****Technical Service Description

Managed Applications on Azure

Table des matières

[1 Definitions 1](#_Toc123118811)

[2 Overview of the Service 1](#_Toc123118812)

[2.1 Overall description 1](#_Toc123118813)

[2.2 Managed and Co-managed Service Strategy 1](#_Toc123118814)

[3 Managed Business Application on Azure 1](#_Toc123118815)

[3.1 Benefits 1](#_Toc123118816)

[3.2 Scope of Work 1](#_Toc123118817)

[3.3 Managing the dependencies 1](#_Toc123118818)

[3.4 Managing the external interfaces 1](#_Toc123118819)

[3.5 Managing the business functions 1](#_Toc123118820)

[3.6 Contribution of the Service Reliability Engineer 1](#_Toc123118821)

[3.7 Summary 1](#_Toc123118822)

[3.8 Pre-requisites 1](#_Toc123118823)

[3.9 Limitations 1](#_Toc123118824)

[3.10 Charging model 1](#_Toc123118825)

[3.11 Changes catalogue – in Tokens, per act 1](#_Toc123118826)

[4 The Service Reliability Engineer 1](#_Toc123118827)

[**4.1** **Deliverables** 1](#_Toc123118828)

[**4.2** **Limitations** 1](#_Toc123118829)

[**4.3** **Implications** 1](#_Toc123118830)

[**4.4** **Charging model** 1](#_Toc123118831)

[5 The co-management run delivery model 1](#_Toc123118832)

[6 Managed Cloud Native Services on Azure 1](#_Toc123118833)

[6.1.1 The cloud native services 1](#_Toc123118834)

[6.1.2 Tasks involved Cloud Native service management 1](#_Toc123118835)

[6.1.1 Table of tasks involved in the management a Cloud Native service 1](#_Toc123118836)

[6.1.2 Tooling used for cloud native managed services 1](#_Toc123118837)

[6.1.3 General pre-requisites to the run of managed services 1](#_Toc123118838)

[6.1.4 Criteria for the run of a managed cloud native service component 1](#_Toc123118839)

[7 The build of services & managed services on Azure 1](#_Toc123118840)

[7.1 The build and deployment scope of work 1](#_Toc123118841)

[7.2 Project management and coordination during the build phase 1](#_Toc123118842)

[7.3 Cloud Expert Services for migration 1](#_Toc123118843)

[7.4 Criteria for assessing the model of build for a resource 1](#_Toc123118844)

[7.4.1 Criteria for qualifying as “backend build” model a.k.a class 2 SoW for a resource: 1](#_Toc123118845)

[7.4.2 Criteria for qualifying as “operations build” model a.k.a class 4 SoW for a resource: 1](#_Toc123118846)

[7.4.3 Criteria for qualifying as “full build” model a.k.a class 5 SoW for a resource: 1](#_Toc123118847)

[7.4.4 Mitigation in case of pre-requisites or criteria not met: 1](#_Toc123118848)

[7.4.5 Charging model for build 1](#_Toc123118849)

[8 Detailed responsibilities and accountabilities 1](#_Toc123118850)

[9 Detailed build methodology 1](#_Toc123118851)

[9.1 Inputs to the build 1](#_Toc123118852)

[9.2 Initialization of the project: building the landing zone, IaC and pipelines on Azure 1](#_Toc123118853)

[9.3 Change management methodology 1](#_Toc123118854)

[9.4 Transition from build to run 1](#_Toc123118855)

[9.5 Security policies and access control 1](#_Toc123118856)

[10 Detailed description of the run tasks per cloud service (Extract) 1](#_Toc123118857)

[10.1 API Management 1](#_Toc123118858)

[10.1.1 Description 1](#_Toc123118859)

[10.1.2 Build to run service included in the OTC 1](#_Toc123118860)

[10.1.3 RUN services included in the MRC 1](#_Toc123118861)

[10.1.4 Charging model 1](#_Toc123118862)

[10.1.5 Changes catalogue – in Tokens, per act 1](#_Toc123118863)

[10.2 Application Gateway 1](#_Toc123118864)

[10.2.1 Description 1](#_Toc123118865)

[10.2.2 Build to run service included in the OTC 1](#_Toc123118866)

[10.2.3 RUN services included in the MRC 1](#_Toc123118867)

[10.2.4 Charging model 1](#_Toc123118868)

[10.2.5 Changes catalogue – in Tokens, per act 1](#_Toc123118869)

[10.3 Application Insights – basic monitoring with class 2 transition 1](#_Toc123118870)

[10.3.1 Description 1](#_Toc123118871)

[10.3.2 Build to run service included in the OTC 1](#_Toc123118872)

[10.3.3 RUN services included in the MRC 1](#_Toc123118873)

[10.3.4 Charging model 1](#_Toc123118874)

[10.3.5 Changes catalogue – in Tokens, per act 1](#_Toc123118875)

[10.4 Application service 1](#_Toc123118876)

[10.4.1 Description 1](#_Toc123118877)

[10.4.2 Build to run service included in the OTC 1](#_Toc123118878)

[10.4.3 RUN services included in the MRC 1](#_Toc123118879)

[10.4.4 Charging model 1](#_Toc123118880)

[10.4.5 Changes catalogue – in Tokens, per act 1](#_Toc123118881)

[10.5 Azure DNS 1](#_Toc123118882)

[10.5.1 Description 1](#_Toc123118883)

[10.5.2 Build to run service included in the OTC 1](#_Toc123118884)

[10.5.3 RUN services included in the MRC 1](#_Toc123118885)

[10.5.4 Charging model 1](#_Toc123118886)

