Mortal Combat:
Azure Automation vs. Functions

Azure Automation key facts

- **cloud-based, cross-platform** automation and configuration service for your Azure and non-Azure environments
- **Key capabilities:** process automation, configuration management and update management
- Windows PowerShell scripts and PowerShell Workflows (+ others)
- Supports **AzureRM and Az modules**
- Automation account
- Built-in integration with **PowerShell Gallery** and **Script Center**
- Source control (**CVS**) integration
- **Authoring and testing:** Portal editor or tools (Windows PowerShell ISE, VS Code)
- Supports **delegated resource management** (Lighthouse)

Azure Functions Key Facts

- Key **serverless** offering in Azure, new programming model based on triggers and bindings
- Languages: C#, F#, JS, Java, **PowerShell**, Python, TypeScript
- Runtime versions: 1, 2, and 3 (all GA), **PowerShell in 2 and 3**
- **Automatic management** of Azure (Az) modules
- Managed **Identity** support
- Supports **only Az modules**
- Native **bindings** to respond to Azure Monitor alerts, events published to Event Grid, HTTP or Timer triggers
- **Hybrid management:** VNet integration, App Service Hybrid Conn
- **Authoring and testing:** Portal or tools (VS, VS Code, any-IDE /w Azure Functions Core Tools
- Runtime is open-sourced on GitHub

Azure Automation Capabilities

Process Automation
Orchestrate processes using graphical, PowerShell, and Python runbooks

Configuration Management
Collect inventory
Track changes
Configure desired state

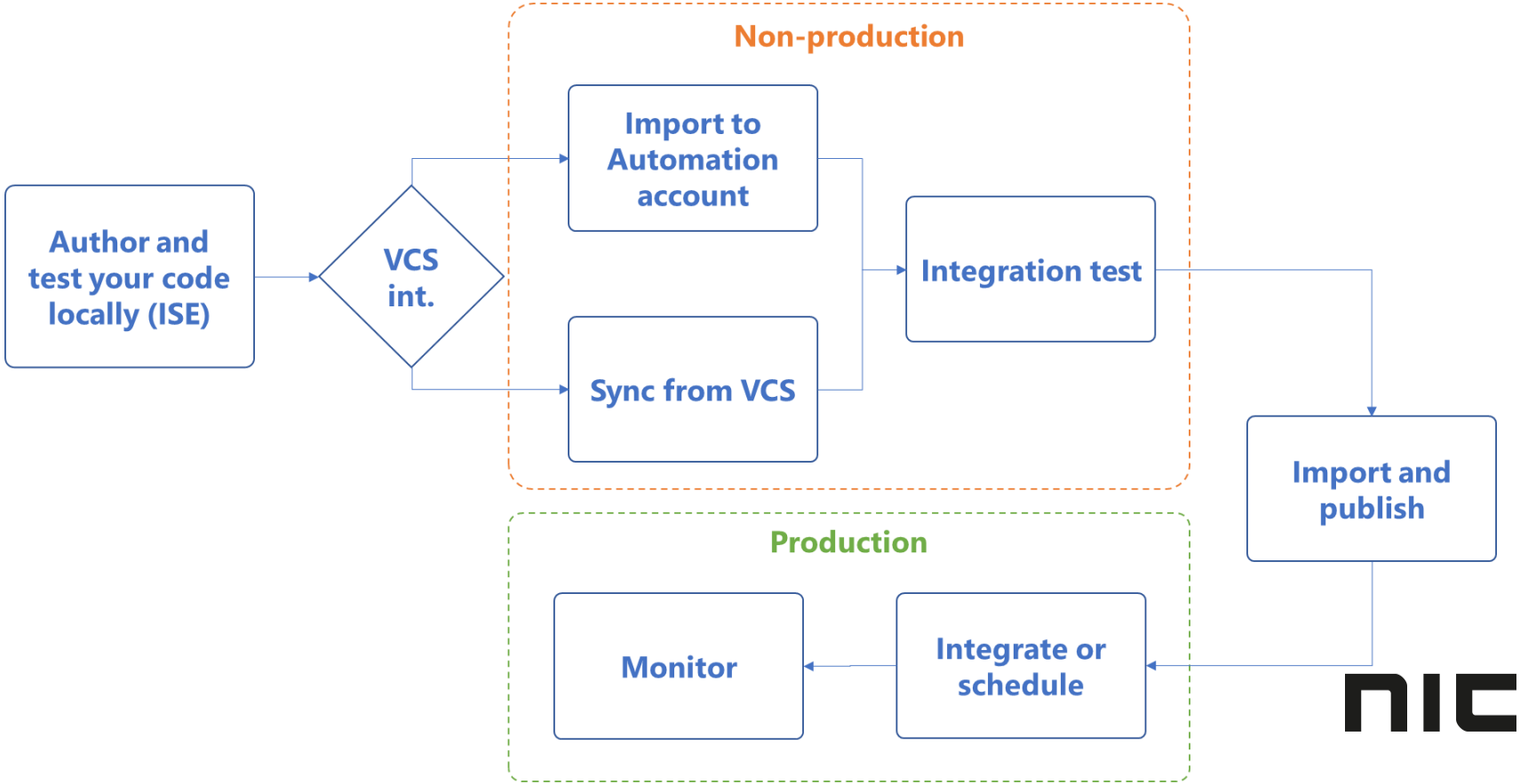
Update Management
Assess compliance
Schedule update installation

Shared capabilities
Role based access control
Secure, global store for variables, credentials, certificates, connections
Flexible scheduling
Shared modules
Source control support
Auditing
Tags

