Study Definition Repository (SDR) Reference Implementation

High Level Design (Infrastructure)

SDR Release V2.0.2, October 2023

TransCelerate BIOPHARMA INC.

Disclaimer

These materials and information, as well as the underlying code/application they relate to, are provided by TransCelerate Biopharma Inc. AS IS. Any party using or relying on this information, these materials, and/or the underlying code/application do so entirely at their own risk. Neither TransCelerate nor its members will bear any responsibility or liability for any harm, including indirect or consequential harm, that a user may incur from use or misuse of this information, materials, or underlying code/application.

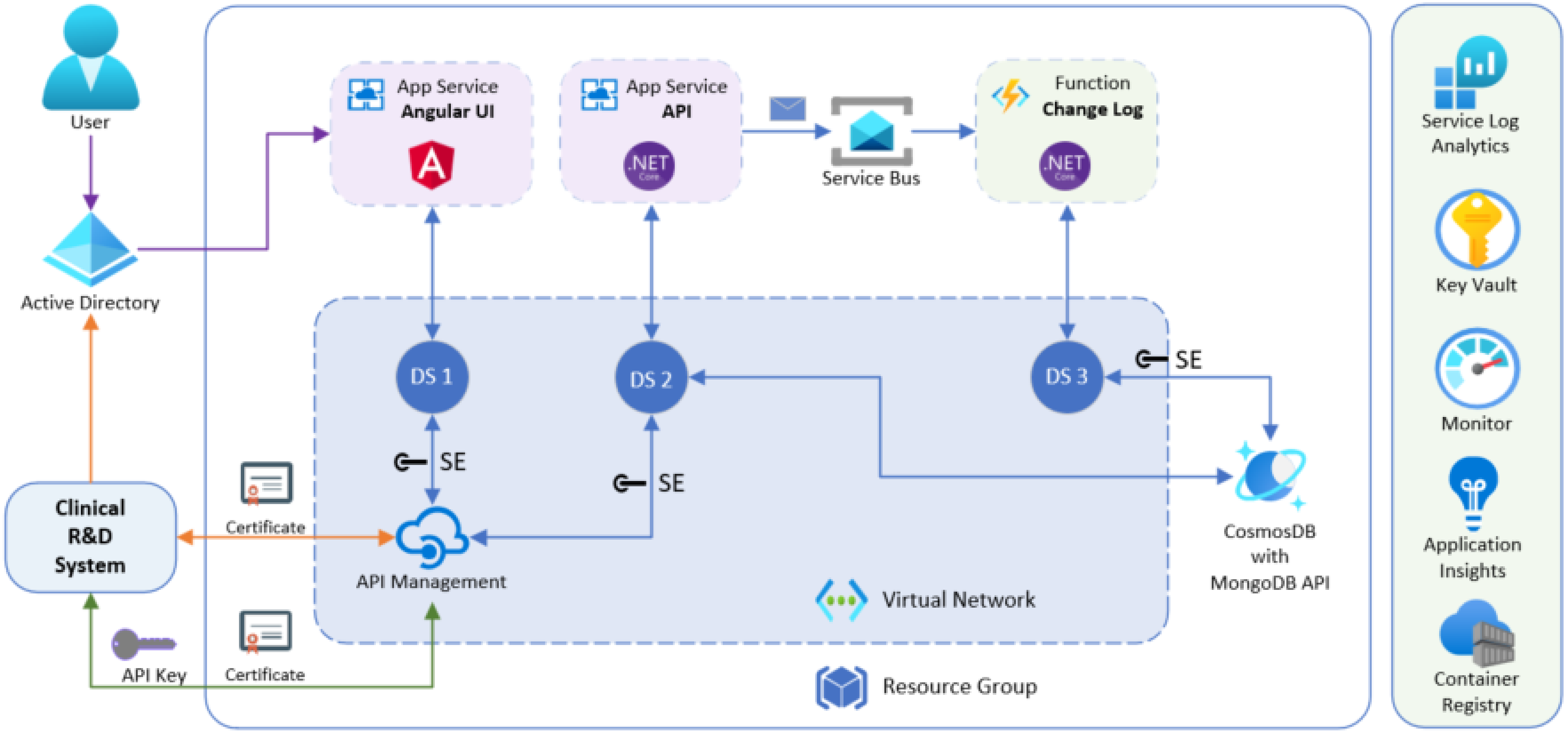
TransCelerate does not endorse any particular software, system, or service. And the use of specific brands of products or services by TransCelerate and its collaboration partners in developing the SDR Reference Implementation should not be viewed as any endorsement of such products or services. To the extent that the SDR Reference Implementation incorporates or relies on any specific branded products or services, this resulted out of the practical necessities associated with making a reference implementation available to demonstrate the SDR’s capabilities.

Table of Contents

Overview

- This document captures High-Level Design of the Infrastructure Platform of Study Definition Repository Reference Implementation (SDR RI) hosted on Azure* Cloud.
- The SDR RI was built to show **one** possible way to implement the SDR. Users remain free to download the SDR source code and design their own implementations.
- To the extent that the SDR Reference Implementation incorporates or relies on any specific branded products or services, this resulted out of the practical necessities associated with making a reference implementation available to demonstrate the SDR’s capabilities and is not an endorsement of those products and services.
- This document outlines each of the architectural construct that make up Azure Fabric (API Management, AppService, AppService Plan, Cosmos DB, Azure Monitor, Application Insights, Key Vault).

SDR RI Platform Architecture on Azure



Legend

DS = Delegated Subnet
SE = Service Endpoint

Governance – Tagging

For SDR Reference Implementation, following tags have been created as part of the Tagging Strategy:

Tag Name	Description	Required/ Optional	Value	Resources
Environment	Specifies the environment to which the resources belong to. Every resource belongs to a unique environment.	Required	E.g.: Dev, Test	Resource Group, VNet, API Management, App Service Plan, Application Insights, App Service, Log Analytics Workspace, Cosmos DB, Key Vault, Container Registries, Service Bus, Function App.
App_Layer	Identifies whether the resource is part of “Front End” or “Back End”	Required	Frontend/ Backend/ ”N/A”	VNet, API Management, App Service Plan, Application Insights, App Service, Log Analytics Workspace, Cosmos DB, Key Vault, Function App, Service Bus.

Governance – Resource Locks

SDR Reference Implementation has Delete lock on all Resource Groups. This includes all resources servicing core functionality such as network connectivity, security, and automation, as well as any resource running a workload that serves either internal or external customers (i.e., that would create impact for multiple users if it were taken offline or deleted). The locks have been applied at the Resource Group level. Locks applied to a Resource Group are to be applied to all the resources that exist within that Resource Group.

Note: *SDR reference implementation has leveraged Read-only locks for archived data that need to be retained for compliance and regulatory needs.*

Lock Level	Description
CanNotDelete (Delete)	Authorized users can read and modify a resource, but they can't delete it
ReadOnly (Read-only)	Authorized users can read a resource, but they can't delete or update it. Applying this lock is like restricting all authorized users to the permissions granted by the Reader role

Governance – Naming Convention

<ResourceType>-<Subscription>-<Purpose/Segment/Environment>-<region>-<Instance Number or Level>

Meta Data	Field Required	Field Length
<ResourceType>	Mandatory	Up to 4 Characters
<Subscription>	Mandatory	Up to 7 Characters
<Environment>	Mandatory	Up to 20 Characters*
<Region>	Mandatory	Up to 6 Characters
<Instance Number or Level>	Mandatory	Up to 3 Characters

Usage

Resource Type	Format	Examples
VNet	vnet-<subscription>	vnet-sdr-test-eastus
Subnet	snet-<subscription>	snet-sdr-test-eastus