[10.5.5 Changes catalogue – in Tokens, per act 1](#_Toc123118887)

[10.6 Azure Firewall 1](#_Toc123118888)

[10.6.1 Description 1](#_Toc123118889)

[10.6.2 Build to run service included in the OTC 1](#_Toc123118890)

[10.6.3 RUN services included in the MRC 1](#_Toc123118891)

[10.6.4 Charging model 1](#_Toc123118892)

[10.6.5 Changes catalogue – in Tokens, per act 1](#_Toc123118893)

[10.7 Azure Function 1](#_Toc123118894)

[10.7.1 Description 1](#_Toc123118895)

[10.7.2 Build to run service included in the OTC 1](#_Toc123118896)

[10.7.3 RUN services included in the MRC 1](#_Toc123118897)

[10.7.4 Charging model 1](#_Toc123118898)

[10.7.5 Changes catalogue – in Tokens, per act 1](#_Toc123118899)

[10.8 Azure storage 1](#_Toc123118900)

[10.8.1 Description 1](#_Toc123118901)

[10.8.2 Build to run service included in the OTC 1](#_Toc123118902)

[10.8.3 RUN services included in the MRC 1](#_Toc123118903)

[10.8.4 Charging model 1](#_Toc123118904)

[10.8.5 Changes catalogue – in Tokens, per act 1](#_Toc123118905)

[10.9 Content Delivery Network (CDN) 1](#_Toc123118906)

[10.9.1 Description 1](#_Toc123118907)

[10.9.2 Build to run service included in the OTC 1](#_Toc123118908)

[10.9.3 RUN services included in the MRC 1](#_Toc123118909)

[10.9.4 Charging model 1](#_Toc123118910)

[10.9.5 Changes catalogue – in Tokens, per act 1](#_Toc123118911)

[10.10 Event Hubs 1](#_Toc123118912)

[10.10.1 Description 1](#_Toc123118913)

[10.10.2 Build to run service included in the OTC 1](#_Toc123118914)

[10.10.3 RUN services included in the MRC 1](#_Toc123118915)

[10.10.4 Charging model 1](#_Toc123118916)

[10.10.5 Changes catalogue – in Tokens, per act 1](#_Toc123118917)

[10.11 Express Route 1](#_Toc123118918)

[10.11.1 Description 1](#_Toc123118919)

[10.11.2 Build to run service included in the OTC 1](#_Toc123118920)

[10.11.3 RUN services included in the MRC 1](#_Toc123118921)

[10.11.4 Charging model 1](#_Toc123118922)

[10.11.5 Changes catalogue – in Tokens, per act 1](#_Toc123118923)

[10.12 Front door 1](#_Toc123118924)

[10.12.1 Description 1](#_Toc123118925)

[10.12.2 Build to run service included in the OTC 1](#_Toc123118926)

[10.12.3 RUN services included in the MRC 1](#_Toc123118927)

[10.12.4 Charging model 1](#_Toc123118928)

[10.12.5 Changes catalogue – in Tokens, per act 1](#_Toc123118929)

[10.13 Key vault 1](#_Toc123118930)

[10.13.1 Description 1](#_Toc123118931)

[10.13.2 Build to run service included in the OTC 1](#_Toc123118932)

[10.13.3 RUN services included in the MRC 1](#_Toc123118933)

[10.13.4 Information on Azure service 1](#_Toc123118934)

[10.13.5 Charging model 1](#_Toc123118935)

[10.13.6 Changes catalogue – in Tokens, per act 1](#_Toc123118936)

[10.14 Load Balancer 1](#_Toc123118937)

[10.14.1 Description 1](#_Toc123118938)

[10.14.2 Build to run service included in the OTC 1](#_Toc123118939)

[10.14.3 RUN services included in the MRC 1](#_Toc123118940)

[10.14.4 Charging model 1](#_Toc123118941)

[10.14.5 Changes catalogue – in Tokens, per act 1](#_Toc123118942)

[10.15 Log Analytics – basic monitoring with class 2 transition 1](#_Toc123118943)

[10.15.1 Description 1](#_Toc123118944)

[10.15.2 Build to run service included in the OTC 1](#_Toc123118945)

[10.15.3 RUN services included in the MRC 1](#_Toc123118946)

[10.15.4 Charging model 1](#_Toc123118947)

[10.15.5 Changes catalogue – in Tokens, per act 1](#_Toc123118948)

[10.16 Logic App 1](#_Toc123118949)

[10.16.1 Description 1](#_Toc123118950)

[10.16.2 Build to run service included in the OTC 1](#_Toc123118951)

[10.16.3 RUN services included in the MRC 1](#_Toc123118952)

[10.16.4 Charging model 1](#_Toc123118953)

[10.16.5 Changes catalogue – in Tokens, per act 1](#_Toc123118954)

[10.17 Network Security Group - Network and Security management services 1](#_Toc123118955)

[10.17.1 Description 1](#_Toc123118956)

[10.17.2 Charging model 1](#_Toc123118957)

[10.17.3 Changes catalogue – in Tokens, per act 1](#_Toc123118958)

[10.18 Service Fabric 1](#_Toc123118959)

[10.18.1 Description 1](#_Toc123118960)

[10.18.2 Build to run service included in the OTC 1](#_Toc123118961)

[10.18.3 RUN services included in the MRC 1](#_Toc123118962)

[10.18.4 Charging model 1](#_Toc123118963)

[10.18.5 Changes catalogue – in Tokens, per act 1](#_Toc123118964)

[10.19 Site Recovery 1](#_Toc123118965)

[10.19.1 Description 1](#_Toc123118966)

