PROGRAM MANAGEMENT

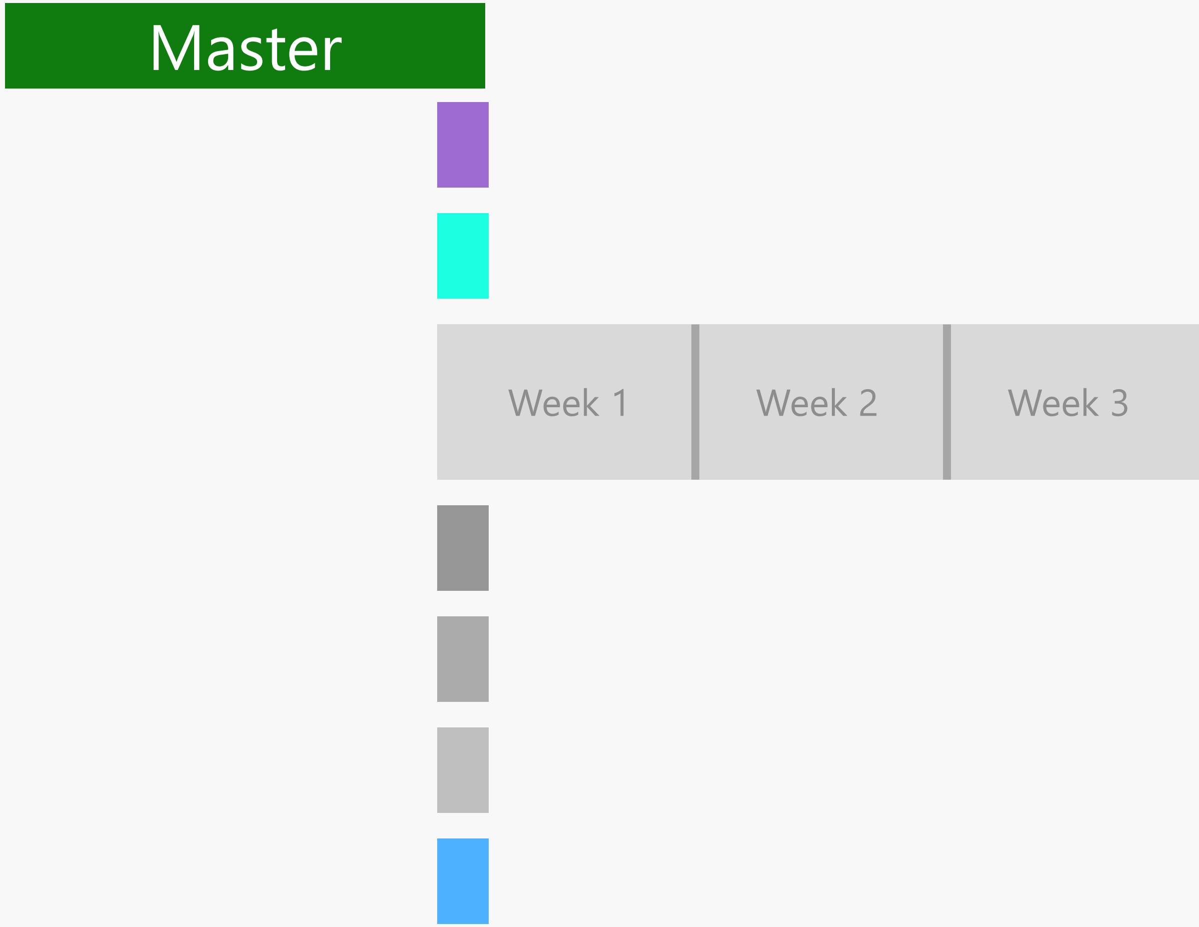
ENGINEERING

OPs

Azure Repos

Source Control

Everyone creates a branch...



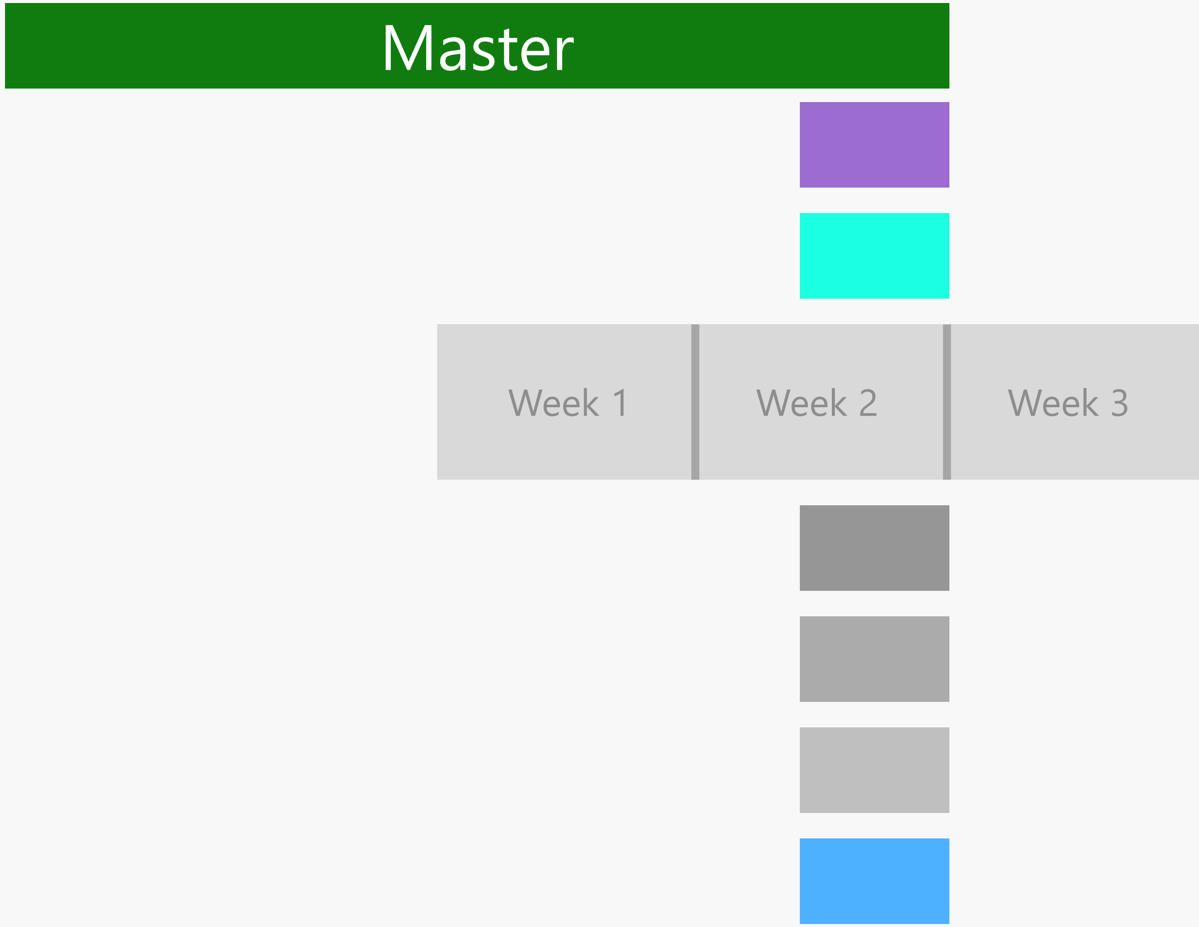
Master

Week 1

Week 2

Week 3

Writes a lot of code…



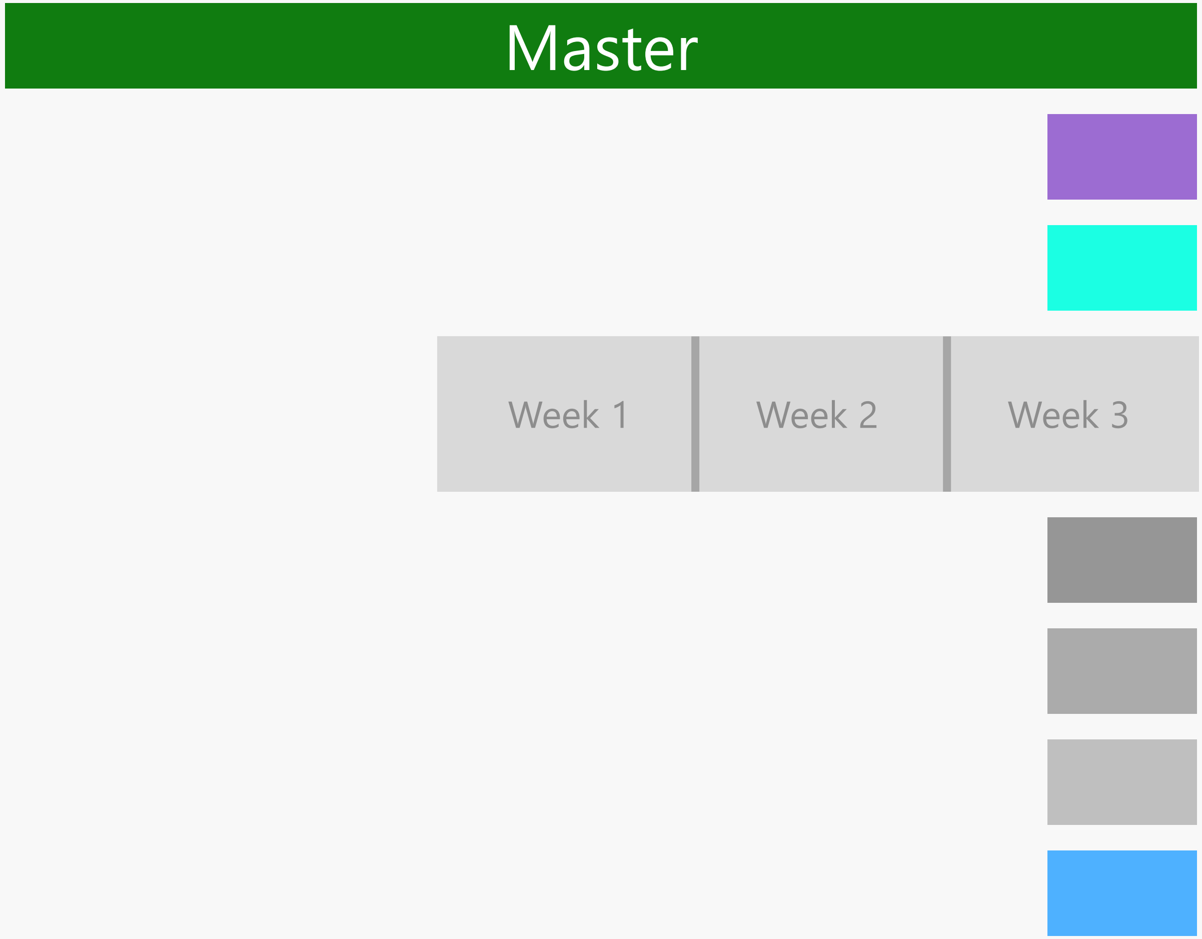
Master

Week 1

Week 2

Week 3

It needs to come together...



