



databricks®



## **Azure Databricks Automation**

---

- Shamir Charania
- [shamir@keepsecure.ca](mailto:shamir@keepsecure.ca)
- @SleepySecNinja

**IT'S NOT THE SIZE OF YOUR DATA THAT MATTERS**



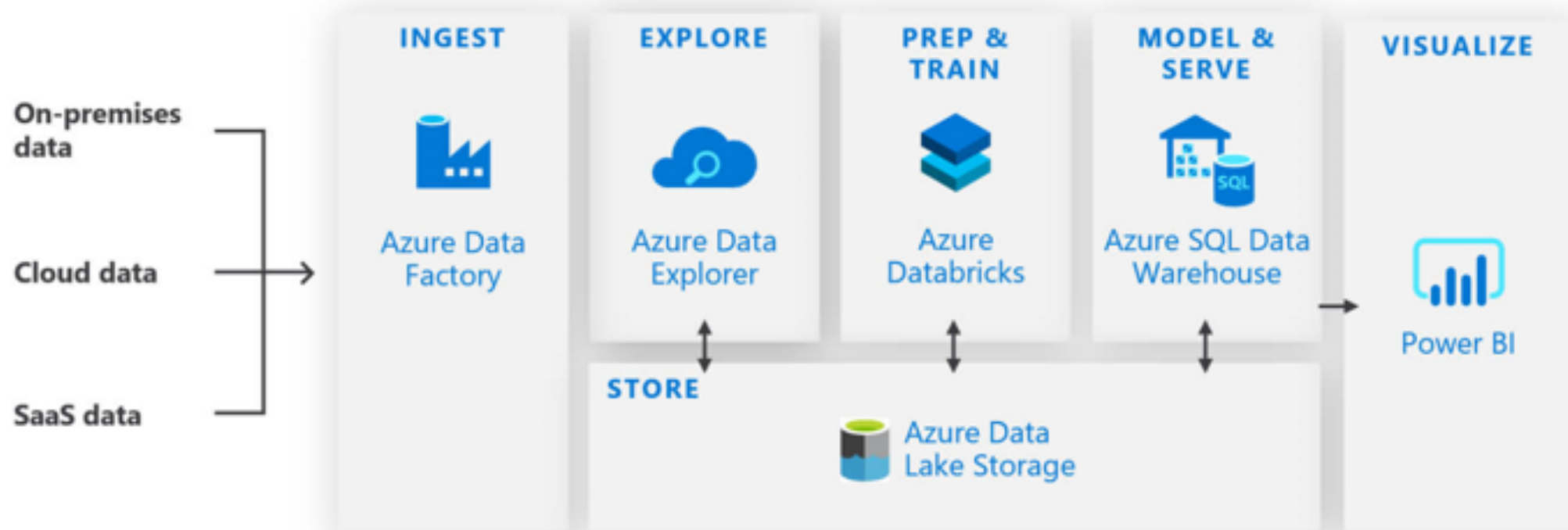
**IT'S HOW YOU USE IT!**

# Agenda

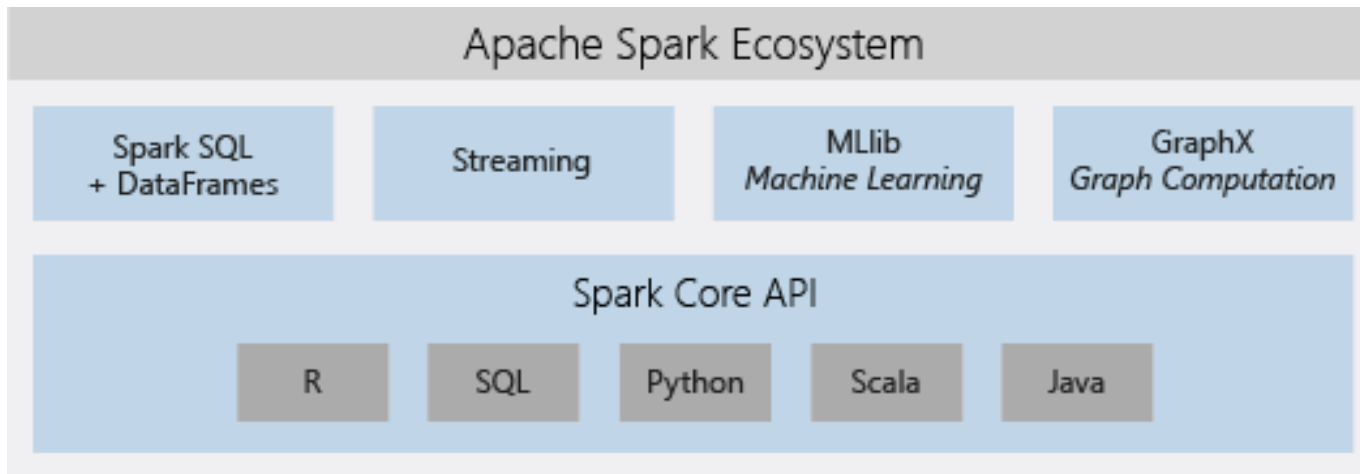
---

1. What is Azure Databricks
2. Key Automation Areas
  - i. Workspaces and Clusters
  - ii. Network Security
  - iii. Databricks Users
  - iv. Permissions
  - v. Integration
3. Some thoughts on Security
4. Questions