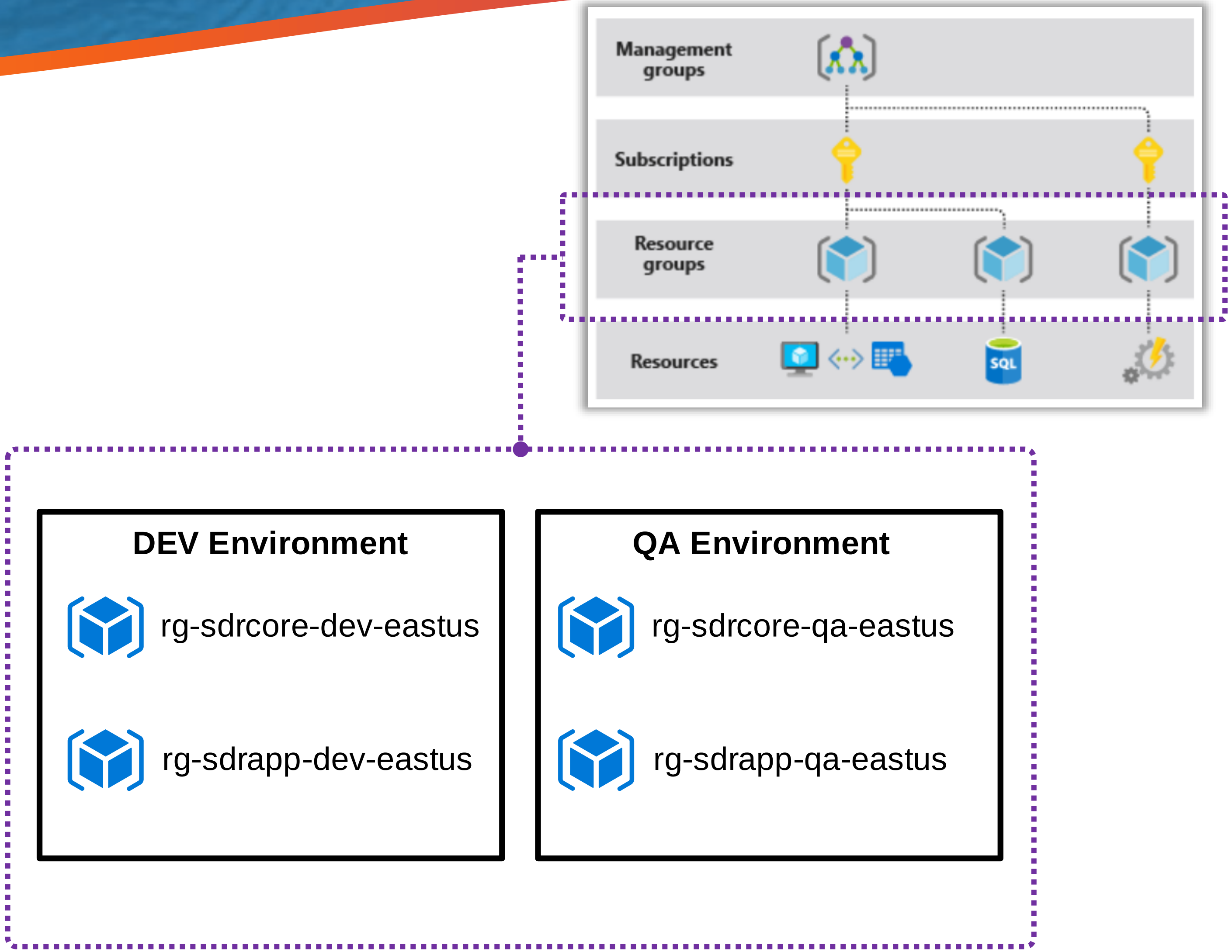
*Note: There are some exceptions to this naming convention where the resource names should be globally unique (like Azure Key Vault, Log Analytic Workspace, Storage Accounts etc.), some which do not allow any special characters (like Storage Accounts) and some have *character limitation.*

Resource Group

- Resources have been grouped in a Resource Group if they have the same life-cycle such as Applications or Services.

- SDR Reference Implementation used 2 different resource groups for core and app for each environment.

- The core resource group includes the resources – Key Vault, Log Analytics Workspace, Virtual Network and all the remaining resources are part of the rg-sdrapp-<env>-eastus resource group.



DEV Environment

rg-sdrcore-dev-eastus

rg-sdrapp-dev-eastus

QA Environment

rg-sdrcore-qa-eastus

rg-sdrapp-qa-eastus

Resources - Decisions

Decision Area	Decision
API Management	• A single API Management is used per environment. • The Application Insights is integrated with API Management. • Developer portal is leveraged for each environment to allow API key-based authentication
App Service Plan	• There are 3 App Service plans used per environment. (one each for frontend UI, backend API and Function App)
Application Insights	• A single Application Insights is deployed per environment.
App Service	• There are 2 App Services used per environment – one for the frontend UI and the other for Backend API.

Resources - Decisions

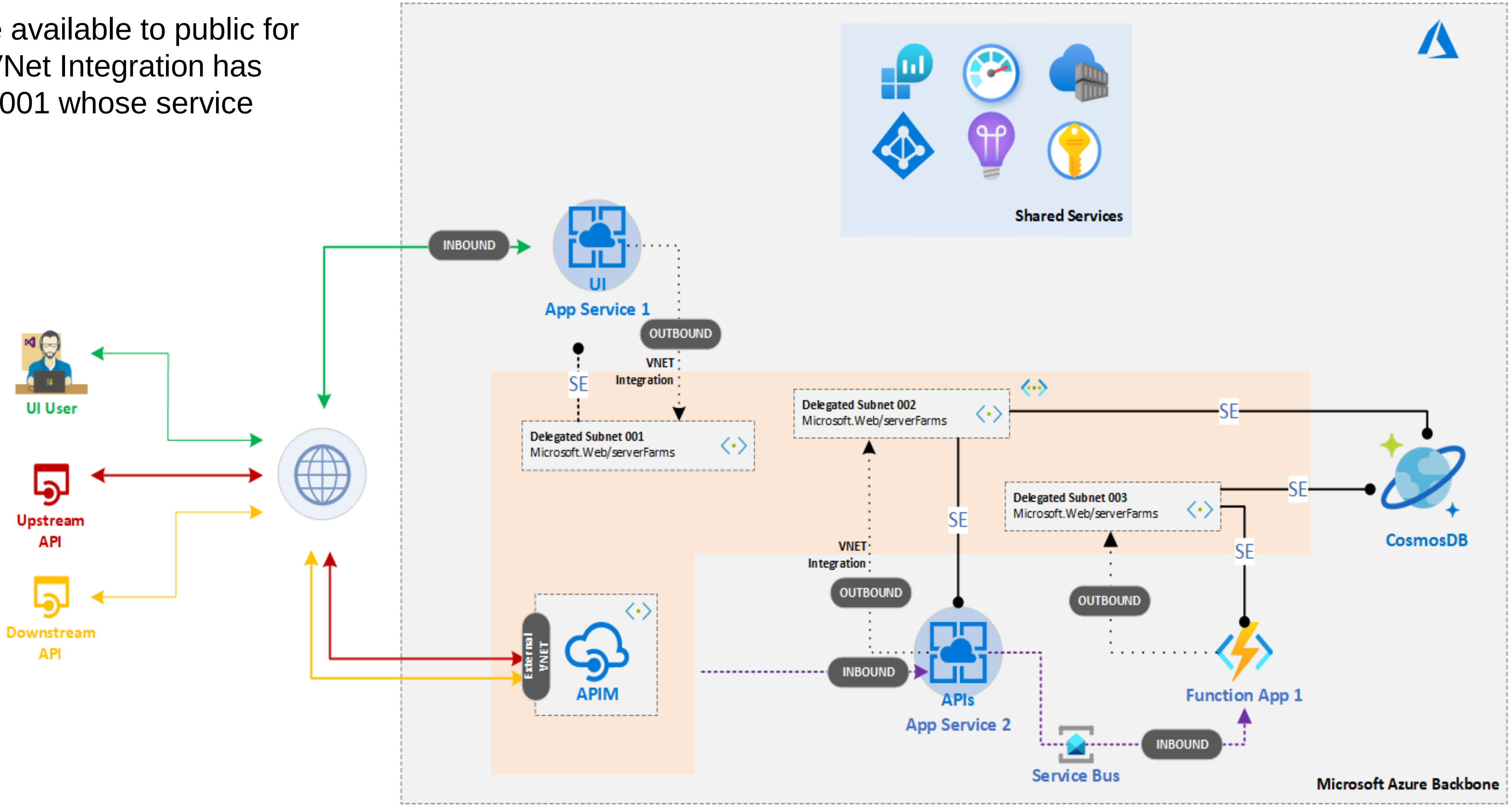
Decision Area	Decision
Cosmos DB	• A single Cosmos DB API for Mongo DB per Environment is used.
Function App	• A single instance of Function App per Environment is used. (Storage account is created by default with a function app)
Service Bus	• A single Service Bus per environment

Networking cont.