[10.19.2 Build to run service included in the OTC 1](#_Toc123118967)

[10.19.3 RUN services included in the MRC 1](#_Toc123118968)

[10.19.4 Charging model 1](#_Toc123118969)

[10.19.5 Changes catalogue – in Tokens, per act 1](#_Toc123118970)

[10.20 Traffic Manager 1](#_Toc123118971)

[10.20.1 Description 1](#_Toc123118972)

[10.20.2 Build to run service included in the OTC 1](#_Toc123118973)

[10.20.3 RUN services included in the MRC 1](#_Toc123118974)

[10.20.4 Charging model 1](#_Toc123118975)

[10.20.5 Changes catalogue – in Tokens, per act 1](#_Toc123118976)

[10.21 Virtual Machines and OS 1](#_Toc123118977)

[10.21.1 Description 1](#_Toc123118978)

[10.21.2 Build to run service included in the OTC 1](#_Toc123118979)

[10.21.3 RUN services included in the MRC 1](#_Toc123118980)

[10.21.4 Charging model 1](#_Toc123118981)

[10.21.5 Changes catalogue – in Tokens, per act 1](#_Toc123118982)

[10.22 VPN Gateway 1](#_Toc123118983)

[10.22.1 Description 1](#_Toc123118984)

[10.22.2 Build to run service included in the OTC 1](#_Toc123118985)

[10.22.3 RUN services included in the MRC 1](#_Toc123118986)

[10.22.4 Charging model 1](#_Toc123118987)

[10.22.5 Changes catalogue – in Tokens, per act 1](#_Toc123118988)

[10.23 Web Application Firewall 1](#_Toc123118989)

[10.23.1 Description 1](#_Toc123118990)

[10.23.2 Build to run service included in the OTC 1](#_Toc123118991)

[10.23.3 RUN services included in the MRC 1](#_Toc123118992)

[10.23.4 Charging model 1](#_Toc123118993)

[10.23.5 Changes catalogue – in Tokens, per act 1](#_Toc123118994)

[**10.24** **Azure Database for MySQL** 1](#_Toc123118995)

[**10.24.1** **Description** 1](#_Toc123118996)

[**10.24.2** **Build to run service included in the OTC** 1](#_Toc123118997)

[**10.24.3** **RUN services included in the MRC** 1](#_Toc123118998)

[**10.24.4** **Charging model** 1](#_Toc123118999)

[**10.24.5** **Changes catalogue – in Tokens, per act** 1](#_Toc123119000)

[10.25 Azure Database for PostgreSQL 1](#_Toc123119001)

[10.25.1 Description 1](#_Toc123119002)

[10.25.2 Build to run service included in the OTC 1](#_Toc123119003)

[10.25.3 RUN services included in the MRC 1](#_Toc123119004)

[10.25.4 Charging model 1](#_Toc123119005)

[10.25.5 Changes catalogue – in Tokens, per act 1](#_Toc123119006)

[10.26 Azure SQL Database 1](#_Toc123119007)

[10.26.1 Description 1](#_Toc123119008)

[10.26.2 Build to run service included in the OTC 1](#_Toc123119009)

[10.26.3 RUN services included in the MRC 1](#_Toc123119010)

[10.26.4 Charging model 1](#_Toc123119011)

[10.26.5 Changes catalogue – in Tokens, per act 1](#_Toc123119012)

[10.27 Azure Cosmos DB 1](#_Toc123119013)

[10.27.1 Description 1](#_Toc123119014)

[10.27.2 Build to run service included in the OTC 1](#_Toc123119015)

[10.27.3 RUN services included in the MRC 1](#_Toc123119016)

[10.27.4 Charging model 1](#_Toc123119017)

[10.27.5 Changes catalogue – in Tokens, per act 1](#_Toc123119018)

[10.28 Azure Database for MariaDB 1](#_Toc123119019)

[10.28.1 Description 1](#_Toc123119020)

[10.28.2 Build to run service included in the OTC 1](#_Toc123119021)

[10.28.3 RUN services included in the MRC 1](#_Toc123119022)

[10.28.4 Charging model 1](#_Toc123119023)

[10.28.5 Changes catalogue – in Tokens, per act 1](#_Toc123119024)

[10.29 Azure Managed Instance for Apache Cassandra 1](#_Toc123119025)

[10.29.1 Description 1](#_Toc123119026)

[10.29.2 Build to run service included in the OTC 1](#_Toc123119027)

[10.29.3 RUN services included in the MRC 1](#_Toc123119028)

[10.29.4 Charging model 1](#_Toc123119029)

[10.29.5 Changes catalogue – in Tokens, per act 1](#_Toc123119030)

[10.30 Azure Cache For Redis 1](#_Toc123119031)

[10.30.1 Description 1](#_Toc123119032)

[10.30.2 Service Tiers 1](#_Toc123119033)

[10.30.3 Build to run service included in the OTC 1](#_Toc123119034)

[10.30.4 RUN services included in the MRC 1](#_Toc123119035)

[10.30.5 Charging model 1](#_Toc123119036)

[10.30.6 Changes catalogue – in Tokens, per act 1](#_Toc123119037)

[10.31 Azure SQL Managed Instance 1](#_Toc123119038)

[10.31.1 Description 1](#_Toc123119039)

[10.31.2 Build to run service included in the OTC 1](#_Toc123119040)

[10.31.3 RUN services included in the MRC 1](#_Toc123119041)

[10.31.4 Charging model 1](#_Toc123119042)

[10.31.5 Changes catalogue – in Tokens, per act 1](#_Toc123119043)

[11 End of the document 1](#_Toc123119044)