# Big Data Pipeline Architecture

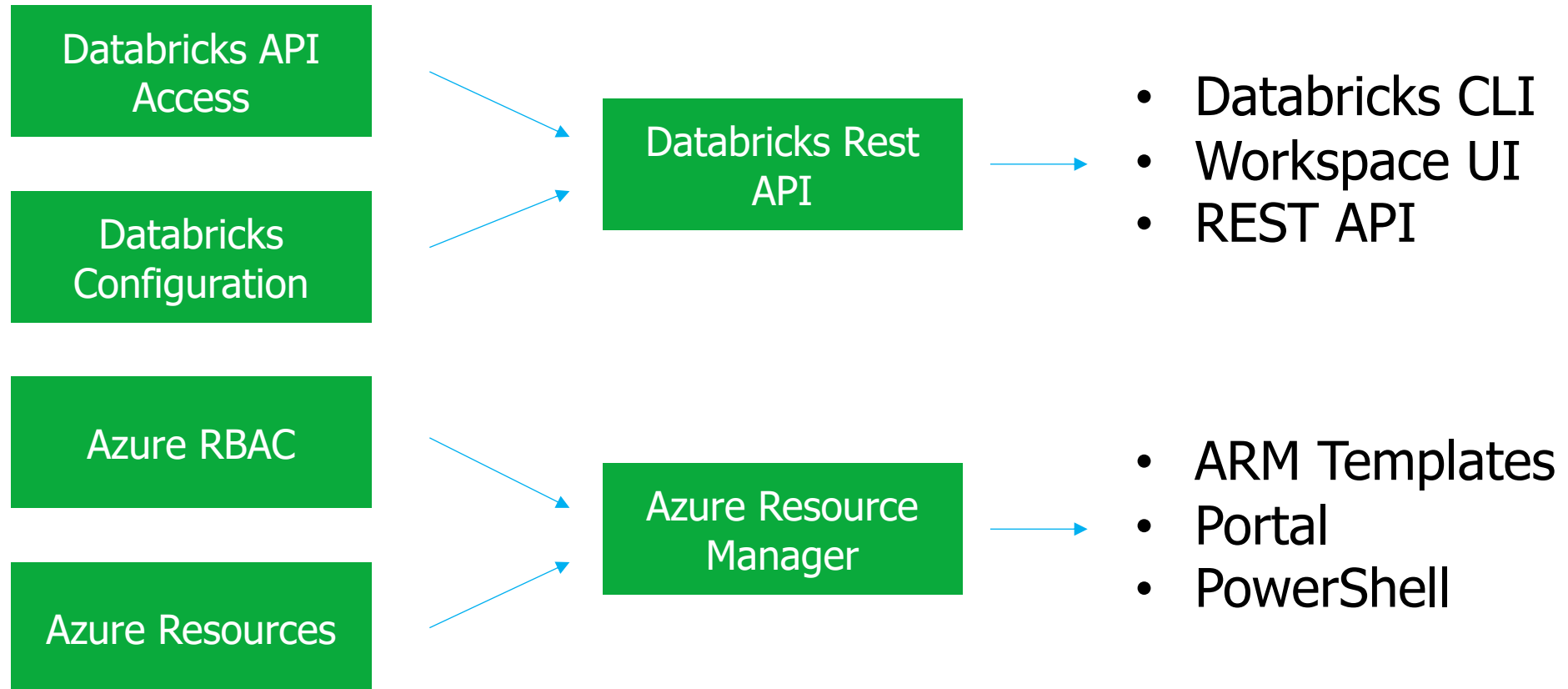


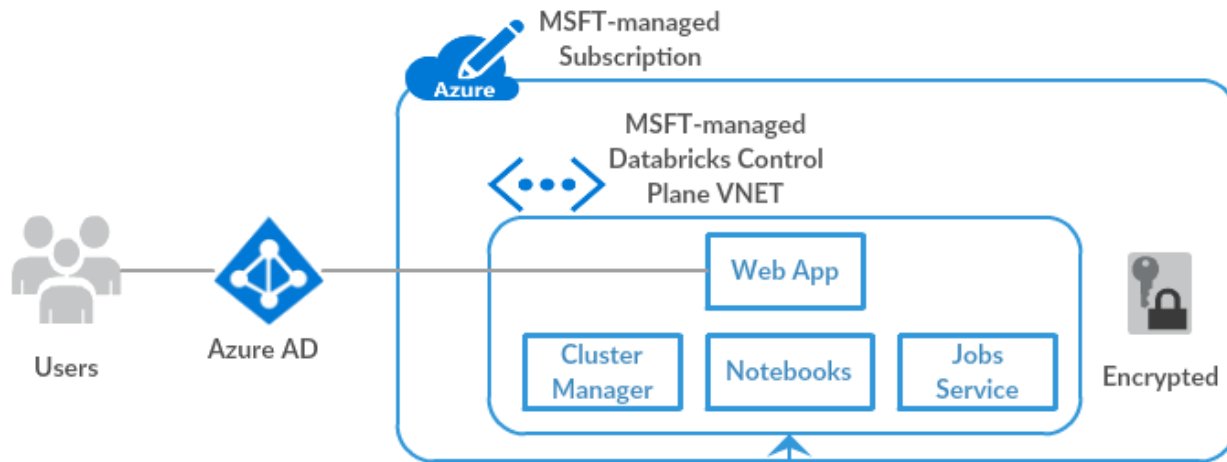
# What is Azure Databricks?



- Fully managed spark clusters
- Interactive workspace for data exploration
- Optimizations for the Azure environment
- Programmatic API access (Jobs)
- Integration with Azure services

# Core Concepts – Configuration Layers

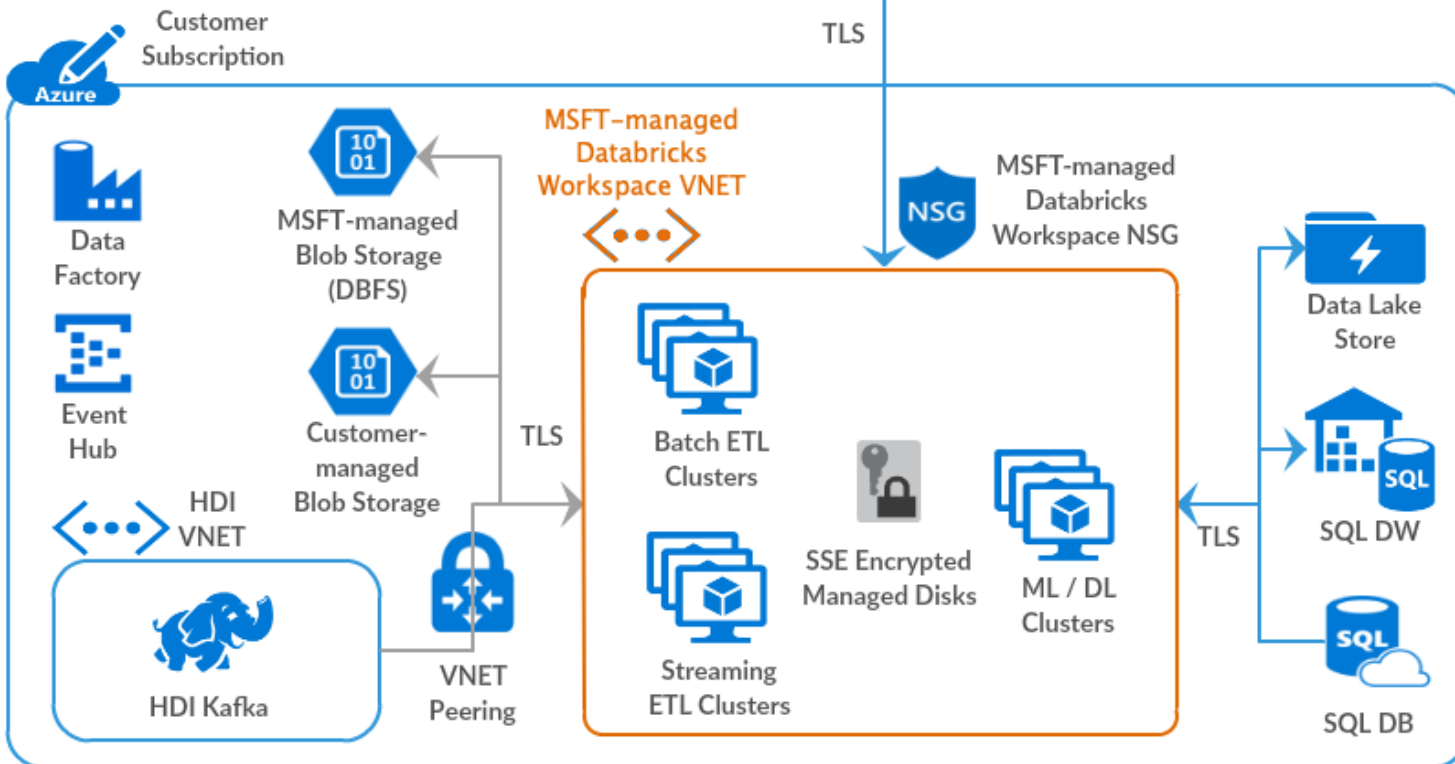




Control Plane

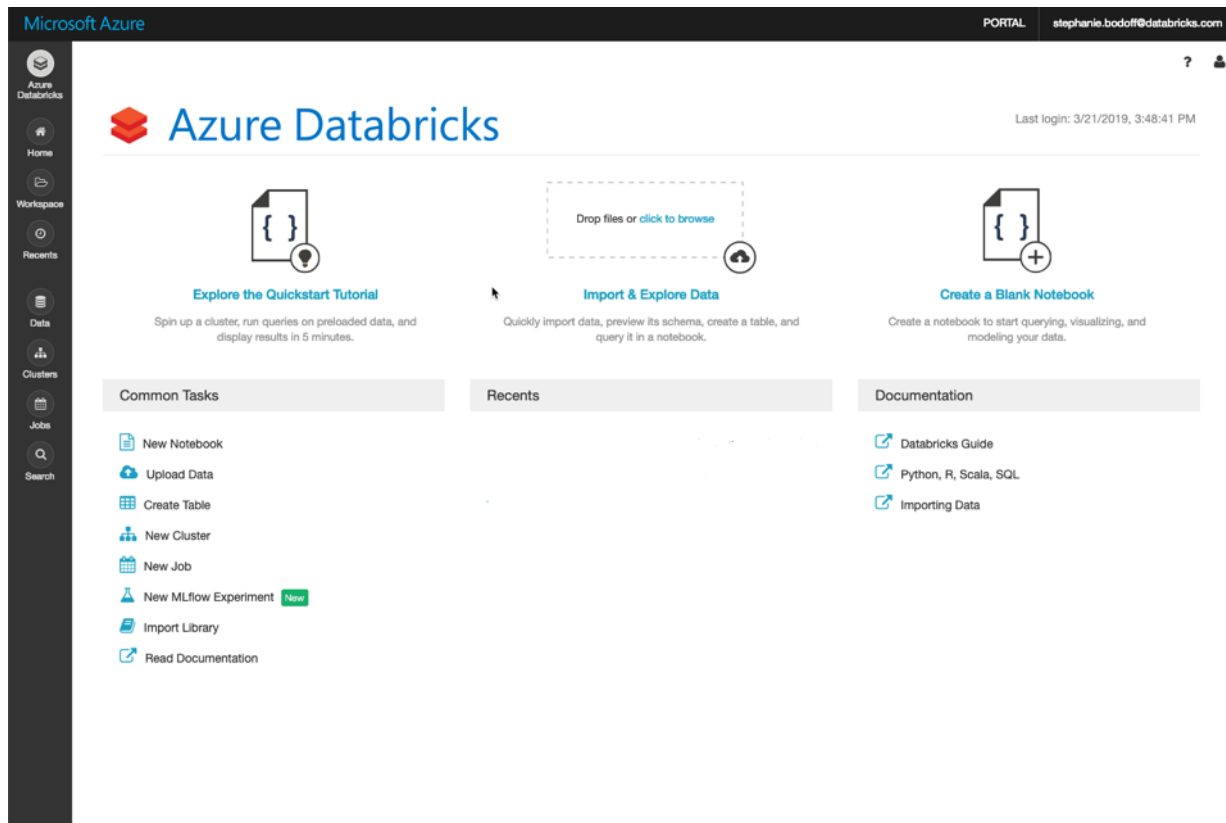
# Azure Databricks – Overall Architecture