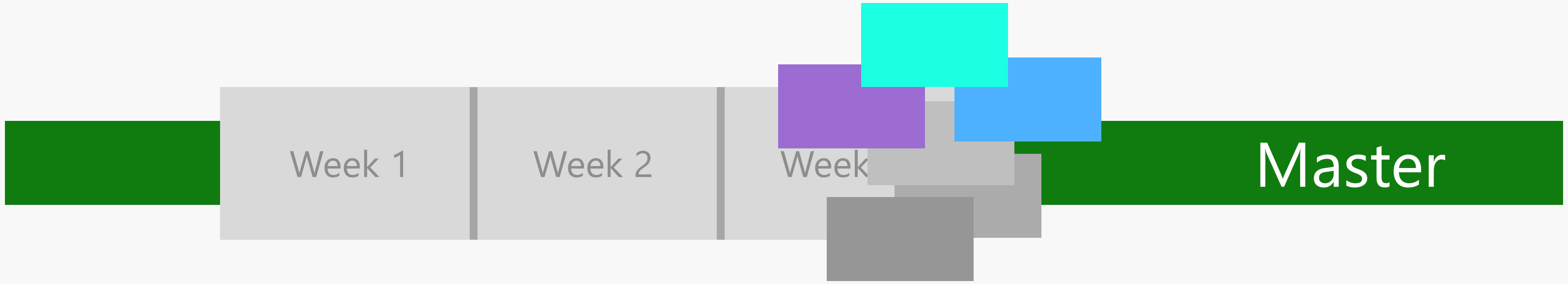
Master

Week 1

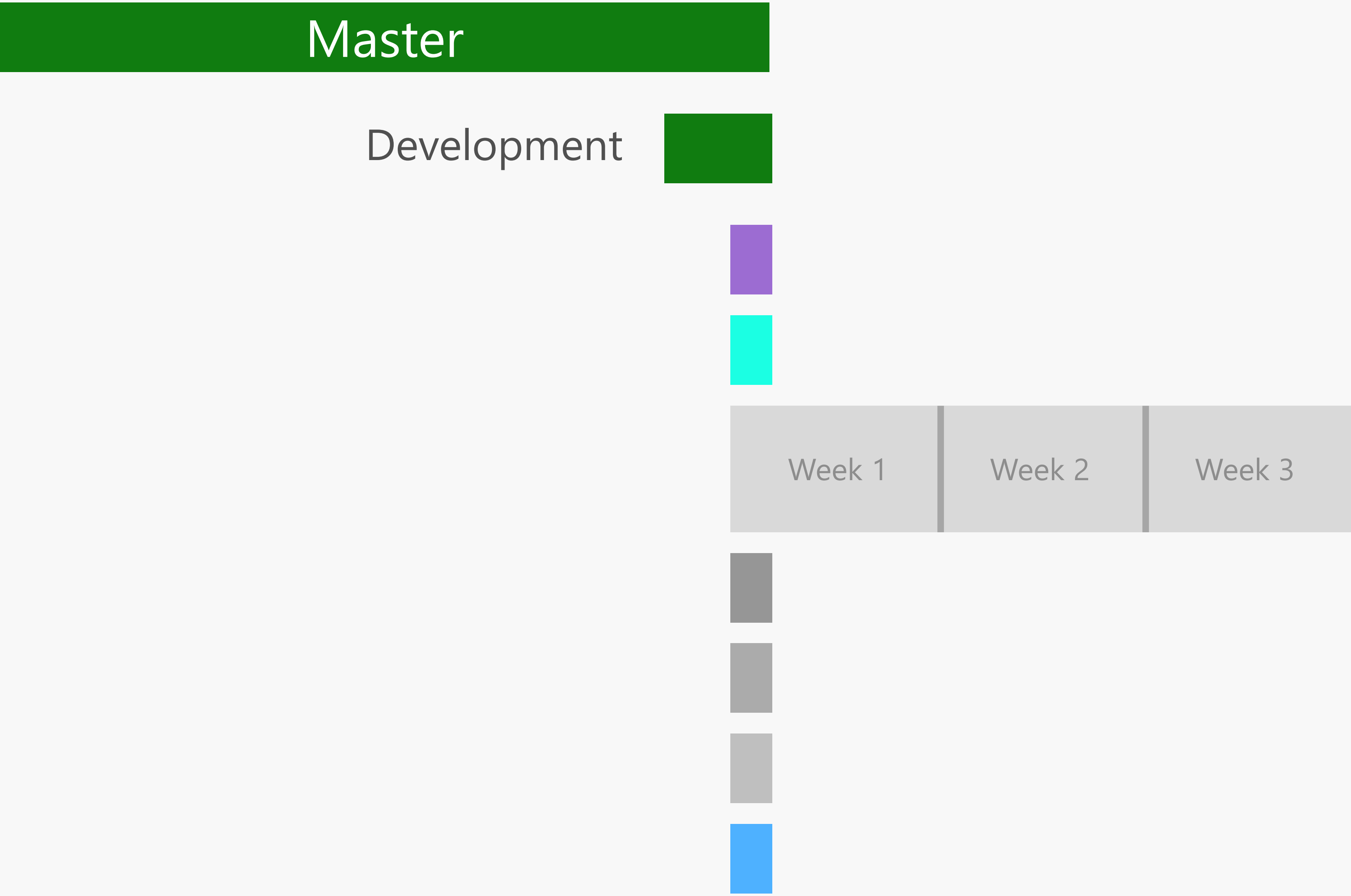
Week 2

Week 3

Merge Debt



Work with Development Branch…



Master

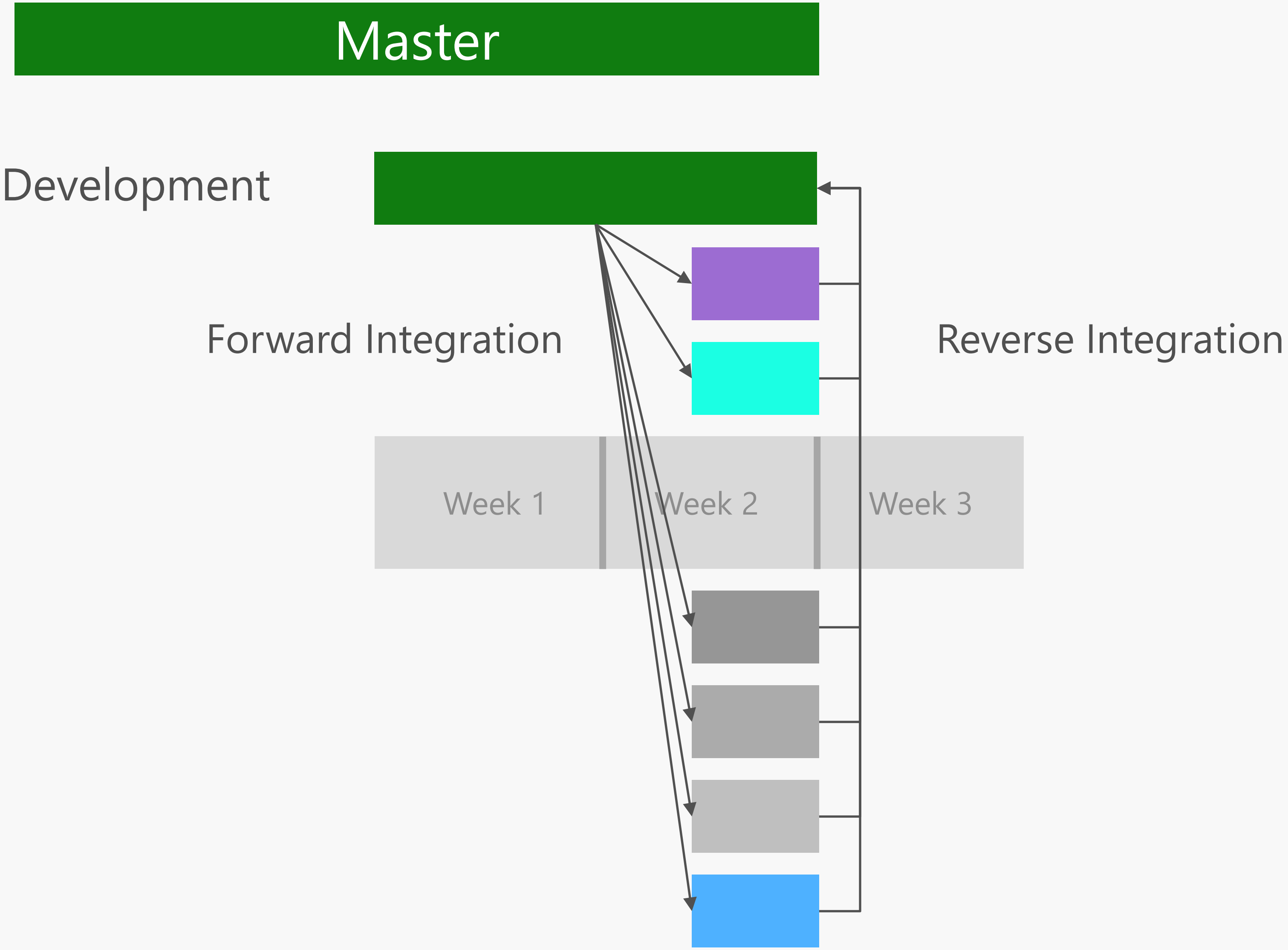
Development

Week 1

Week 2

Week 3

Integrate continuously…



Master

Development

Forward Integration

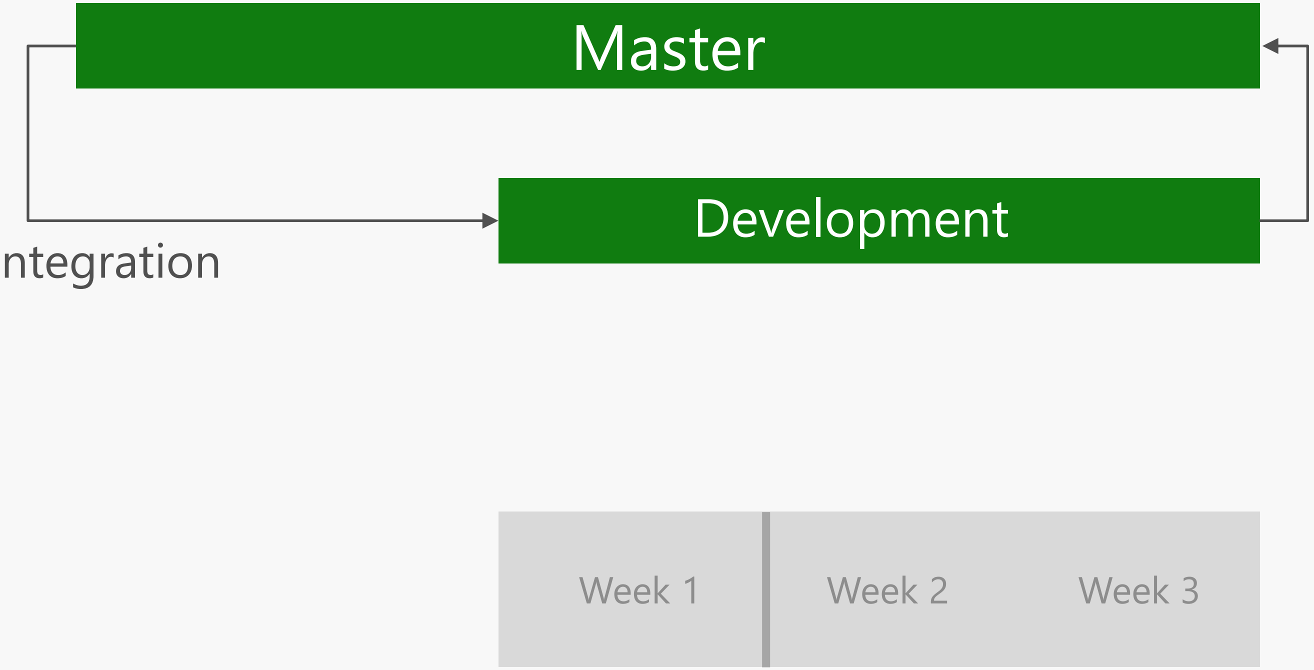
Reverse Integration

Week 1

Week 2

Week 3

Integrate Development Branch…



Master

Reverse Integration

Development

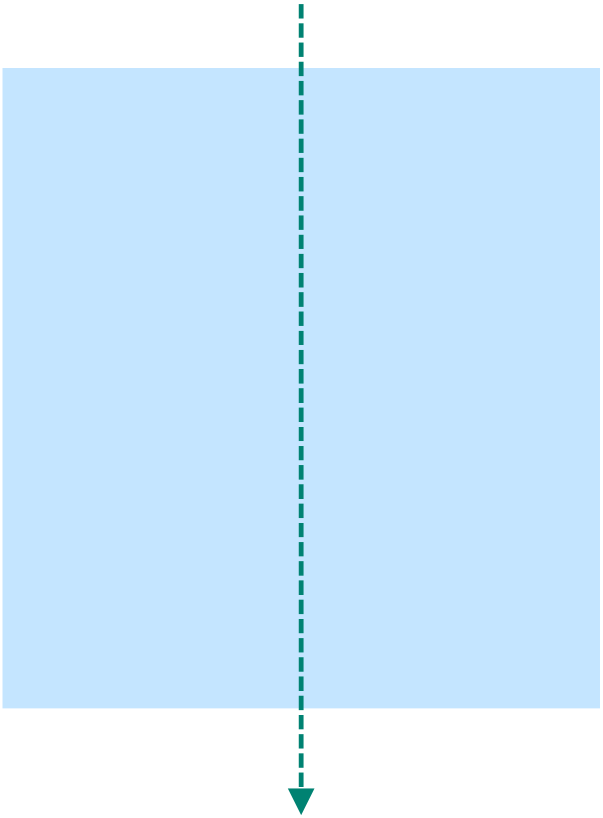
Forward Integration

Week 1

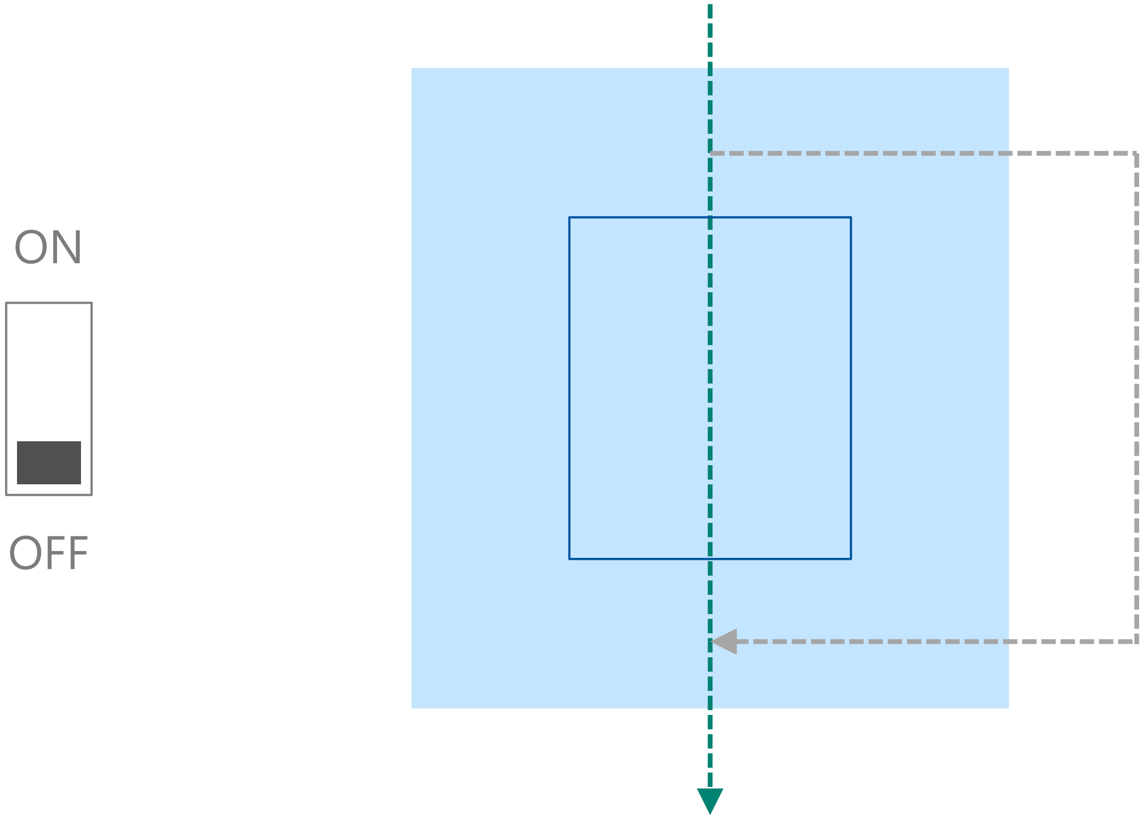
Week 2

Week 3