Heterogenous
Windows & Linux
Azure and on-premises

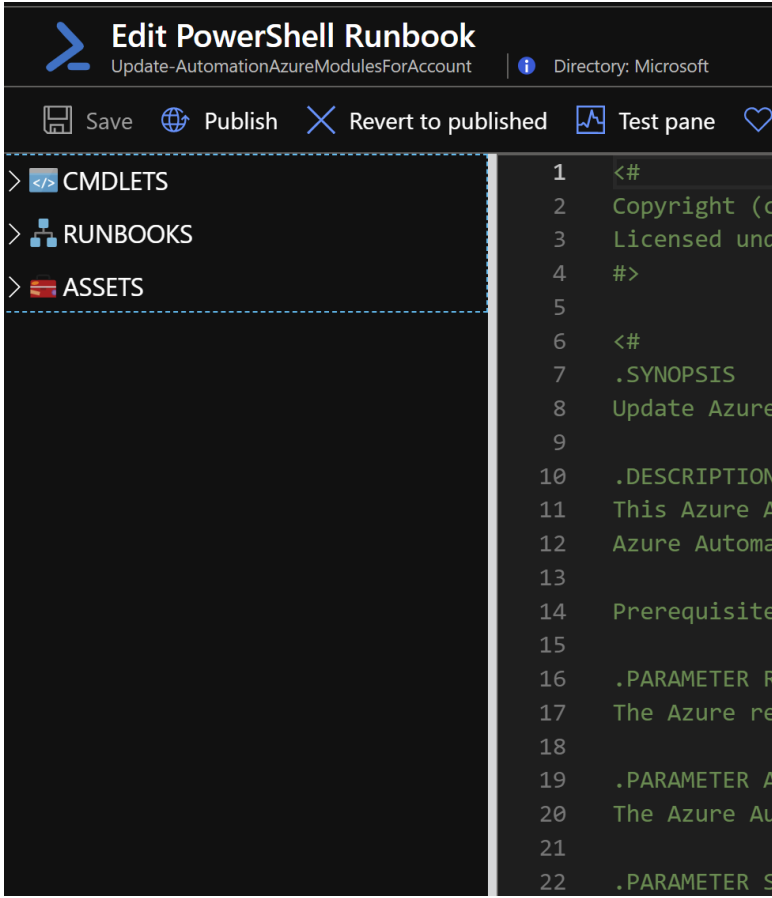
Round #1
Code authoring, workflows, and tools

Typical workflow for Runbooks

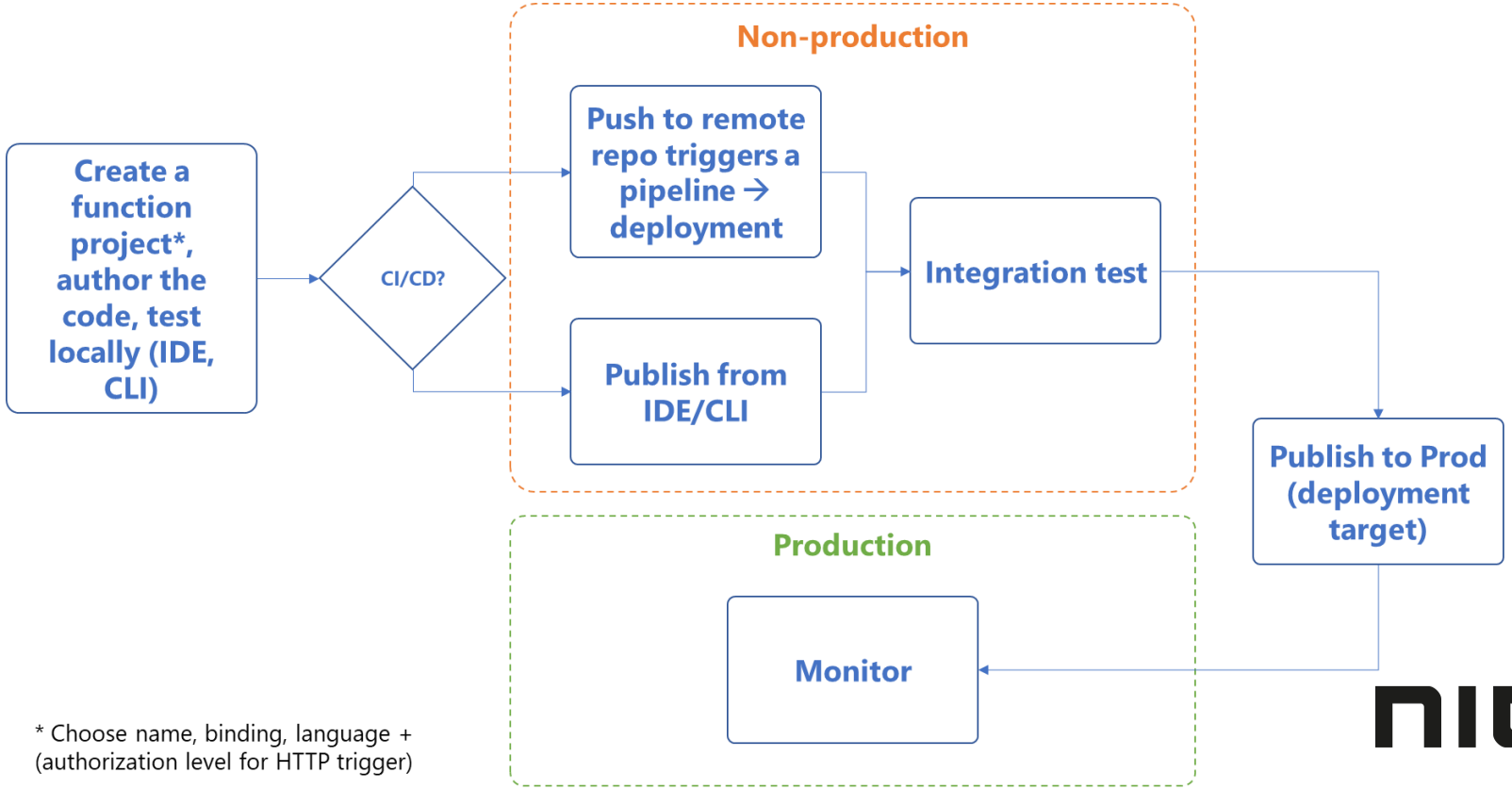


Azure Automation Tools

- **PowerShell ISE Add-on**
  - still works, open-sourced on GitHub, last release 10/2017, build for VS Code
- **Editor in the Portal**
  - Author and test, pane with cmdlets/runbooks/assets
- **Other IDE/ISE + PSH cmdlets**
  - no CLI support