Data Plane



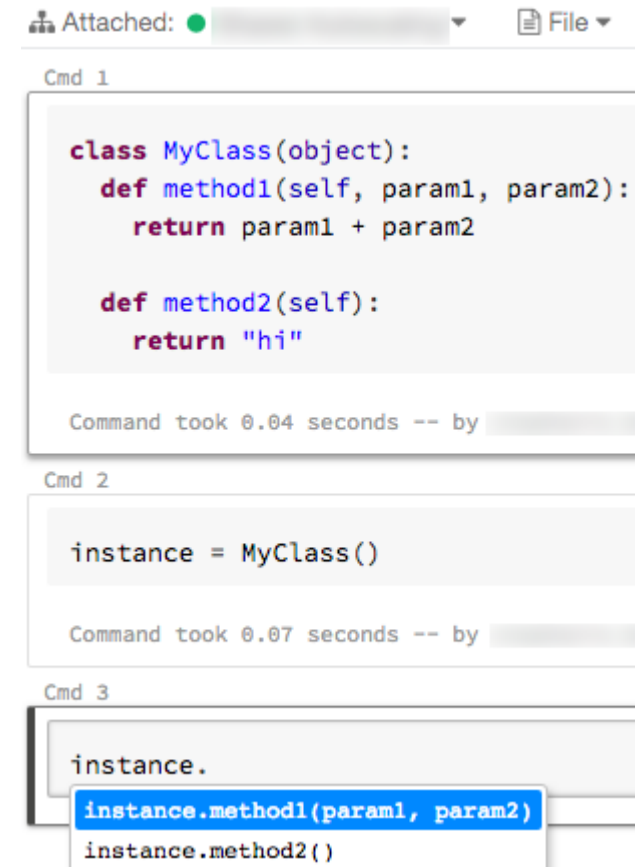
# Core Concepts – Workspaces and Notebooks

## Workspaces



The screenshot shows the Azure Databricks workspace interface. At the top, there's a header with "Microsoft Azure" and "Azure Databricks". Below this, the main area is divided into sections. On the left, there's a sidebar with navigation options: Home, Workspace, Recents, Data, Clusters, Jobs, and Search. The main content area features three large cards: "Explore the Quickstart Tutorial", "Import & Explore Data", and "Create a Blank Notebook". Below these cards, there are three tabs: "Common Tasks", "Recents", and "Documentation". The "Common Tasks" tab is active, showing a list of tasks: "New Notebook", "Upload Data", "Create Table", "New Cluster", "New Job", "New MLflow Experiment", "Import Library", and "Read Documentation".

## Notebooks



The screenshot shows a Databricks notebook interface. At the top, there's a header with "Attached:" and a "File" dropdown. Below this, there are three command blocks, each labeled "Cmd 1", "Cmd 2", and "Cmd 3".

```
Cmd 1
class MyClass(object):
    def method1(self, param1, param2):
        return param1 + param2

    def method2(self):
        return "hi"

Command took 0.04 seconds -- by
```

```
Cmd 2
instance = MyClass()

Command took 0.07 seconds -- by
```

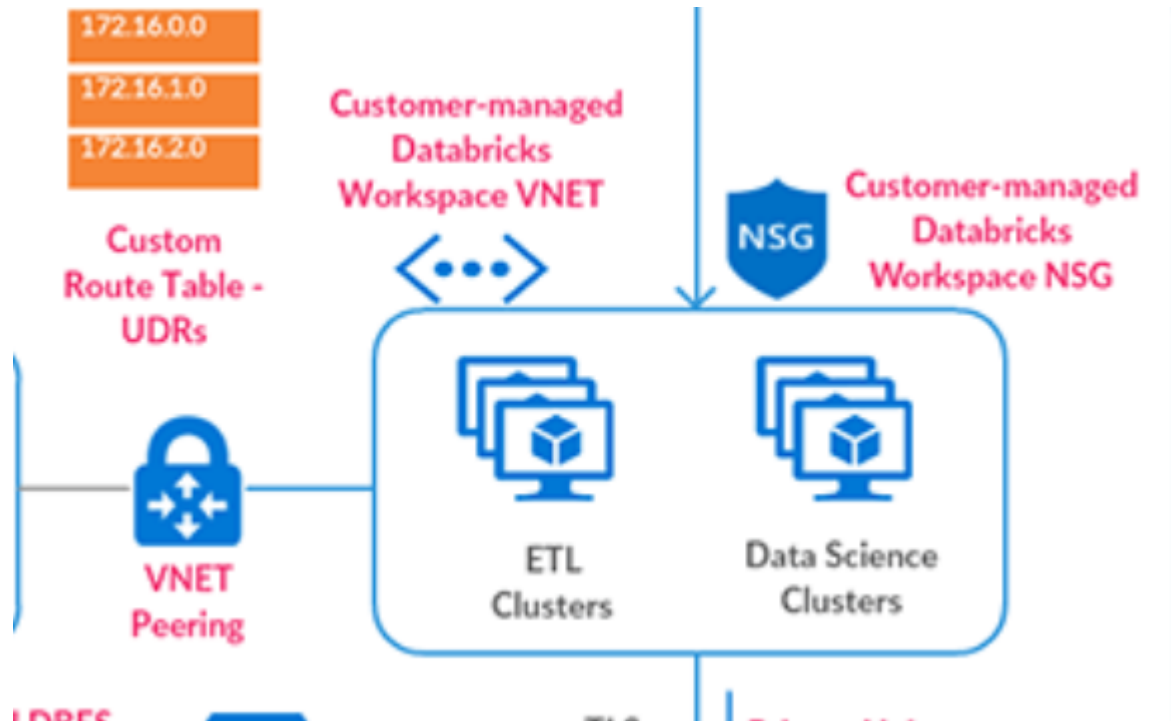
```
Cmd 3
instance.
instance.method1(param1, param2)
instance.method2()
```

# Workspace Automation – ARM Templates

```
{
  "name": "string",
  "type": "Microsoft.Databricks/workspaces",
  "apiVersion": "2018-04-01",
  "tags": {},
  "location": "string",
  "properties": {
    "managedResourceGroupId": "string",
    "parameters": {},
    "uiDefinitionUri": "string",
    "authorizations": [
      {
        "principalId": "string",
        "roleDefinitionId": "string"
      }
    ]
  },
  "sku": {
    "name": "string",
    "tier": "string"
  }
}
```

- Control where “data plane” will live
- Can assign a specific user to use to host management activities
- Can define service targets