Feature Flags :Everything is going to production



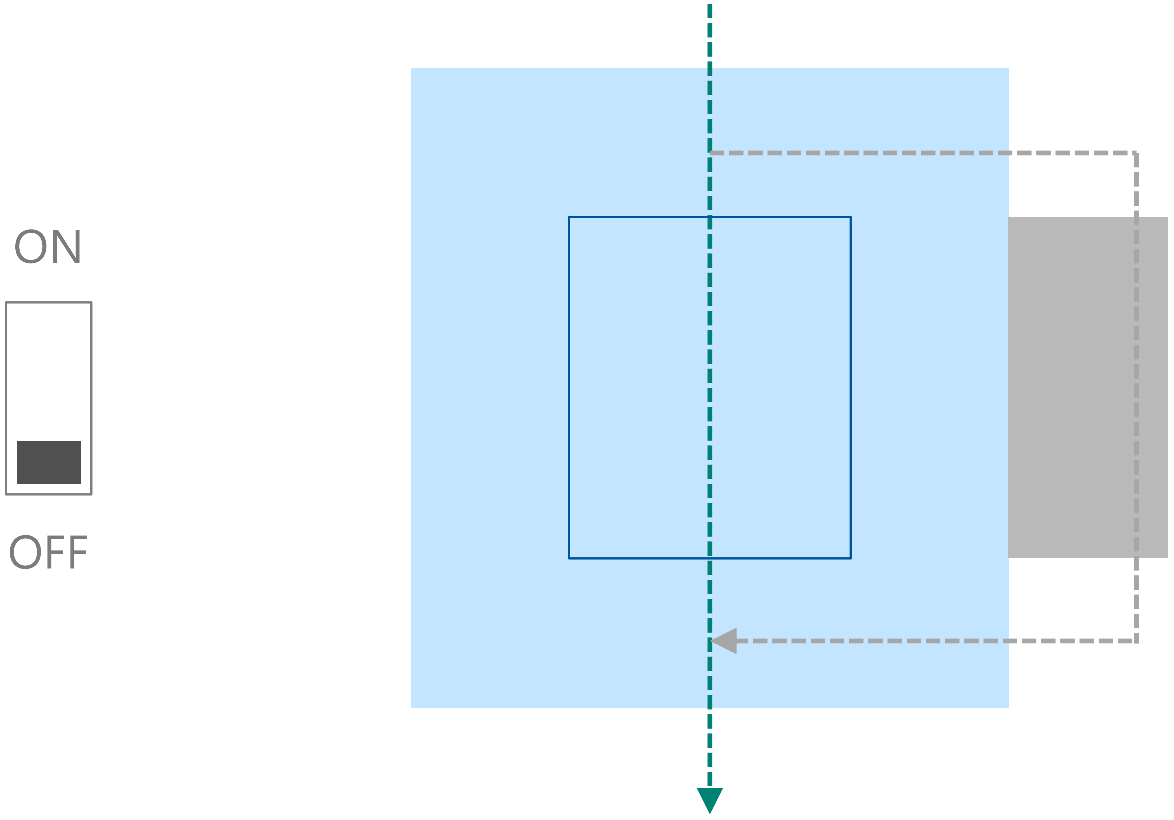
Feature Flags



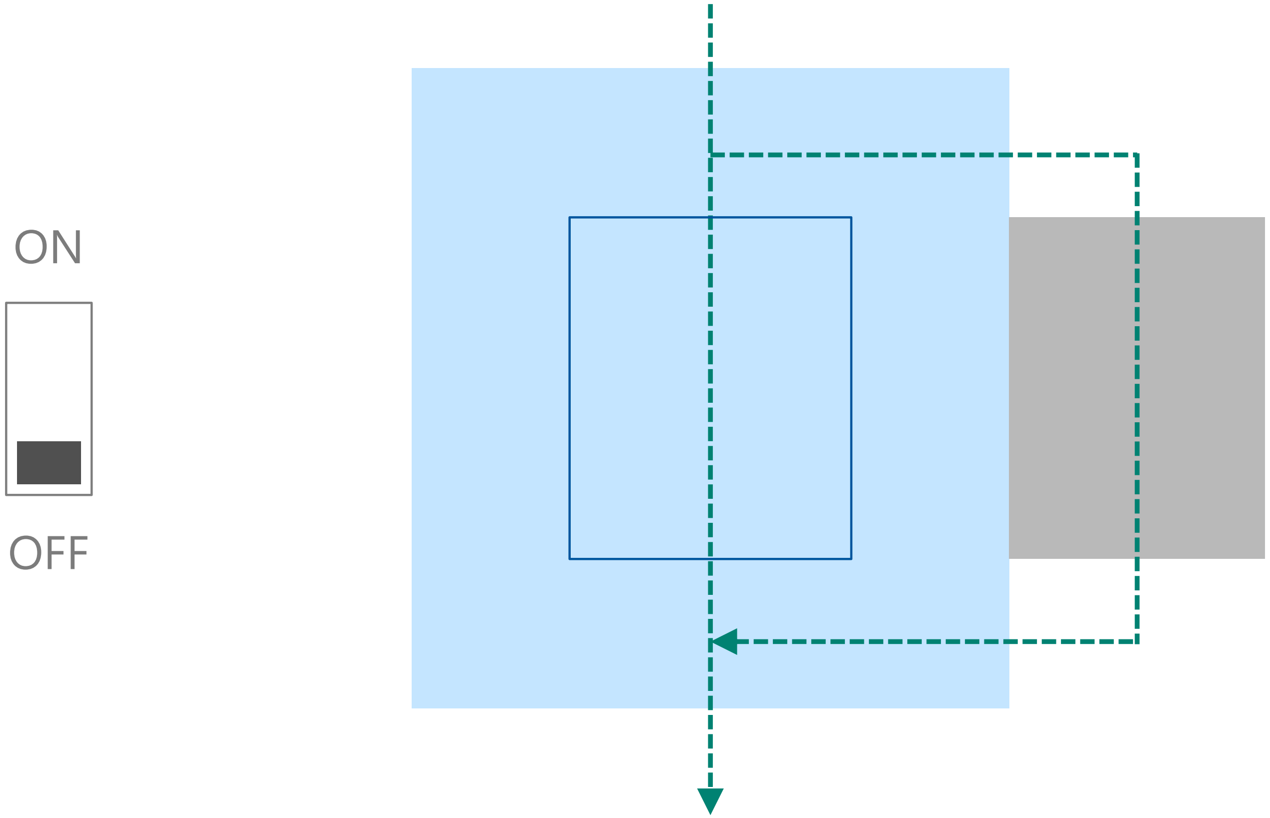
ON

OFF

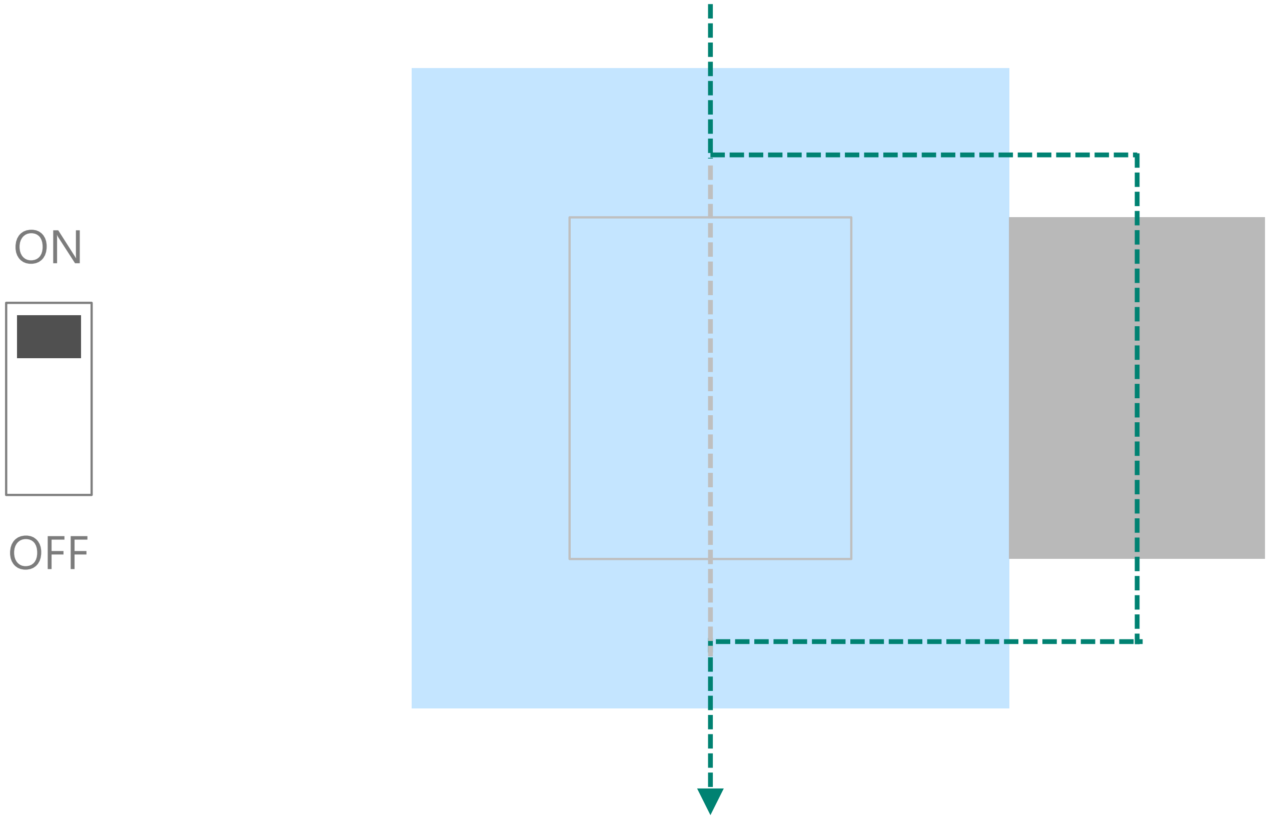
Feature Flags



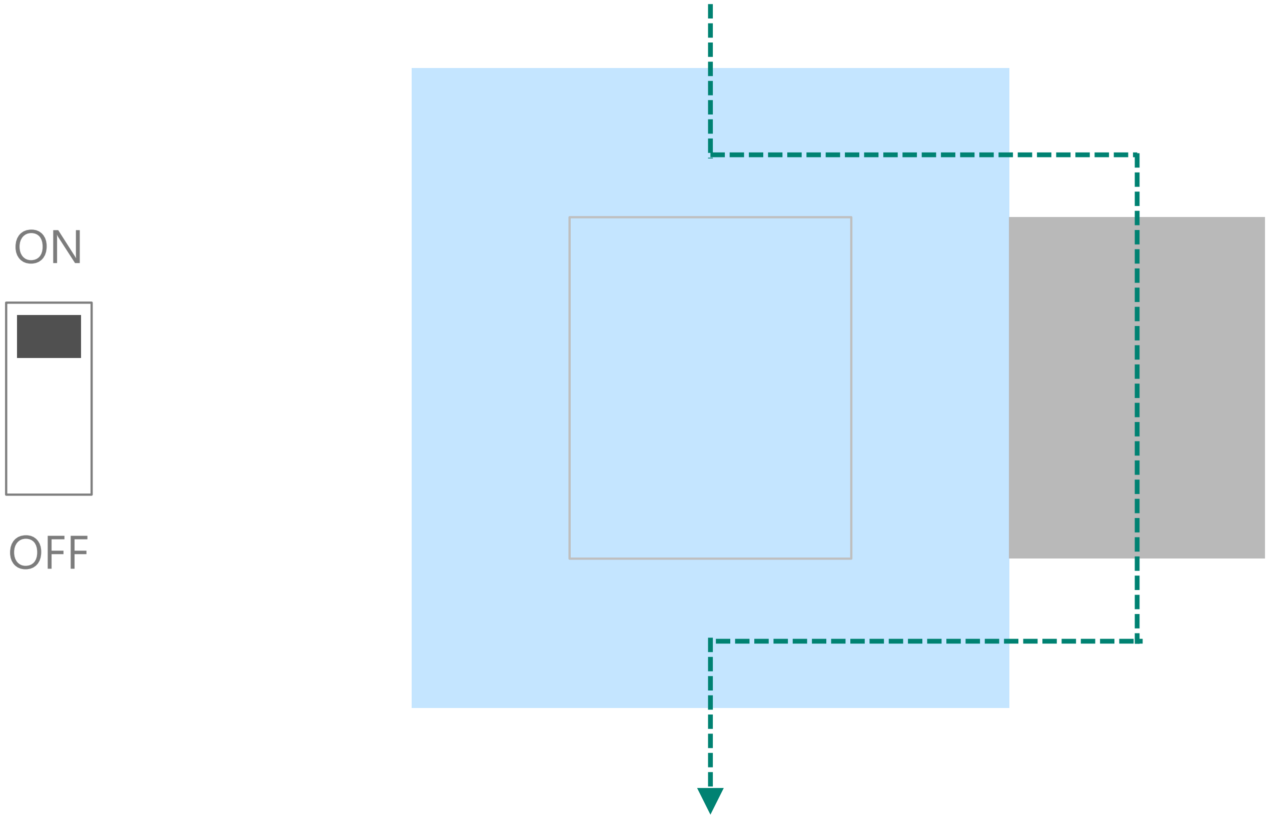
Feature Flags



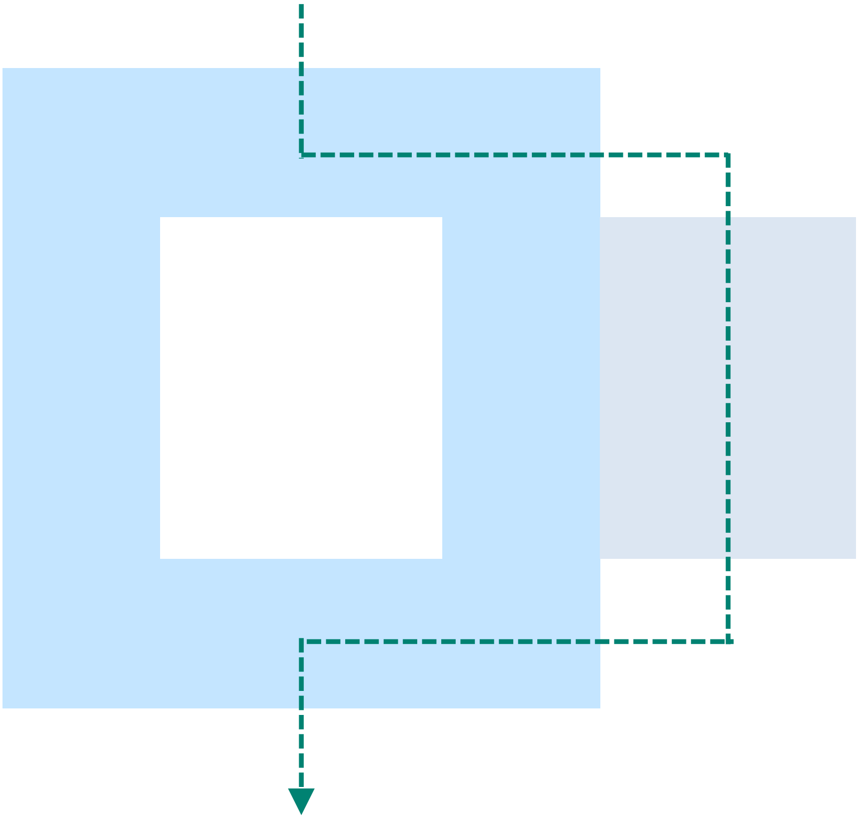
Feature Flags



Feature Flags

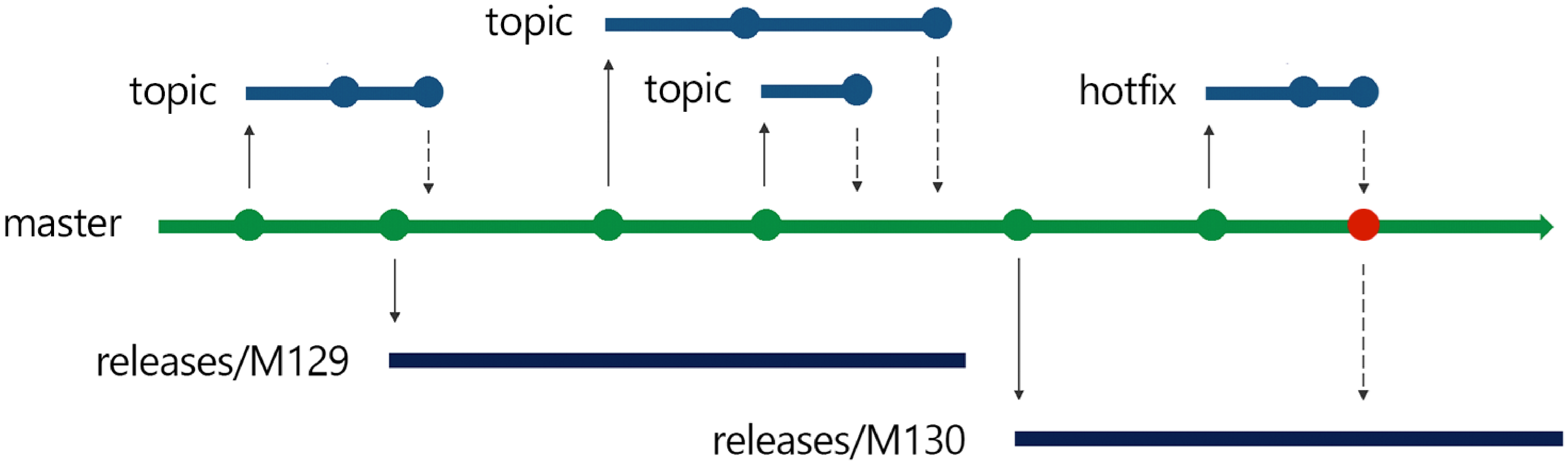


Feature Flags



Release Flow

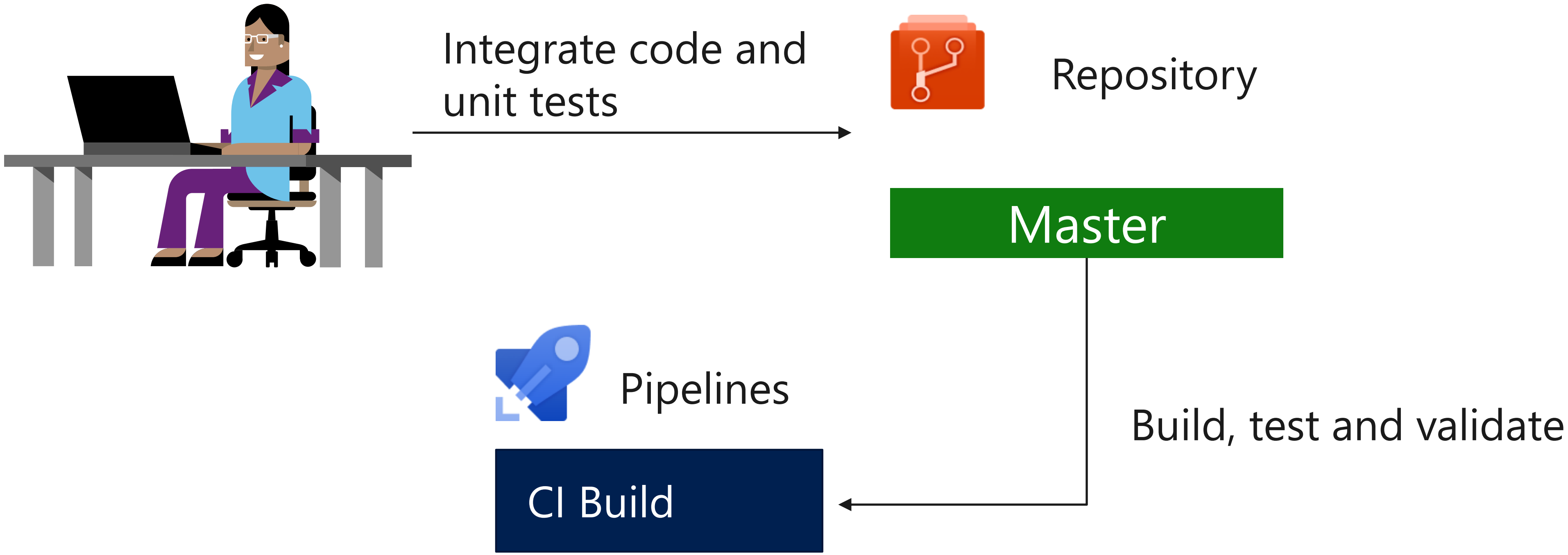
Using Trunk Based Development to avoid Merge Debt



Azure Pipelines

CI/CD

Continuous Integration (CI)



Integrate code and unit tests

Repository

Master

Build, test and validate