# Definitions

**Git** refers to code versioning software

**CI / CD** refers to continuous Integration, Continuous Deployment chain and tooling

**IaC** refers to Infrastructure As Code.

**MRC** refers to Monthly Recurring Charge

**OBS** refers to Orange Business Services

**OTC** refers to One Time Charge

**RACI** refers to definition of responsibilities: Responsible, Accountable, Contributor, Informed

**SIEM** refers to Security Information and Event Management

# Overview of the Service

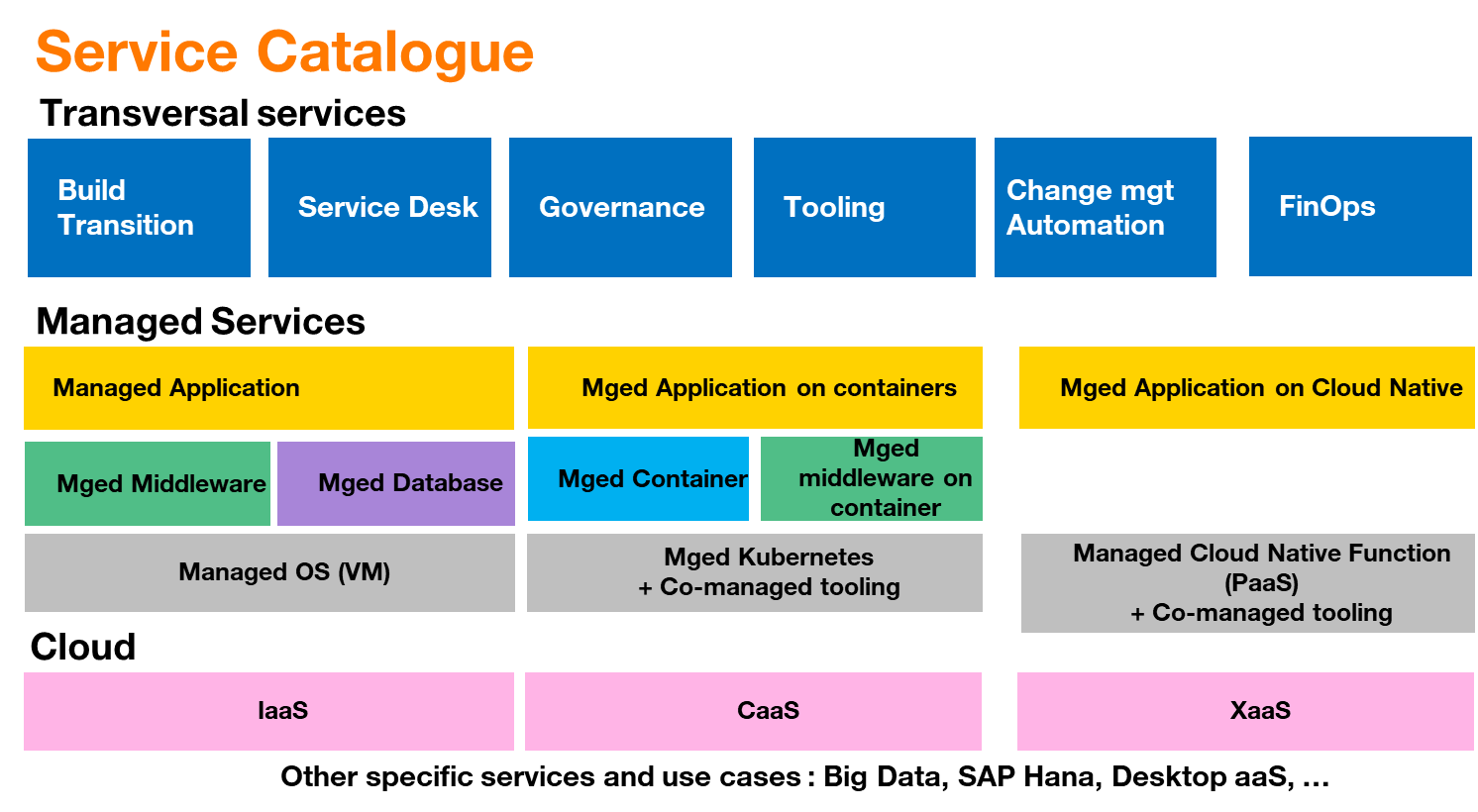
## Overall description

The document is an appendix to the Managed Application Service Description. It provides service description and further details for the

* MANAGED BUSINESS APPLICATION ON AZURE
* MANAGED CLOUD NATIVE SERVICES ON AZURE

The description adds to the managed services already described in the other document called Managed Applications Service Description:

* MANAGED OS FOR CLOUD SERVERS
* MANAGED DATABASE
* MANAGED MIDDLEWARE
* MANAGED KUBERNETES
* MANAGED CONTAINER
* MANAGED SAP
* CAASCAD SERVICE
* LOG AS A SERVICE (LAAS)



## Managed and Co-managed Service Strategy

Orange Business Services can flexibly support Customers in different ways for their use of the cloud.

The **Fully Managed** option is a model by which OBS takes care of all the deployment, the supervision, the operations of the workloads of the Customer. The Customer has the responsibility to provide a fully tested workload. This model is best suitable for the stable applications and workloads with a low amount of modifications and transformation. It provides operational efficiency.