- The Inbound access has been made available to public for App Service1 and Outbound has a VNet Integration has been done to the Delegated Subnet 001 whose service endpoint is Microsoft.Web.

- The Inbound access for App Service2 and Function App has been restricted to VNet through Access Restriction and Outbound has a VNet Integration done to the Delegated Subnet 002 and Delegated Subnet 003 respectively, whose service endpoints are Microsoft.AzureCosmosDB (for both DS) and Microsoft.Web (for DS 002 only)

- API Management is the API Gateway for the Upstream and Downstream APIs. It has been deployed in the VNet and made External.



Networking - Key Decisions

Decision Area	Decision
VNet Architecture	Each environment has its own VNET
Subnet	Each environment has a single subnet assigned.
Delegated Subnet	Each environment will have 3 delegated subnets with Microsoft.Web, Microsoft.AzureCosmosDB, Microsoft.Storage as the service endpoints.

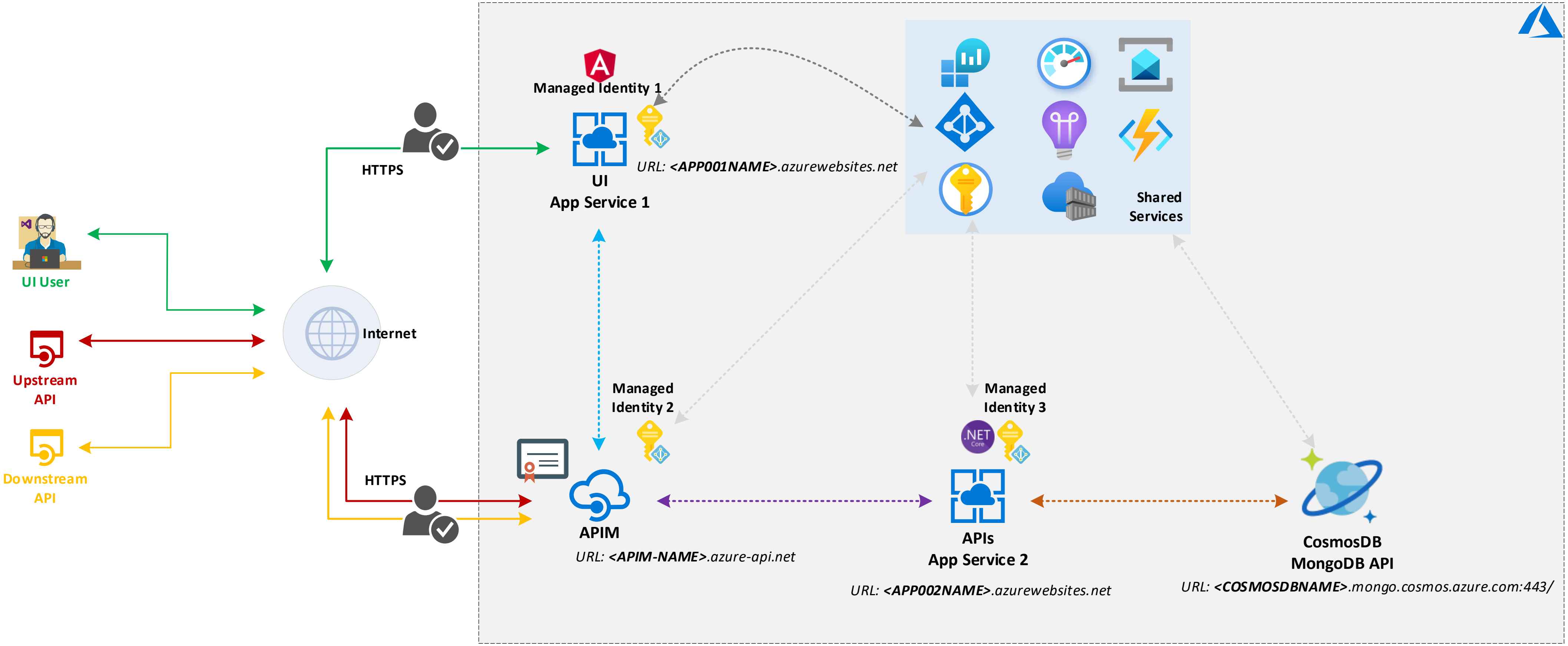
Connectivity

- Principle of least privilege access has been applied. Only minimal permissions needed to accomplish the task have been granted.
- Process has been established around this, as well as a process around periodic access reviews to ensure users who don’t need elevated roles are removed.

Communication Flow in SDR

- Communication from Internet to UI App Service is enabled through HTTPS Protocol
- Upstream communication through the internet is handled by APIM which in turn communicates with API App Service. The API App Service forwards the call further to Cosmos DB.
- Downstream communication is also handled by APIM via Internet. APIM handles the calls to UI App Service and API App Service.
- Function App trigger is a queue-based trigger and can be invoked only from messages pushed to the Service Bus.
- API App Service, and CosmosDB Mongo API App cannot be reached directly from the internet. All the calls to these resources are handled by APIM.
- The shared services in Azure are enabled to handle the requests from UI App Service, APIM, API App Service for Logging, Monitoring etc.