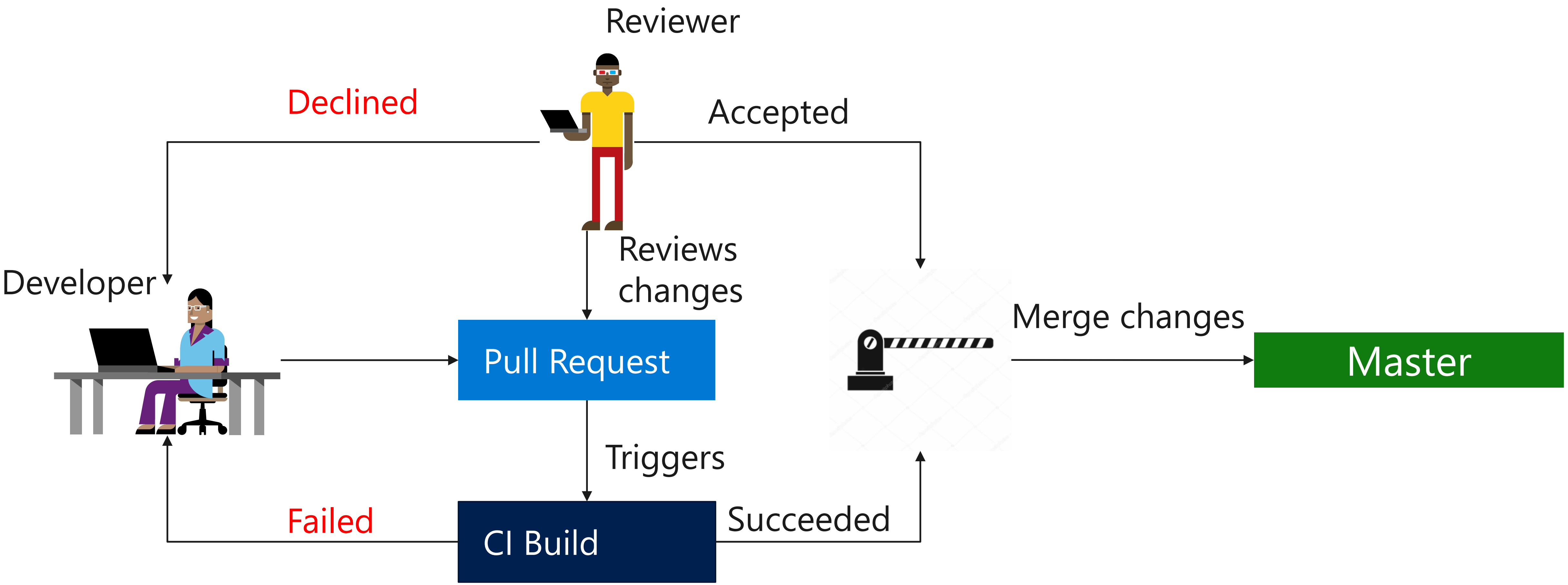
Pipelines

CI Build

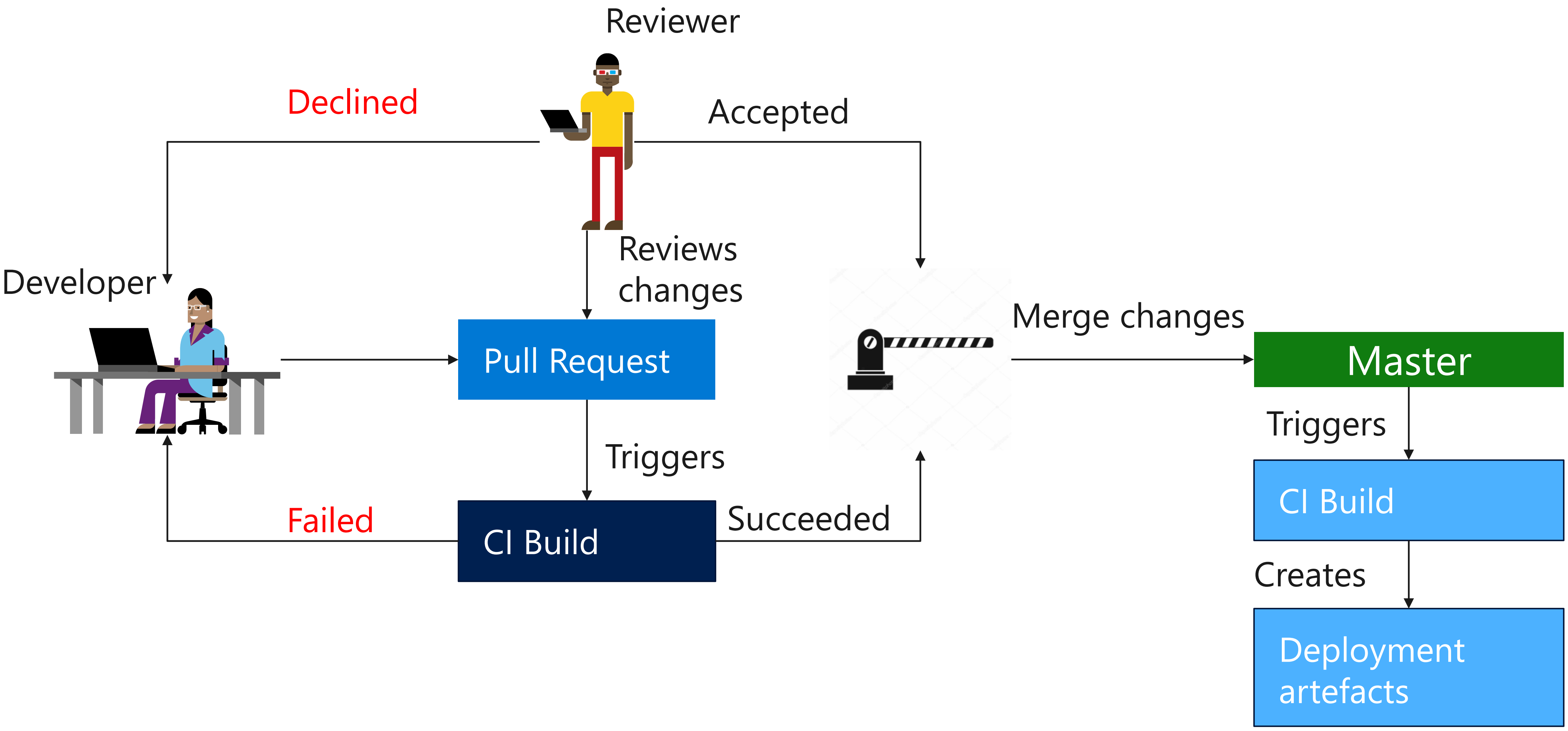
Pull Request

- CI keeps the master branch clean.
- Developer submits a “Pull Request” when a feature is completed and, on approval of the pull request, changes get merged to the master branch.
- A pull request is approved by reviewers
- A pull request is validated by a CI build

Pull request



CI Build to create deployment artefacts

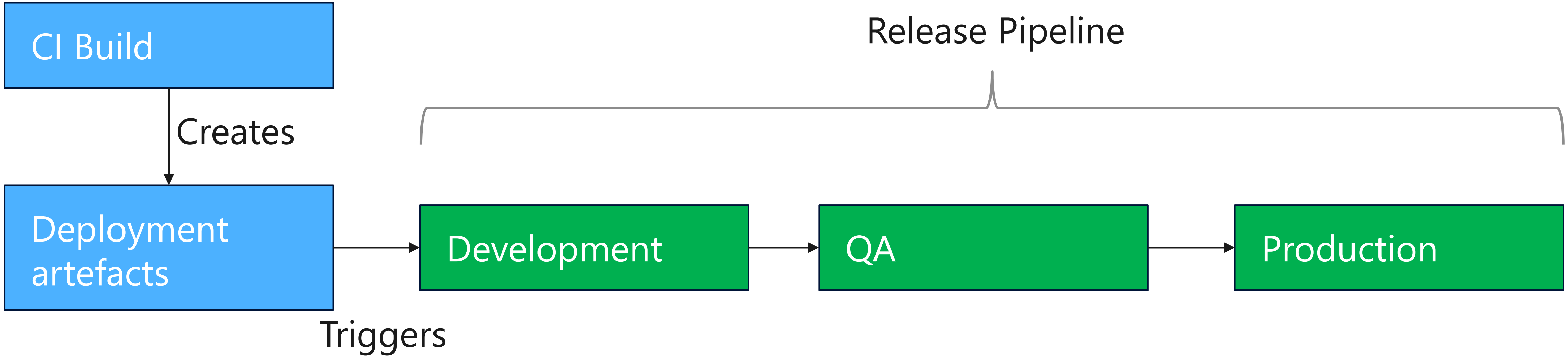


Continuous Deployment (CD)

Continuous Deployment is the process to build, test and deploy from a build to a production environment.