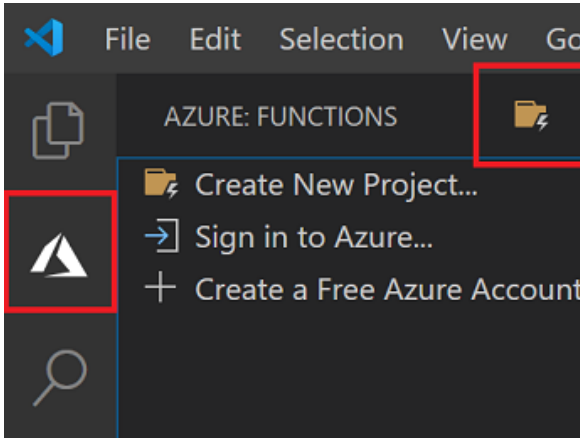
Typical workflow for Functions



* Choose name, binding, language + (authorization level for HTTP trigger)

Azure Functions Tools

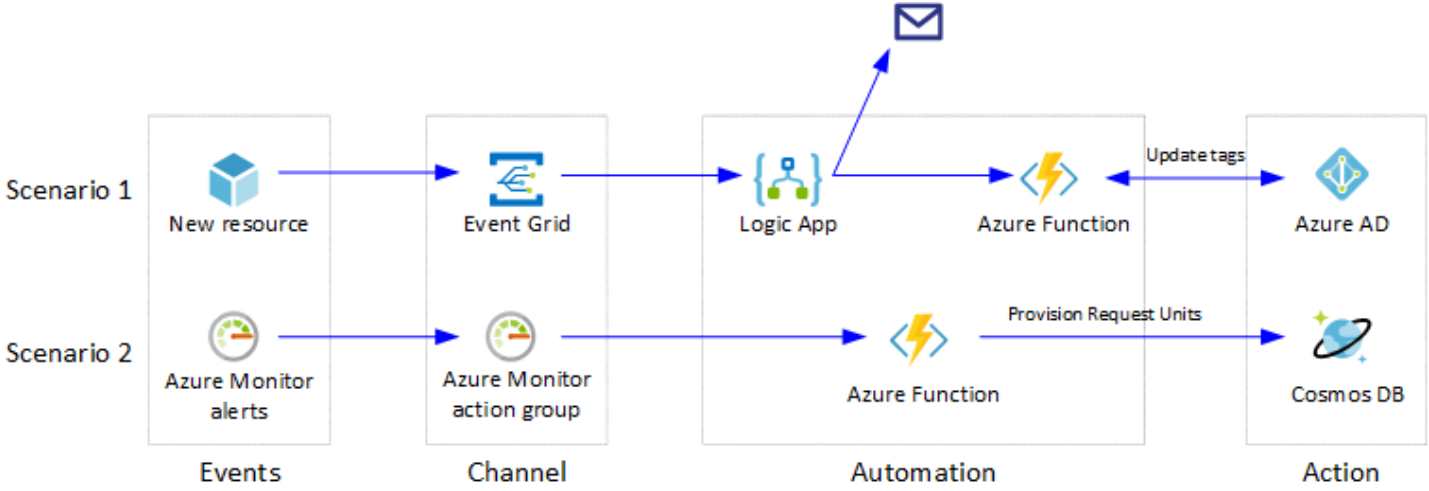
- Visual Studio (Azure development workload)
- Visual Studio Code (Azure Functions extension)
- Editor in the Portal - Author and test
- Other IDE/ISE – Azure Functions Core Tools
  - Node.js, .NET Core, PowerShell Core SDK
- Azure Cloud Shell
- Visual Studio Online
- *Do not mix local development with portal development in the same function app!*