# Workspace Architecture – Key Considerations



- ➡ Network Permissions
- ➡ Secrets
- ➡ Mount Points

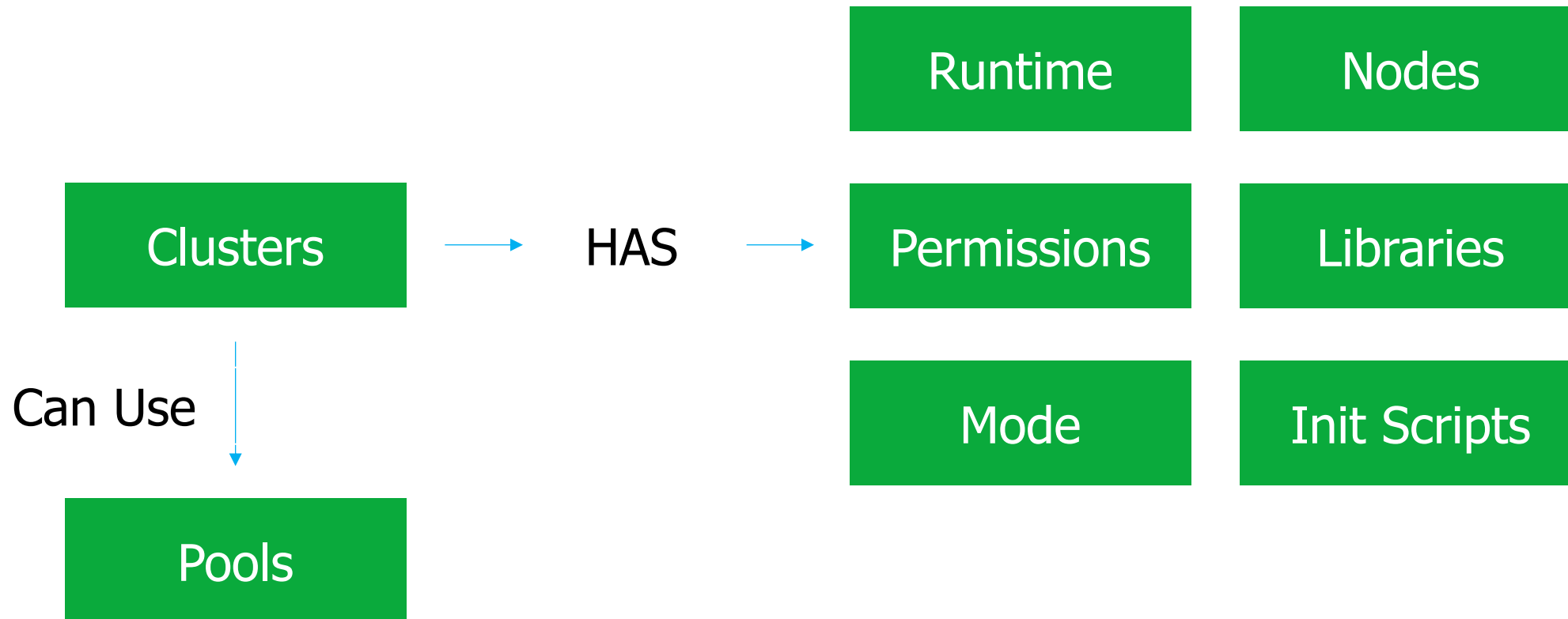
<https://databricks.com/blog/2020/03/27/data-exfiltration-protection-with-azure-databricks.html>

# Core Concept – Databricks Rest API 2.0

Clusters	Libraries	Workspace
DBFS	MLFlow	Permissions
Groups	SCIM	Instance Pools
Jobs	Secrets	Token

- Most components can be configured via REST
- Authentication done via PAT, preview for OAuth
- Databricks CLI wrapper over REST API, effectively just passing JSON objects (request/response)
- Consider creating a Service Account with PAT to handle automation

# Core Concepts - Clusters

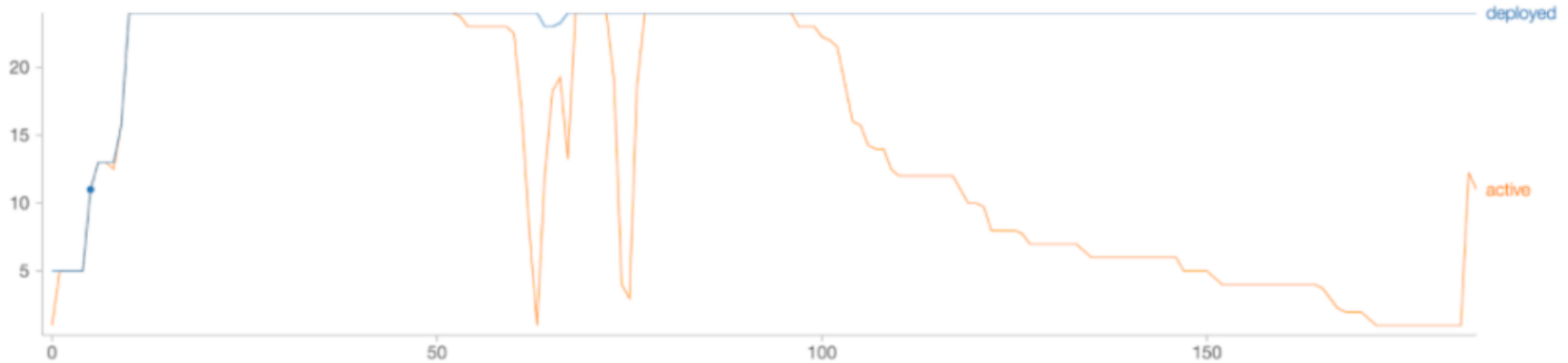


# Cluster Automation – Key Considerations

```
{  
  "cluster_name": "TheBestestCluster",  
  "spark_version": "latest-stable-scala2.11",  
  "node_type_id": "Standard_DS3_v2",  
  "num_workers": 5,  
  "autotermination_minutes": 30,  
  "idempotency_token": "arandomstring",  
  "custom_tags": [  
    {  
      "key": "Importance",  
      "value": "Really low"  
    }  
  ],  
  "spark_conf": {  
    "spark.databricks.delta.preview.enabled": "true",  
    "spark.databricks.cluster.profile": "serverless",  
    "spark.databricks.service.server.enabled": "true",  
    "spark.databricks.repl.allowedLanguages": "sql,python,r"  
  }  
}
```

- ➡ Nodes vs Autoscale
- ➡ Init Scripts
- ➡ Libraries
- ➡ Idempotency\_token
- ➡ Custom Tags
- ➡ Auto-termination Minutes

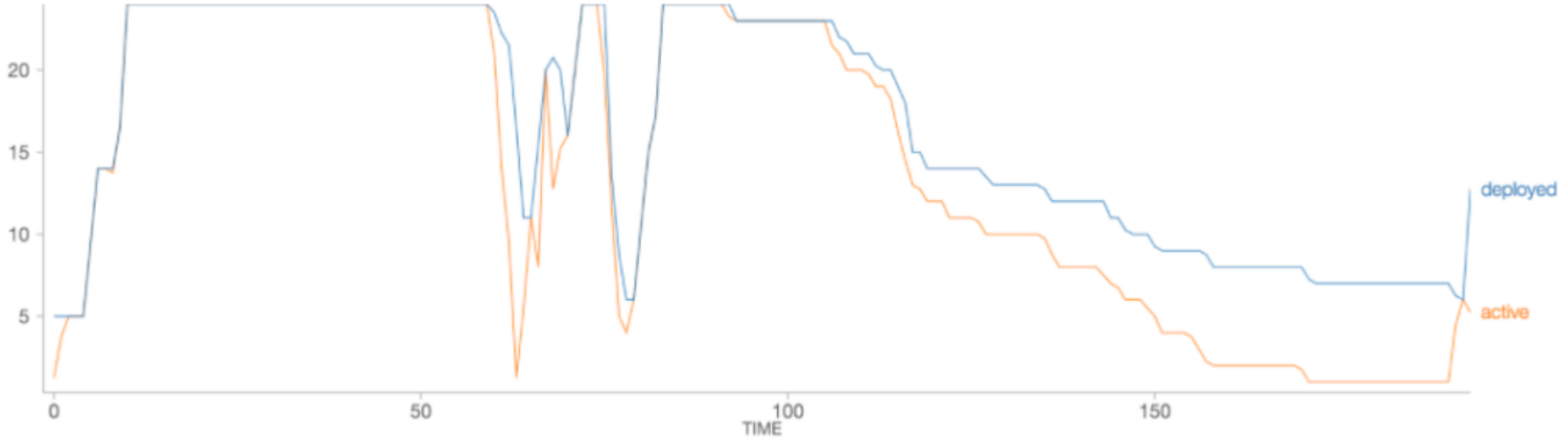
# Cluster Considerations – Standard Autoscaling