Connectivity- Diagram



INDEX

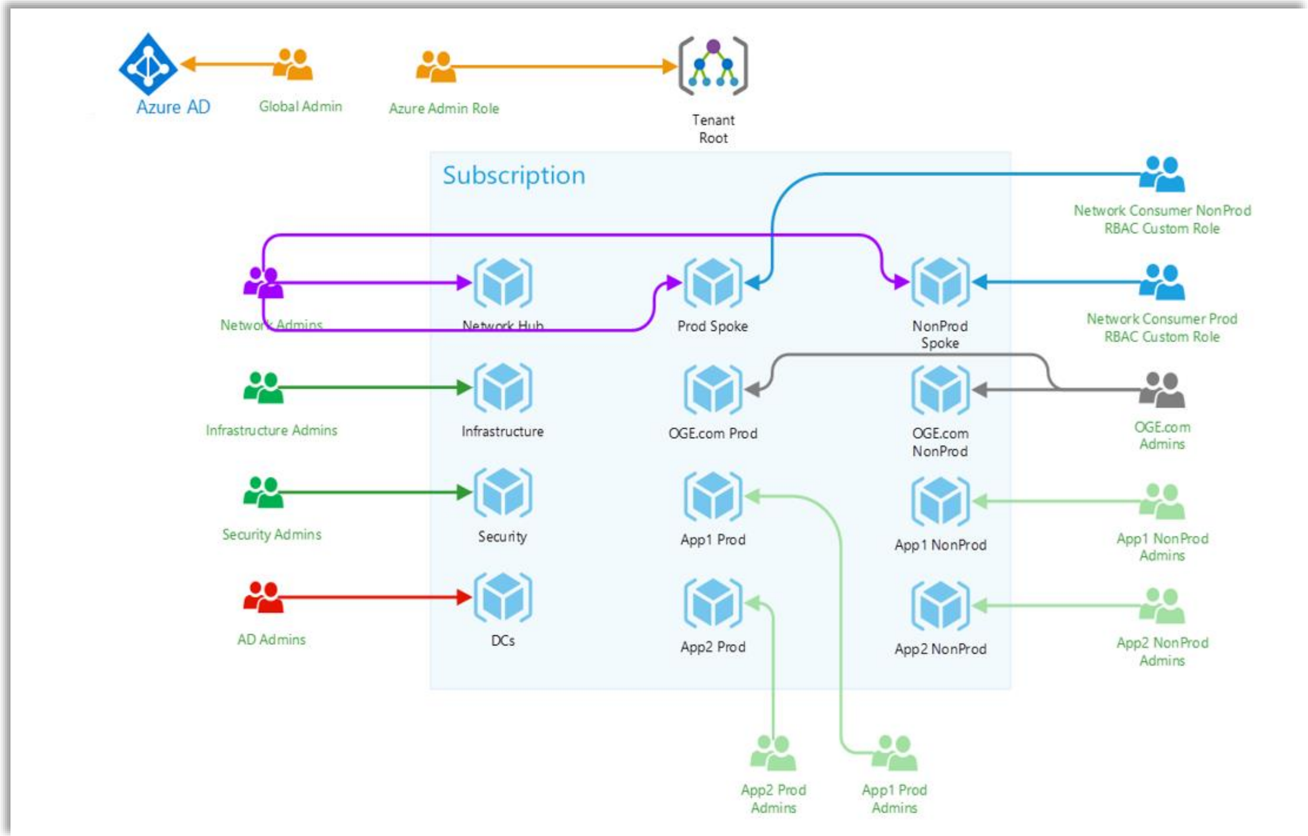
- ↔ *End User call to UI App Service*
- ↔ *Upstream call API to APIM*
- ↔ *Downstream API call to APIM*
- ↔ *UI App Service call to APIM*
- ↔ *APIM call to API App Service*
- ↔ *API App Service call to Cosmos DB*
- ↔ *Virtual network*
- ↔ *OAuth 2.0*

Identity - Decisions

Decision Area	Options	Decision
Azure AD	• Use Existing Azure AD Tenant • Use New Azure AD Tenant • AD Integration/ Synchronization	• The SDR Reference Implementation application has leveraged Azure AD for Authentication.
Admin Scope	• Limited owner permissions	• Subscription owner permissions assigned to limited users.
Service Principal	• Integration with GitHub Actions • Authentication/Authorization of App	• For IaC code deployment, a service principal is configured with appropriate privileges to handle the resource deployment in azure.
Managed Identity	• System assigned • User assigned	• System Managed Identities are used, and it’s configured for 3 resources – APIM, UI App Service, API App Service.
RBAC	• Principle of least privilege • Establish process around to strive this goal	• Security Groups are leveraged • Multiple groups have been created to handle the access requirements and segregate the user permissions accordingly

Role based access control

- Principle of least privilege access has been applied while giving access. Minimal permissions have been granted that are needed to accomplish the task.
- Process has been established around this, as well as a process around periodic access reviews to ensure users who don’t need elevated roles are removed.
- All Users have not been given access to the resources. Privileged access for select users has been given with assigned roles via groups and are revoked once the necessary work is completed.

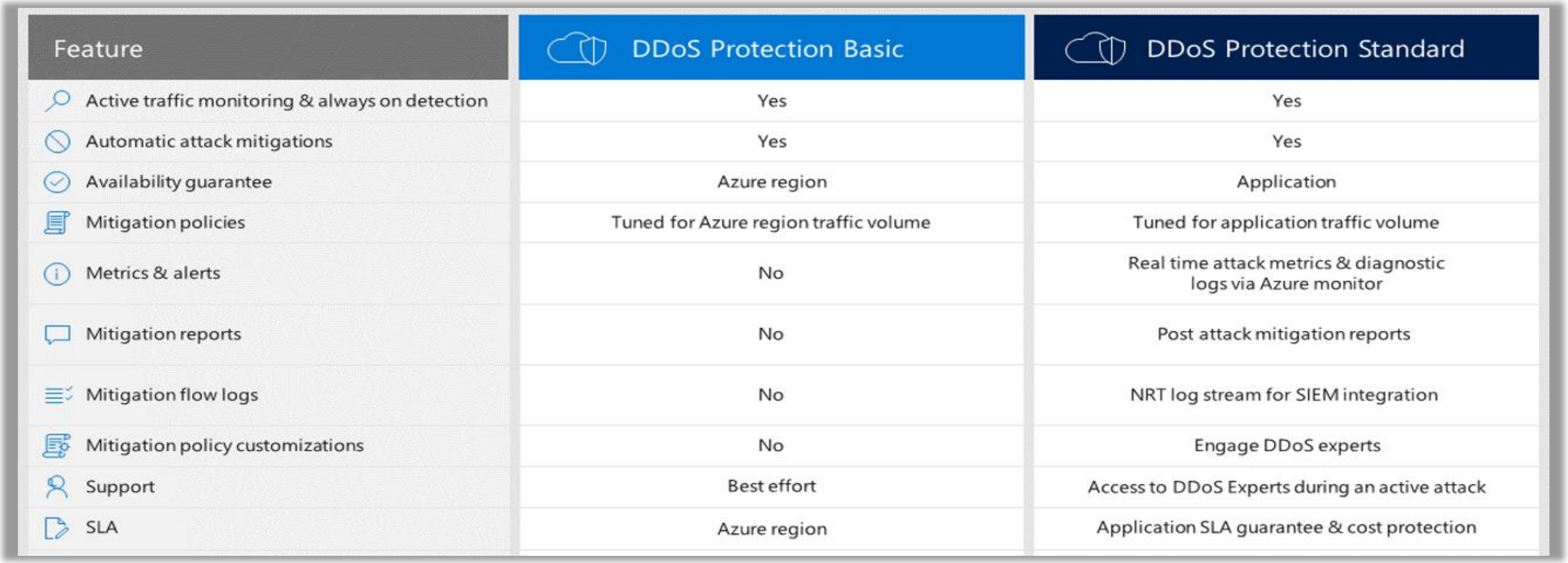


Security - Decisions

Decision Area	Options	Decision
Azure Key Vault	• Key Vault Breakdown based on Environment	• Cloud native SDR deployment has single Key Vault per Environment
Security Monitoring	• User Management & Activity Monitoring • Restriction of Security log Access • Regulatory Compliance Monitoring	• Application and User Activity Monitoring - Cloud Native Tool – Azure Monitor has been configured • Compliance Monitoring is being performed by Accenture Cloud Platform
Certificates	• Certificate based authentication • Password based authentication	• Certificates through APIM have been leveraged. • Root CA and Client certificates are configured in APIM