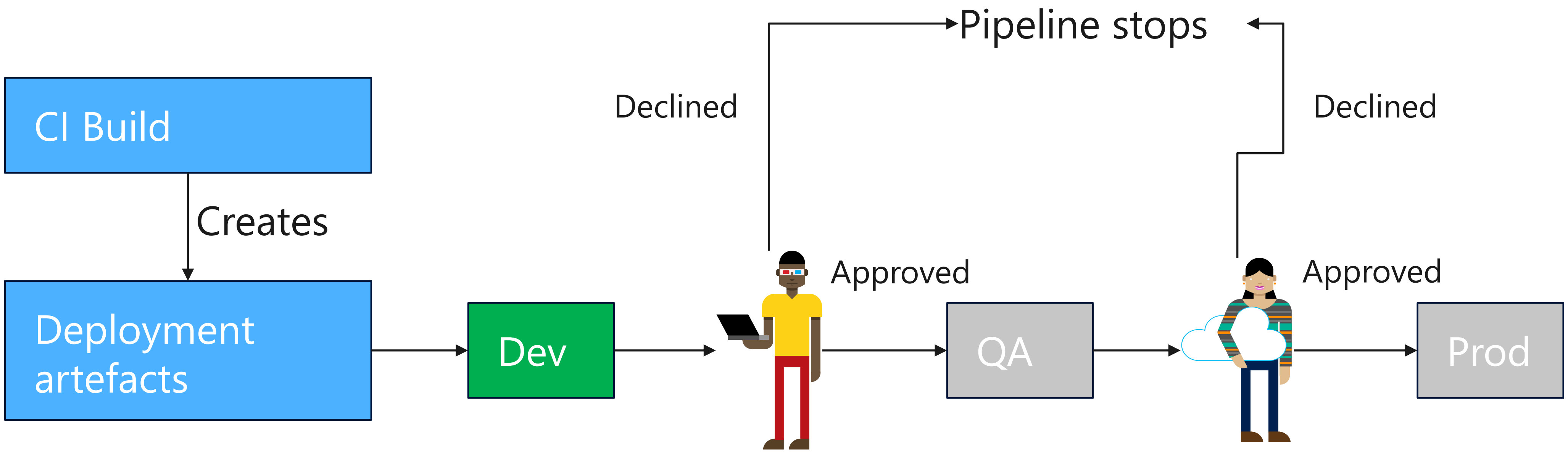
Multiple testing or staging environment create a *Release Pipeline* to automate the creation of infrastructure and deployment of applications.

Continuous Integration starts the CD process.

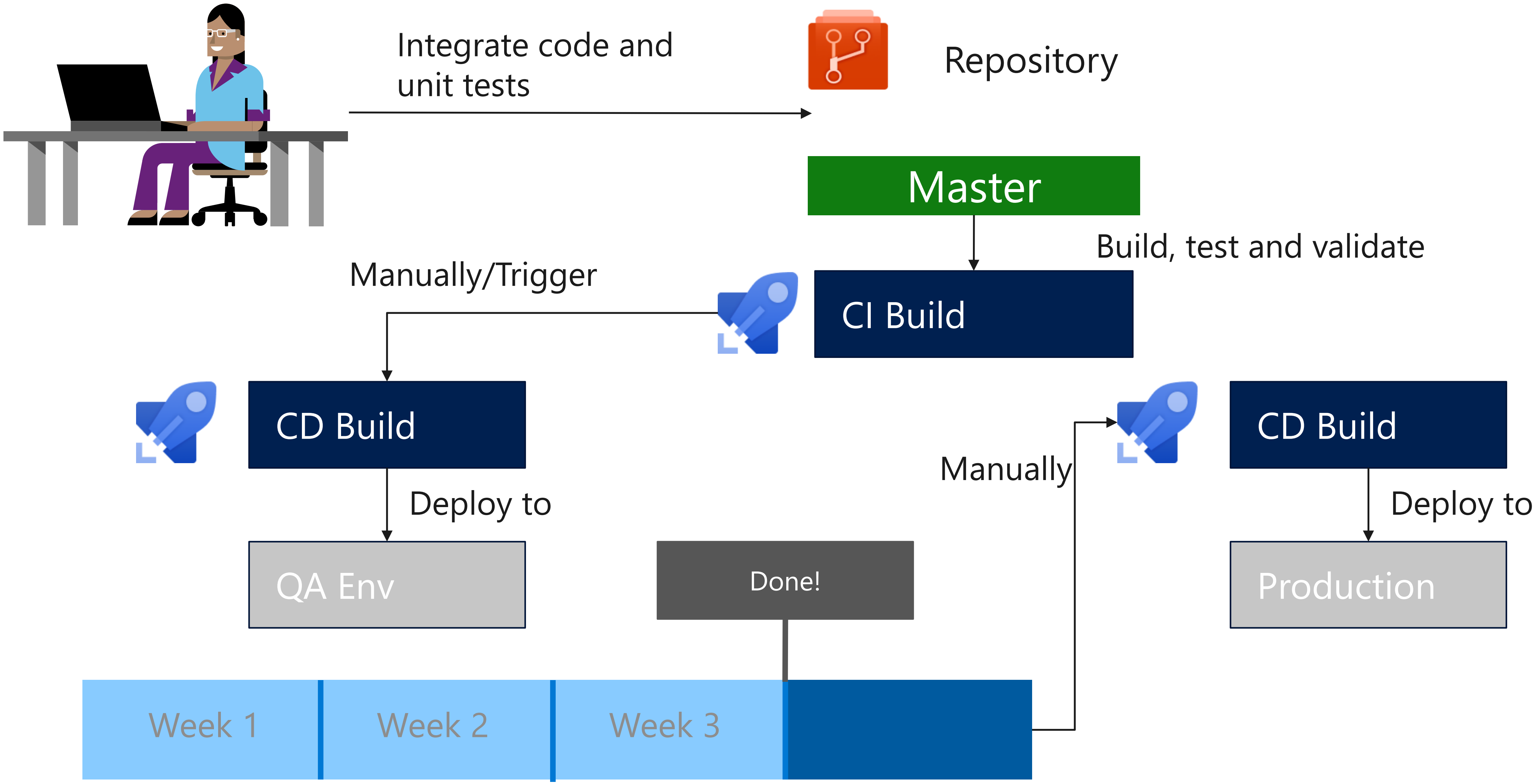


Approval Step

Before a deployment is rolled out to another environment in the pipeline an approval step can be used in which a decision maker signs off on the changes electronically.

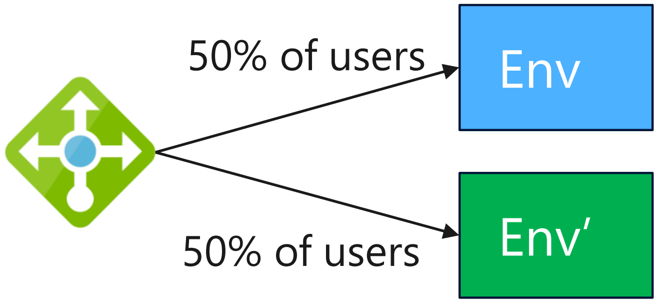


Continuous Delivery

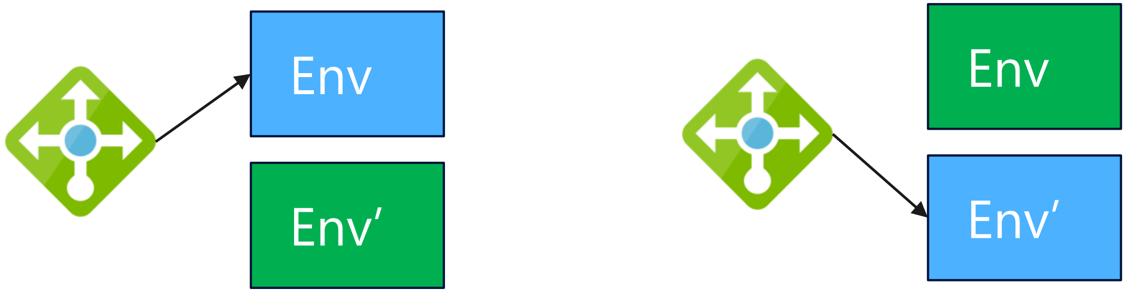


CD Strategies

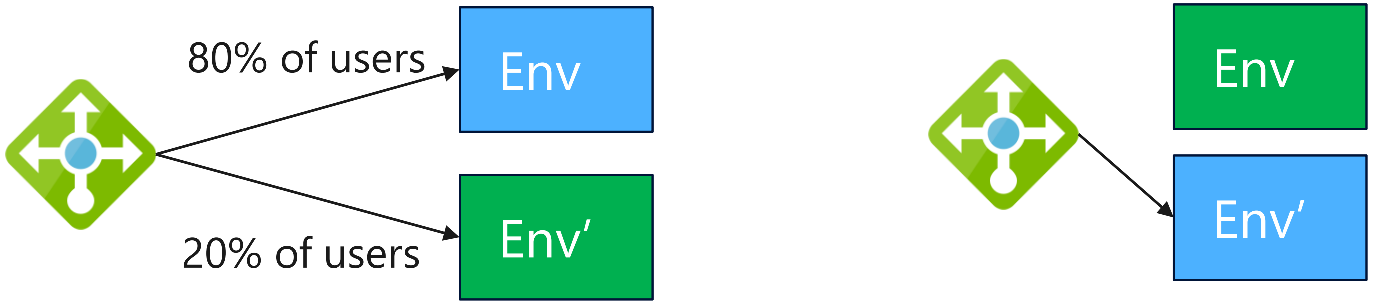
- Test A/B



- Blue/Green



- Canary Release



DevOps on Azure

DevOps on Azure

First Party Tools

End to end solutions, enterprise grade

- Azure DevOps
- Team Foundation Server
- Visual Studio App Center

Third Party Integrations

Integrations with industry leading tools

HashiCorp Terraform

Jenkins

CHEF

ANSIBLE

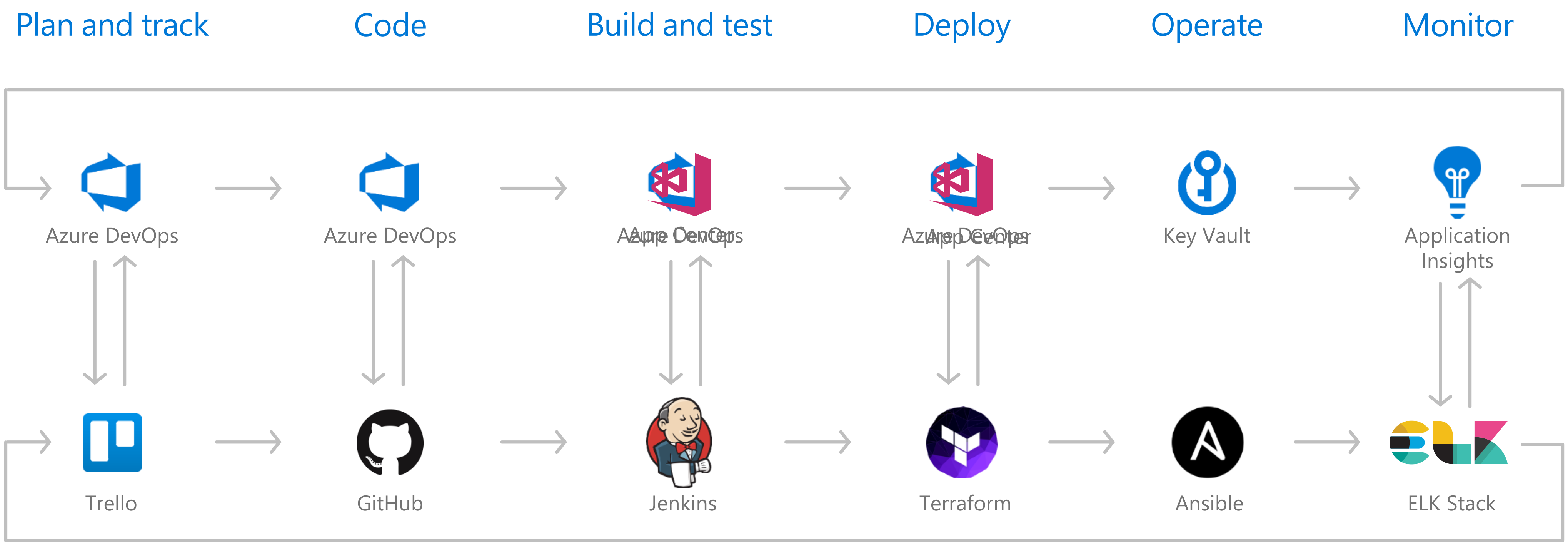
puppet

Azure Capabilities

Built into Azure, out-of-the-box capabilities

- ARM Templates
- Azure Monitor
- Application Insights
- Log Analytics
- Azure Key Vault