- ➡ Scaling type in the "Standard" cluster mode
- ➡ Scale-down only occurs when cluster is completely idle

- ➡ Add/Removes nodes using exponential algorithms
- ➡ Scales down after 10 minutes of "underutilization"

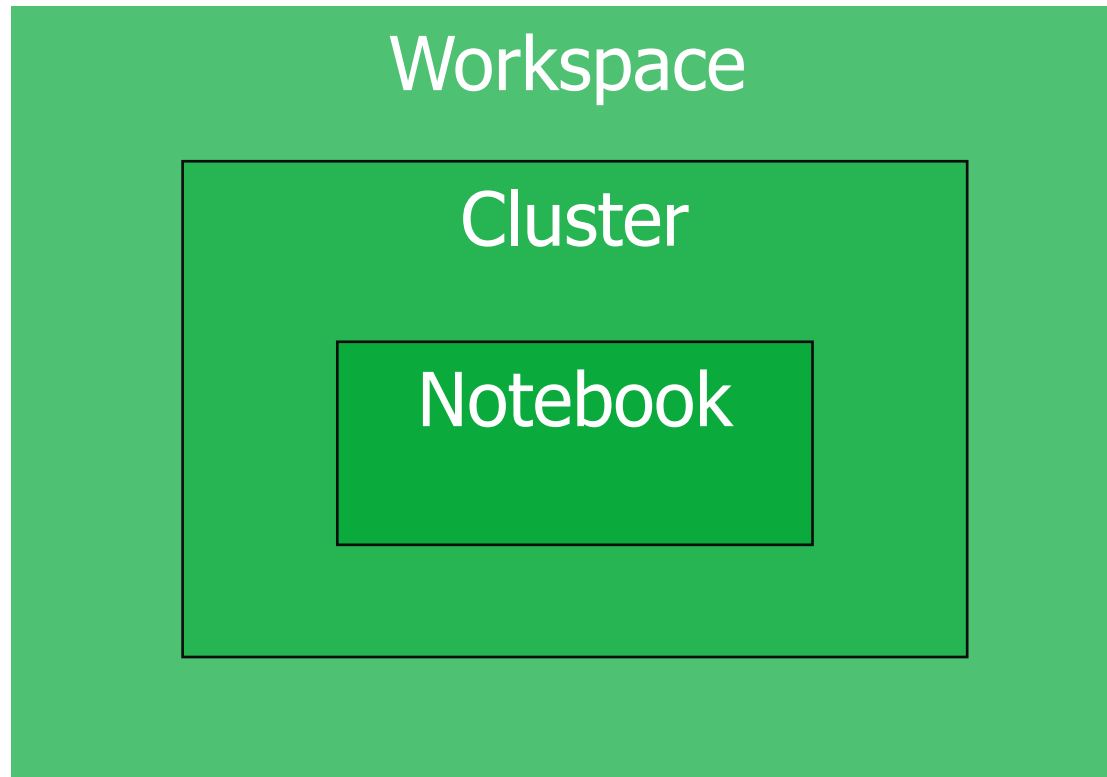
# Cluster Considerations – Optimized Auto scaling



- ➡ Scaling type in the "Premium" cluster mode
- ➡ Jobs are always run with Optimized Auto Scaling
- ➡ Can scale down while workloads running by moving shuffle files
- ➡ Makes scale down decisions after 150 seconds

# Cluster Considerations – Init Scripts and Libraries

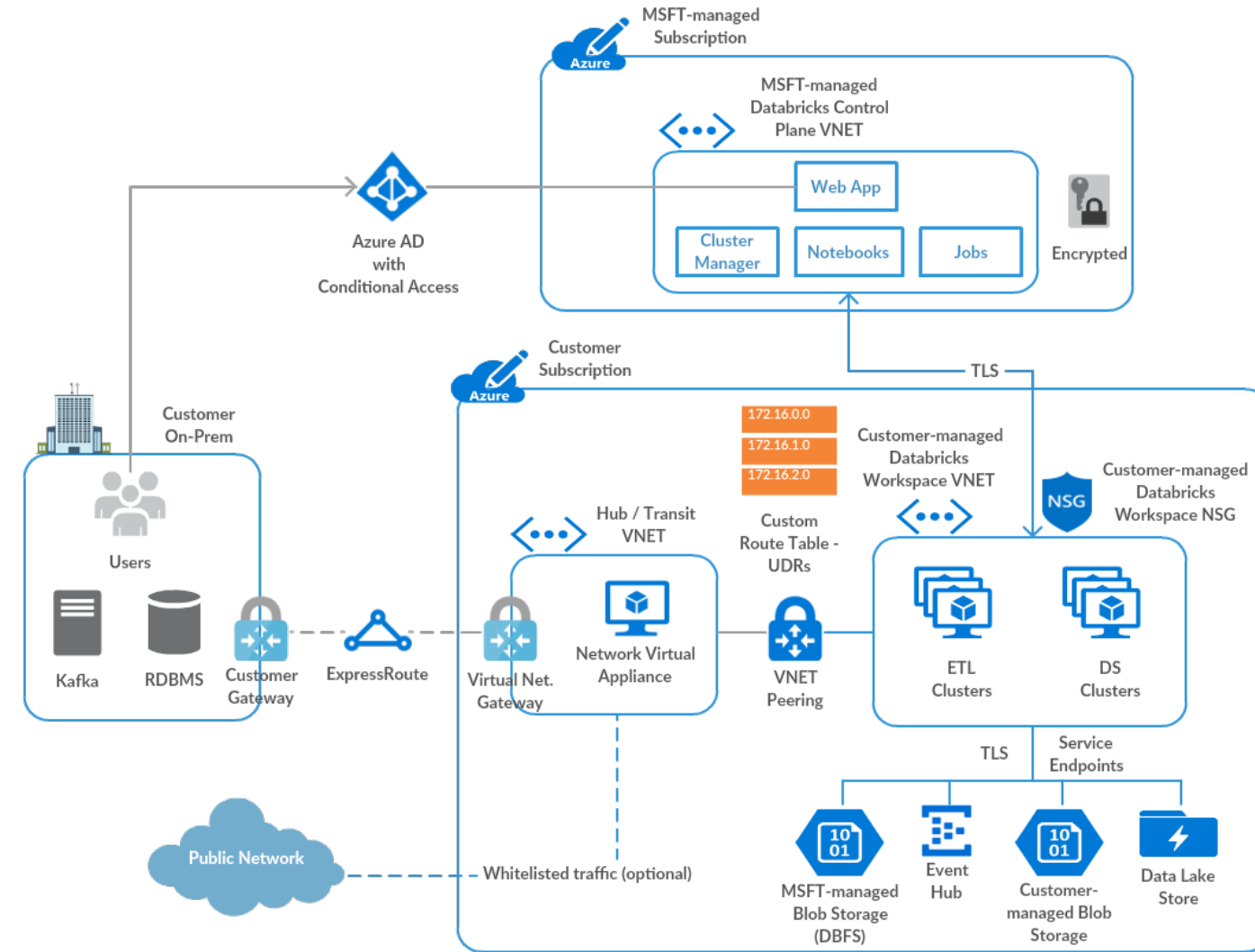
## SCOPES



⇒ Can require local DBFS

⇒ Consider using CI/CD Process to handle complexities

# Core Concept – Databricks Vnet Injection



- Makes use of Azure Resource Delegation
- Network security group configured with Azure service tags
- Requires 2 subnets (public and private)
- Makes use of other Azure networking features (on-prem access, service endpoints, etc)