Perimeter Security – Decisions

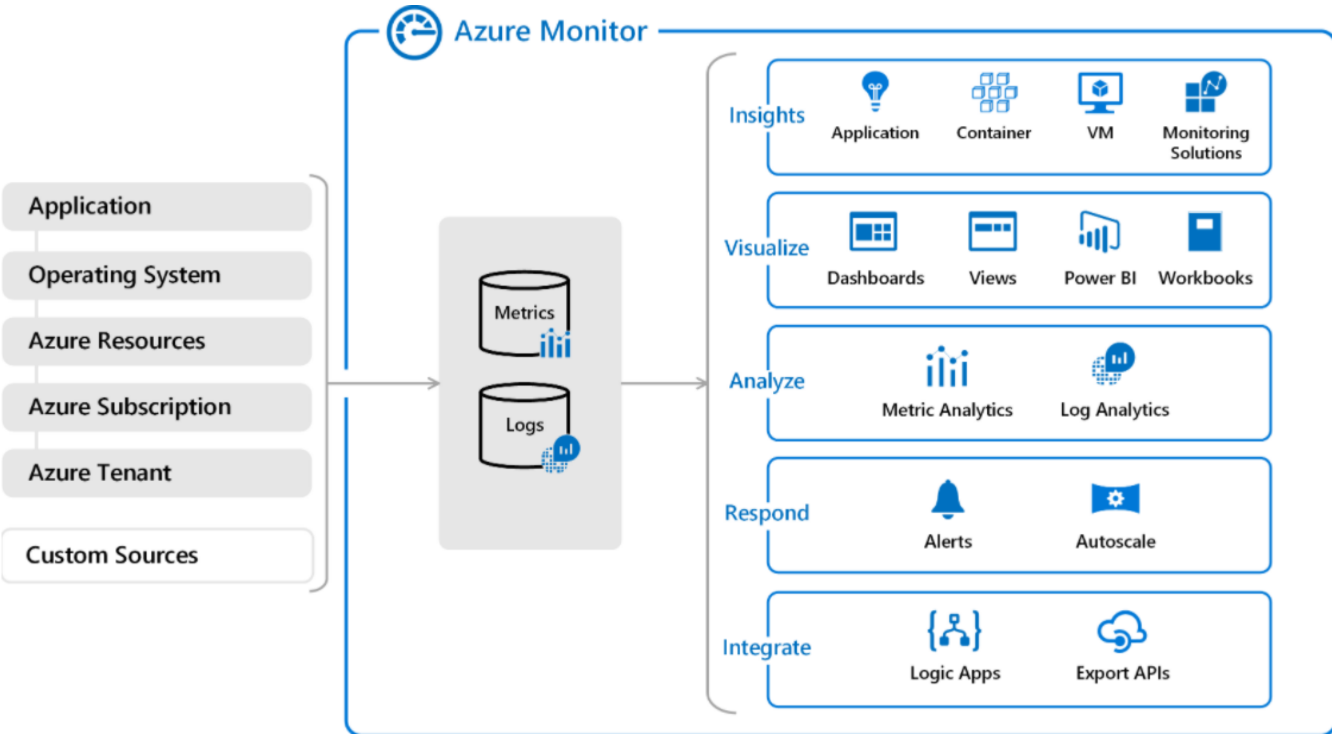
Area	Features	Decision
DDoS Protection	• DDoS protection @ edge	• Azure Basic DDoS Protection has been used



Feature	DDoS Protection Basic	DDoS Protection Standard
Active traffic monitoring & always on detection	Yes	Yes
Automatic attack mitigations	Yes	Yes
Availability guarantee	Azure region	Application
Mitigation policies	Tuned for Azure region traffic volume	Tuned for application traffic volume
Metrics & alerts	No	Real time attack metrics & diagnostic logs via Azure monitor
Mitigation reports	No	Post attack mitigation reports
Mitigation flow logs	No	NRT log stream for SIEM integration
Mitigation policy customizations	No	Engage DDoS experts
Support	Best effort	Access to DDoS Experts during an active attack
SLA	Azure region	Application SLA guarantee & cost protection

Security – Logging & Monitoring – Azure Monitor

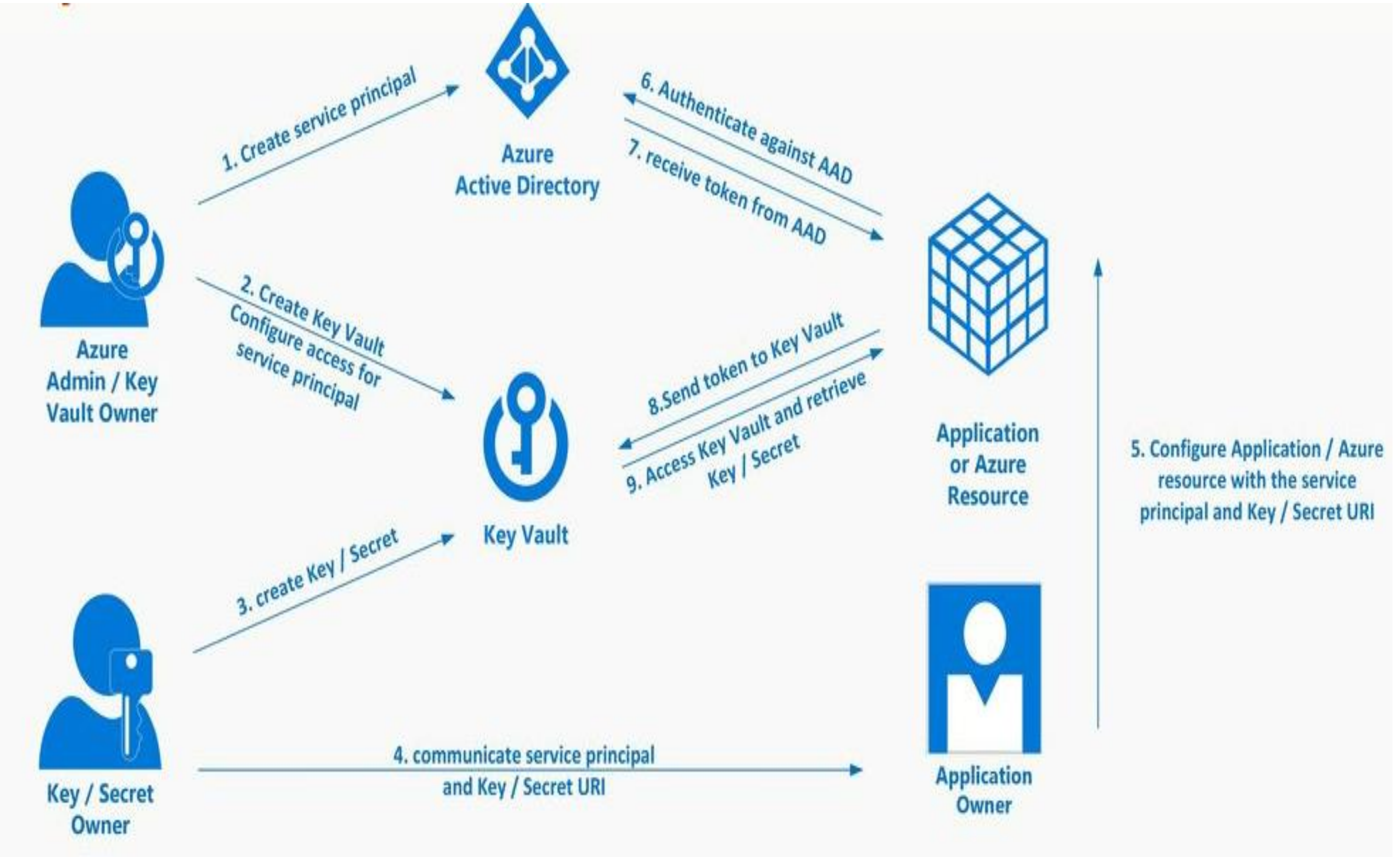
- Centralized Log Analytics workspace per environment to log all the events and metrics.
- SDR Reference Implementation has leveraged built-in cloud native tools Azure Monitor, Application Insights for Monitoring with its default configuration.



Security - Azure Key Vault

Key Information

- Key Vaults have been separated by environment since environments should not share any credentials.
- Access policies have been configured for each vault to determine what operations a user, group, app, or Service Principal can perform.
- Key Vaults have been locked to CanNotDelete. Purge protection and soft delete have been enabled for recovery in case of a malicious or accidental deletion attempt.
- SDR RI UI and API have been integrated with KeyVault to store or retrieve secrets like connection strings, credentials etc.