The **Co-managed** model, whereby the Customer and OBS share the responsibilities the deployment, the supervision, the operations of the applications and workloads. In this model, the Customer is taking care of the development and testing of the application. The Customer can propose deployment templates thanks to change process. OBS is responsible for the monitoring and maintenance 24 x 7 inclusive of non-business hours and non-business days and / or 8 x 5 for less critical workloads. Both collaborate using a Git referential, a Continuous Integration and Deployment Chain and a shared tooling for monitoring, logging, alerting, dashboards and communication. The **Service Reliability Engineers** will take part into Customer’s development team to contribute to the readiness of the operability, and its continuous improvement. This model is an efficient compromise for getting the development agility, transformation of the application and frequent push to production, while keeping an efficient service assurance through delegation to OBS and preserving the critical development resources towards most value-added development tasks. The Co-managed model can be complemented with Cloud Center of Excellence or Cloud Expertise.

The **Full DevOps** model, whereby the development team of the Customer is fully responsible for the development, the deployment, the supervision, the operations of the workloads. In that model, OBS can propose professional services to the Customer, typically under the form of a **Cloud Center of Excellence or Cloud Expertise** to help the Customer setting the DevOps pipelines, tooling, landing zone and build to run activity. In this model, OBS can also propose the Service Reliability Engineering. This model is best suitable for applications and workloads subject to intensive development and modifications with frequent deployments to production. Drawback is that the developers need to be mobilized in 24x7 and need to take care for some operational aspects during the whole project life.

Each customer case is specific, yet the Co-managed model is taking momentum especially with the use of Cloud Native functions / PaaS functions of the clouds.

The service description in this document **applies to both full managed and co-managed services**. During the presale or consulting phase, the Customer and OBS will agree to the model of managed services required and adapt the RACI accordingly. This may vary from one Customer to another, from one service or application to another.

# Managed Business Application on Azure

## Benefits

The customer can delegate the supervision of the business applications services for an always-on service 24h/7d. The customer development teams can continue to evolve the software and architecture of the business application. In the co-managed mode, OBS Service Reliability Engineer will participate to customer’s scrum team meetings to contribute to the enablers necessary to manage the business application.

The customer and the SRE can concentrate on defining the observability and management procedures for the business functions and rely on standard managed services for the management of underlying dependencies.

## Scope of Work

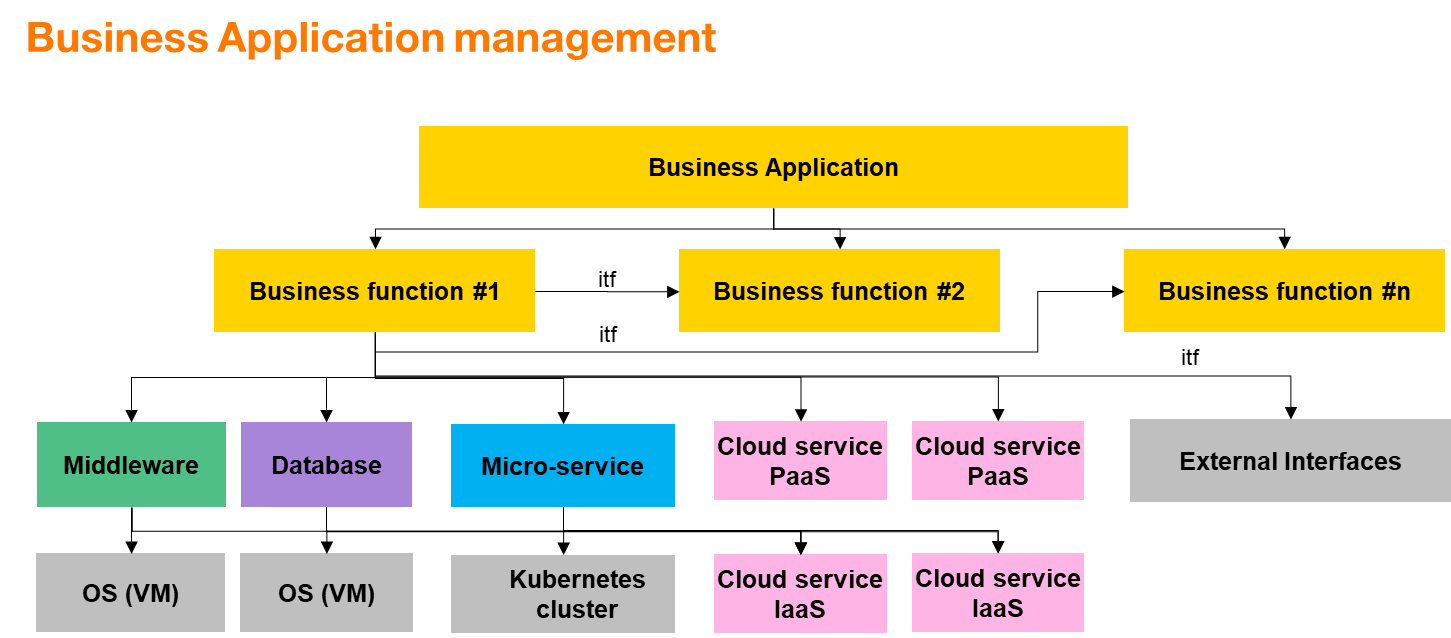
The Scope of Work for co-management of the Business Application is defined between the Customer and OBS.

The architecture of the Application is explained by the customer to OBS expert to identify which components need to be supervised and maintained among:

The business functions

Their dependencies on interfaces with other Application business functions and with external services.

Their dependencies on operating systems, middleware, databases, micro-services, Kubernetes services, cloud services, big data services



The main assumption for the co-management of Business Application is that the software is coded by the developers of the Customer or by a software 3rd party supplier to the customer. The customer is accountable and responsible - by himself or through his supplier - for the business application software and architecture maintenance, for the business application functioning and testing on the cloud environment prior to transition to OBS managed service.