Azure DevOps framework



Plan and track

Code

Build and test

Deploy

Operate

Monitor

Azure DevOps

Azure DevOps

Azure DevOps
App Center

Azure DevOps
App Center

Key Vault

Application Insights

Trello

GitHub

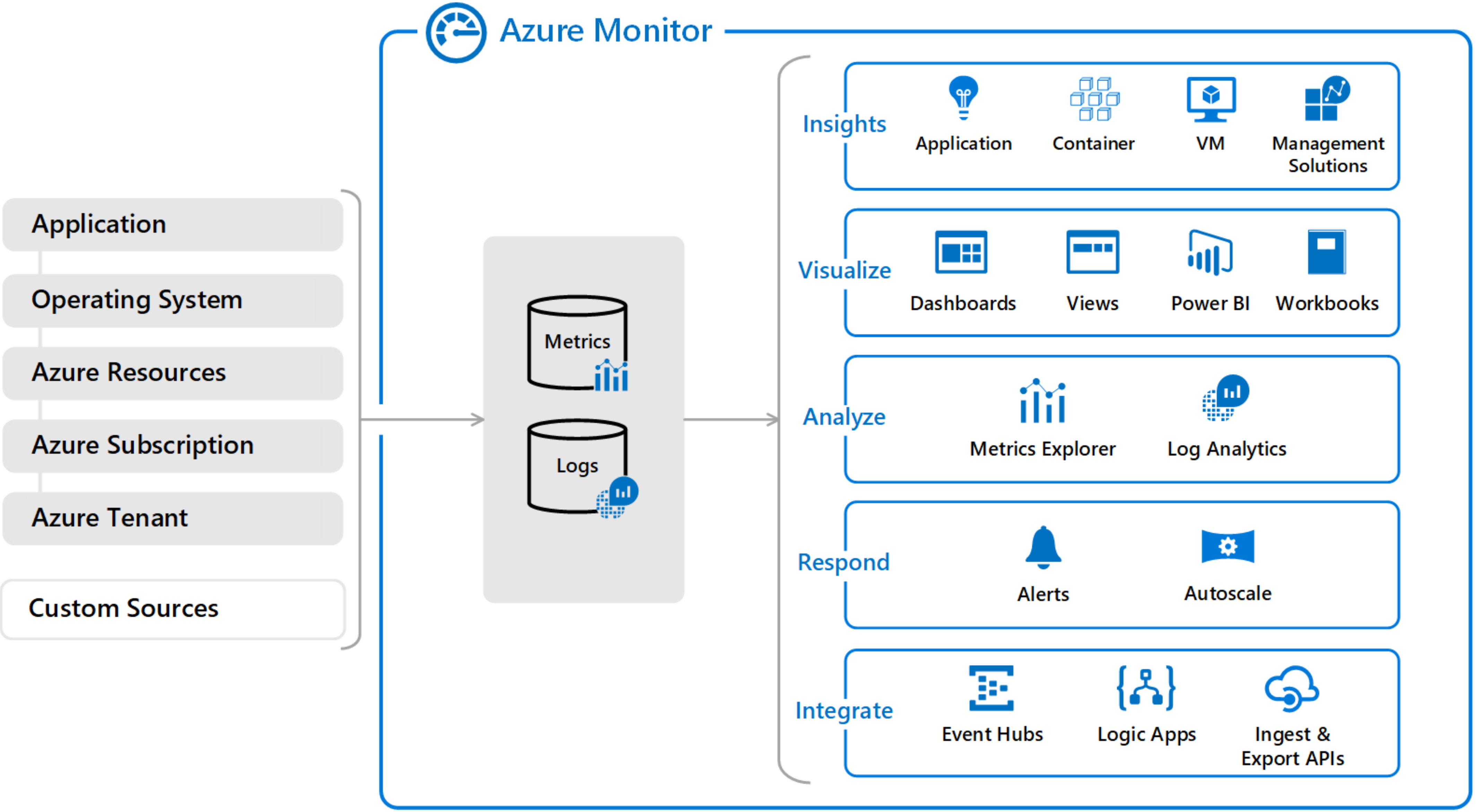
Jenkins

Terraform

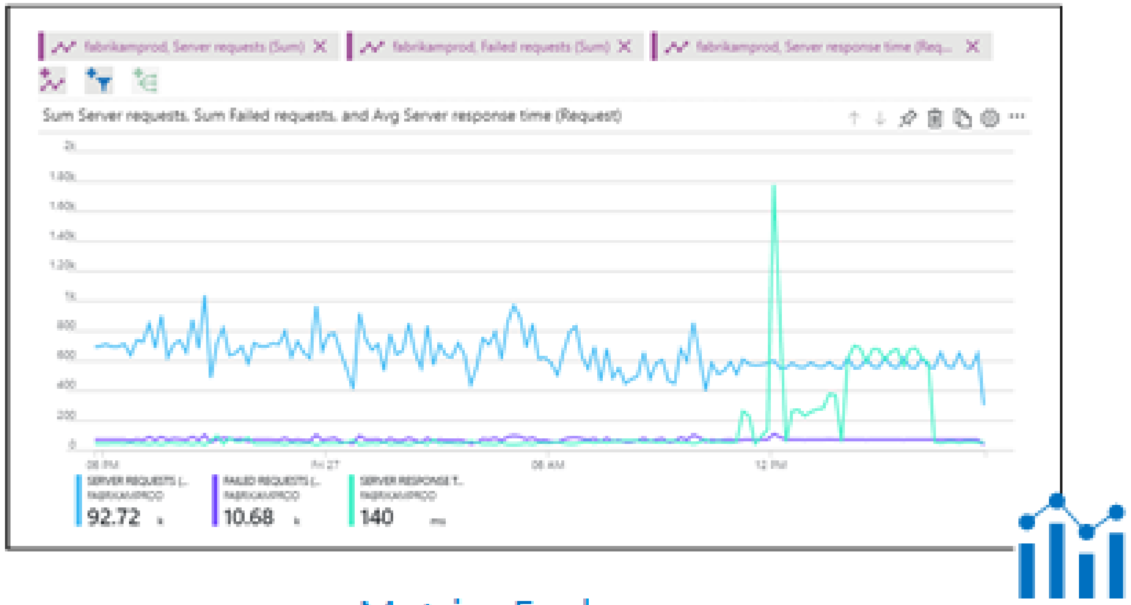
Ansible

ELK Stack

Azure Monitor



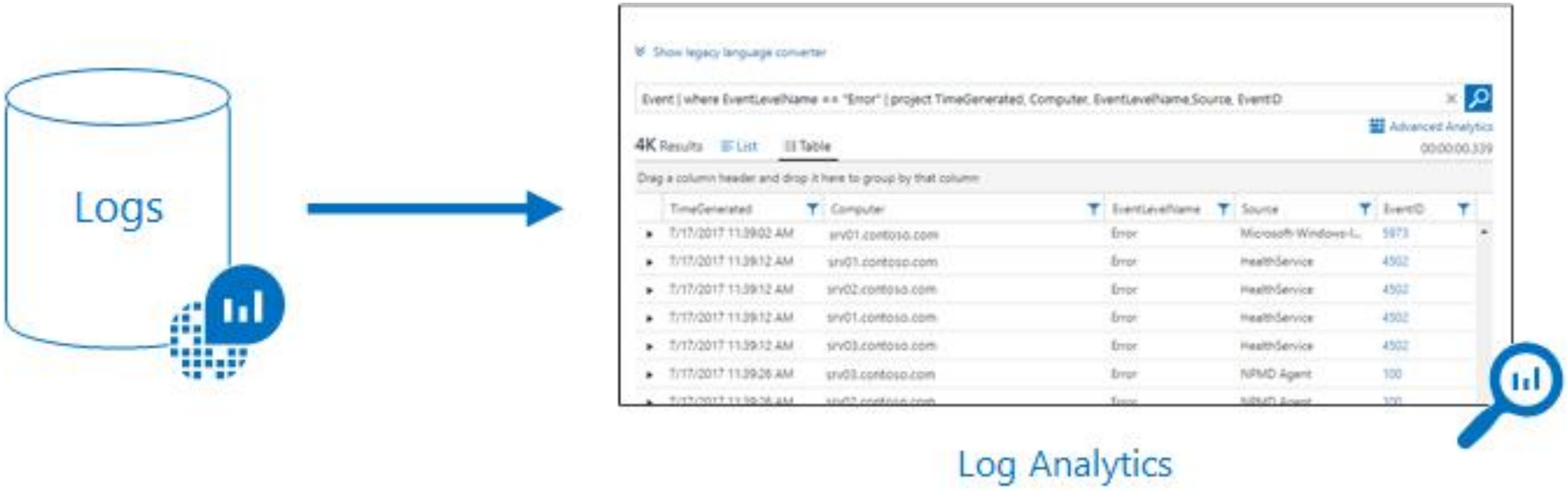
Metrics


fabrikamprod, Server requests (Sum)
fabrikamprod, Failed requests (Sum)
fabrikamprod, Server response time (Req...
Sum Server requests, Sum Failed requests, and Avg Server response time (Request)
SERVER REQUESTS (...
FABRIKAMPROD
92.72 k
FAILED REQUESTS (...
FABRIKAMPROD
10.68 k
SERVER RESPONSE T...
FABRIKAMPROD
140 ms

Metrics Explorer

Metrics are numerical values that describe some aspect of a system at a particular point in time. They are lightweight and capable of supporting near real-time scenarios.

Logs



```
Event | where EventLevelName == "Error" | project TimeGenerated, Computer, EventLevelName,Source, EventID
```

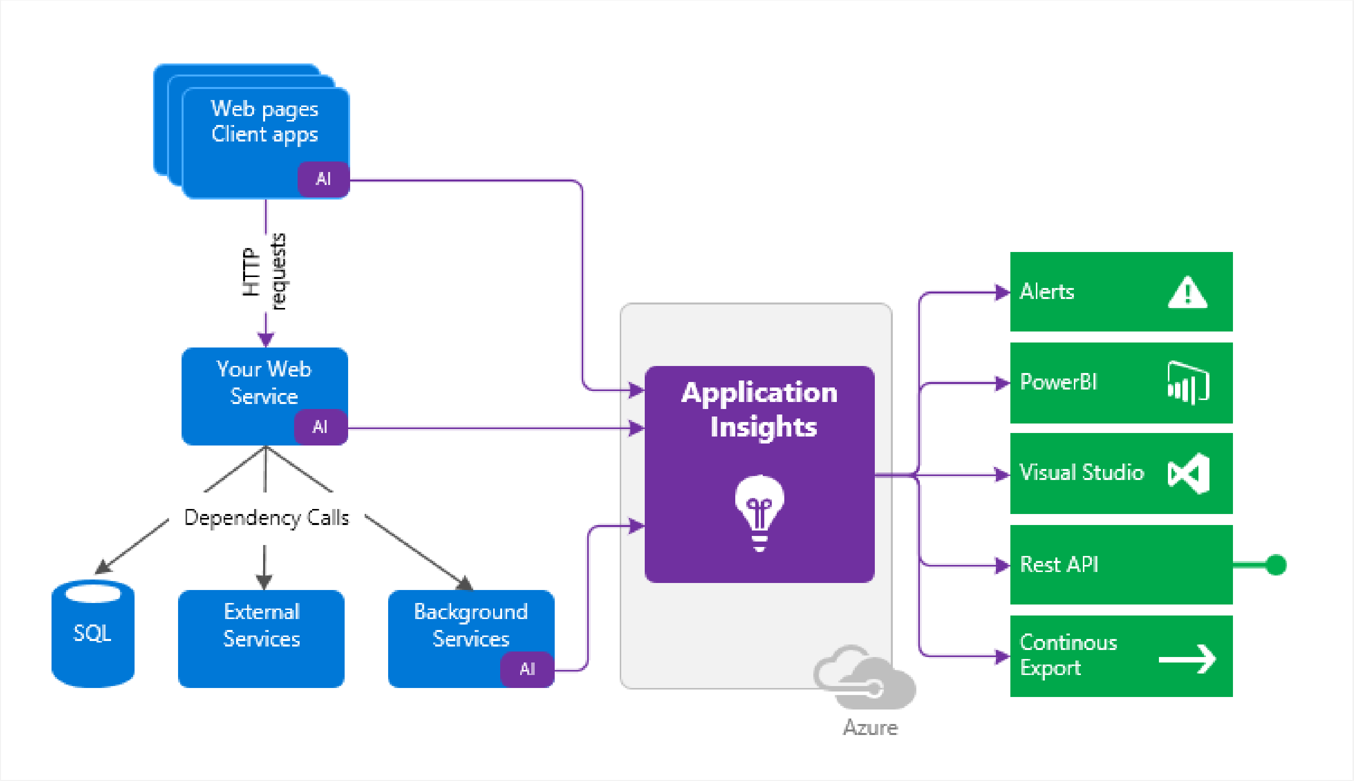
00:00:00.339

Drag a column header and drop it here to group by that column

TimeGenerated	Computer	EventLevelName	Source	EventID
7/17/2017 11:39:02 AM	srv01.contoso.com	Error	Microsoft-Windows-I...	5973
7/17/2017 11:39:12 AM	srv01.contoso.com	Error	HealthService	4502
7/17/2017 11:39:12 AM	srv02.contoso.com	Error	HealthService	4502
7/17/2017 11:39:12 AM	srv01.contoso.com	Error	HealthService	4502
7/17/2017 11:39:12 AM	srv03.contoso.com	Error	HealthService	4502
7/17/2017 11:39:26 AM	srv03.contoso.com	Error	NPMD Agent	100
7/17/2017 11:39:26 AM	srv02.contoso.com	Error	NPMD Agent	100

Logs contain different kinds of data organized into records with different sets of properties for each type.

Application Insights



What does Application Insights monitor?

Request rates, response times, and failure rates - Find out which pages are most popular, at what times of day, and where your users are. See which pages perform best. If your response times and failure rates go high when there are more requests, then perhaps you have a resourcing problem.

Dependency rates, response times, and failure rates - Find out whether external services are slowing you down.

Exceptions - Analyze the aggregated statistics, or pick specific instances and drill into the stack trace and related requests. Both server and browser exceptions are reported.

Page views and load performance - reported by your users' browsers.

AJAX calls from web pages - rates, response times, and failure rates.

User and session counts.

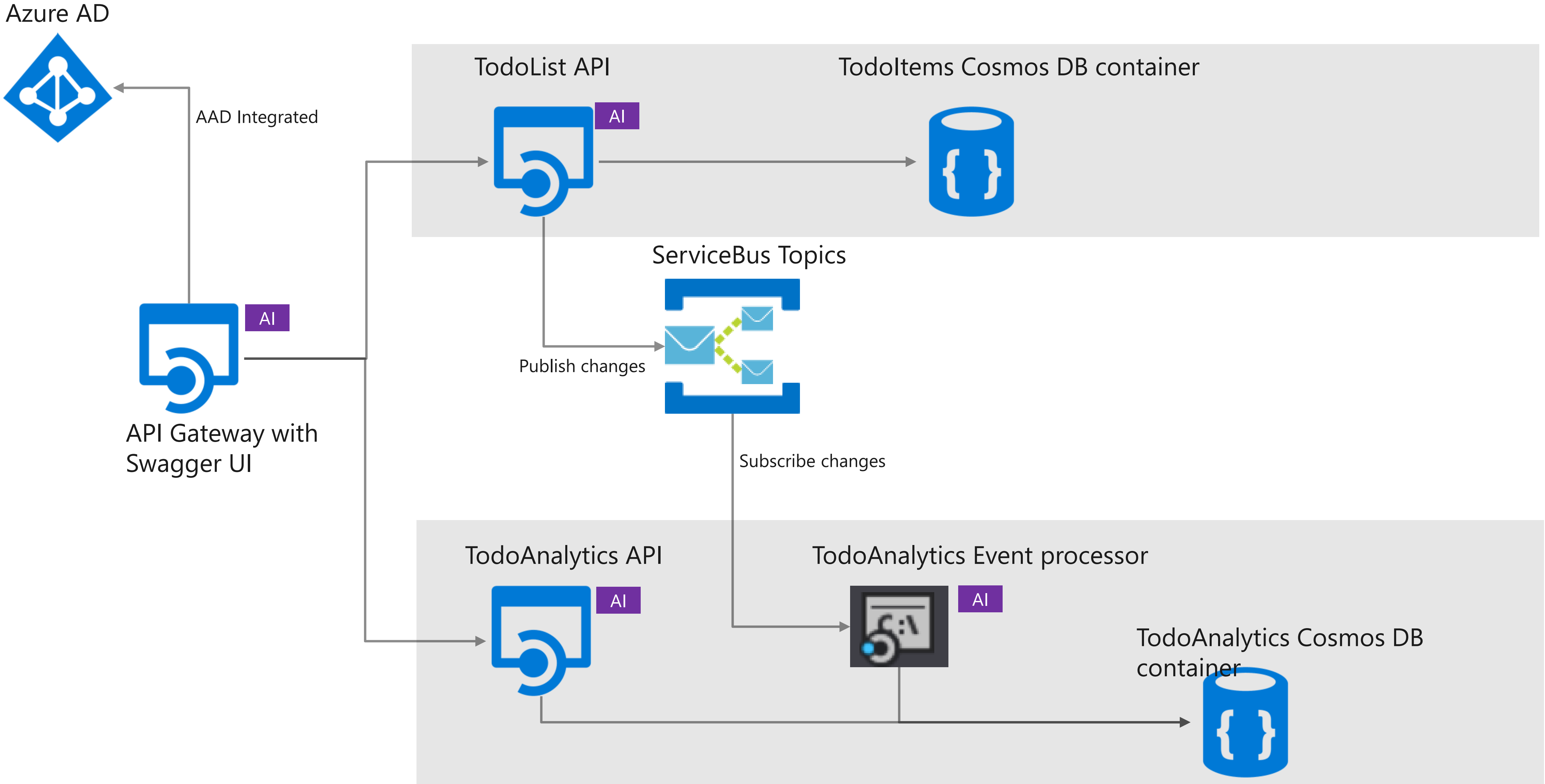
Performance counters from your Windows or Linux server machines, such as CPU, memory, and network usage.