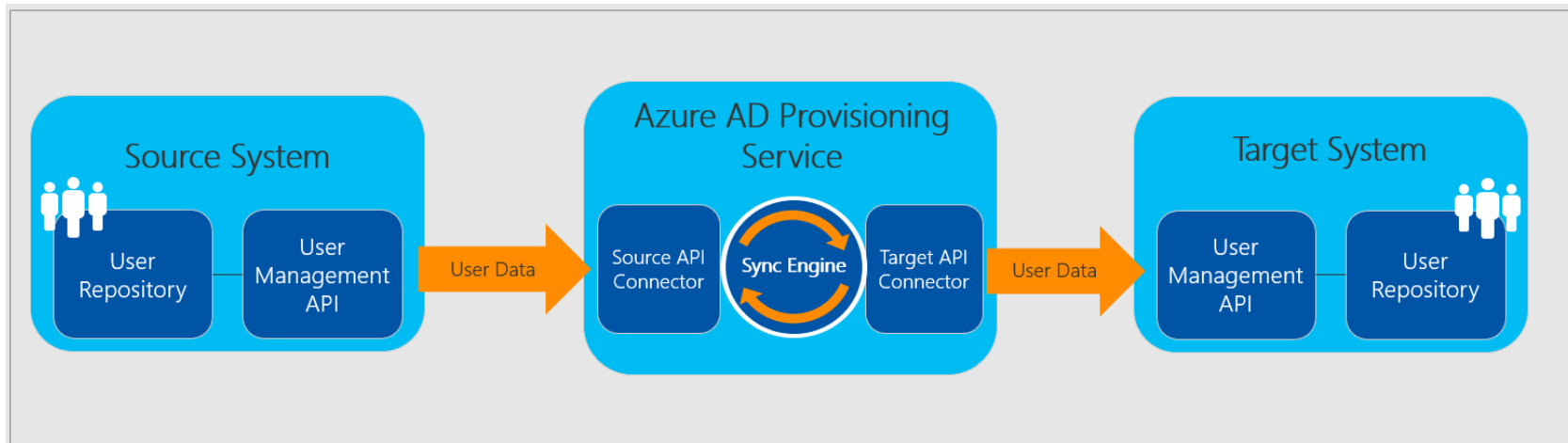
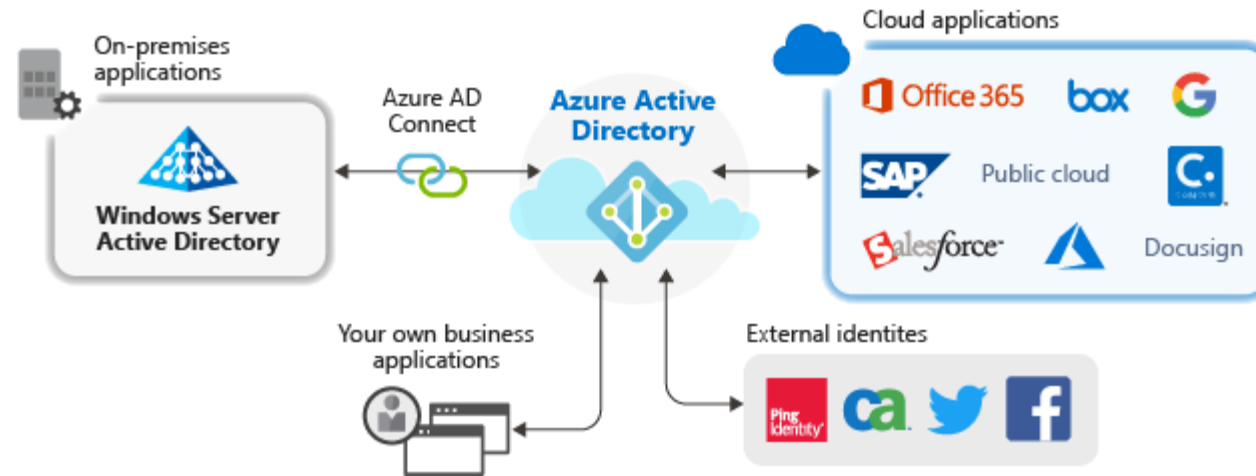
# Network Security – Enable Delegation

```
{
  "name": "publicsubnet",
  "properties": {
    "addressPrefix": "[parameters('publicSubnetPrefix')]",
    "networkSecurityGroup": {
      "id": "[resourceId('Microsoft.Network/networkSecurityGroups', concat(parameters(
'virtualNetworkName'), '-pubnsg'))]"
    },
    "delegations": [
      {
        "name": "databricks-del-public",
        "properties": {
          "serviceName": "Microsoft.Databricks/workspaces"
        }
      }
    ]
  },
}
```

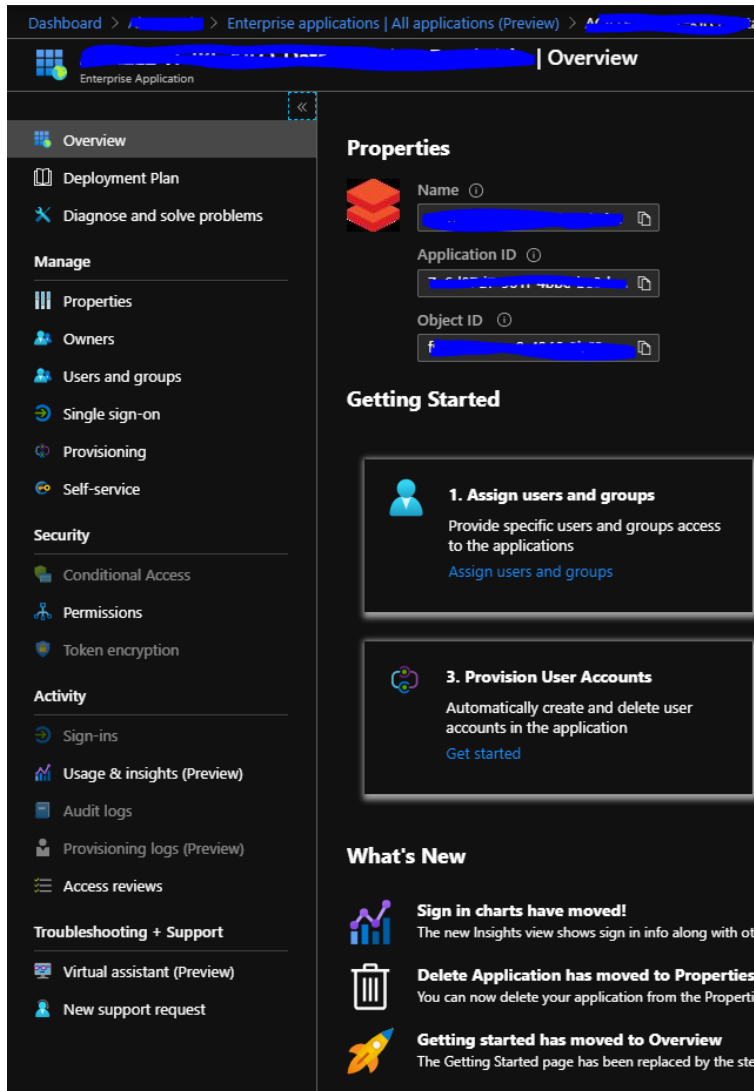
# Network Security – Add VNET Injection

```
{
  "type": "Microsoft.Databricks/workspaces",
  "properties": {
    "ManagedResourceGroupId": "[variables('managedResourceGroupId')]",
    "parameters": {
      "customVirtualNetworkId": {
        "value": "[parameters('customVirtualNetworkId')]"
      },
      "customPublicSubnetName": {
        "value": "[parameters('customPublicSubnetName')]"
      },
      "customPrivateSubnetName": {
        "value": "[parameters('customPrivateSubnetName')]"
      }
    }
  }
}
```

# Core Concept: System for Cross Identity Management



# Databricks Users via SCIM



- “Append only” type functionality
- Can configure users/groups to provision
- Sync happens every 20-40 minutes
- Requires Personal Access Tokens

# Core Concepts: Permissions

Jobs

Clusters

Directories

Pools

Notebooks

- Generally configured via UI, but API in preview
- Only available in the premium plan
- Usually applied after the target has been created