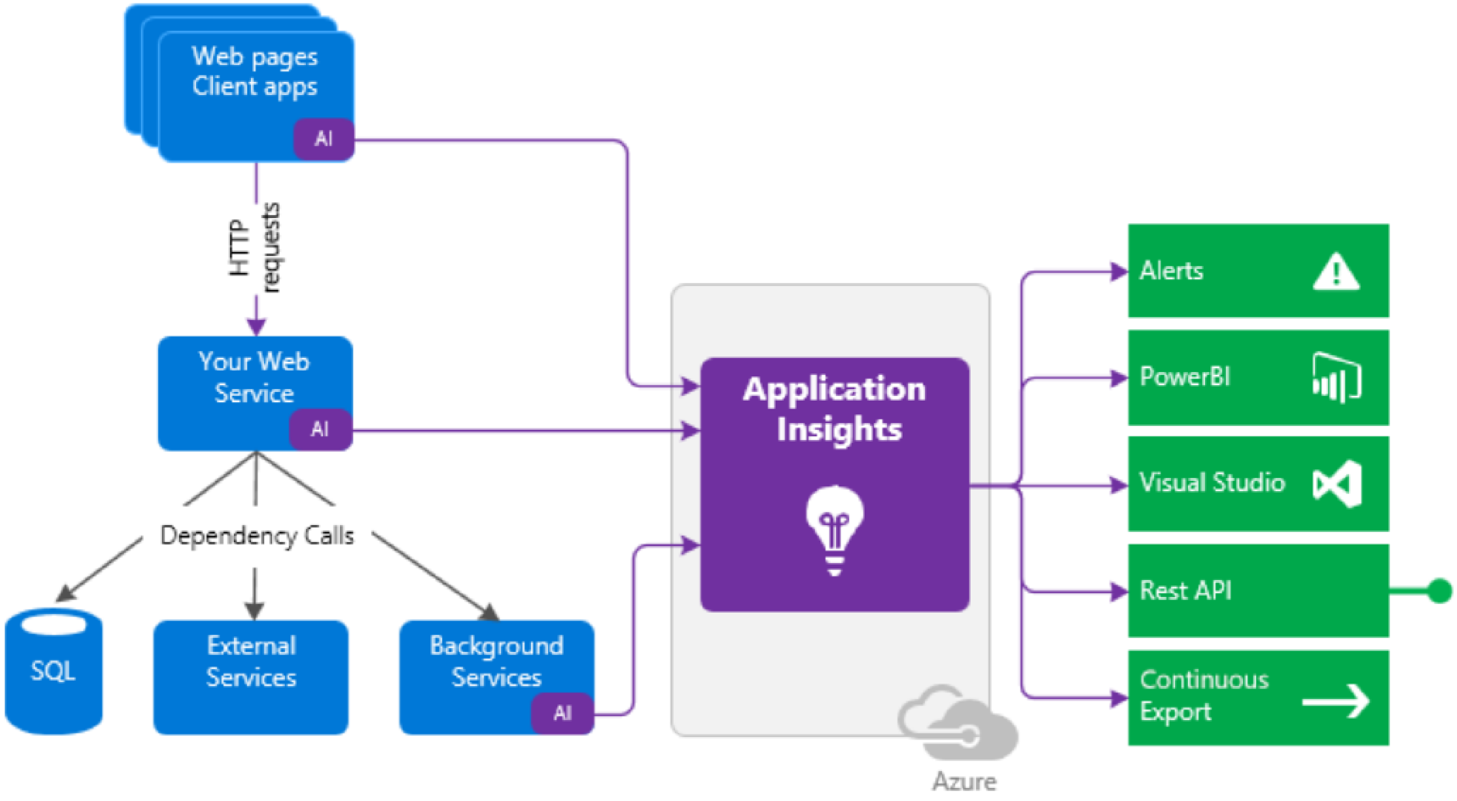
Azure Active Directory
1. Create service principal
6. Authenticate against AAD
7. receive token from AAD
Azure Admin / Key Vault Owner
2. Create Key Vault Configure access for service principal
8.Send token to Key Vault
9. Access Key Vault and retrieve Key / Secret
Application or Azure Resource
Key Vault
5. Configure Application / Azure resource with the service principal and Key / Secret URI
3. create Key / Secret
Key / Secret Owner
4. communicate service principal and Key / Secret URI
Application Owner

API Management

- The SDR RI APIs are exposed via API Management
- SDR RI UI and external API - Users can interact with APIs only through APIM.
- The SDR Reference Implementation architecture prevents calling APIs directly. All web service requests are channeled through the API Management layer.
- APIM developer portal is configured to allow vendors generate API keys for system-to-system integration with SDR APIs.

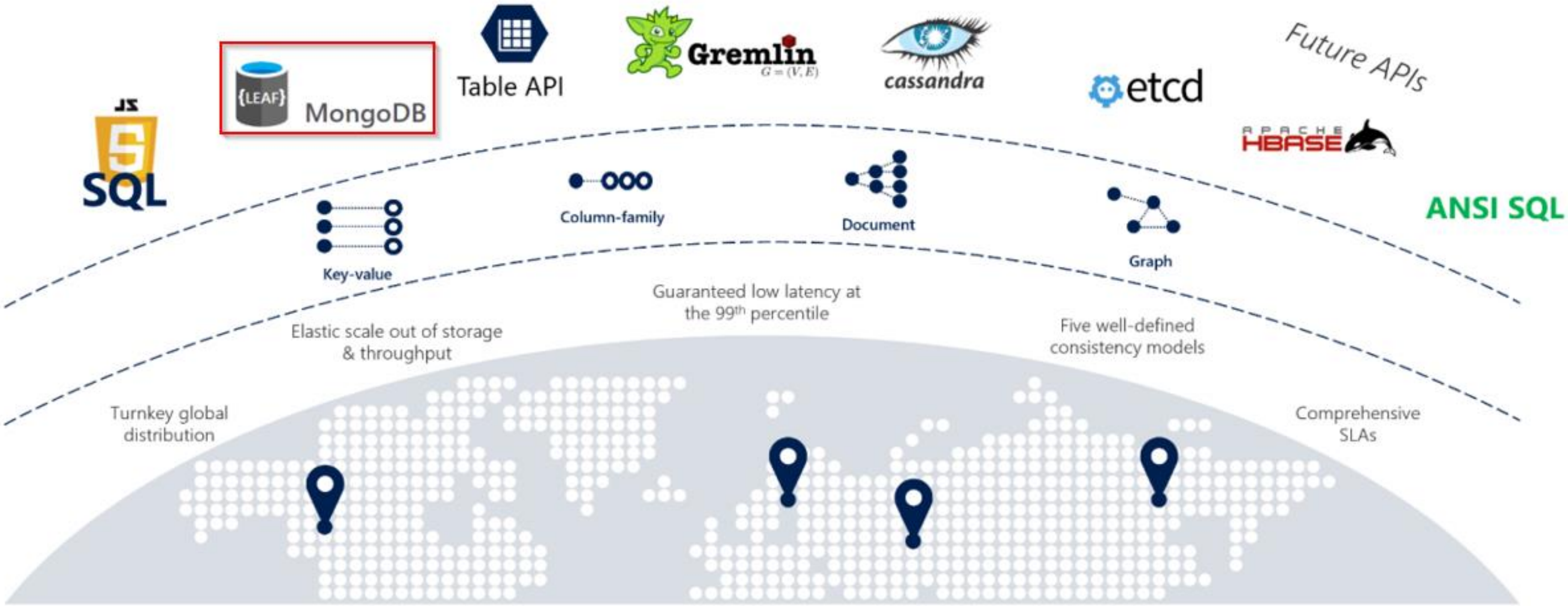
Application Insights

- SDR Reference Implementation utilizes 1 Application Insights per Environment. Application Insights, a feature of Azure Monitor, is an extensible Application Performance Management (APM) service for developers and DevOps professionals. It is used to monitor live applications.
- Automatically detects performance anomalies, and includes powerful analytics tools to help diagnose issues and to understand what users do with your app. It's designed to help continuously improve performance and usability.