## Managing the dependencies

For managing the business functions properly, the dependencies need to be managed. The catalogue of managed services includes pre-defined work units for the known middleware, databases, micro-services Kubernetes clusters, OS and native cloud services. The customer and OBS identify the necessary dependencies to be managed and add them to the Scope of Work as per service definition for the Manage Application catalogue of service (please refer to this document and to Managed Application Service Description).

## Managing the external interfaces

Dependencies may include interfaces towards external systems or interfaces towards other business functions. Those shall be supervised as well to rapidly detect and identify root cause. The responsibility for repairing the external system is not part of the scope of work.

## Managing the business functions

For the business applications functions, the scope of work shall be established based on the following inputs and deliverables from the customers:

How is the business function supervised? what is the RACI between the customer and OBS.

What is the criticality of the business function for the service?

What are the known issues? what procedure shall be used to recover?

How is the business function recovered in case of failure? Is it based on redeployment from Infra as Code? Is it based on restore from backup? what is the procedure?

Are there specific routines to be run?

How is the business function created and deployed? What is the chain dev, pre-prod, prod? what is the RACI between the customer and OBS on the various environments.

What are the security policies to be applied and firewalling rules?

Is a disaster plan needed? How is it achieved?

Are there other services required from OBS? advisory, health check, performance, capacity?

What is the frequency of incidents and changes on the element?

What is the frequency of release roll-out?

What is the maturity of the element?

Since the specificities of the Business Applications are specific knowledge of the customer and its 3rd party supplier, the customer is responsible for the level 3 support for the business application (potentially through its 3rd party supplier).

OBS level 1 and level 2 tasks consists in:

* supervising the business functions agreed in the scope of work
* applying the remediation procedures if an incident occurs
* resolving an incident on a managed dependence
* notifying and escalading to customer’s level 3 if resolution is not possible thanks to the procedure

## Contribution of the Service Reliability Engineer

As not all metrics, alerts, remediation procedures for managing the business application may not be known at once by the customer’s development team at the beginning of the project. The Service Reliability Engineer participates (remotely) to customer’s development team scrum meetings to contribute to the observability and automation specification and/or development. The SRE brings his expertise and experience about running applications on the cloud and facilitates the transition to the run.

As time goes, the customer’s development team and the SRE:

* Identify new pertinent metrics to be monitored by the run team
* Develop new exporters to have them supervised
* Document and automates troubleshooting of known problems
* Implement dashboards to be used by the development team for trend and behavior analysis
* Implement procedures based on logs for troubleshooting
* Implement various test routines

The ultimate goal is the improvement of the reliability and availability of the business application.

## Summary

The following table summarizes the service

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Service | Type | Configuration | Monitoring and alerts configured in Azure Monitor | Backup configured in Azure Backup | Recovery procedure | Patch management | Antivirus management | Specificities |
| Business Function supervision | Managed | Deployment, redeployment of the Business Application is based on Time and Material. Pre-requisite: Image or deployment script provided by the customer. The Business App has been successfully tested on the infrastructure prior to transition to run. | Metrics exporters, alerters to Azure monitor or Prometheus provided by customer as a prerequisite. | Backup and restore is an option. Customer to identify backup procedures necessary to protect business application data. And confirm whether backup of the underlying components is sufficient or not. | Troubleshooting and recovery procedure provided by the Customer. Procedure shall last less than15 min. Otherwise would be charged time based. The Customer is performing Level 3 support. | Customer is responsible for the software and software patching. | Customer is responsible for the software antivirus of the Business Application. | Optional: Scope of work to be defined with Customer Pre-requisite: dependencies shall be managed. |
| Supervision of an External Interface | Managed | A pre-requisite is that the external interface is exposed and reachable. Out of scope of the Managed Service. | A part of the software or a probe tests the availability of the external interface. | n/a | Customer is notified when the external interface down. The support of the external interface is out of scope of MA service. | n/a | n/a | n/a |
| Cloud Services dependencies | Managed | Refer to each cloud managed service (as per catalogue) on which the Business App is dependant. | | | | | | |

## Pre-requisites

The architecture of the application and deployment on the cloud shall be defined. Architecture is out of scope of the service.

The application shall be deployed and tested by the customer prior to transitioning to the run team. Typically, successful testing in pre-production, with a pre-production environment iso-production.

The business application exports metrics towards Azure Monitor or an agreed Prometheus.

The data backup strategy and disaster recovery strategy shall be provided by the customer.

The troubleshooting and service restoration procedures shall be provided by the customer.

Whereas a procedure requires logs or dashboard those shall have been developed by the customer prior to the service.

A remedial procedure on incident shall not last more than 15 minutes. Beyond, that time amount, the effort would be charged on time base.

Customer shall have subscribed to the managed applications service for the underlying components on which the business application is dependent.

## Limitations

The services needed from the Cloud Service Providers for observability, logs, monitoring, backup are not included in the service and therefore are charged as part of CSP subscription.

The business application software and Third-party Application Maintenance are out of scope. Application patching, vulnerability, virus free is customers responsibility.

The scope of work of the Managed Business Application service and OBS’s responsibility with regards to security is limited to configuration of the CSP firewalls and policy groups as per specifications by the customer. Should more security services be required, it shall be part of an optional mutually agreed scope of work.

Business Application end-users are not managed nor supported.

Application Performance Management is a specific Scope of Work and Quote.

Build, pipeline and deployment of the application is out of scope of the standard work unit. A specific scope of work shall be established.