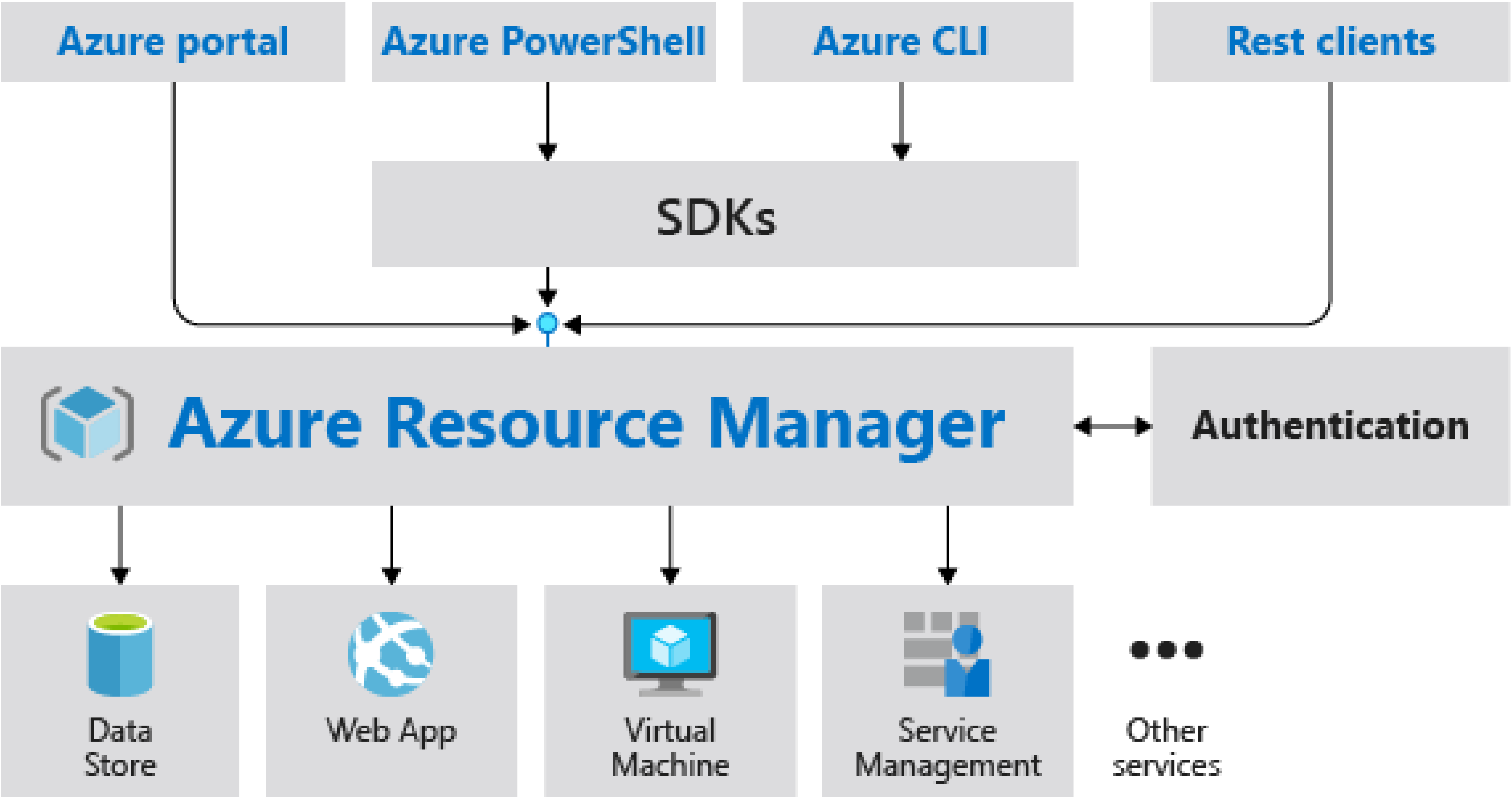
Host diagnostics from Docker or Azure.

Diagnostic trace logs from your app - so that you can correlate trace events with requests.

Custom events and metrics that you write yourself in the client or server code, to track business events such as items sold or games won.

Demo AKS Microservice Architecture

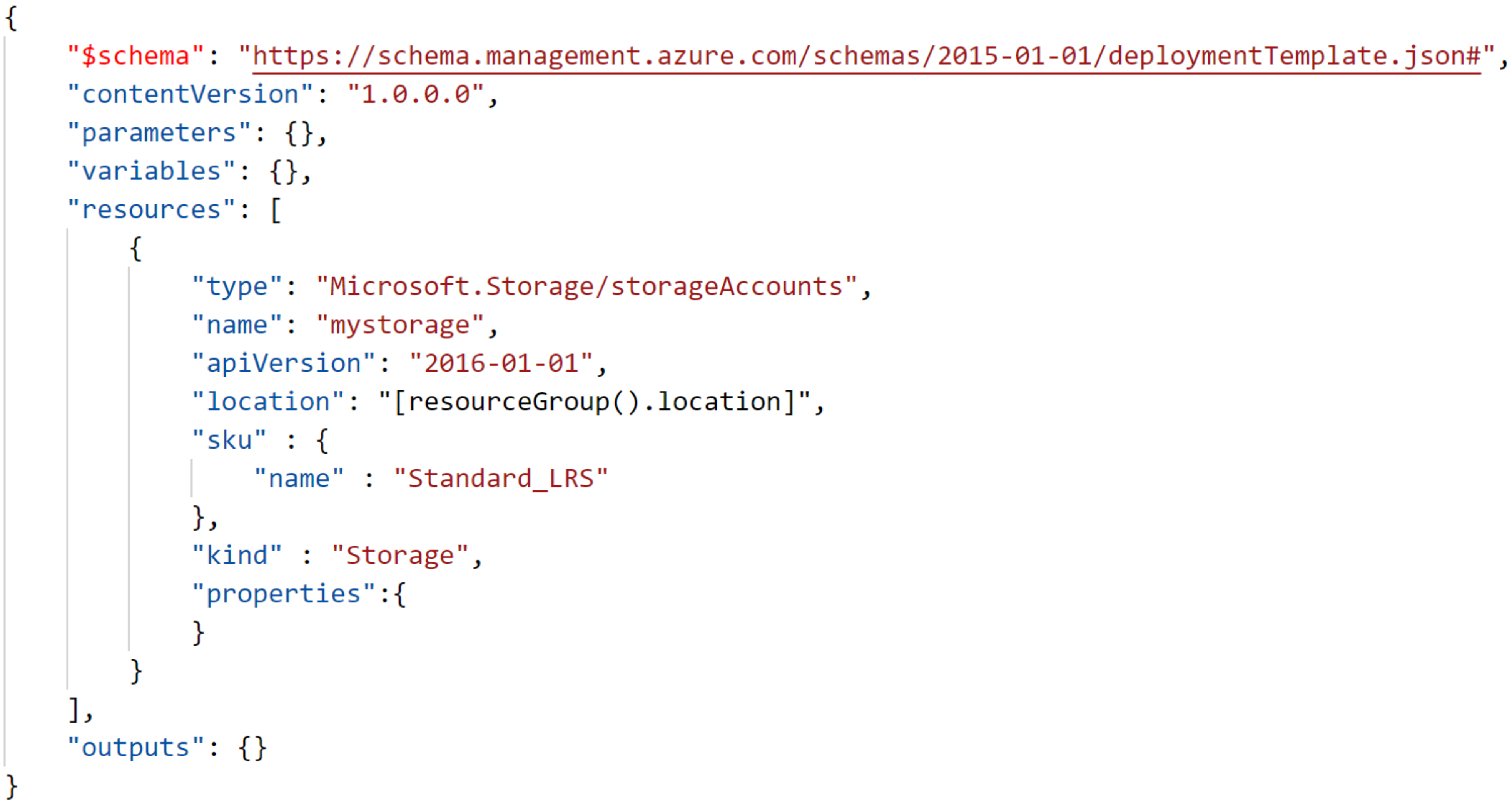

Azure AD
AAD Integrated
TodoList API
AI
TodoItems Cosmos DB container
ServiceBus Topics
Publish changes
AI
API Gateway with Swagger UI
Subscribe changes
TodoAnalytics API
AI
TodoAnalytics Event processor
AI
TodoAnalytics Cosmos DB container


Azure portal
Azure PowerShell
Azure CLI
Rest clients
SDKs
Azure Resource Manager
Authentication
Data Store
Web App
Virtual Machine
Service Management
Other services

Terminology

- resource A manageable item that is available through Azure. Virtual machines, storage accounts, web apps, databases, and virtual networks are examples of resources
- resource group - A container that holds related resources for an Azure solution. The resource group includes those resources that you want to manage as a group. You decide how to allocate resources to resource groups based on what makes the most sense for your organization
- resource provider - A service that supplies Azure resources. For example, a common resource provider is Microsoft.Compute, which supplies the virtual machine resource. Microsoft.Storage is another common resource provider

ARM Template

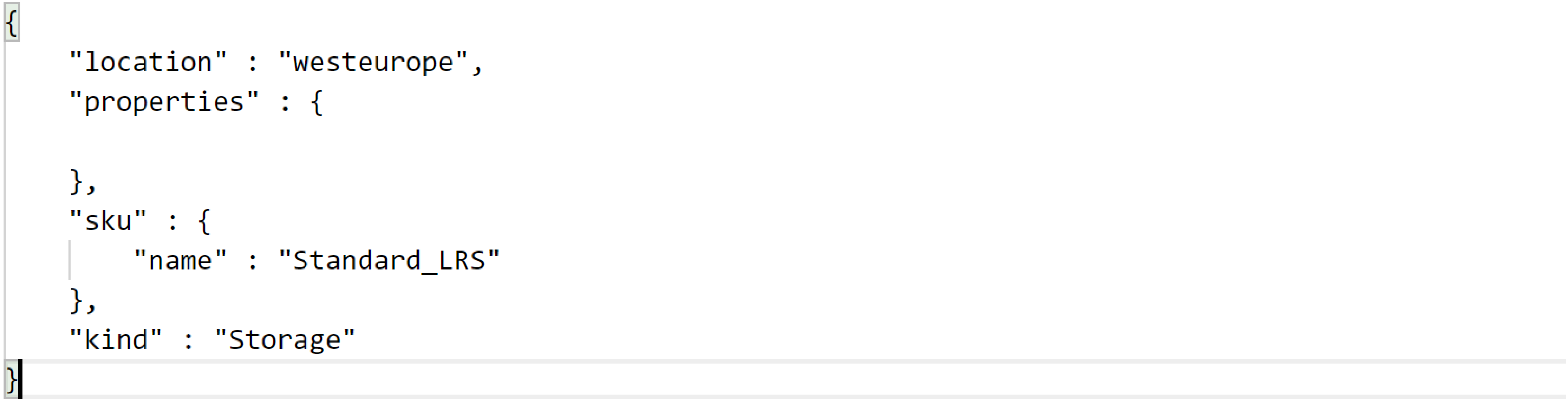


```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {},
    "variables": {},
    "resources": [
        {
            "type": "Microsoft.Storage/storageAccounts",
            "name": "mystorage",
            "apiVersion": "2016-01-01",
            "location": "[resourceGroup().location]",
            "sku" : {
                "name" : "Standard_LRS"
            },
            "kind" : "Storage",
            "properties":{
            }
        }
    ],
    "outputs": {}
}
```

ARM REST Api Call

```
PUT
https://management.azure.com/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/
providers/Microsoft.Storage/storageAccounts/mystorageaccount?api-version=2016-01-01
```

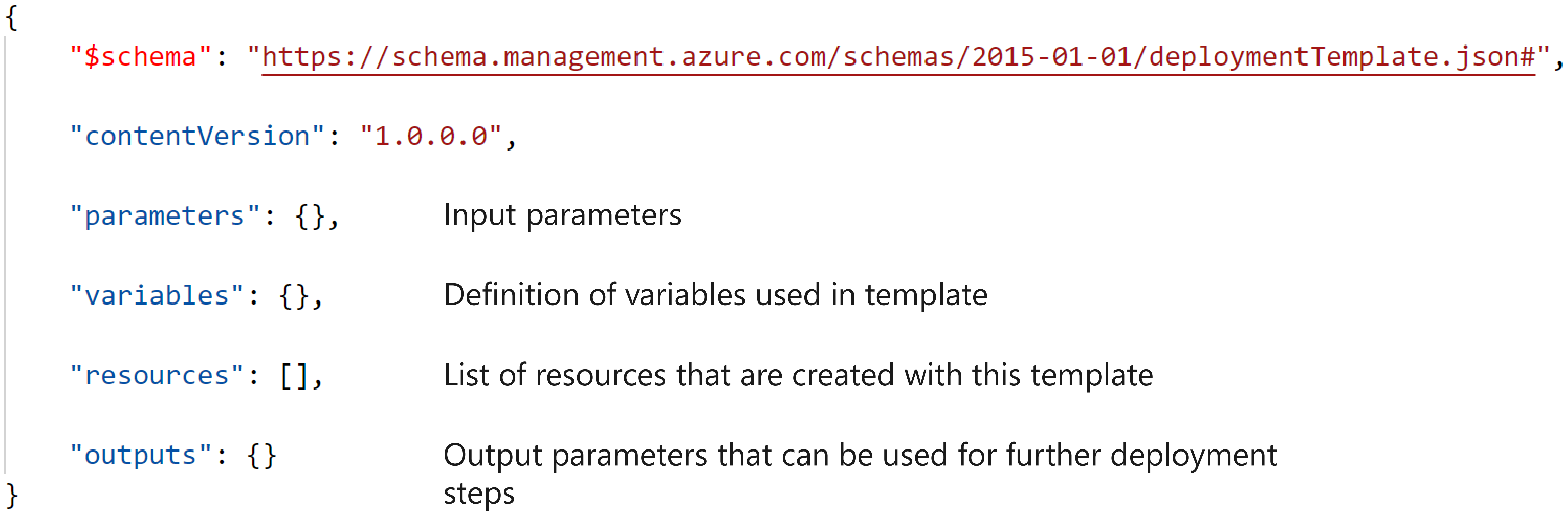
REQUEST BODY



```
{
    "location" : "westeurope",
    "properties" : {

    },
    "sku" : {
        "name" : "Standard_LRS"
    },
    "kind" : "Storage"
}
```