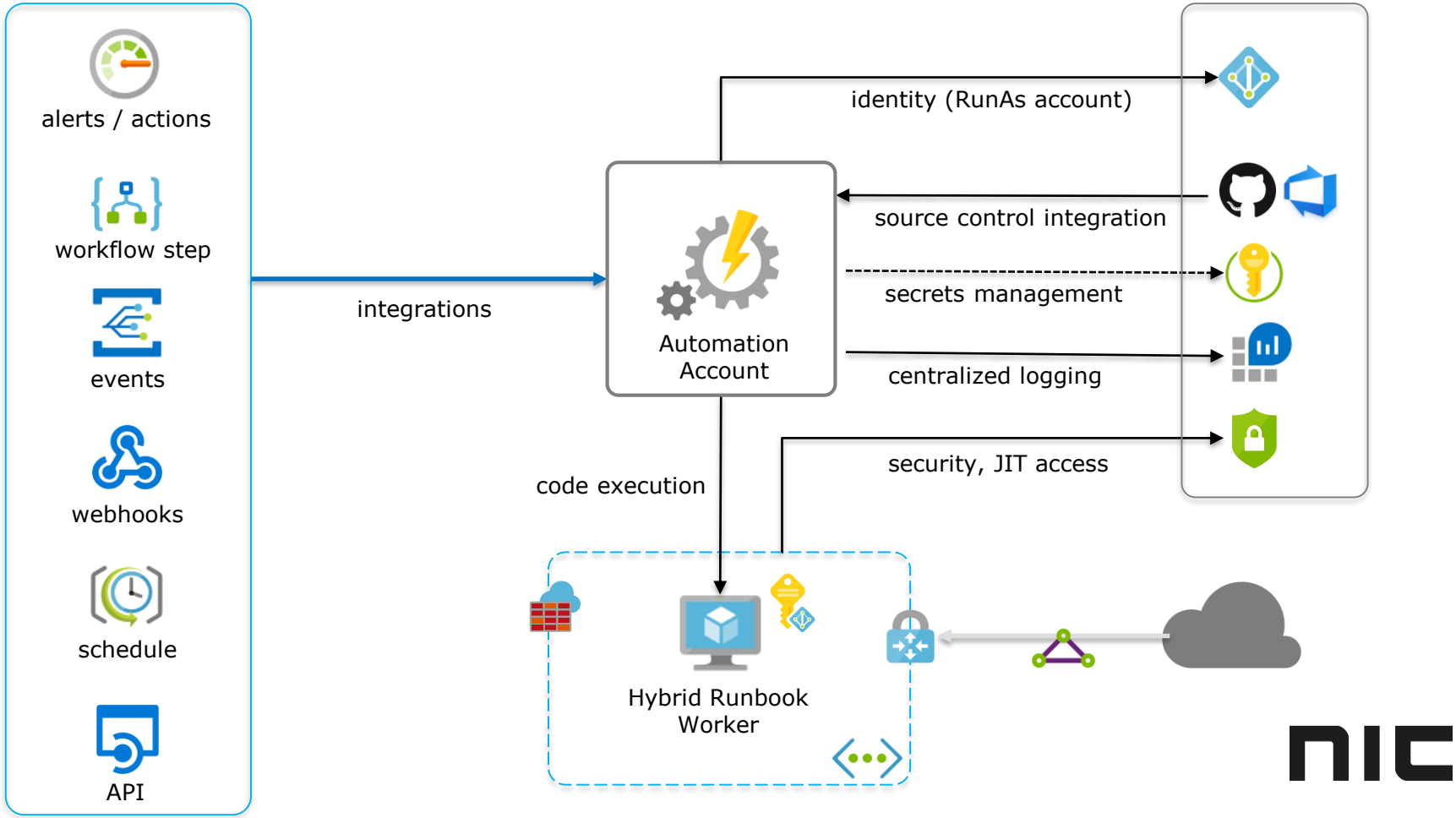


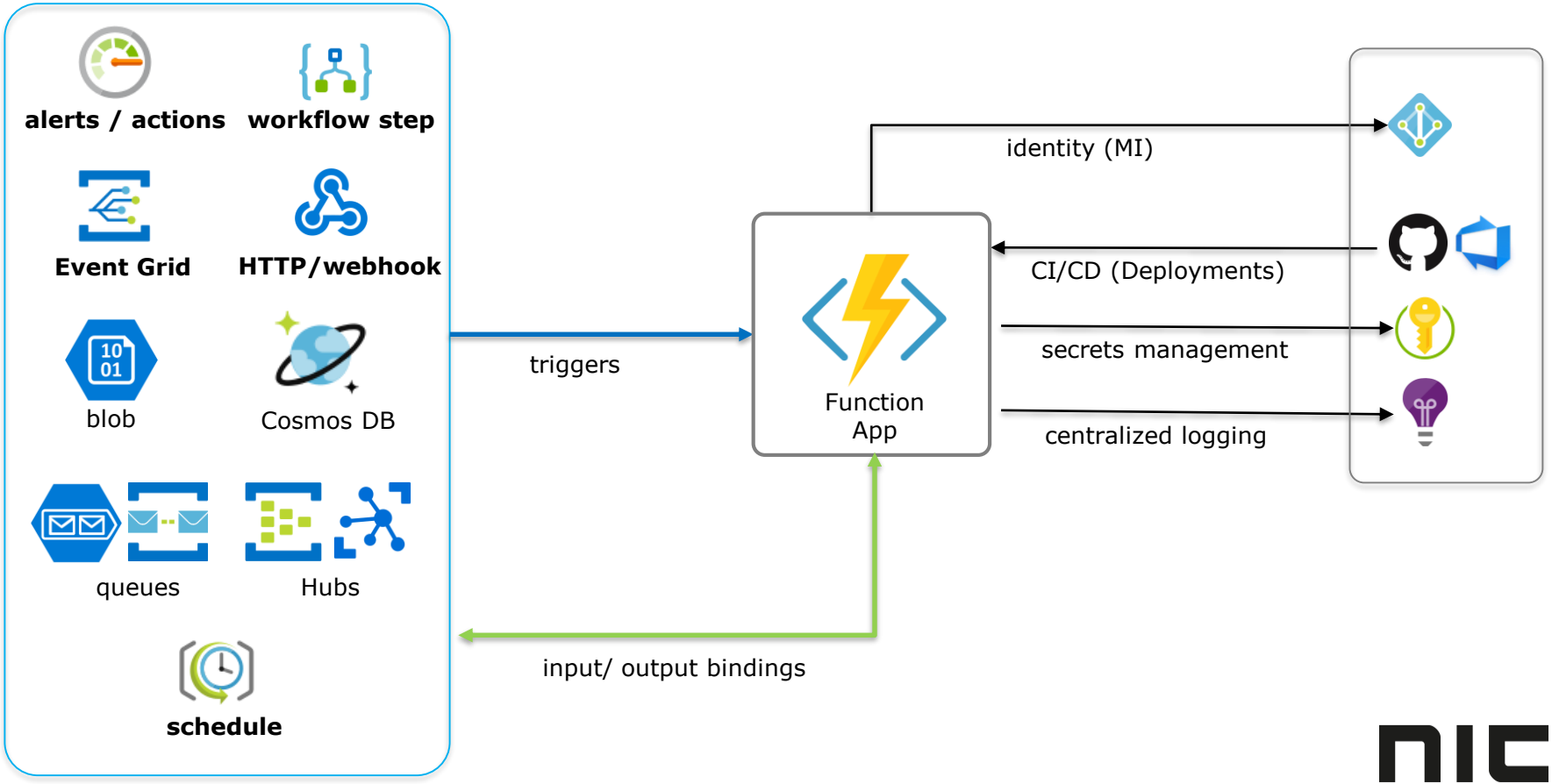
Round #2
Use cases and integrations

Patterns in event-based automation

- Respond to events on resources – uses Event Grid
- Scheduled tasks – timer-trigger function
- Process Azure alerts – Azure Monitor alerts / action groups
- Orchestrate with external systems – uses Logic Apps