# Permissions: Example



Job Runner

Jobs

CAN\_MANAGE\_RUN

Directories

CAN\_VIEW



Data Analytics

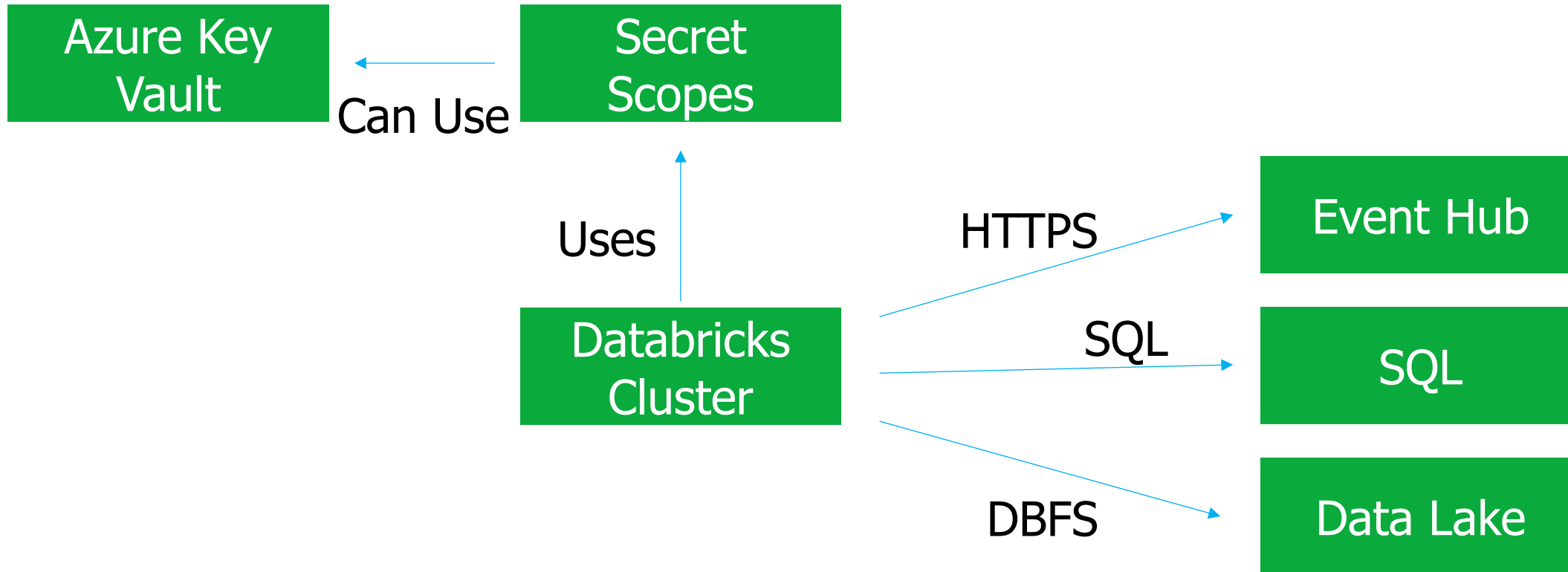
Clusters

CAN\_RESTART

Notebooks

CAN\_EDIT

# Core Concept: Integrations



# Who Dun It?

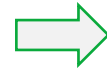
## Secret Scopes



Can use ACLs (Premium Only)



Build specific scopes for users/groups



Azure Key Vault provides separation of duties

## Data Lake

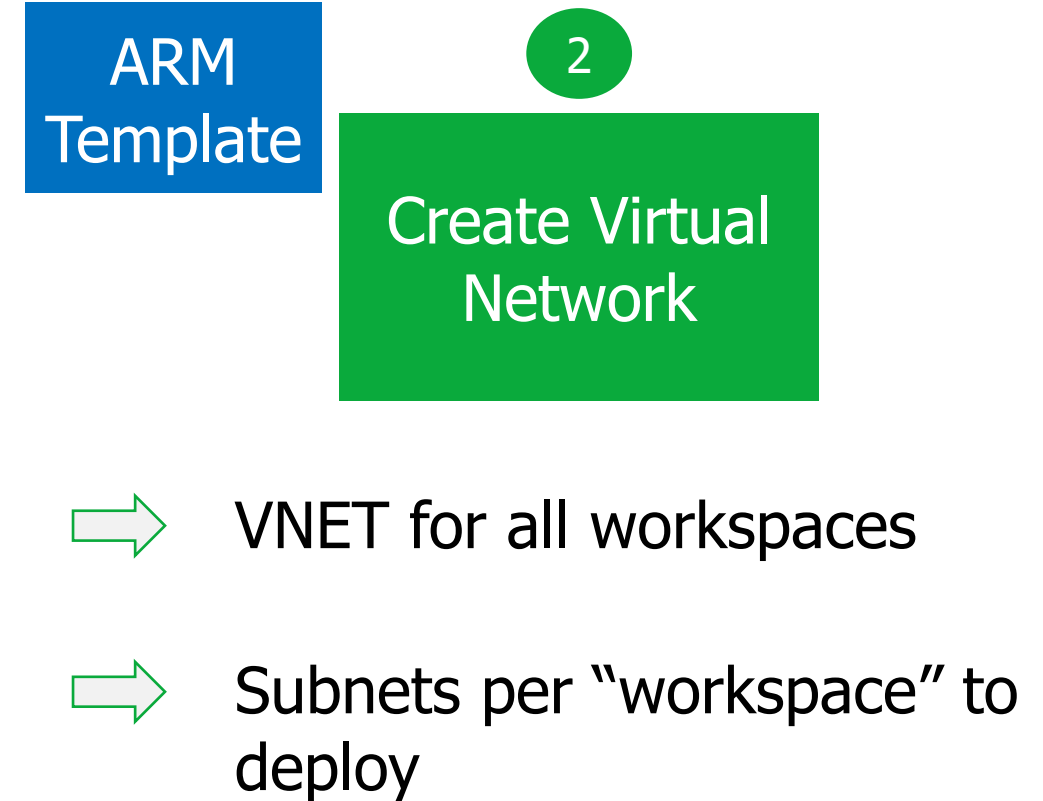
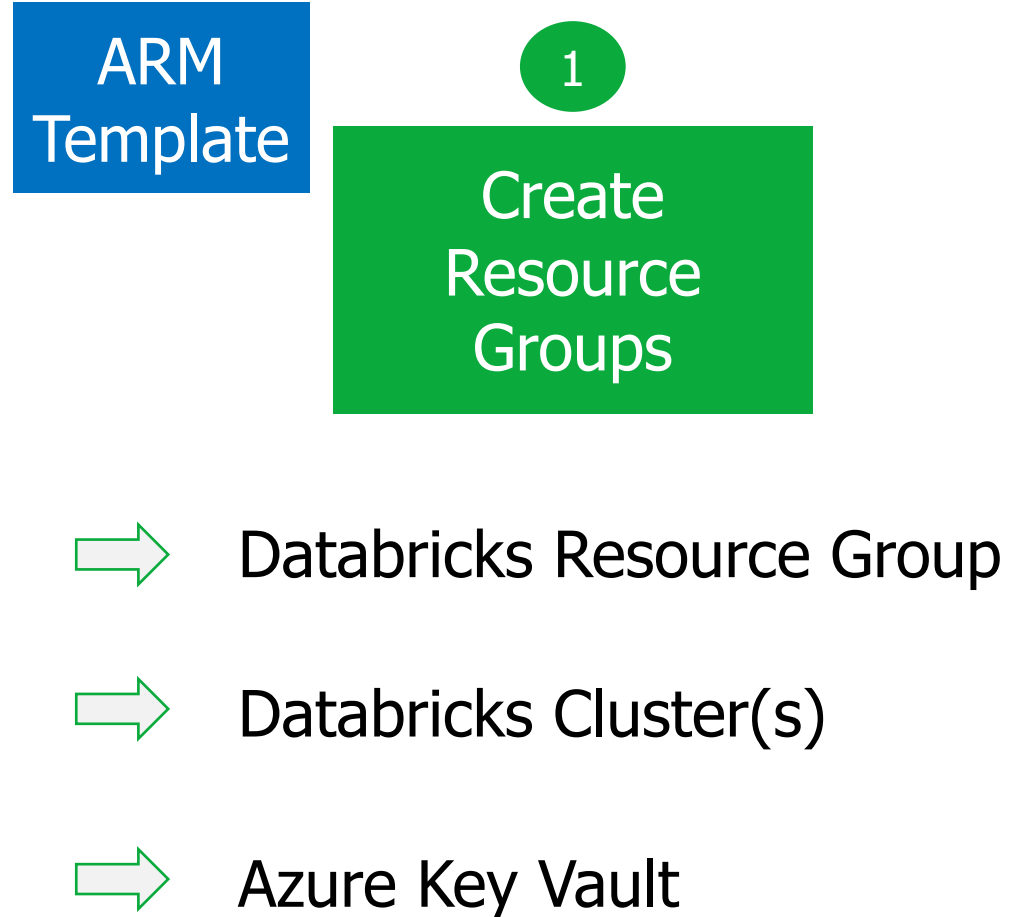