Blank ARM Template



```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",

    "contentVersion": "1.0.0.0",

    "parameters": {},

    "variables": {},

    "resources": [],

    "outputs": {}
}
```

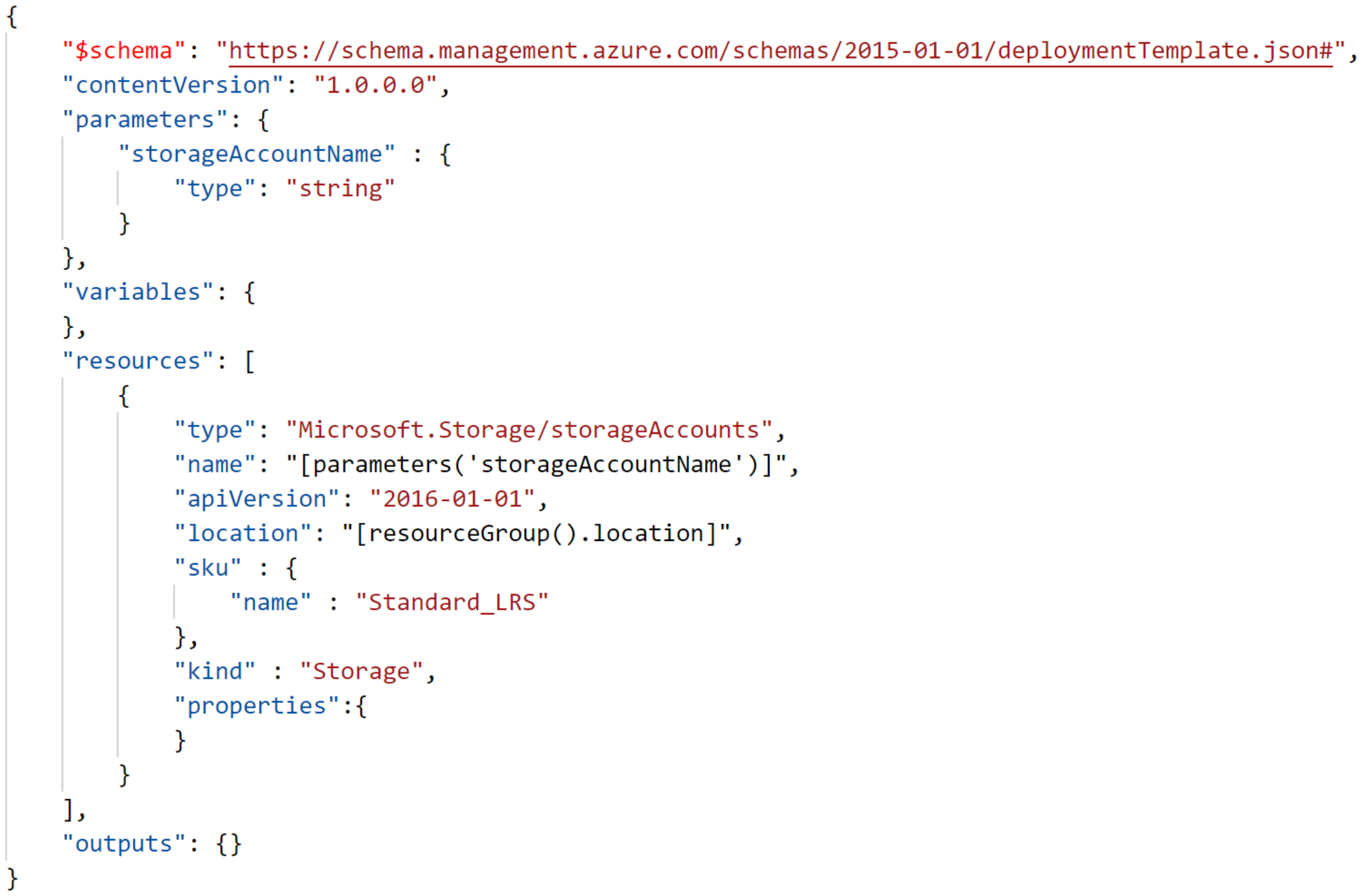
"parameters": Input parameters

"variables": Definition of variables used in template

"resources": List of resources that are created with this template

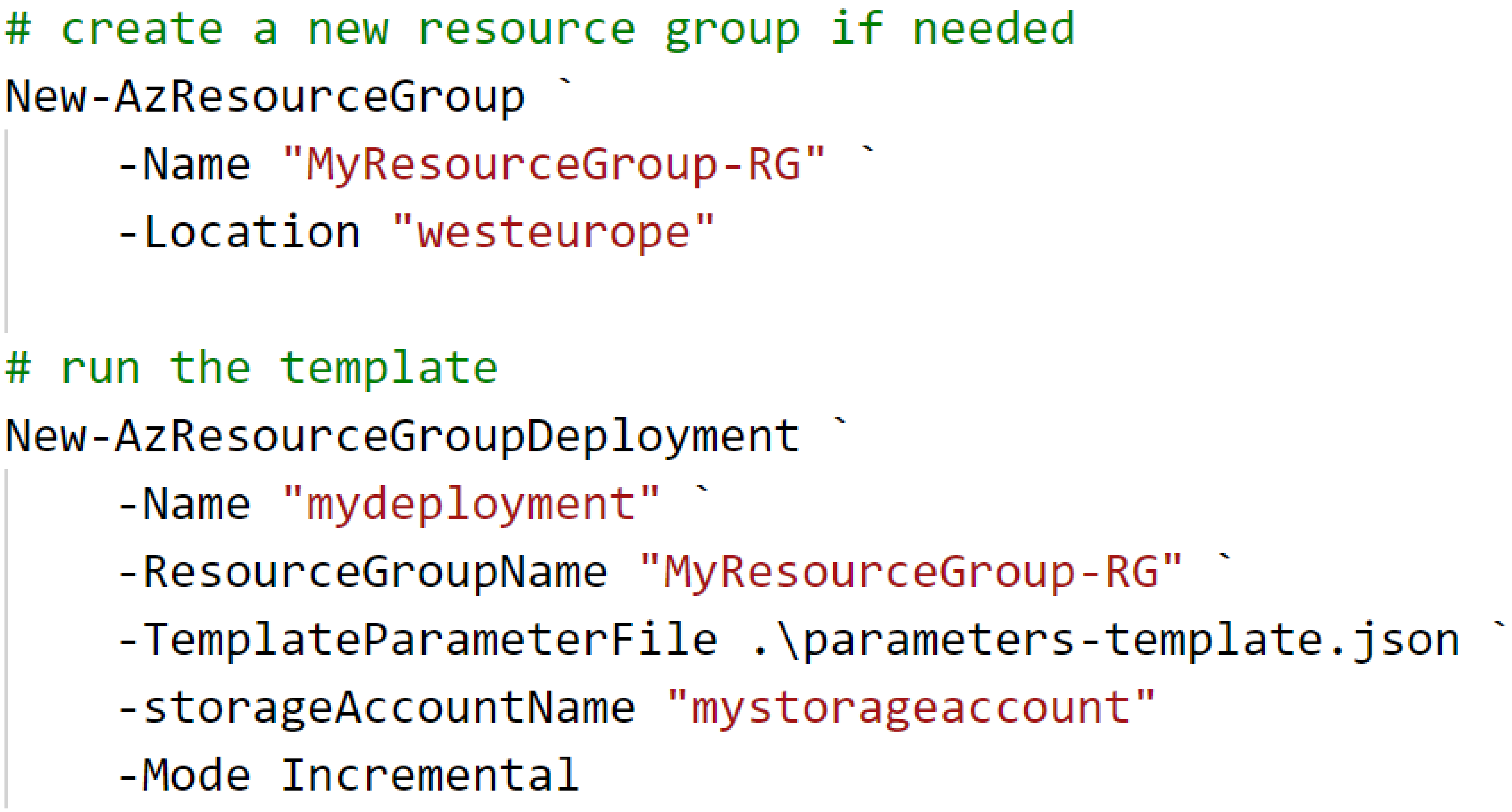
"outputs": Output parameters that can be used for further deployment steps

ARM Template Parameters



```
{
    "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",
    "contentVersion": "1.0.0.0",
    "parameters": {
        "storageAccountName" : {
            "type": "string"
        }
    },
    "variables": {
    },
    "resources": [
        {
            "type": "Microsoft.Storage/storageAccounts",
            "name": "[parameters('storageAccountName')]",
            "apiVersion": "2016-01-01",
            "location": "[resourceGroup().location]",
            "sku" : {
                "name" : "Standard_LRS"
            },
            "kind" : "Storage",
            "properties":{
            }
        }
    ],
    "outputs": {}
}
```

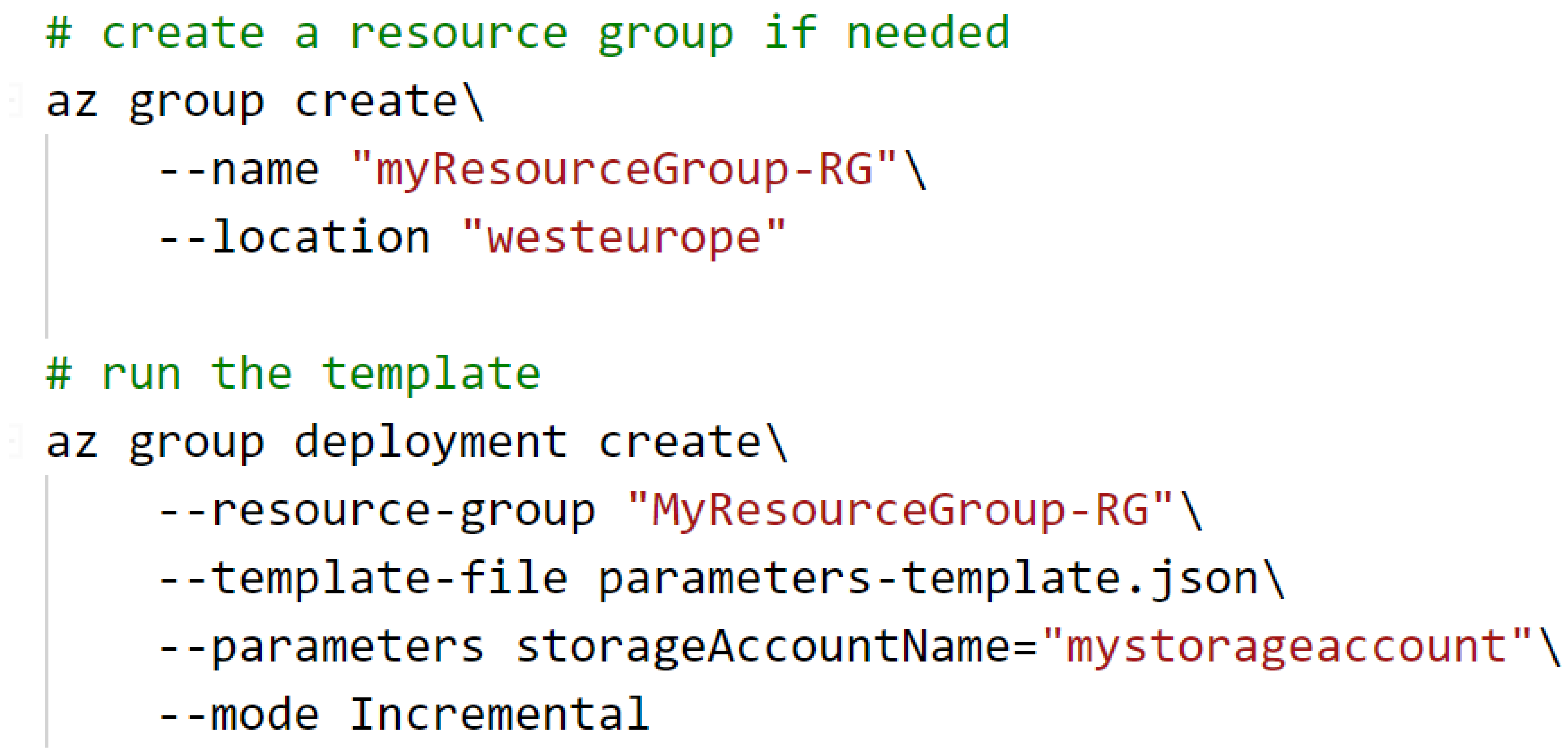
Run an ARM Template



```
# create a new resource group if needed
New-AzResourceGroup `
    -Name "MyResourceGroup-RG" `
    -Location "westeurope"

# run the template
New-AzResourceGroupDeployment `
    -Name "mydeployment" `
    -ResourceGroupName "MyResourceGroup-RG" `
    -TemplateParameterFile .\parameters-template.json `
    -storageAccountName "mystorageaccount"
    -Mode Incremental
```

Run an ARM Template



```
# create a resource group if needed
az group create\
     --name "myResourceGroup-RG"\
     --location "westeurope"

# run the template
az group deployment create\
     --resource-group "MyResourceGroup-RG"\
     --template-file parameters-template.json\
     --parameters storageAccountName="mystorageaccount"\
     --mode Incremental
```

Links

- Azure Resource Manager Documentation https://docs.microsoft.com/en-us/azure/azure-resource-manager/
- Azure Quickstart Templates https://azure.microsoft.com/en-us/resources/templates/?sort=Popular
- Azure Quickstart Template on GitHub https://github.com/Azure/azure-quickstart-templates

Demo

- Parts Unlimited
- MSENG