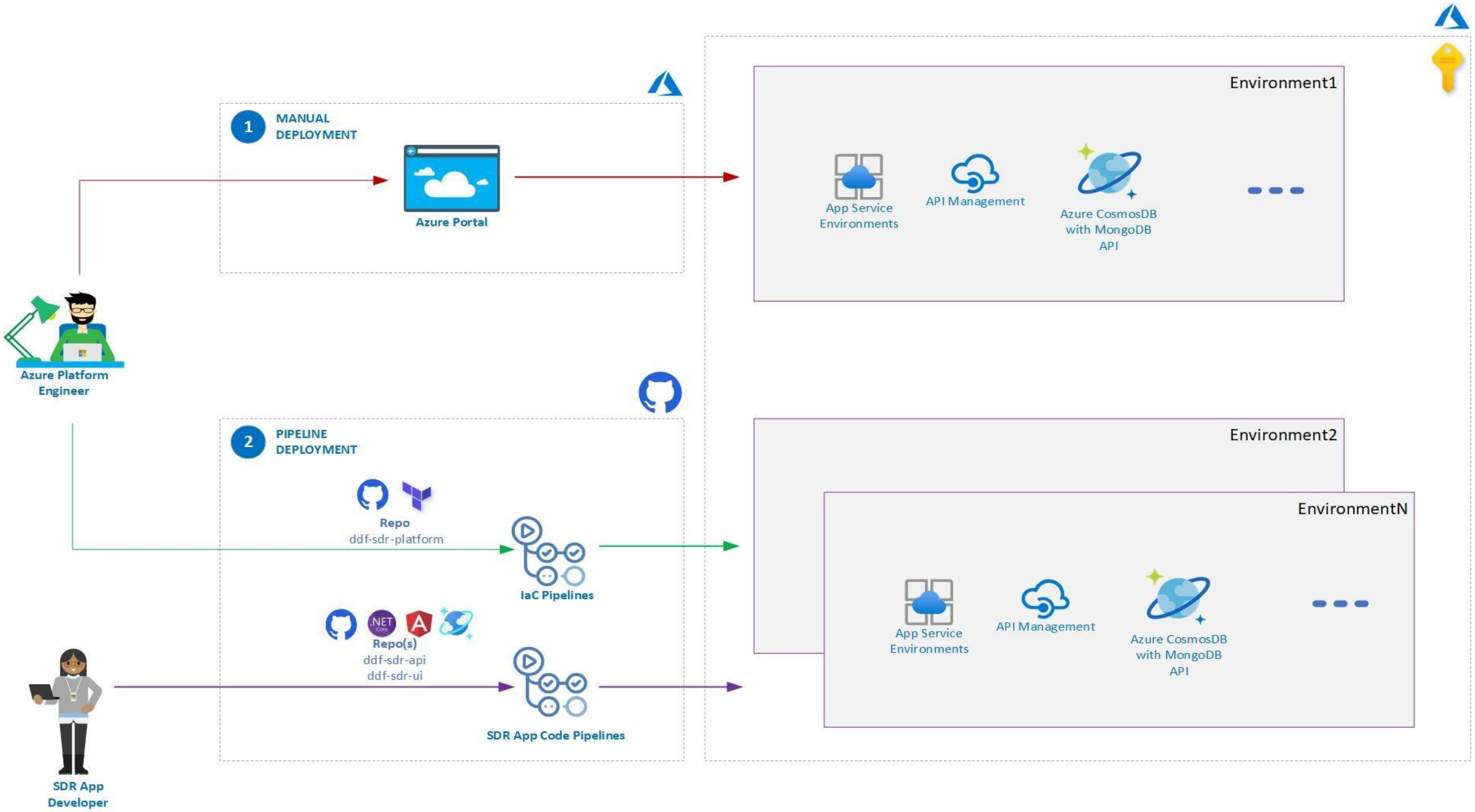
COSMOS DB

- SDR Reference Implementation is built on one Azure Cosmos DB API for MongoDB per Environment.



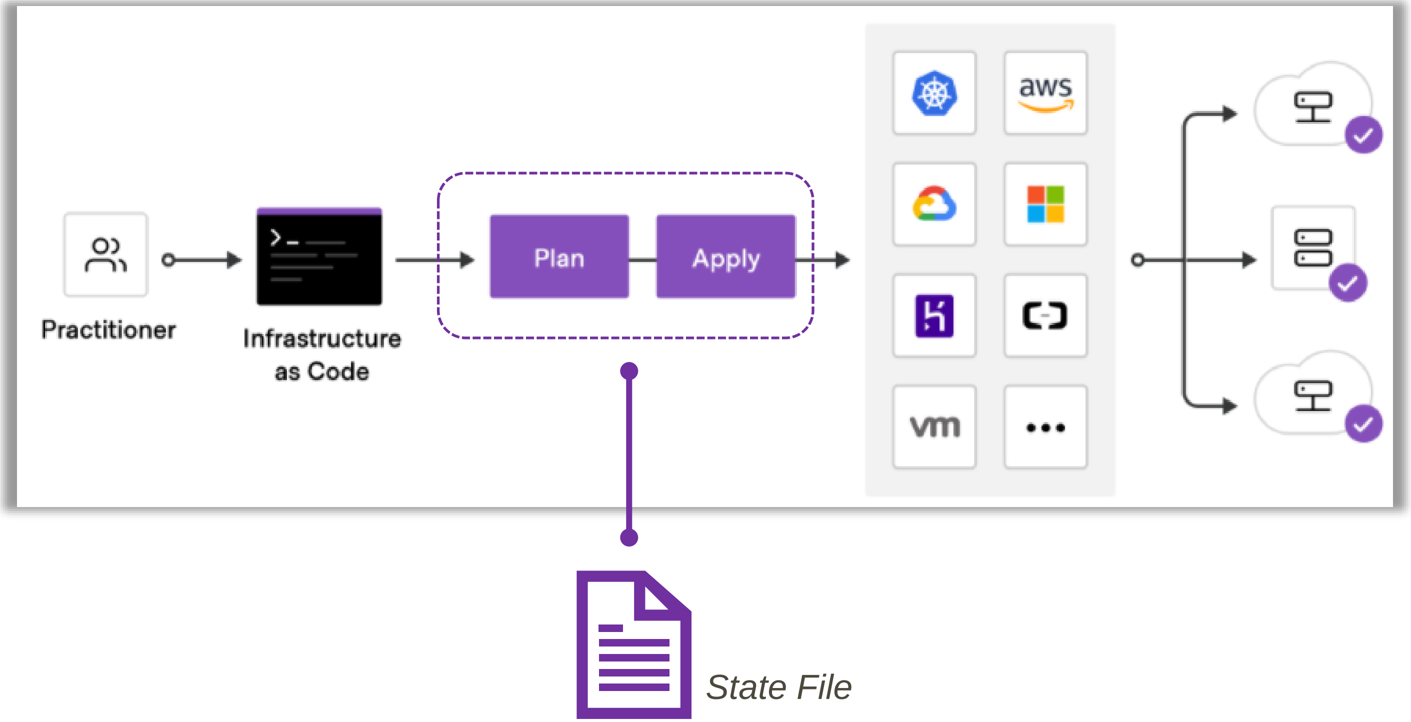
JS
SQL
{LEAF}
MongoDB
Table API
Gremlin
G = (V, E)
cassandra
etcd
Future APIs
APACHE HBASE
ANSI SQL
Key-value
Column-family
Document
Graph
Elastic scale out of storage & throughput
Guaranteed low latency at the 99th percentile
Five well-defined consistency models
Turnkey global distribution
Comprehensive SLAs