Can use AAD Passthrough



Combine with ACLs on the Data Lake

# Databricks Setup Steps



# Databricks Setup Steps

ARM  
Template

3

Create Key  
Vaults

- ➡ Consider secret scopes architecture
- ➡ Provide necessary key vault policies to users/groups/admins

ARM  
Template

4

Create  
Workspaces

- ➡ Consider mounts
- ➡ Consider tiers (Standard vs Premium)
- ➡ Consider VNET/Subnet Setups

# Databricks Setup Steps

SCIM

5

Assign  
Databricks  
Users

- ➡ Consider users/groups per workspace
- ➡ Tie in with other AD Features (dynamic groups, etc)

API

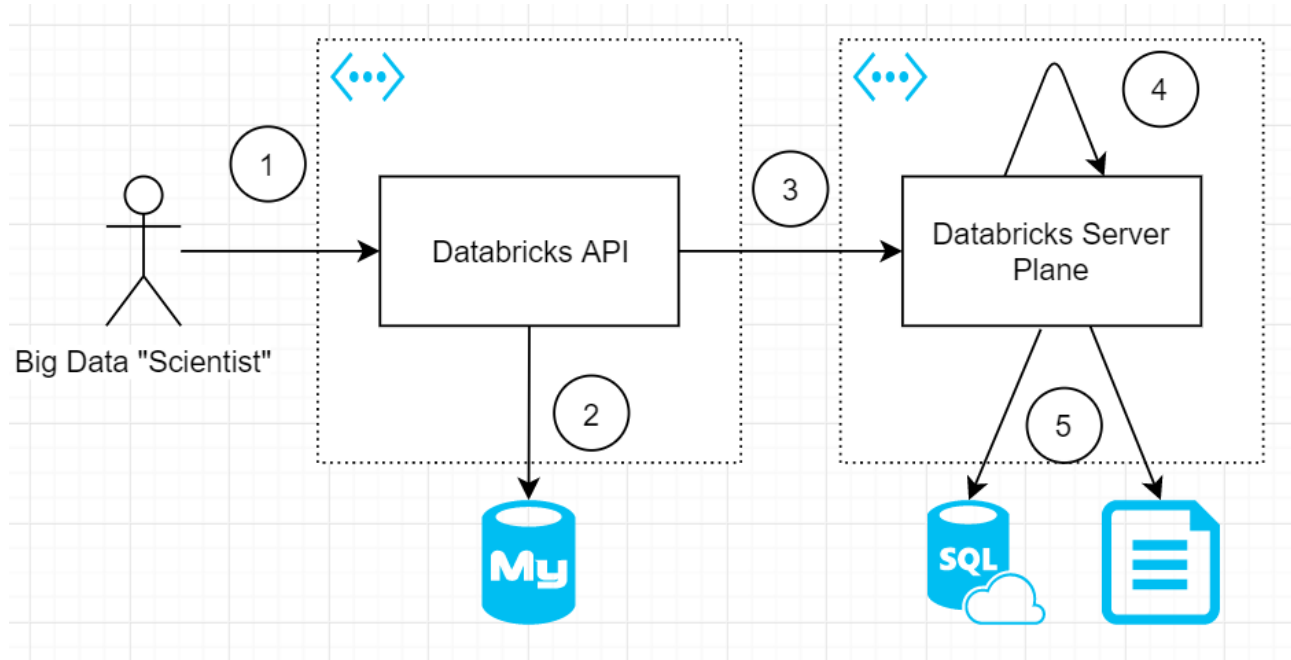
6

Create  
Databricks  
Objects

- ➡ Notebooks / Directories / Secret Scopes / etc
- ➡ Assign appropriate permissions

7 Profit, I think....

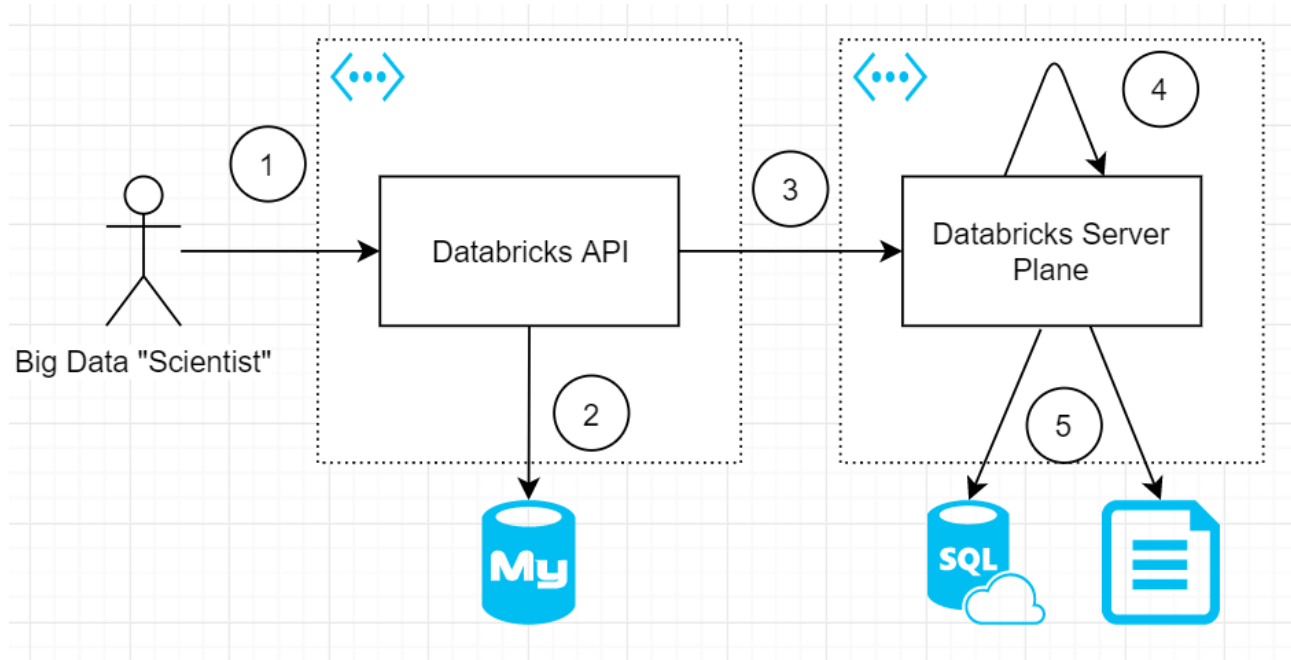
# Security 1 – Control Plane Data



- ➡ Notebook results are stored in MySQL database in the control plane
- ➡ Option to use customer managed keys in key-vault
- ➡ Increased latency / key caching

<https://docs.microsoft.com/en-us/azure/databricks/security/keys/customer-managed-key-notebook>

# Security 2 – Cluster Communication



- ➡ Communication to the cluster is encrypted
- ➡ Communication between cluster nodes is not encrypted
- ➡ Init Script to enable cluster communication to use TLS

<https://docs.microsoft.com/en-us/azure/databricks/security/encryption/encrypt-otw>



# Thank You

---

 Shamir Charania

 [shamir@keepsecure.ca](mailto:shamir@keepsecure.ca)

 <https://www.keepsecure.ca>