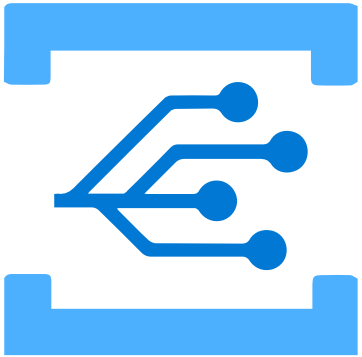



alerts / actions
workflow step
events
webhooks
schedule
API
integrations
Automation Account
identity (RunAs account)
source control integration
secrets management
centralized logging
security, JIT access
code execution
Hybrid Runbook Worker



https://docs.microsoft.com/en-us/azure/azure-functions/functions-triggers-bindings

Azure Event Grid



Simplify event-based automation with a **publish-subscribe model**

- Simple HTTP-based event delivery.
- Build better, more reliable automation through reactive programming.

Common scenarios:

- Add tags with looked up values when resource is created.
- Grant access to resource group to ops / dev teams when it is created.
- Send teams event when resource is deleted.
- Respond to forwarded VM maintenance notification (scheduled events)
  https://github.com/Azure-Samples/virtual-machines-python-scheduled-events-central-logging

Azure Monitor

Respond to Azure alerts to remediate or escalate to external system.

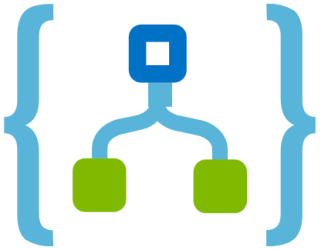
- Native integration with action groups in Azure Monitor.
- Respond to metrics or based on log search query.

Common scenarios:

- Send teams event when Azure functions are failing.
- Restart service inside a VM when it is stopped.
- Truncate table when SQL database reaches maximum size.

Integrating with external systems with Logic Apps

Orchestrate processes across various systems to perform end to end automation.



- Over 300 connectors
- Visual designer to focus on business process
- Call Azure function to run automation code.

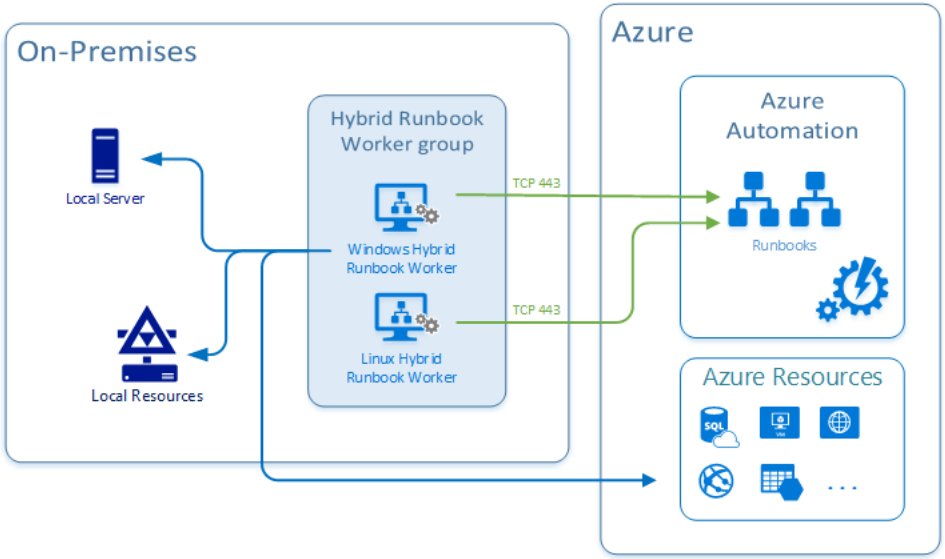
Common scenarios:

- Fulfill request based on approval in ServiceNow system
- Send customized email notification when automation task is completed

Round #3
Hybrid Environment Automation

Azure Automation

- Sandboxes (hosted workers)
- Hybrid Runbook Worker (hosted in Azure, on-prem, other hosting options)

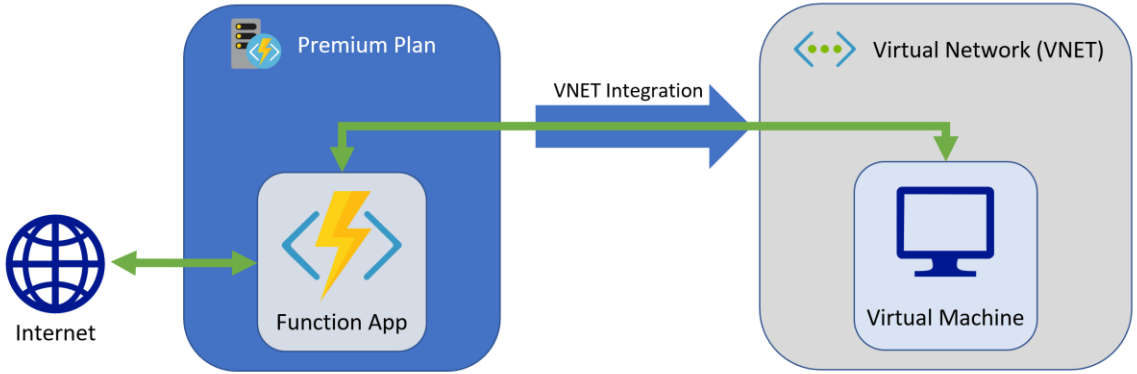


Hybrid Runbook Worker Benefits

- No “Fair share” limits (180 min)
- **Complete control** over the host, it’s config and capacity
- Pre-install all PSH modules and other tools → **speed**
- Utilize Azure VM extensions *(e.g. can be domain-joined, if needed)*
- Control **network traffic**: private VNet, Azure Firewall & NSGs, connectivity to on-prem, service endpoints & private link
- **Compliance** – use in-guest policies or Azure DSC, onboard VM to Security Center, Azure Arc
- **Better logging and monitoring** – diagnostic logs and metrics to Log Analytics
- Managed **Identity** (vs. RunAs) and Key Vault integration
- **Scale** – HWR group

