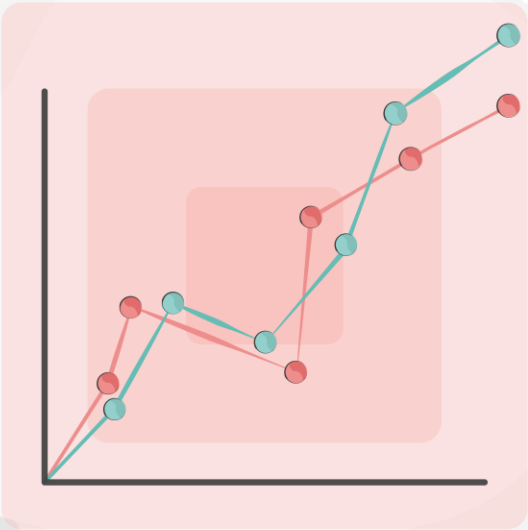
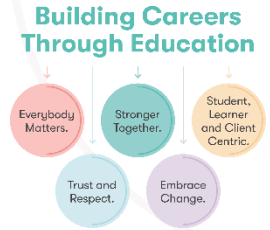
Level 5 Data Engineer
Module 5 Topic 6

Cloud solutions integration

Welcome to today's webinar.

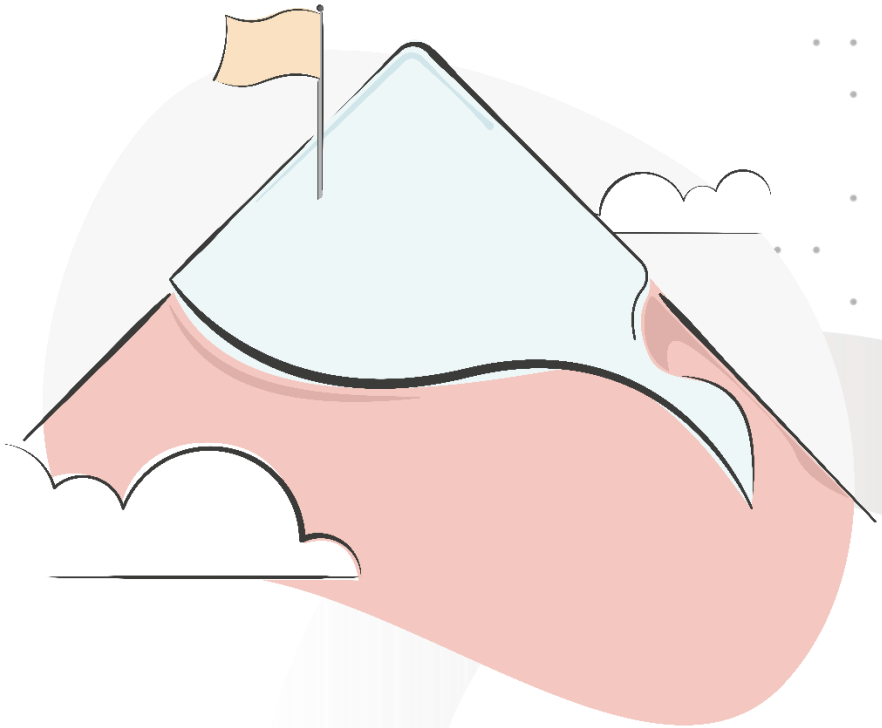


Session objectives



This webinar supports the following learning outcomes:

- Document and describe complex integrated cloud architectures.
- Explain the significance of Infrastructure as Code (IaC) in cloud integration and how it enhances efficiency and sustainability.
- Appraise the role of tools like Ansible and Teraform in automating cloud integrations.
- Demonstrate integrating multiple cloud providers into a hybrid cloud environment, including on-premises systems
- Explain the benefits of using comprehensive third-party tools like Snowflake and Databricks to integrate and manage data across cloud platforms.
- Explain how you would integrate visualisation tools (like Power BI) with a cloud data product.

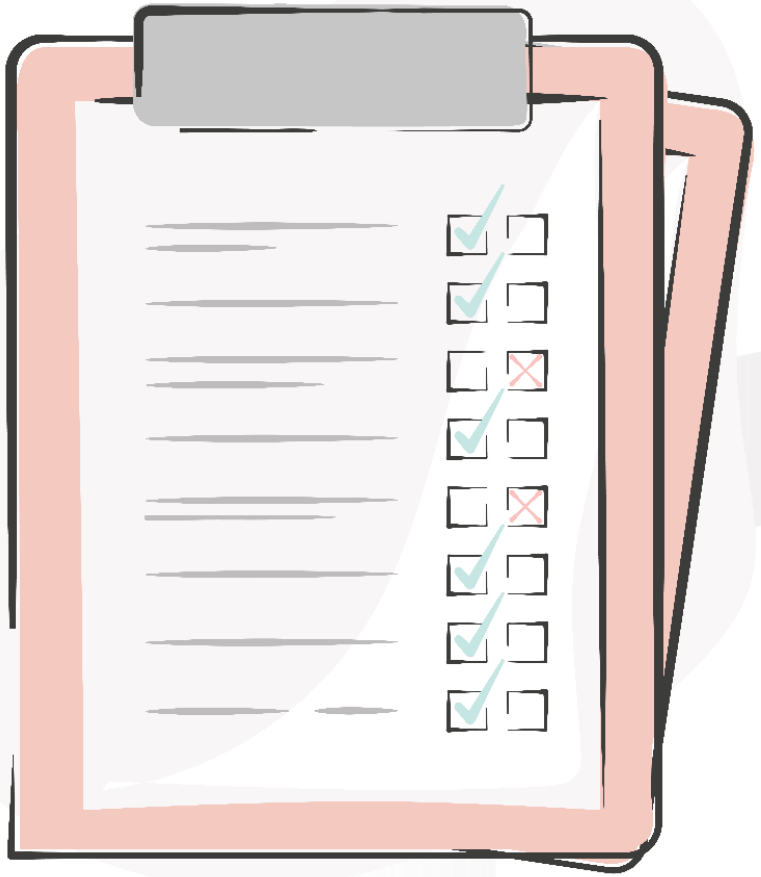


BPP

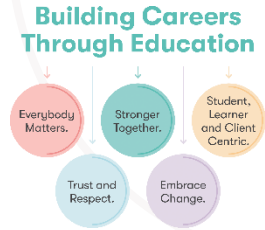
Webinar Agenda

What we will cover in the webinar:

1. Architecting a Cloud-Based Big Data Solution
2. Documenting & Visualising Architectures
3. Investigating Performance & Scaling
4. 4. Automating integrated deployments



Discussion



Cloud service providers

1. Which cloud service provider would you pick for a big data project? AWS, Azure or GCP?

2. In a real-time project, which these two options are more relevant for your organisation: dashboards or real-time alerting, and why?

3. To build either dashboards or real-time alerting, why would you normally have to **integrate** multiple data sources or multiple cloud solutions?

Submit your responses to the chat!

BPP

Cloud architecting

Goals…

- Architectures describe key information about a solution:
    - How data moves through the solution
    - What services will be/are utilised by the solution
    - What requirements the solution enables
- Architectural designs can contain varied levels of detail
    i. Data sources and storage destinations
    ii. Specific processing steps
    iii. Key services/software used
    iv. Performance expectations
    v. Costs
    vi. Users and use cases