## Charging model

|  |  |
| --- | --- |
| Service | Work Unit |
| Service Reliability Engineer | Time and material |
| Business function – supervision – low priority | Per supervision source |
| Business function – supervision - standard | Per supervision source |
| Business function – supervision - critical | Per supervision source |
| External interface – supervision – low priority | Per supervision source |
| External interface – supervision - standard | Per supervision source |
| External interface – supervision - critical | Per supervision source |
| Data backup | Scope of Work |
| Disaster Recovery | Scope of Work |
| Cloud Services dependencies: OS, middleware, database, Kubernetes, microservices, big data | Work Unit of the managed service catalogue |
| Incident raised by customer | Per incident ticket |

## Changes catalogue – in Tokens, per act

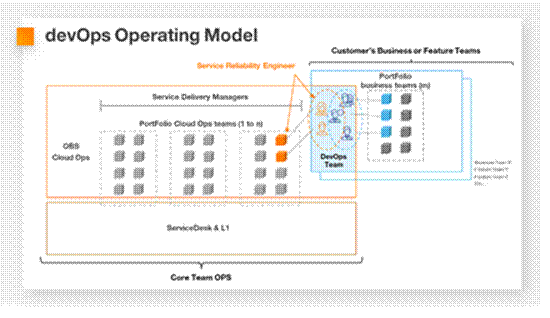
|  |  |
| --- | --- |
| Changes examples | Effort |
| Adding a new alarm | On quote or estimation in tokens based on time spent. Additional recurring work unit |
| Deploying the business application | On quote or estimation in tokens based on time spent |
| Adding a new troubleshooting procedure to the operational knowledge database | On quote or estimation in tokens based on time spent |
| Troubleshooting beyond 15 mins due to lengthy procedure | Estimation in tokens based on over-time spent |
|  |  |

# The Service Reliability Engineer

The Service Reliability Engineer is a key actor of the managed services on public clouds especially in the DevOps model and co-managed model.

The SRE is a named expert, knowledgeable about operations and software engineering, simultaneously participating to the run of the managed service within OBS operations team and working closely with the customer’s development team.

The SRE works closely with Customer's development Team to identify and implement the observability indicators, automation of operations and infrastructure as code to meet business needs. He contributes his expertise to the development team for delivering the enablers necessary to a reliable run. On longer term, the SRE contributes to continuous improvement of the reliability of the business application and its operations.

****

The SRE participates (remotely) to regular meetings with Application owners for continuous improvement Alignments.

* 1. **Deliverables**

The SRE contributes together with the development team to the following deliverables:

Guidelines for DevOps automation (Infra as Code, Integration, Blue-Green deployment, etc.) according to customer’s team maturity

Infra as Code necessary to deploy / redeploy the resources in case of service loss or misconfiguration

Identification and implementation of observability metrics necessary to monitor the business activity

Define and manage SLO, SLI

Implementation of automated dashboards allowing analysis of metrics and trends. Pieces of advice for the tooling for implementing them.

Identification of alarms / thresholds on metrics and alarm collection mechanism

Identification of necessary backup procedures and security measures for the application and data to meet customer's needs

Write-up of main procedures necessary to handle the known incidents. Procedures which will be handoff to the level 1 & 2 core operation teams.

Simple procedures are typically integrated in the infrastructure as code to accelerate the remedial actions.

Review/ validation of technical procedures for the changes proposed by the Service Delivery Manager for inclusion in the change catalogue.

Identification and implementation of log collection to detect anomalies and ease troubleshooting for the business application. Setup of automated correlations and alerts from logs analysis.

Cold analysis of dashboards, logs for preventive maintenance when requested.

Configuration of security tooling and SIEM.

Definition and write-up of recurring check procedures when necessary.

Criteria for “go” to pre-production. RACI between the customer and OBS for the deployment to pre-preproduction. Automation of deployment if requested.

Criteria for “go” to production taking into consideration technical and business constraints (deployment time, particular events, etc.). RACI between the customer and OBS for the deployment to production. Automation of deployment if requested.

* 1. **Limitations**

The build and design of the architecture, including disaster recovery, HLD and LLD is the responsibility of the customer or of an architect i.e. the Technical Design Authority. Following their validation by OBS, the SRE maintains architecture, HLD and LLD during RUN phase and identifies the necessary updates in terms of fault tolerance, auto healing, resilience and reliability to meet new business needs.

* 1. **Implications**

Customer’s development team decides jointly with the SRE of the improvements to bring to the management and reliability of the application. Such decision could lead to an additional use of cloud services and to additional scope of work for operations team (additional business functions monitored, additional resources monitored, additional backup routines) which may lead to additional monthly recurring charges.

* 1. **Charging model**

|  |  |
| --- | --- |
| Service | Work Unit |
| Service Reliability Engineer | Time and material (or Change Tokens) |
| Additional CSP services used | Additional recurring fees (CSP contract) |
| Additional Managed Services scope | Additional recurring fees for Managed Services |

# The co-management run delivery model

The scope of work and the delivery model for the run is established during the pre-sales phase depending on customer’s needs. The present chapter establishes guidelines which will be adjusted depending on the customer scope of work.

The Level 3 are the experts about the service component. They are the most knowledgeable to troubleshoot and resolve an incident on the service component. They implement the observability, the alerting and the procedures for troubleshooting, the backup and the procedure for repairing the service component. They validate the procedures and handoff those procedures to the Level 2. They troubleshoot incidents and problems in last resort when the Level 2 cannot fix it.