Azure Functions

- VNet integration in Premium Plan
  - Create an empty subnet (dedicated for function app)
- App Service Hybrid Connections
- Isolated App Service Plan (ASE)



Round #4
Dependency Management

Azure Automation

- Import from PSH Library or your own repo
- **Azure modules** - default is AzureRM, you can install Az modules side-by-side (you can't delete modules provided out-of-the-box)
- Azure modules **auto-update**
  - https://github.com/Microsoft/AzureAutomation-Account-Modules-Update
  - Create / Import a runbook, parameters
  - Can update Azure, AzureRM, and Az modules
- #Requires –Module Az.Compute in your code

Azure Functions

- PowerShell modules can be managed by service **automatically**
- Service will keep the function app updated with the latest dependencies as they ship.
- Control major version upgrade of the dependencies.
- Custom modules upload

Host.json

```
1 {
2   "version": "2.0",
3   "managedDependency": {
4     "Enabled": true
5   }
6 }
```

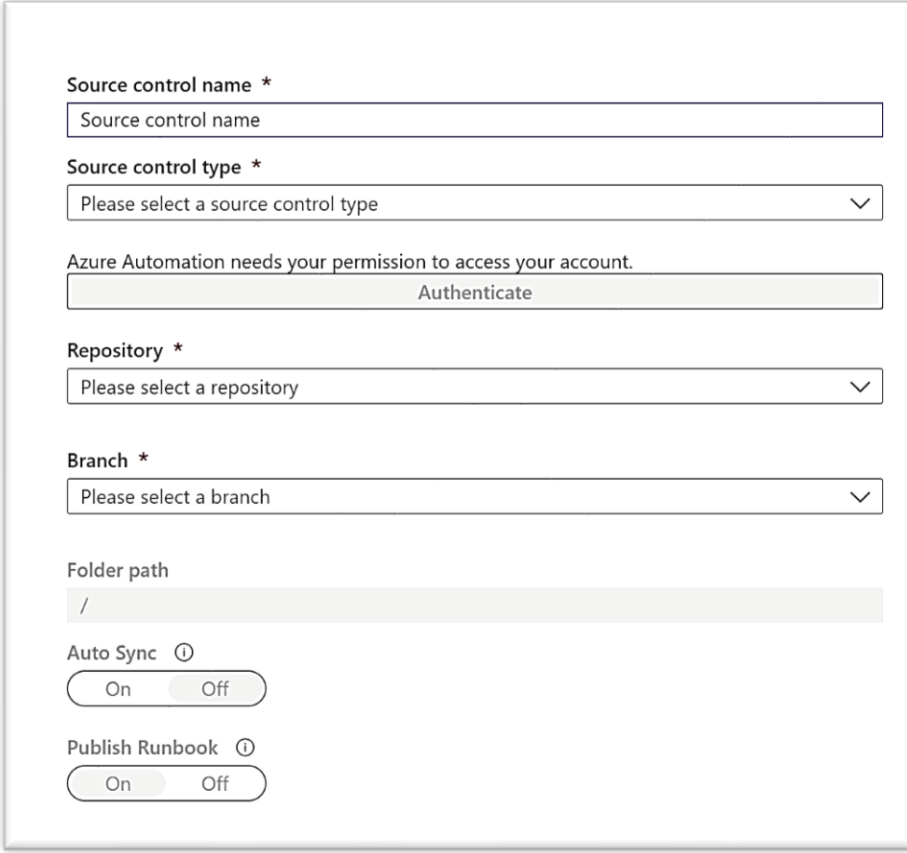
Requirements.psd1 (PowerShell)

```
1 @{
2     Az = '1.*'
3 }
4
```