BPP

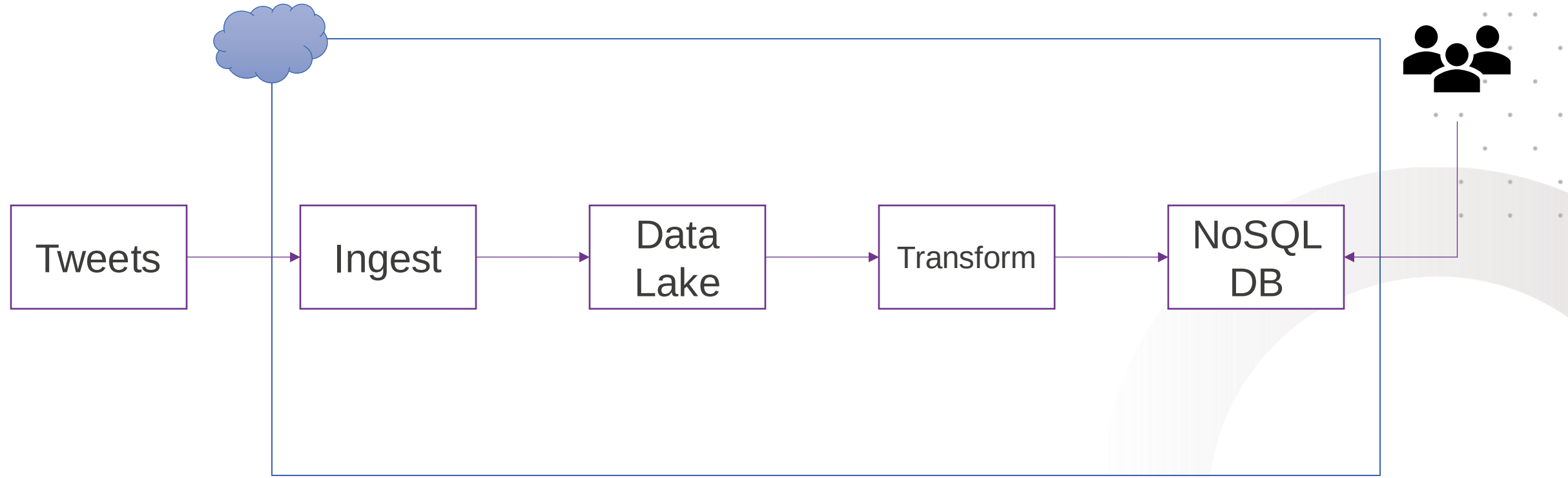
Cloud architecting

Simplified X Arch…

BPP

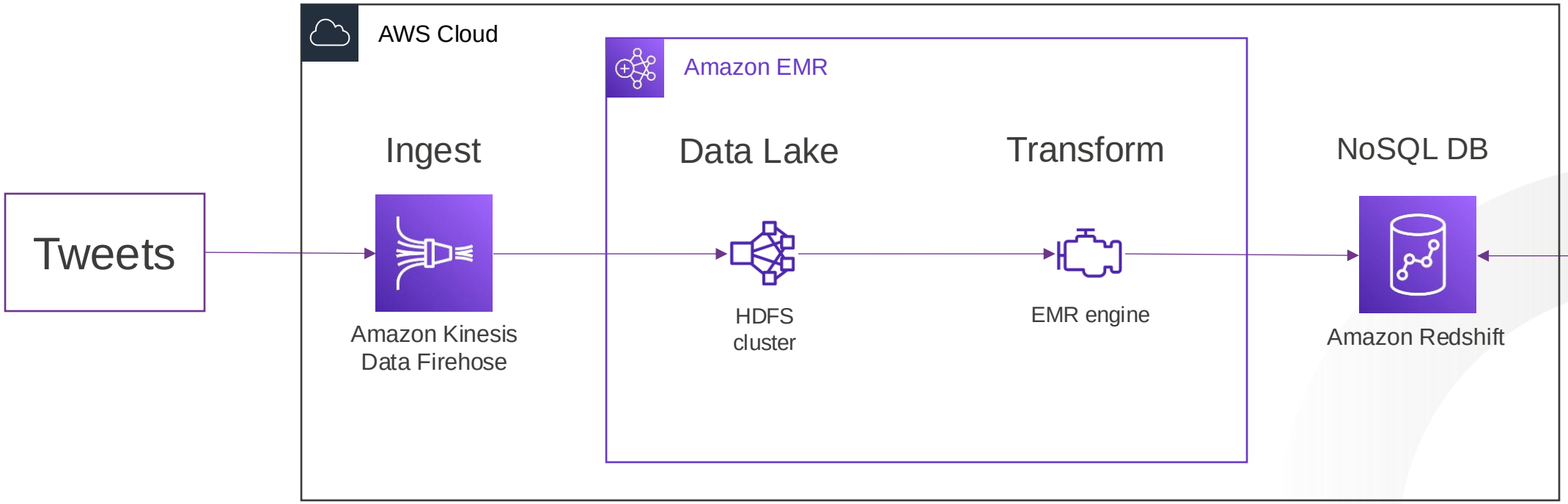
Cloud architecting

Generic cloud X Arch…



Cloud architecting

AWS Twitter architecture…



Activity

Cloud architecting

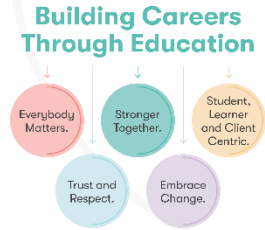
1. To convey useful information, all architectural diagrams must be fully detailed. True or false? Why (or why not)?

2. Architecture diagrams are also used for “traditional” data systems. What might differentiate a big data architectural diagram from a “small” data architectural diagram?

3. **Extension:** In the examples shown so far, we have only used abstract arrows to connect services, without any further details. What useful information could be attached to those arrows?

Submit your responses to the chat!

Activity



Cloud architecting

The following slides will present examples of big data architecture diagrams provided by various CSPs. For each:

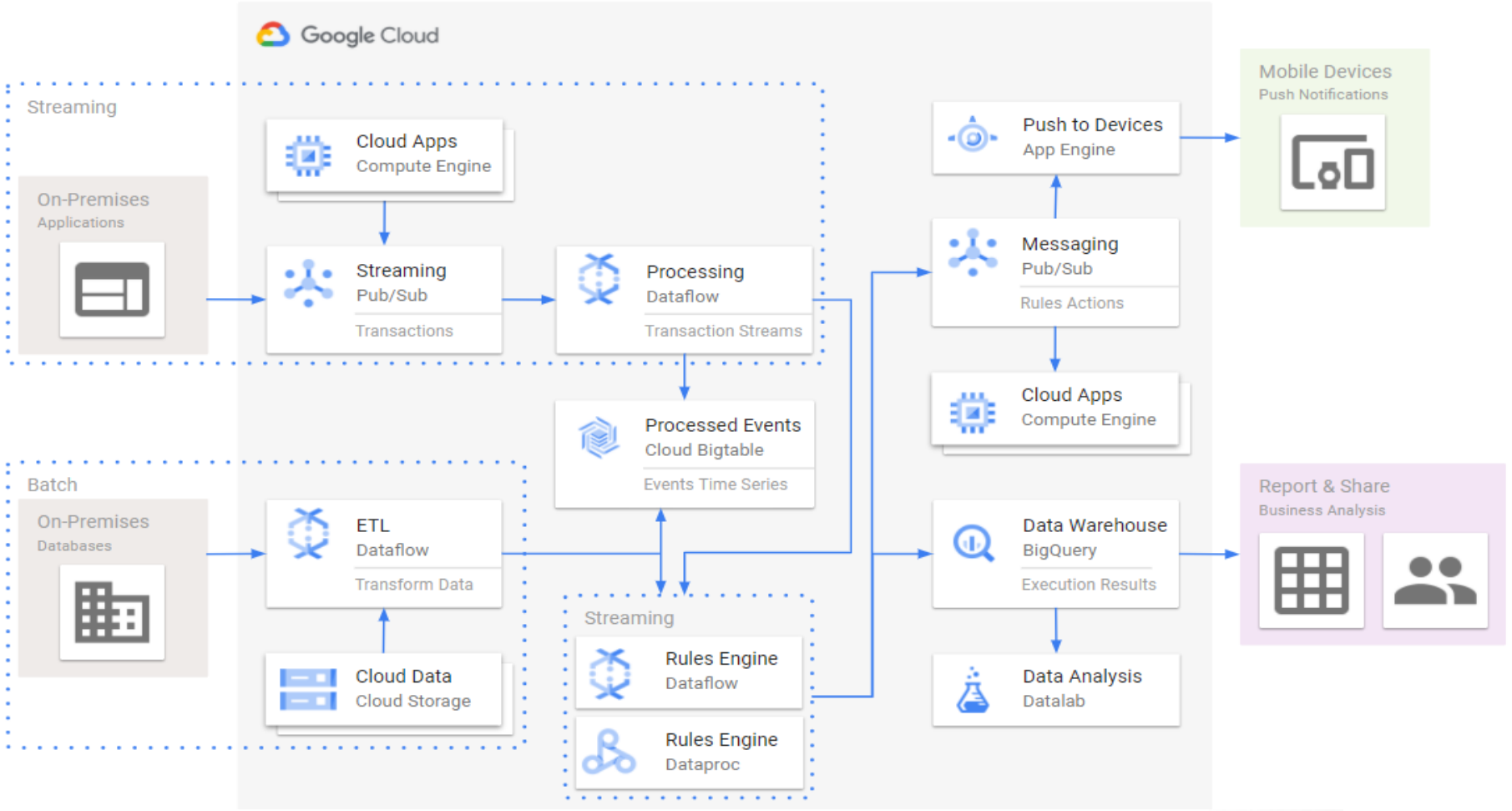
1. What data is being ingested?
2. Where in the cloud is the data being stored?
3. What are the key cloud services being used?
4. Is a useful architectural diagram? Why or why not?

Submit your responses to the chat!