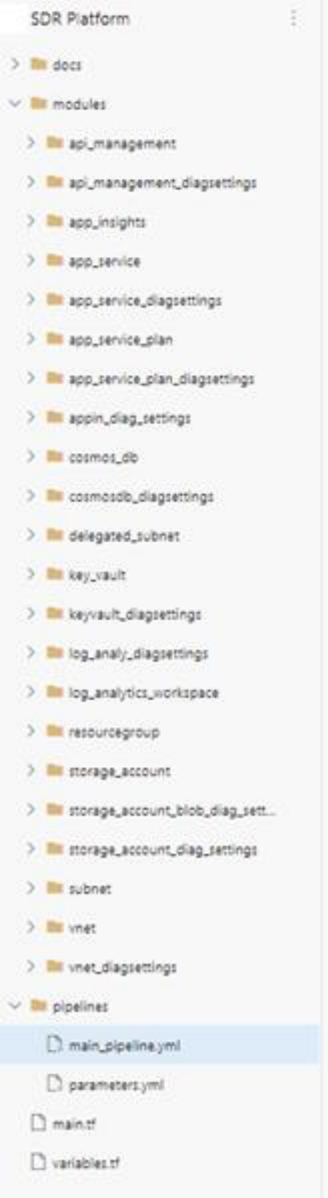
Infrastructure as Code (IaC) – Page 1/2



- **Environment Build Reliability**
  - *Idempotency*
  - *Immutability*
- **Speed**
- **Modularity**

Infrastructure as Code (IaC) – Page 2/2



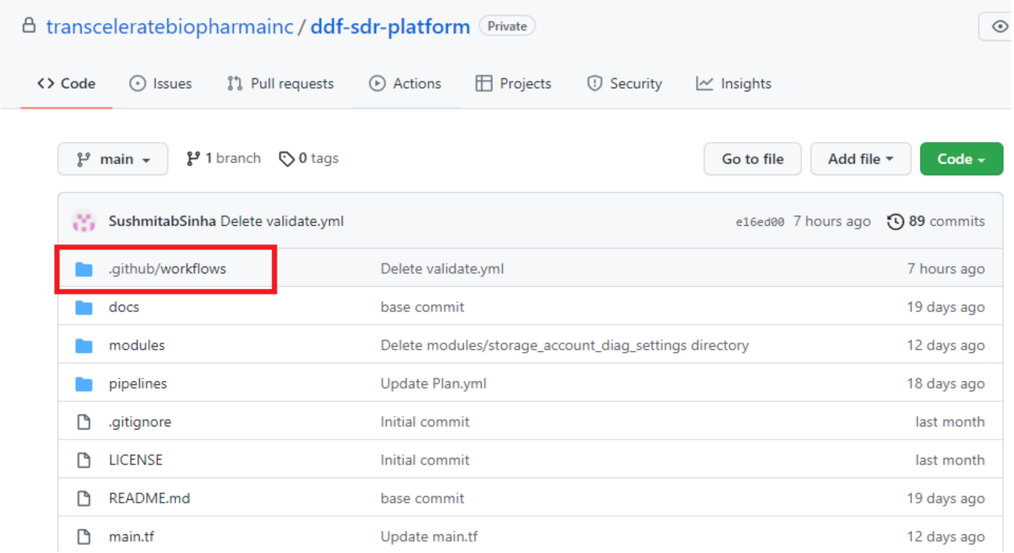

```
SDR Platform
> docs
v modules
  > api_management
  > api_management_diagsettings
  > app_insights
  > app_service
  > app_service_diagsettings
  > app_service_plan
  > app_service_plan_diagsettings
  > appin_diag_settings
  > cosmos_db
  > cosmosdb_diagsettings
  > delegated_subnet
  > key_vault
  > keyvault_diagsettings
  > log_analy_diagsettings
  > log_analytics_workspace
  > resourcegroup
  > storage_account
  > storage_account_blob_diag_sett...
  > storage_account_diag_settings
  > subnet
  > vnet
  > vnet_diagsettings
v pipelines
    main_pipeline.yml
    parameters.yml
  main.tf
  variables.tf
```

- **IaC Components**
  - *State File*
  - *Plan & Apply Code*
  - *Environment Variables*

- **Capabilities**
  - *Environment Build*
  - *Environment Destroy*

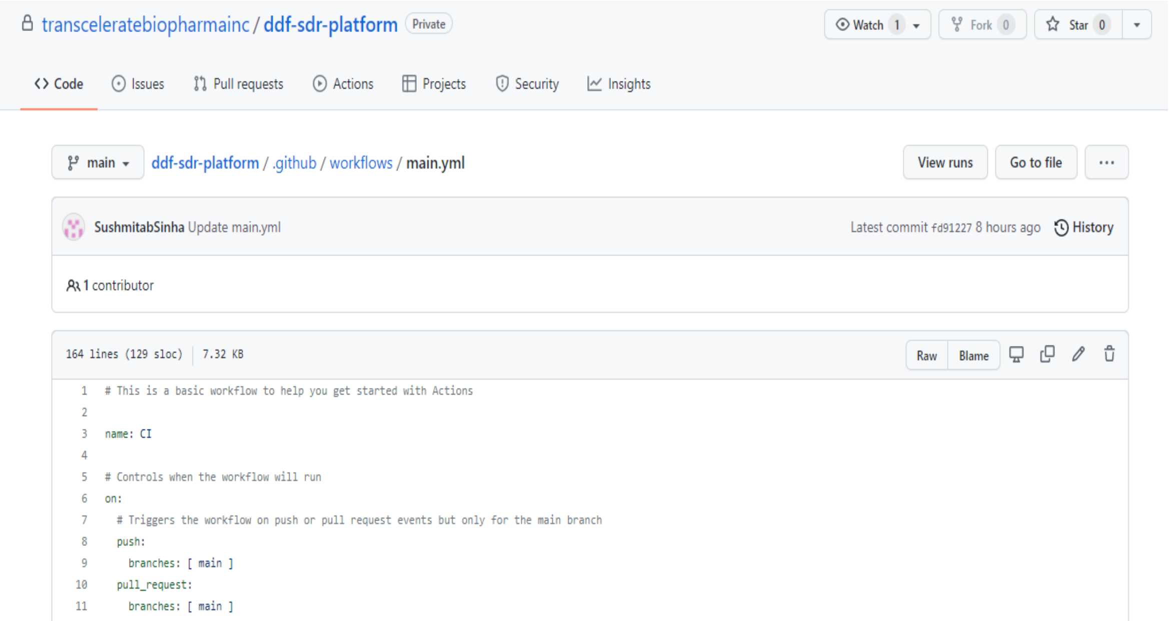
IAC Code Deployment

- SDR RI IAC code is deployed using GitHub Workflows & Actions:
- The GitHub actions are run directly in the GitHub environment which would automatically deploy the resources in Azure based on the configuration in the “main.yml” and “main.tf” file.



SushmitabSinha Delete validate.yml e16ed00 7 hours ago 89 commits

.github/workflows	Delete validate.yml	7 hours ago
docs	base commit	19 days ago
modules	Delete modules/storage_account_diag_settings directory	12 days ago
pipelines	Update Plan.yml	18 days ago
.gitignore	Initial commit	last month
LICENSE	Initial commit	last month
README.md	base commit	19 days ago
main.tf	Update main.tf	12 days ago



SushmitabSinha Update main.yml Latest commit fd91227 8 hours ago History

1 contributor

164 lines (129 sloc) | 7.32 KB

```
# This is a basic workflow to help you get started with Actions

name: CI

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the main branch
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
```

Operations - Decisions

Decision Area	Options	Decision	Rationale

Infrastructure Logging & Monitoring	• Azure Resources (IaaS, PaaS)
	  ◦ Application Logging & Monitoring
	  ◦ Endpoint Logging & Monitoring
		• Cloud Native Infrastructure Monitoring (Log Analytics & Azure Monitor)	• Diagnostic Monitoring
			• App Insights Monitoring
			• Log Analysis

Cosmos DB Backup	• Backup Interval • Backup Retention • Backup Storage Redundancy	• Continuous Restoration policy enabled. • Allows to do restore to any point of time within the last 30 days. • Geo-redundant backup storage enabled.	