Round #5
DevOps

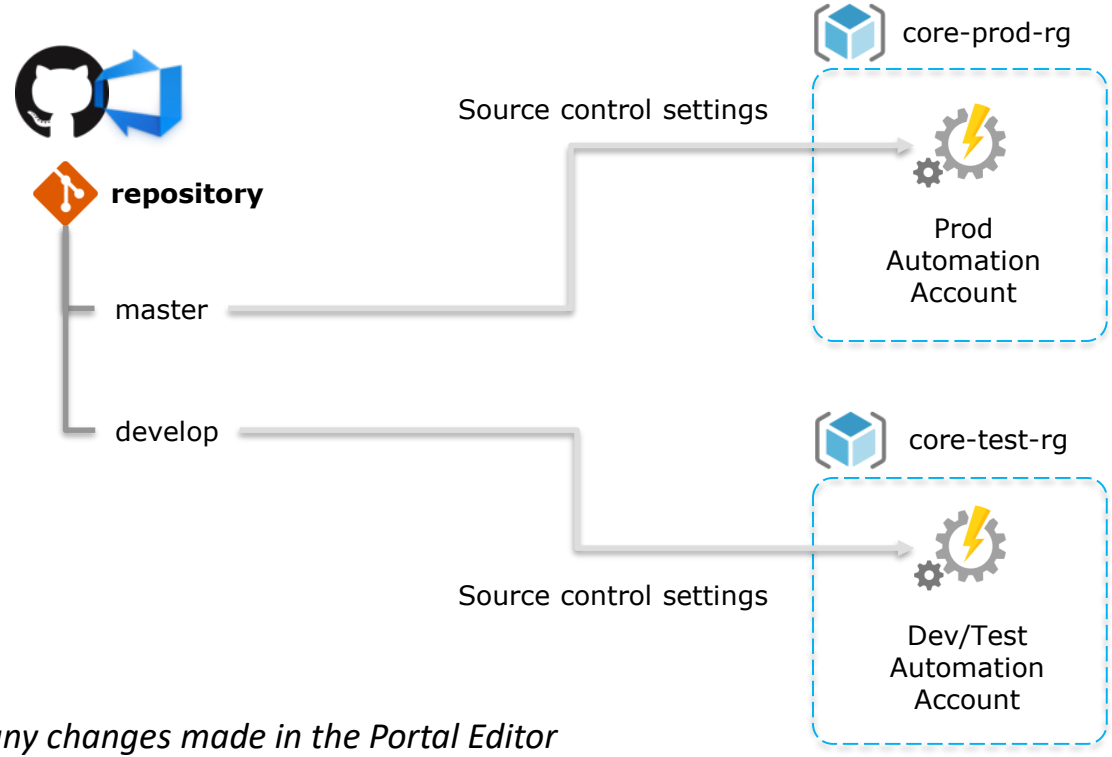
Azure Automation

- **CVS integration:**
  - Built-in vs. DIY
  - GitHub | Azure Repos (Git, TFVC)
  - Auto Sync & Auto Publish
- **Infra-as-Code / Config-as-Code:**
  - ARM templates * | Terraform
  - (RunAs account, HRW)
  - Variables
- **CI/CD:**
  - No GitHub Actions



* https://docs.microsoft.com/en-us/azure/templates/microsoft.automation/allversions

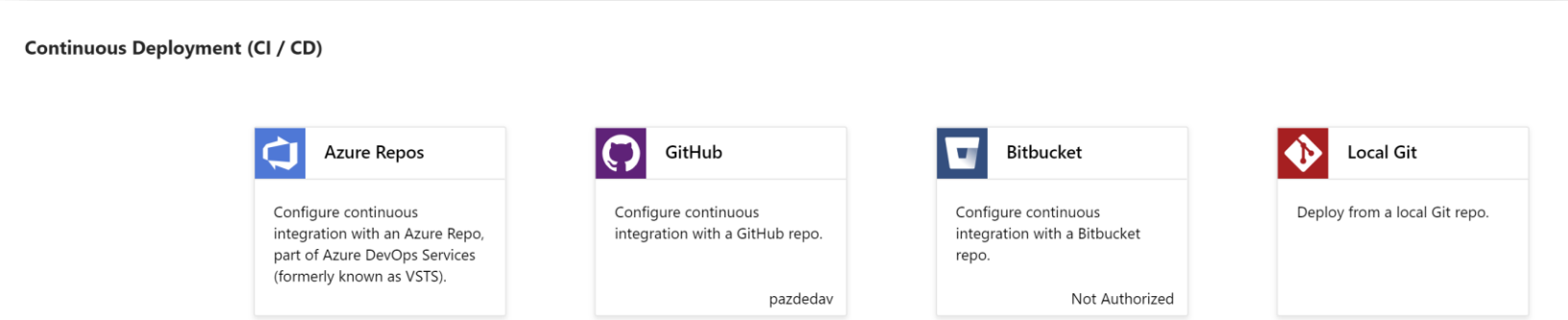
Example setup for CVS integration



Note: sync overrides any changes made in the Portal Editor

Azure Functions

- Deployment Center
- **Infra-as-Code / Config-as-Code:**
  - ARM templates | Terraform support
  - App Settings (local.settings.json)
- **CI/CD:**
  - Azure Pipelines | GitHub Actions
  - Deployment slots



Round #6
Security

Azure Automation

- **Azure identity and secrets for runbooks**
  - RunAs account (+ MSI for HRW)
  - Credentials and certificates in Shared Resources + Key Vault
- **Secure assets in Automation**
  - credentials, certificates, connections, and encrypted variables
  - Microsoft-managed-keys vs. BYOK (Preview) *
- **Access control**
  - 3 built-in roles (Automation Operator, Automation Job Operator, Automation Runbook Operator)
- **Webhooks**

* https://docs.microsoft.com/en-us/azure/automation/automation-secure-asset-encryption

Azure Functions

- **Azure identity and secrets for functions**
  - Managed Identity
  - App Settings with Key Vault references
- **Access control**
  - No Functions or App Service specific role
- **HTTP triggers**
  - OAuth: Active Directory, Facebook, Google, Twitter, and MSA

profile.ps1

```
if ($env:MSI_SECRET -and (Get-Module -ListAvailable Az.Accounts)) {
    Connect-AzAccount -Identity
}
```

KV reference in App Settings

```
@Microsoft.KeyVault (SecretUri= https://myvault.vault.azure.net/secrets/mysecret/ec96f0208)
```

* https://docs.microsoft.com/en-us/azure/automation/automation-secure-asset-encryption

Round #7
Pricing model

Azure Automation

- Process automation (example for West Europe in NOK)
- HRW: infra costs

Pricing details

Process automation

Process automation includes runbook jobs and watchers. Billing for jobs is based on the number of job run time minutes used in the month and for watchers is based on the number of hours used in a month. Charges for process automation are incurred whenever a job or watcher runs. You will be billed only for minutes/hours that exceed the free included units.

	FREE UNITS INCLUDED (PER MONTH)**	PRICE
Job run time	500 minutes	kr0.017/minute
Watchers	744 hours	kr0.017/hour

Azure Functions

- Pricing model depends on selected **hosting plan**
- **Consumption plan**: Azure provides all of the necessary computational resources. You don't have to worry about resource management, and only pay for the time that your code runs.
- **Premium plan**: You specify a number of pre-warmed instances that are always online and ready to immediately respond. When your function runs, Azure provides any additional computational resources that are needed. You pay for the pre-warmed instances running continuously and any additional instances you use as Azure scales your app in and out.
- **App Service plan**: Run your functions just like your web apps. If you use App Service for your other applications, your functions can run on the same plan at no additional cost.