BPP

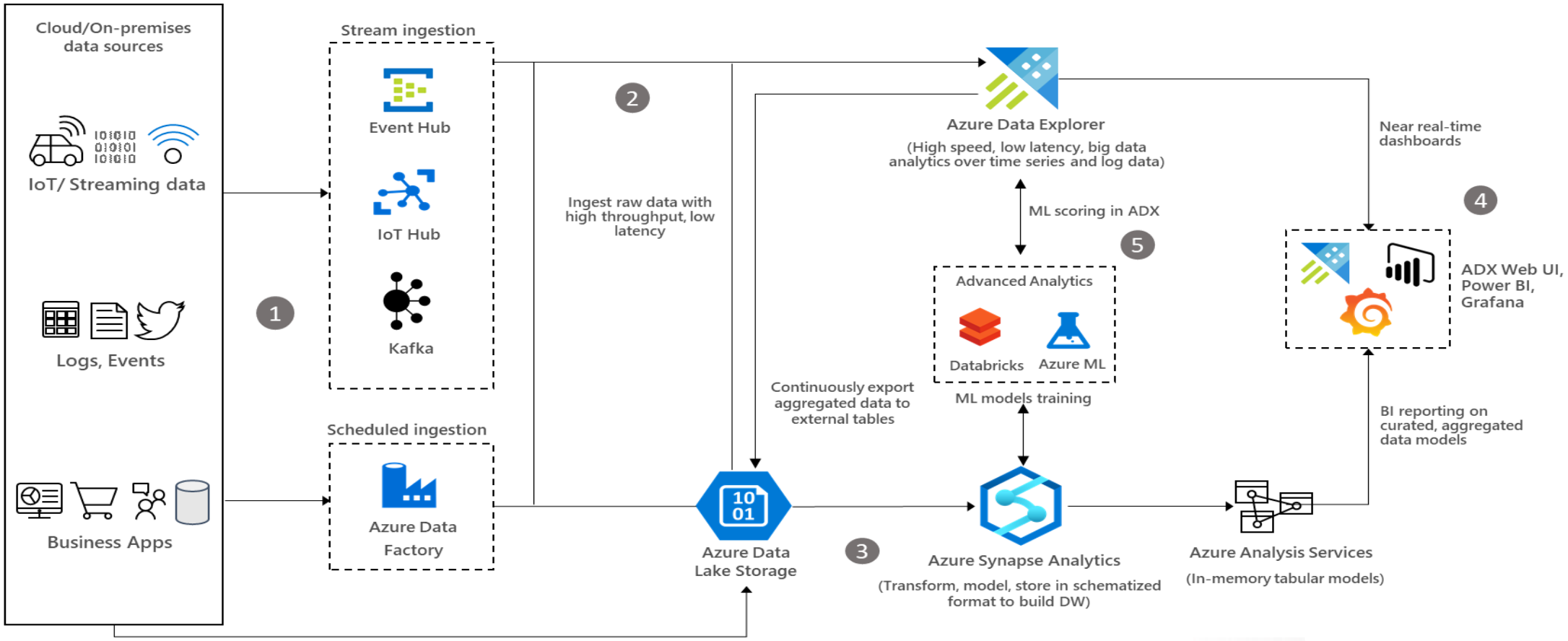
Cloud architecting

Google Cloud…



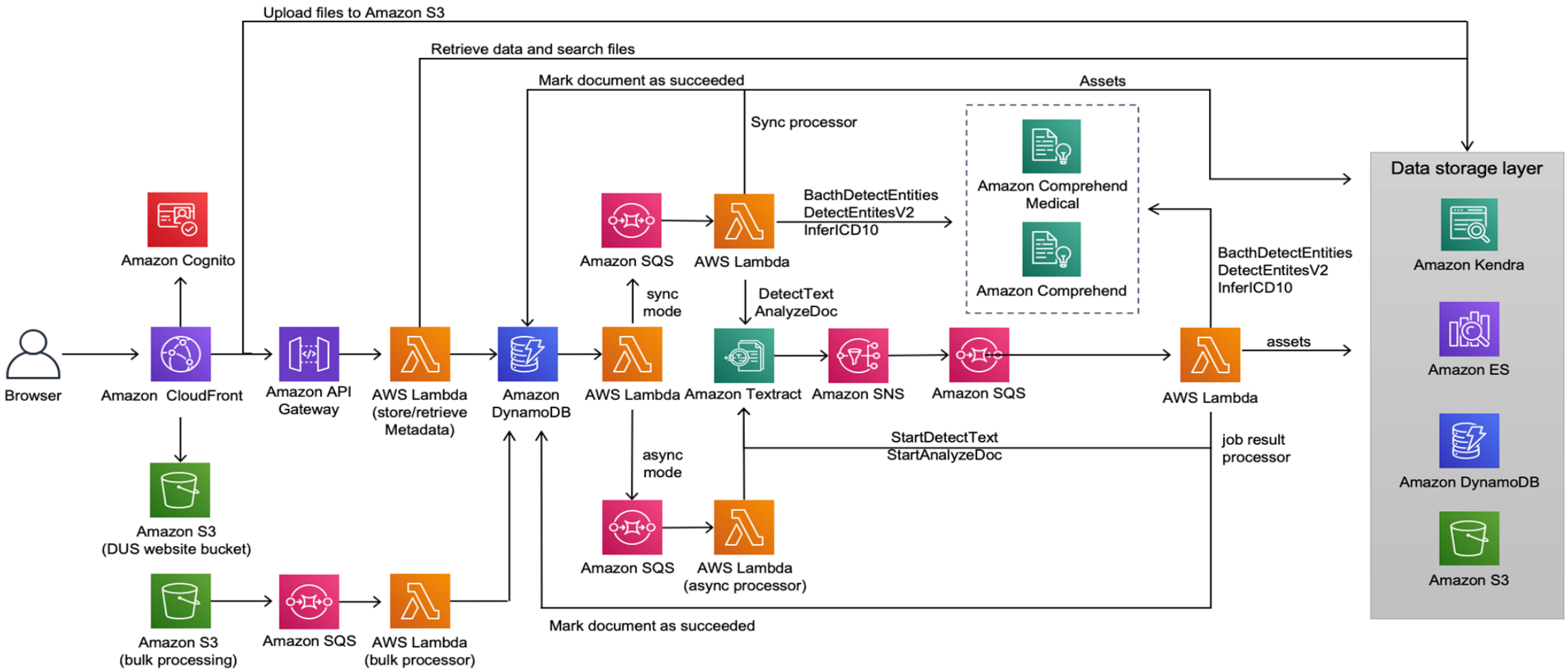
Cloud architecting

Azure Cloud…



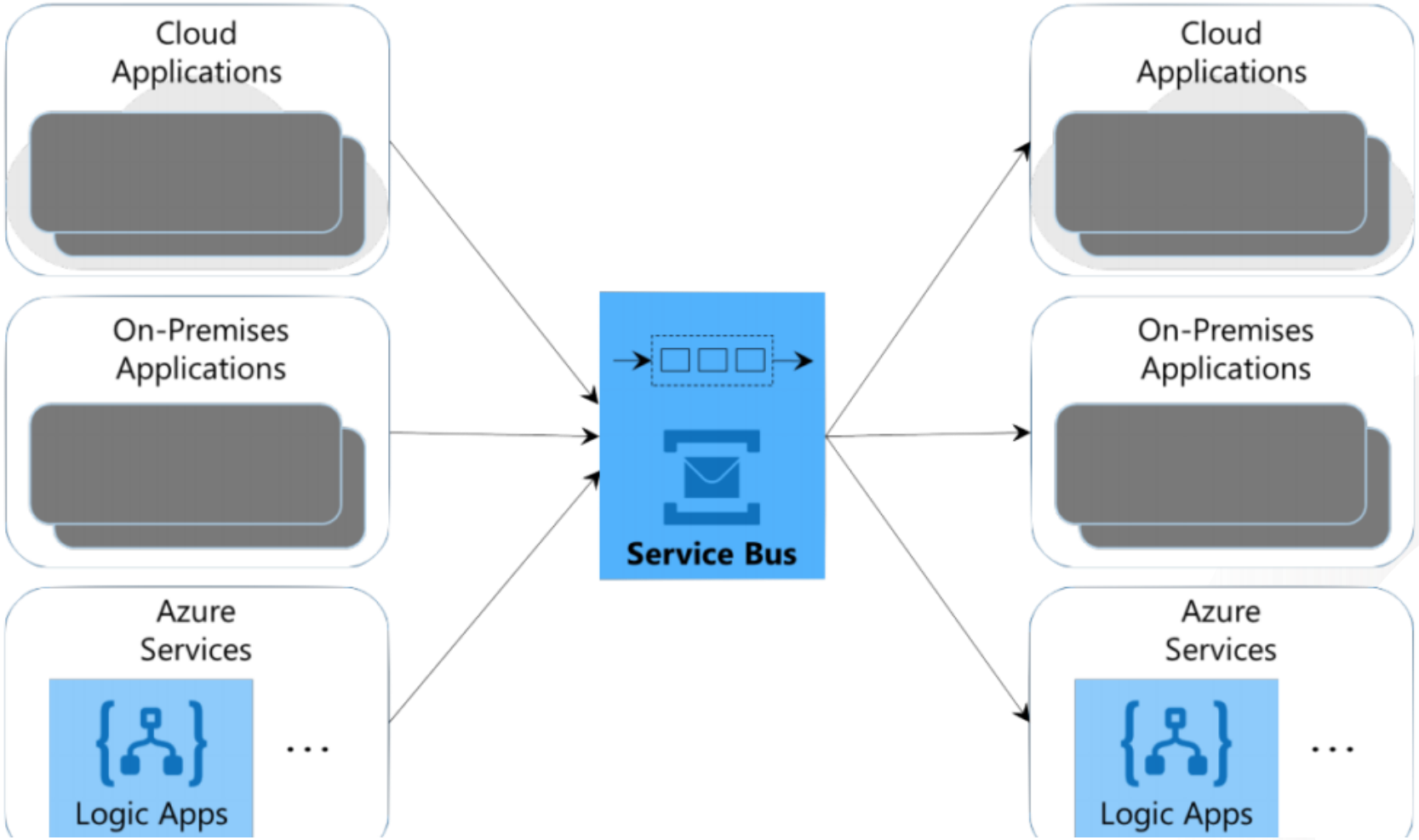
Cloud architecting

AWS Cloud…



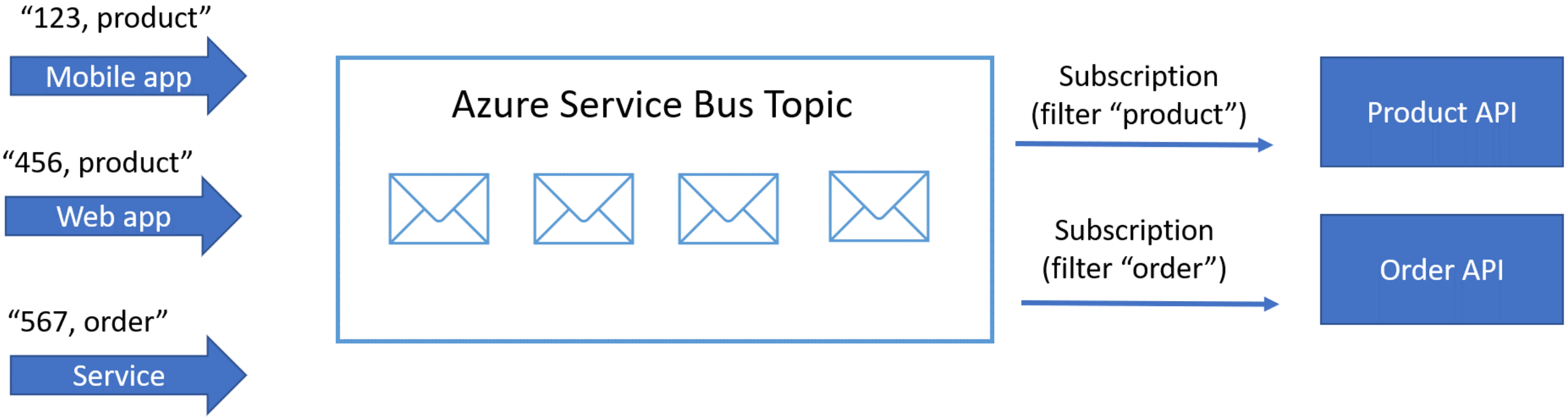
Cloud architecting

Azure Cloud…



Cloud architecting

Azure Cloud…



Discussion – EPA Prep

Why integrate On-Premise with Cloud?

Potential answers:

- Preserve existing investments.
- Handle sensitive data on-premises.
- Gradual migration to cloud.

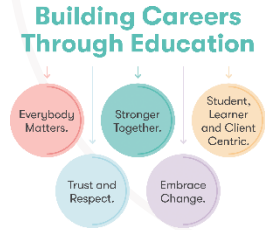
Potential Challenges and Considerations:

- Secure connectivity (VPNs, ExpressRoute).
- Data synchronisation methods.
- Unified identity management.

Submit your responses to the chat!

BPP

Discussion – EPA Prep



Why integrate On-Premise with Cloud?

Question: **Why don’t we implement integrations manually?** (remember Topic 3)

Managing infrastructure manually leads to:

- **Inconsistencies** between environments.
- **Slow deployment** processes.
- **Higher risk** of errors and outages.

It is like building a skyscraper without blueprints—chaotic and error-prone.

There's a need for a structured, repeatable approach to infrastructure management: IaC.

Submit your responses to the chat!

BPP

IaC scenario

A financial services firm automates cloud resource provisioning…

Action: Infrastructure is set up at **7 am** and torn down at **7 pm** daily using IaC.

Benefits:

- **Cost Reduction:** 40% reduction in cloud infrastructure costs.
- **Energy Consumption:** 35% decrease in energy usage.
- **Deployment Speed:** Provisioning time reduced from hours to minutes.

Sustainability Impact:

- Supports **net-zero** objectives by minimising energy consumption.

- Reduces carbon footprint associated with cloud computing

BPP

IaC scenario

Integration and E2E (end-to-end) Testing in Cloud Environments…

Having implemented and automated a cloud integration, we now need to ensure different services and components work together seamlessly.

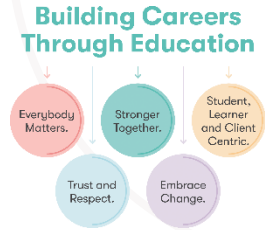
Importance in Cloud:

- Especially important with complex microservices architectures.
  - Check that the microservices can “talk to each other”
- Third-party integrations.
  - Check end-to-end integration.

Benefits:

- Integration tests, and E2E tests, can be automated.
- This leads to early detection of issues.
- Improved reliability and performance.
- Supports continuous integration/continuous deployment (CI/CD) pipelines.

Discussion – EPA Prep



Why are open-source standards important?

- Promotes interoperability and flexibility.
- Reduces vendor lock-in.

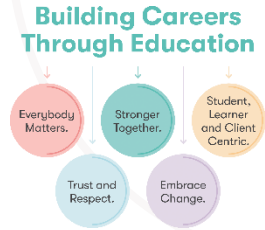
Benefits:

- **Innovation:** Collaborative community improvements.
- **Cost Savings:** Avoid proprietary licensing fees.
- **Transparency:** Enhanced security and trust.

Submit your responses to the chat!

BPP

Discussion – EPA Prep



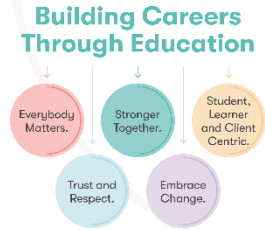
Explain the benefits of using comprehensive third-party tools like Snowflake and Databricks to integrate and manage data across cloud platforms.

- Central data management --> consistency, ease of management
- Automatic scaling without degradation in performance
- Encourages a data-driven culture by making complex analytics accessible to users with varying skill levels.
- Flexible pricing and lack of up-front infrastrucure investment

Submit your responses to the chat!

BPP

Discussion – EPA Prep



Explain how you would integrate visualisation tools (like Power BI) with a cloud data product.

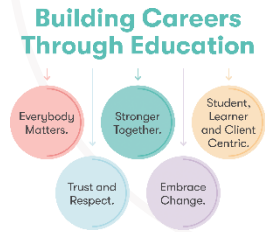
- **Connectivity:** Establish direct connectivity using native connectors
- Utilise APIs (for pushing or pulling data)
- Streamline datasets for maximum performance
- Remember about security: encryption, etc.
- Follow all necessary policies and regulation (GDPR)

Cache frequent queries and visual elements to improve responsiveness and user experience.

Submit your responses to the chat!

BPP

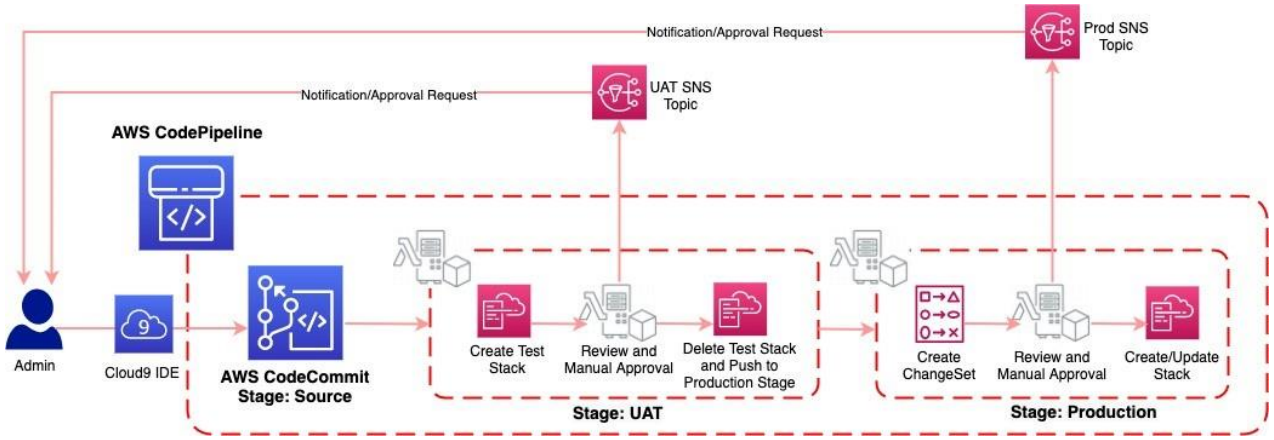
Methods for infrastructure automation



Environment deployment…

The NEW - Infrastructure as code:

- Consistent servers between environments
- Environments created or scaled easily
- Fully automate creation and updates of environments
- Transition to immutable infrastructure
- Use blue/green deployments



The OLD - Manual deployment:

- Deployment steps vary by environment
- More verification steps and more elaborate manual processes
- Increased documentation to account for differences
- Deployment on weekends to allow time to recover from errors

Slower release cadence to minimize pain and long weekends

BPP

Methods for infrastructure automation

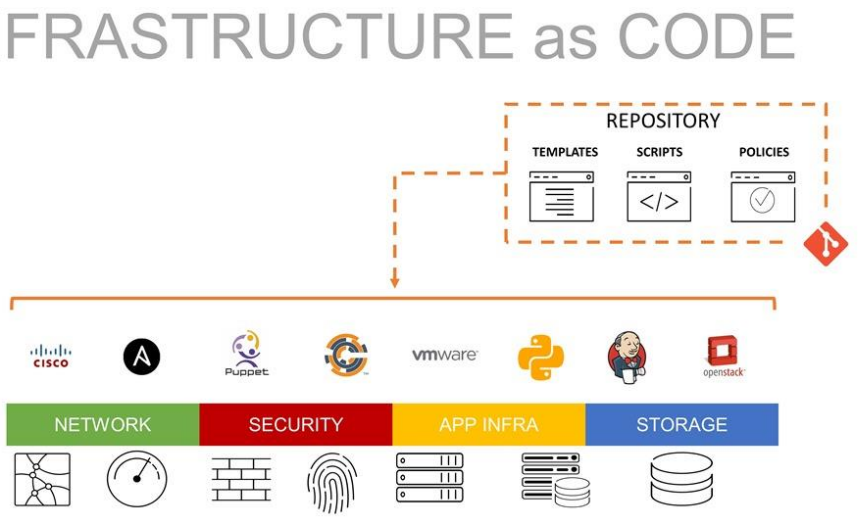
Environment configuration…

Manual configuration:

- Configuration bugs difficult to Identify
- Error prone
- More verification steps and more elaborate manual processes
- Increased documentation
- Deployment on weekends to allow time to recover from errors
- Slower release cadence to minimize requirement for long weekends

Configuration as code:

- Bugs easily reproducible
- Consistent configuration
- Increase deployment cadence to reduce amount of incremental change
- Treat environment and configuration as executable documentation



Infrastructure options for deployment

Imperative versus declarative configuration…

Approaches to implementing infrastructure and configuration as code

Declarative:

- Functional
- Defines **what** the final state should be

Imperative:

- Procedural
- Defines **how** to achieve that final state

Different platforms support different, and often multiple, file formats, such as YAML, JSON, and XML.

BPP

Infrastructure options for deployment

Imperative versus declarative configuration…

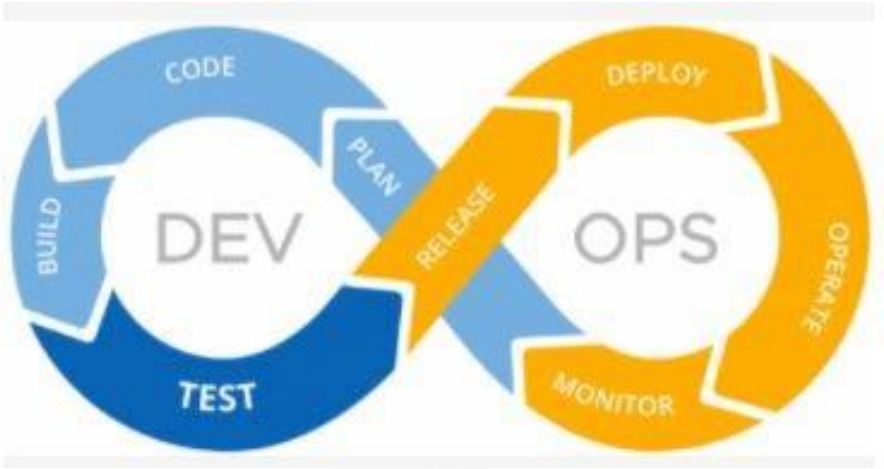
Idempotence – definition:

- Mathematical term used in the context of infrastructure and configuration as code

- Ability to apply one or more operations against a resource, resulting in the same outcome

To attain idempotence:

- Automatically configure and reconfigure an existing set of resources, or
- Discard existing resources and spin up a fresh environment



BPP

Modularising template

Best practice: Modularise templates into individual components:

- Use linked templates to break the solution into individual pieces
- Reuse those elements across different deployments

JSON Copy

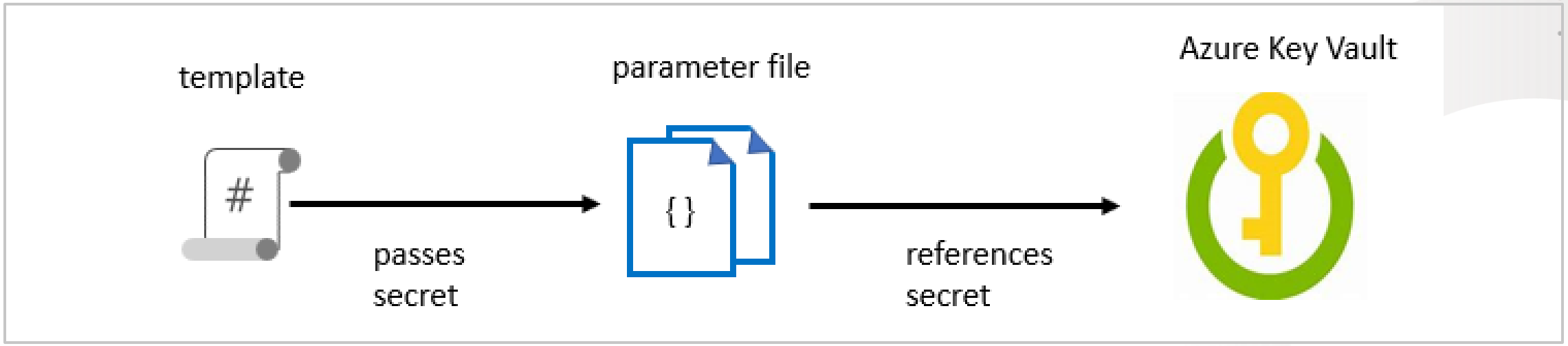
```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {},
  "variables": {},
  "resources": [
    {
      "type": "Microsoft.Resources/deployments",
      "apiVersion": "2020-10-01",
      "name": "linkedTemplate",
      "properties": {
        "mode": "Incremental",
        "templateLink": {
          "uri":"https://mystorageaccount.blob.core.windows.net/AzureTemplates/newStorageAccount.json"
          "contentVersion":"1.0.0.0"
        }
      }
    }
  ],
  "outputs": {
  }
}
```

BPP

Managing secrets in templats

When passing a secure value (e.g., a password) as a parameter during deployment:

- Create a key vault and secret using Azure CLI or PowerShell
- Enable Azure Resource Manager access for template deployment
- Reference the key pair in the parameter file, **not** the template
- Enable access to the secret. **Owner** and **Contributor** roles grant access
- Deploy the template and pass in the parameter file



What is Azure CLI?

Command-line program to connect to Azure (Azure Cloud Shell, PowerShell, or Bash)…

Execute administrative commands on Azure resources through a terminal, command-line prompt, or script, instead of a web browser:

For example, to restart a VM use the command:

```
az vm restart -g MyResourceGroup -n MyVm
```

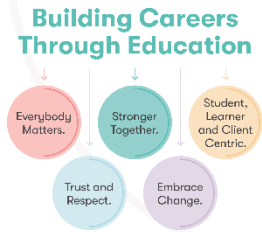
Can be installed on Linux, macOS, or Windows computers, and added as a module to PowerShell

Can be used interactively or scripted:

- *Interactive*: Issue commands directly at the shell prompt
- *Scripted*: Assemble the CLI commands into a shell script and then execute the script

BPP

What is Azure CLI?



Commands in the CLI are structured in groups and subgroups…

- Use az find to find commands you need
  az find blob

- Use the help argument to get more detail about the command
  az storage blob –help

Creating a new Azure resource typically involves the following process:

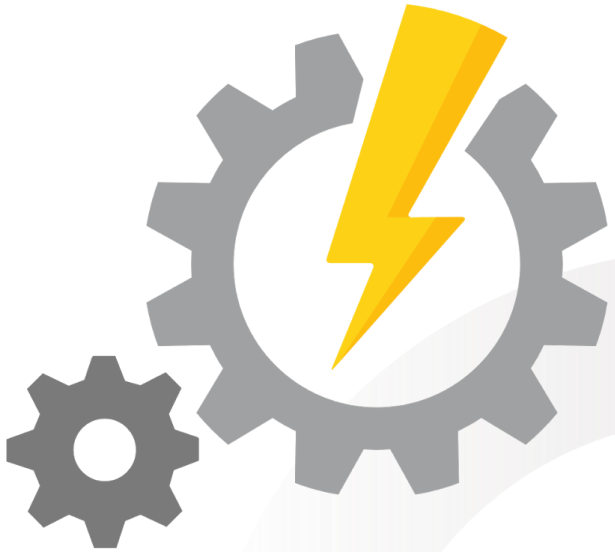
BPP

What is Azure Automation?