More info: https://azure.microsoft.com/en-us/pricing/details/functions/

Azure Functions

- **Consumption plan**
  - Billed based on per-second resource consumption and executions
  - Extra charge for storage and egress

METER	PRICE	FREE GRANT (PER MONTH)
Execution Time*	kr0.000130/GB-s	400,000 GB-s
Total Executions*	kr1.623 per million executions	1 million executions

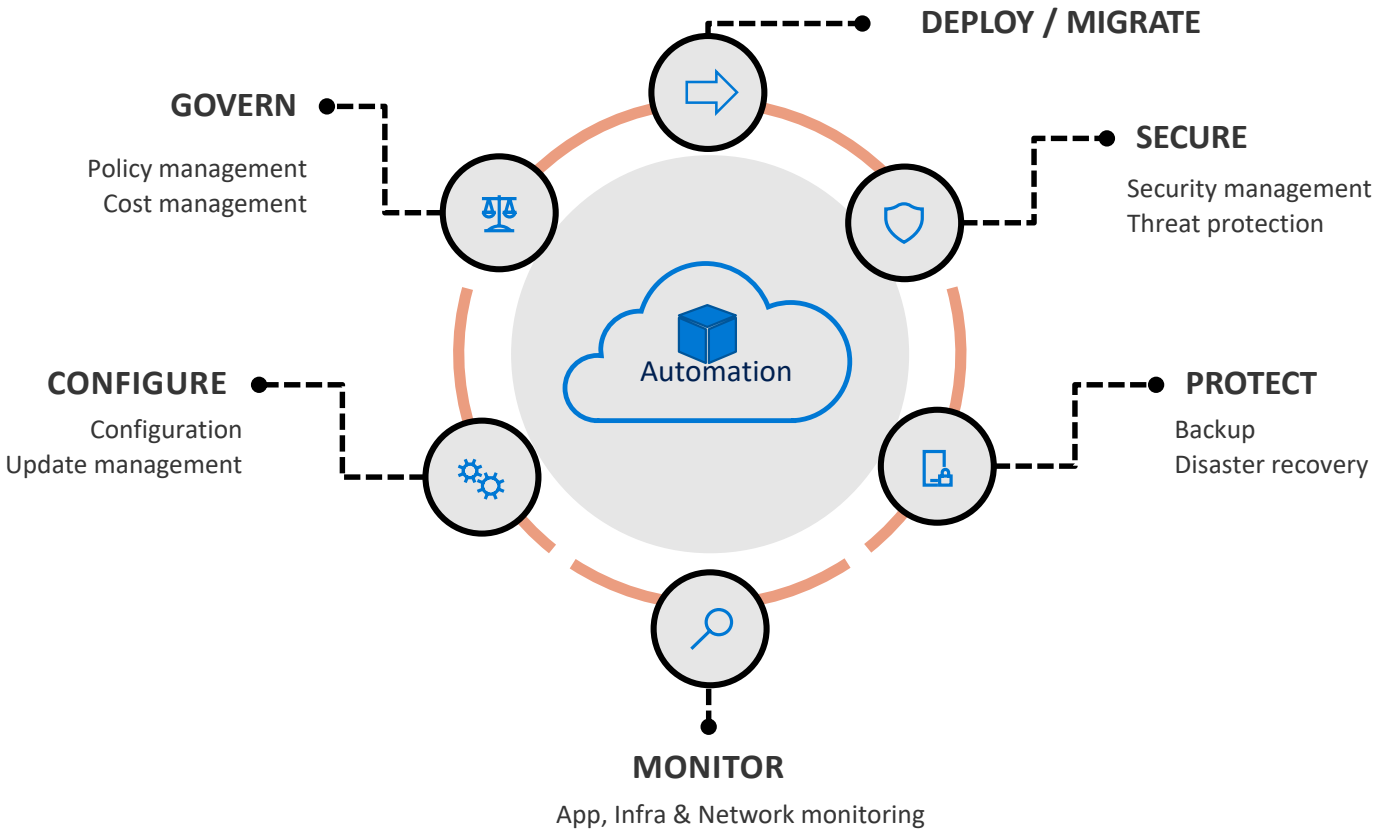
- **Premium plan**
  - Billed based on the vCPU and memory your functions consume

METER	PRICE
vCPU duration	vCPU: ~kr1,001.036517 vCPU/month
Memory duration	Memory: ~kr71.671846 GB/month

More info: https://azure.microsoft.com/en-us/pricing/details/functions/

Final round
When to choose what?

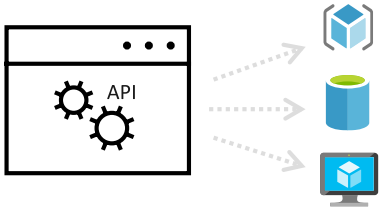
Automation across Azure lifecycle



Automation in Azure

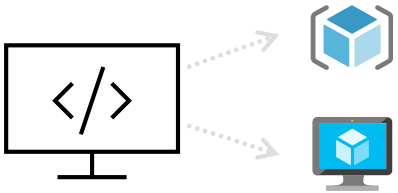
Deploy and operate infrastructure and applications in Azure using domain specific services

Deploy



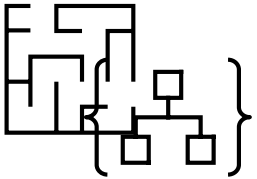
Deliver repeatable and consistent infrastructure as code.

Respond



Create event-based automation to diagnose and resolve issues.

Orchestrate



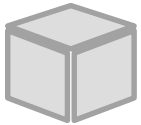
Orchestrate your automation across Azure and 3rd party systems.

DevOps

Deployment Manager

DSC

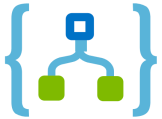
Policy



Resource Manager

Functions

Blueprints



Logic Apps

Resources

Azure PowerShell Functions Developer Guide

https://docs.microsoft.com/en-us/azure/azure-functions/functions-reference-powershell

Event-based Cloud Automation (Reference Architecture)

https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/serverless/cloud-automation

Serverless Library

https://serverlesslibrary.net/

More sessions on NIC 20/20

Event-based Automation with PowerShell in Azure Functions

Aleksandar Nikolic, 6.2. 4-5 PM, Room 5

Azure serverless for IT Pros

Martin Ehrnst, 6.2. 2.40-3.40 PM, Room 4

NIC

Slides and demos from the conference will be available at

https://github.com/nordicinfrastructureconference/2020

February 6th-7th

nic

20/20 VISION

Oslo Spektrum