An Automation service integrated with Microsoft Azure for automating and simplifying the creation, deployment, monitoring and maintenance of Azure resources and resources external to Azure.

Azure Automation Capabilities include:

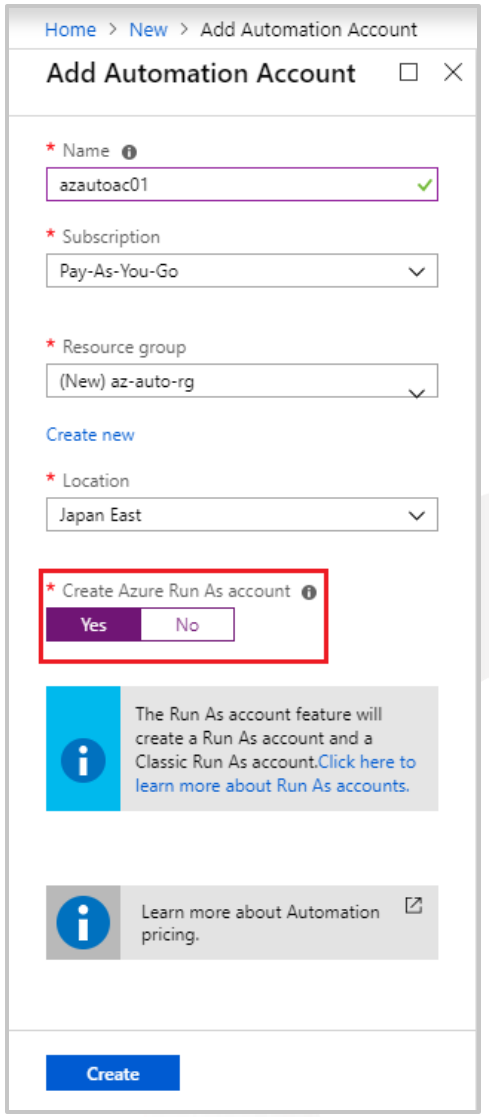
- Manage Shared resources
- State configuration
- Integration with GitHub, Azure DevOps Git/TFVC
- Update management
- Can automate Windows or Linux environments
- Can apply to any system that exposes an API over internet protocols



BPP

Automation accounts

1. To use Azure Automation you must create an Automation account
2. Automation account acts as a container in which you store, manage and use automation artifacts
3. Provides a way to separate your environments or further organize your Automation workflows and resources.
4. Requires subscription-owner level access as provides access to all Azure resources via an API
5. Need at least one automation account but should have multiple for access control
6. Run As account:
7. Creates a service principal in Azure AD which allows access to Azure resources when running automation



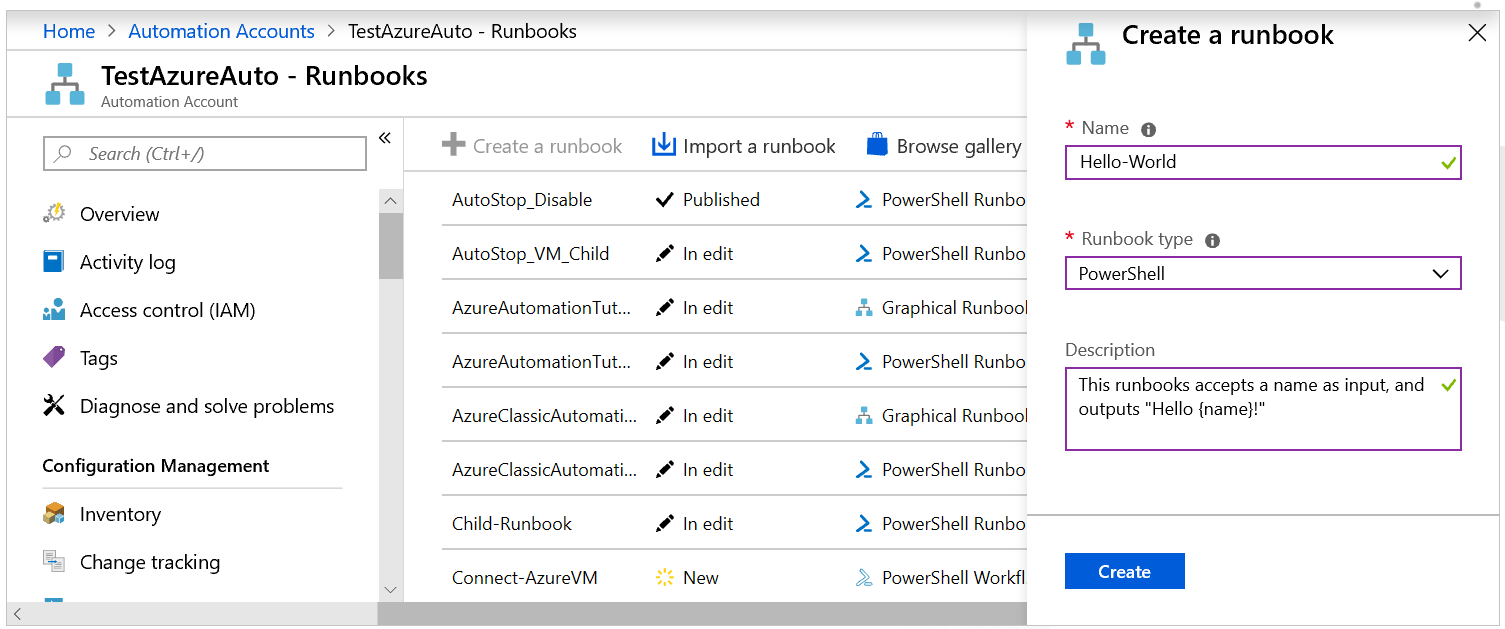
What is a runbook?

A *runbook* is a set of tasks that perform some automated process in Azure Automation.

Runbooks serve as repositories for your custom scripts and workflows
Can create your own or import and modify from community via Runbook Gallery.

Runbook Types available:

- Graphical runbook
- PowerShell runbooks
- PowerShell Workflow runbooks
- Python runbooks



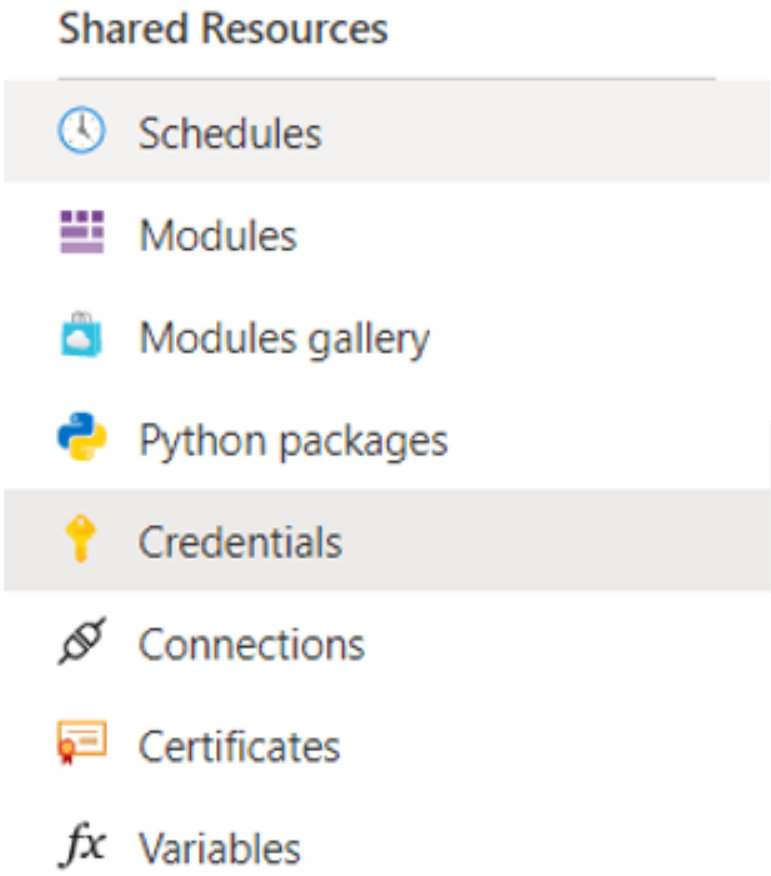
BPP

Automation shared resources

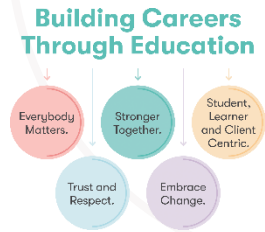
Azure Automation contains shared resources that are globally associated available to be used in, or with a runbook:

Currently Eight Categories:

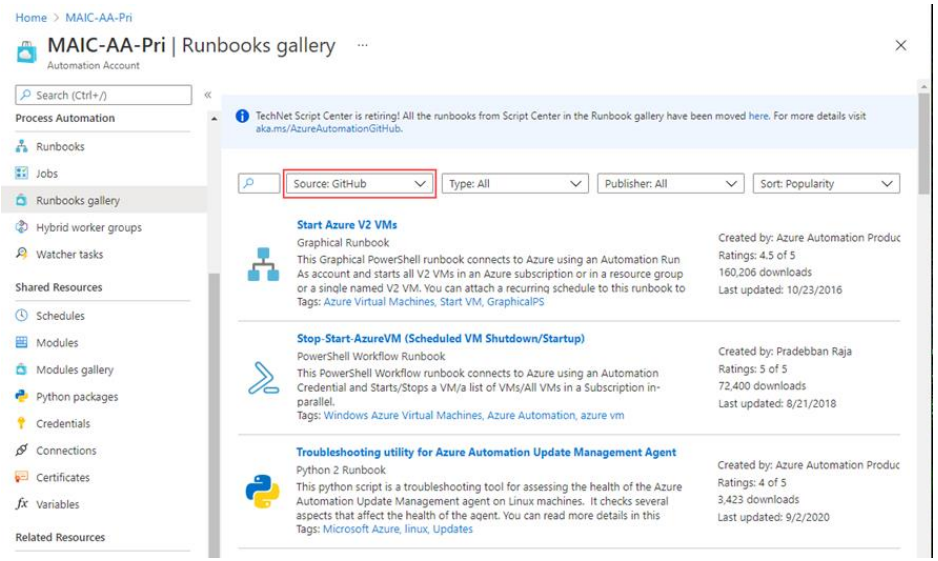
- Schedules
- Modules
- Modules gallery
- Python 2 packages
- Credentials
- Connections
- Certificates
- Variables



Runbook gallery



- Can import pre-existing runbooks from the runbook gallery at the Microsoft Script Center
- Runbooks provided to help eliminate the time it takes to build custom solutions
- Already been built by Microsoft and the Microsoft community
- Can be used with or without modification
- Can review the code or a visualisation of the runbook code on the gallery as well as see source projects, rating, etc.



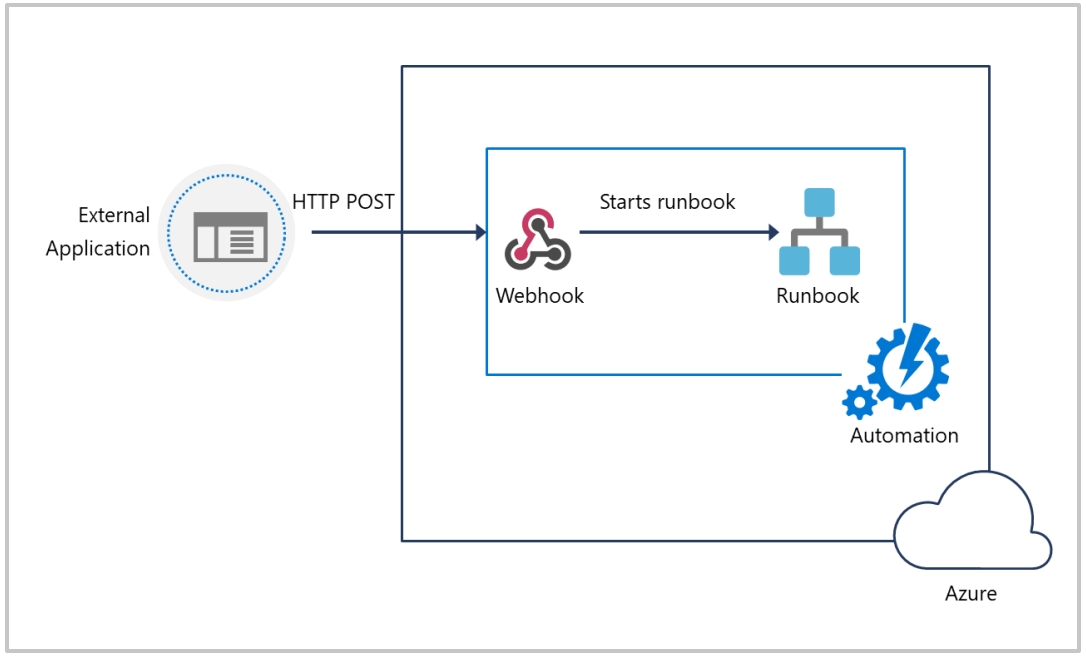
Considerations:

- Python runbooks are also available from the script center gallery. To find them, filter by language and select Python
- You cannot use PowerShell to import directly from the Runbook Gallery

BPP

Webhooks

- Automate the process of starting a runbook either by scheduling it, or by using a webhook.
- Uses a HTTPS request to start a runbook
- Reduces complexity and allows external services such as Azure DevOps, GitHub, or custom applications to use webhooks
- Webhook Syntax: http://< Webhook Server >/token?=< Token Value >



Source control integration

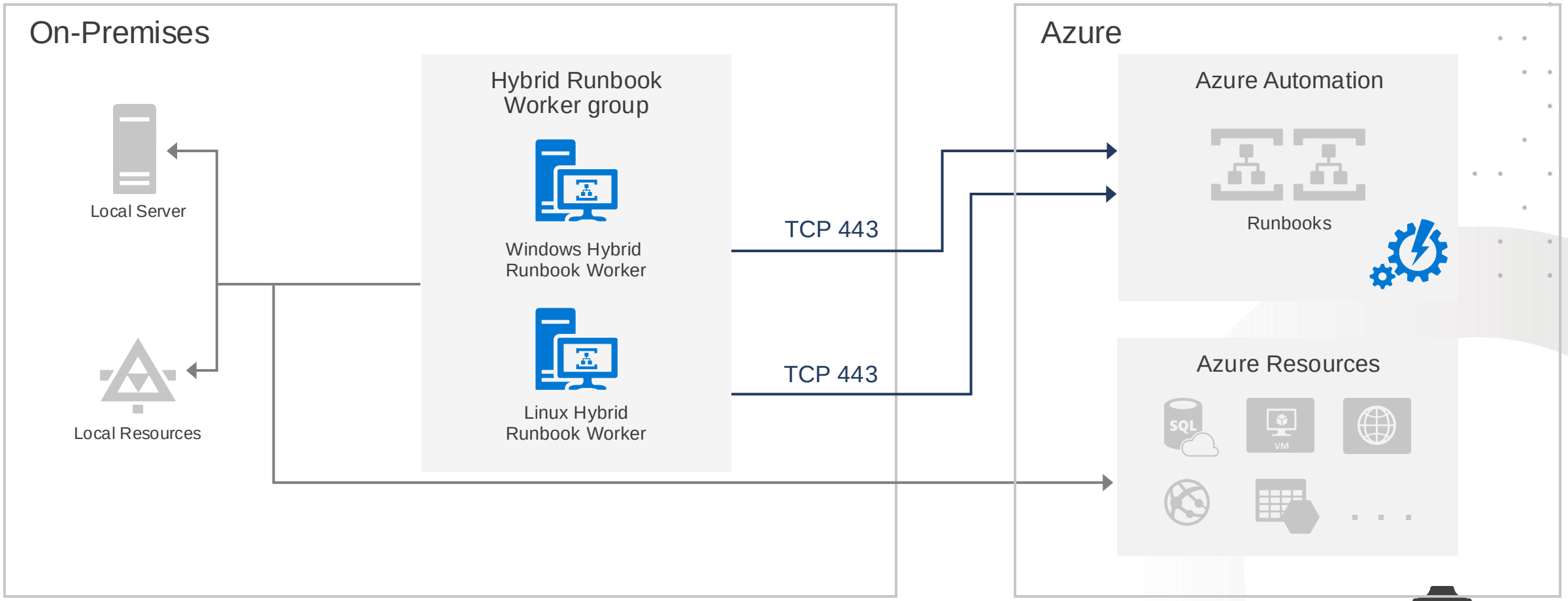
Azure Automation supports the following source Control options:

- GitHub
- Azure DevOps (Git)
- Azure DevOps (TFVC)

- Azure Automation supports source control integration
- Easier collaboration
- Increased auditing and traceability
- Roll back to earlier versions of your runbooks
- Can push code from Azure Automation to source control or pull your runbooks from source control to Azure Automation

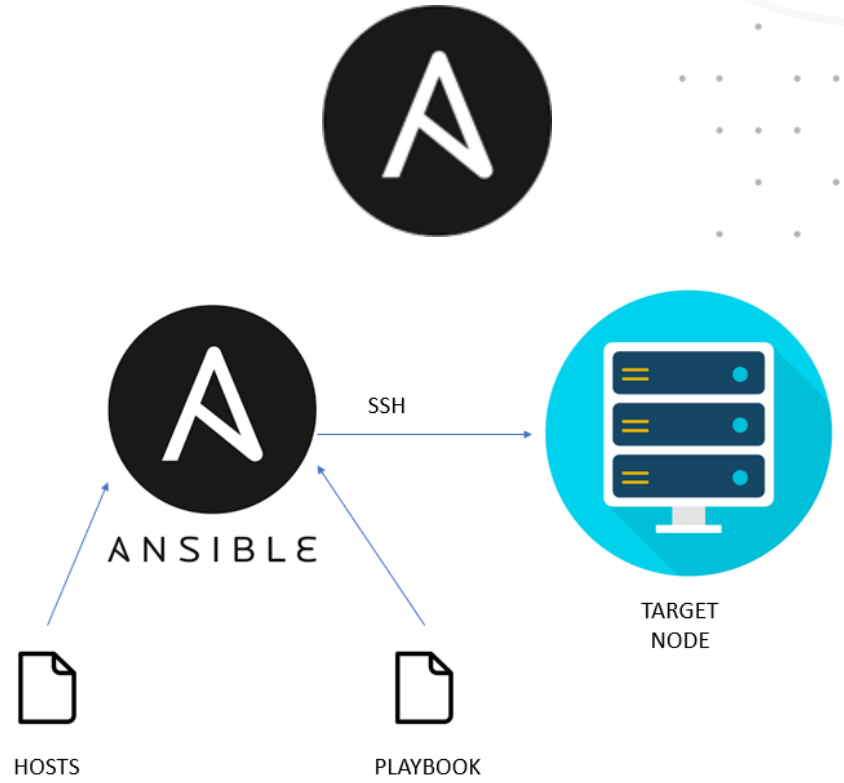
BPP

Hybrid management



Ansible

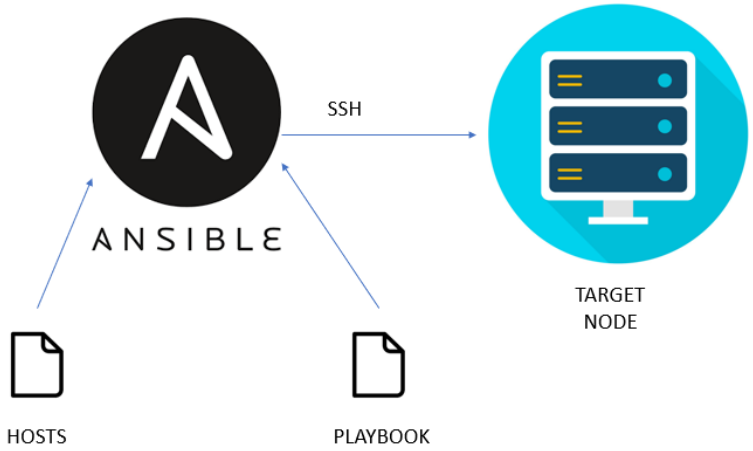
- Ansible is an open-source platform that automates cloud provisioning, configuration management, and application deployments
- Allows you to automate deployment and configuration of resources in your environment such as virtual networks, storage, subnets, and resources groups
- Unlike Puppet or Chef, Ansible is agentless, so you do not have to install software on the managed machines
- Ansible also models your IT infrastructure by describing how all of your systems interrelate, rather than managing just one system at a time



Ansible components

Some of the core components of Ansible include:

- **Control Machine.** This is the machine from which the configurations are run
- **Managed Nodes.** These are the devices and/or machines and environments that are being managed
- **Playbooks.** Playbooks are ordered lists of tasks, written in YAML, that have been saved so you can run them in the same order repeatedly
- **Modules.** Ansible works by connecting to your nodes and then pushing out to the node's small programs (or *units of code*), called *modules*. *Modules* are the units of code that define the configuration. They are modular, and can be re-used across playbooks



BPP

Installing Ansible

To enable a machine to act as the control machine from which to run playbooks, you need to install both Python and Ansible

Python:

Must install either Python 2 (version 2.7), or Python 3 (versions 3.5 and higher)

Ansible:

Only need to install Ansible on one machine, which could be workstation or a laptop—you can manage an entire fleet of remote machines from that central point. Can be Linux, macOS or Windows

No database is installed as part of the Ansible setup

No daemons are required to start or keep running

BPP

Ansible demo

Demonstration: Run Ansible in Azure Cloud Shell

Create a resource group in Azure using Ansible in Azure Cloud Shell with bash. Azure Cloud Shell, has Ansible pre-installed, so you do not have to install or configure anything to be able or run Ansible

You can complete this walkthrough task by going to https://github.com/MicrosoftLearning/AZ400-DesigningandImplementingMicrosoftDevOpsSolutions

BPP

Terraform

HashiCorp Terraform is an open-source tool that allows you to provision, manage, and version cloud infrastructure

Create a resource group in Azure using Ansible in Azure Cloud Shell with bash. Azure Cloud Shell, has Ansible pre-installed, so you do not have to install or configure anything to be able or run Ansible

You can complete this walkthrough task by going to https://github.com/MicrosoftLearning/AZ400-DesigningandImplementingMicrosoftDevOpsSolutions

BPP

Terrarform components

Some of the core components of Terraform include:

Configuration files – Text-based configuration files allow you to define infrastructure and application configuration, and end in the .tf or .tf.json extension

Terraform CLI – A command-line interface from which you run configurations. You can run command such as Terraform apply and Terraform plan, along with many others

Modules – Self-contained packages of Terraform configurations that are managed as a group

1. **Write** - Author infrastructure as code.
2. **Plan** - Preview changes before applying.
3. **Apply** - Provision reproducible infrastructure.

BPP

HCL


```
resource "aws_lambda_function" "my_lambda" {
  filename      = "lambda_function_payload.zip"
  function_name = "lambda_function_name"
  role          = "${aws_iam_role.iam_for_lambda.arn}"
  handler       = "exports.main"
  source_code_hash = "${filebase64sha256("lambda_function_payload.zip")}"
  runtime = "nodejs12.x"
}
```

Terrarform on Azure and AWS

Options for deploying Terraform in Azure include:

Azure Marketplace:

Offers a fully-configured Linux image containing Terraform

Azure virtual machines:

Deploy a Linux or Windows VM in Azure VM's IaaS service, install Terraform and the relevant components, and then use that image

Azure Cloud Shell:

Terraform is installed by default in Azure Cloud Shell

Terrarform on Azure and AWS

Options for deploying Terraform in Azure include:

Azure Marketplace:

Offers a fully-configured Linux image containing Terraform

Azure virtual machines:

Deploy a Linux or Windows VM in Azure VM's IaaS service, install Terraform and the relevant components, and then use that image

Azure Cloud Shell:

Terraform is installed by default in Azure Cloud Shell

BPP

Installing Terraform

Options for deploying Terraform in Azure include:

You must install Terraform on the machine from which you are running the Terraform commands.

Can be installed on Windows, Linux or macOS environments

You can download appropriate Terraform install package from https://www.terraform.io/downloads.html

Authenticate Terraform with Azure using:

The Azure CLI

A Managed Service Identity (MSI)

A service principal and a client certificate or secret

BPP

Demonstration

Run Terraform in Azure Cloud Shell…

Create a create a resource group in Azure using Terraform in Azure Cloud Shell

You can complete this demonstration task by following the steps outlined in the course, or you can simply read through them, depending on your available time

BPP

Key Learning Summary

The key takeaways from this session are as follows:

- **Infrastructure as Code (IaC):** IaC involves managing and provisioning infrastructure through machine-readable definition files, enhancing consistency, efficiency, and reducing errors compared to manual configurations

- **Integration Testing:** Critical in cloud environments, integration testing ensures that different components of an application work together correctly, reducing the risk of deployment failures and downtime.

- **Hybrid Cloud Solutions:** Combining public and private clouds allows organisations to leverage the benefits of multiple cloud providers, optimizing costs, enhancing performance, and increasing resilience.

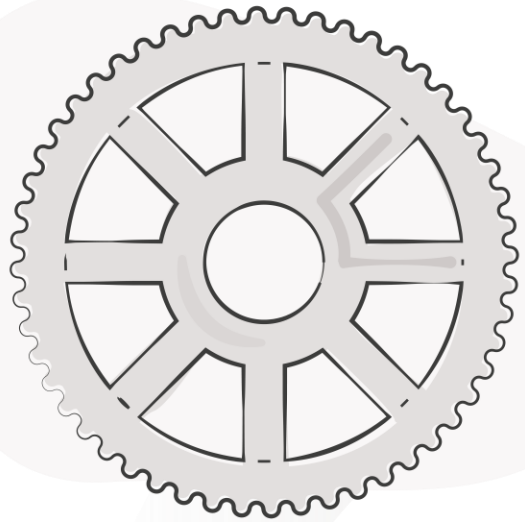
- **Multi-Cloud Management with Terraform:** Terraform enables consistent management of infrastructure across multiple cloud providers, simplifying deployment, reducing errors, and increasing efficiency.

BPP

Post-webinar tasks

Apply…

See document “L5DE M5T6 Activity – Terraform and Azure Pipelines”



BPP

Thank you

Do you have any questions, comments, or feedback?