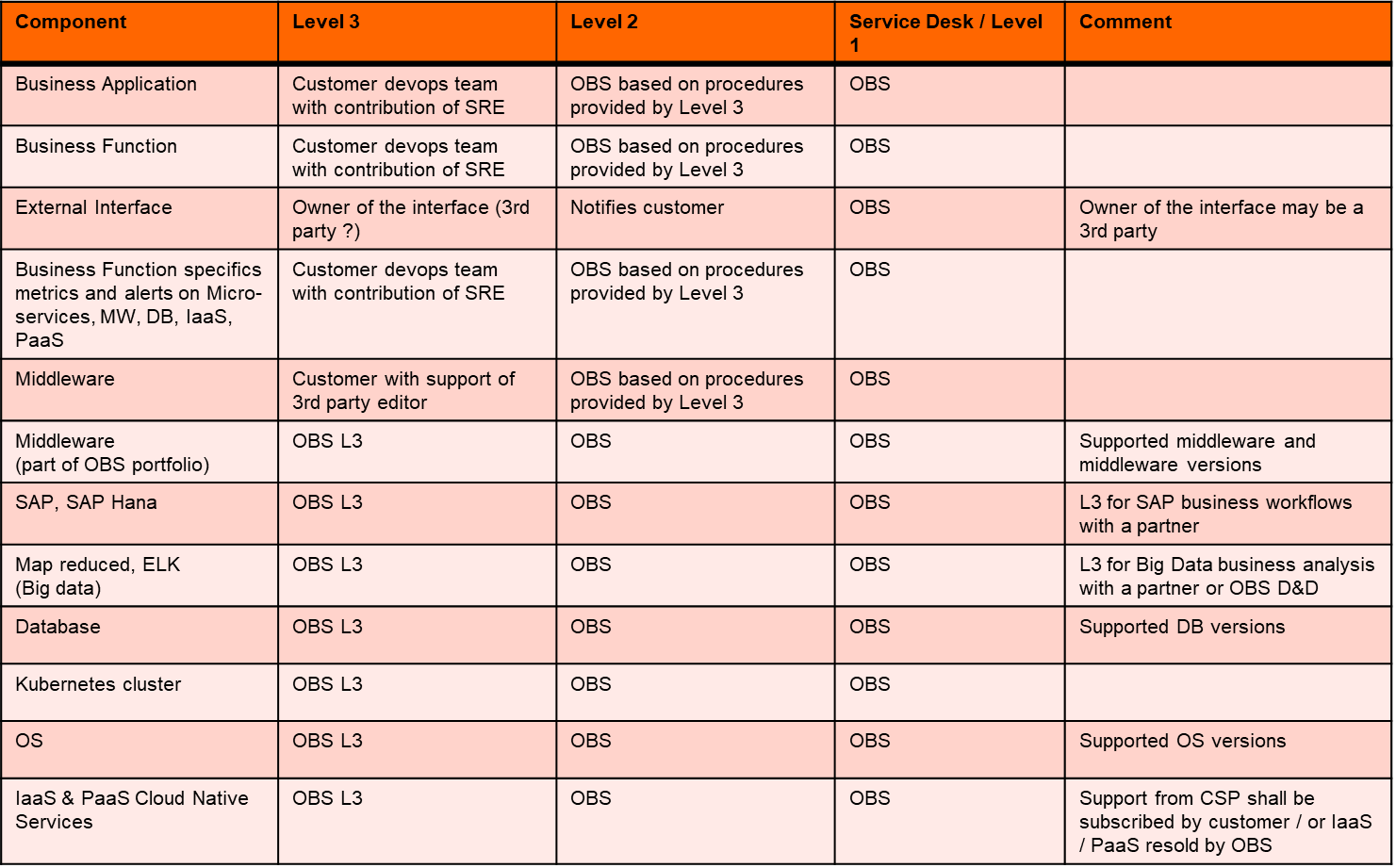
For a Business Functions of the Application, the Level 3 is typically customer’s responsibility: Customer’s development and test teams - or 3rd party teams subcontracted by the customer - who have coded, deployed and tested the Business Application and the Business Function. Customer may contract support to the software editor of the Business Application and Business Function.

Customer can subscribe SRE and Cloud Expert Services from OBS to strengthen the team.

Would the customer request OBS to take responsibility for the Level 3 of a 3rd party Business Application (with a potential partner for 3rd party application maintenance), a specific scope of work and agreement shall be established.

For the IaaS, PaaS, OS, Database, it is typical that the customer may rely on OBS to take care of the Level 3. However, other patterns are possible and can be discussed and established during the presale phase.

Here is an example of responsibility pattern for a project



**Example of responsibility matrix**

Note: for some solutions in OBS portfolio, typically Managed SAP, Managed Computer Vision, Flexible Web Platform, Hub EDI, Corporate e-Invoicing, LogaaS, OBS takes the responsibility for the Level 3 and has established partnership and support agreement with the involved 3rd party applications maintenance or software suppliers.

# Managed Cloud Native Services on Azure

Customer’s business application deployed on Azure are dependent on Azure Cloud Native Services (IaaS, PaaS). Orange Business Services provides the managed services necessary to ensure service assurance and change management for those dependences, as well as the configuration and deployment for building and recovering them.

### The cloud native services

One can typically distinguish 3 categories of services:

* The user plane services: if a business application depends on it, the business application is likely to be affected by a defect of it. The service does not have persistent data, therefore the recovery does not necessitate data restore.
* The data services: if a business application depends on a data service, the business application is likely to be affected by a defect of it. The service has persistent data, therefore a recovery may necessitate data restore. Data loss, data corruption may affect the business application as well.
* The other services: the business application does not depend on them. Most of those services are used for automation, observation, migration. The loss of the service is not likely to affect the business application. Some of the services are used for managing the user plane and data plane services of the business application, some others have specific usage for which a scope of work shall be established would the customer requires OBS to leverage them as part of the managed service provided.

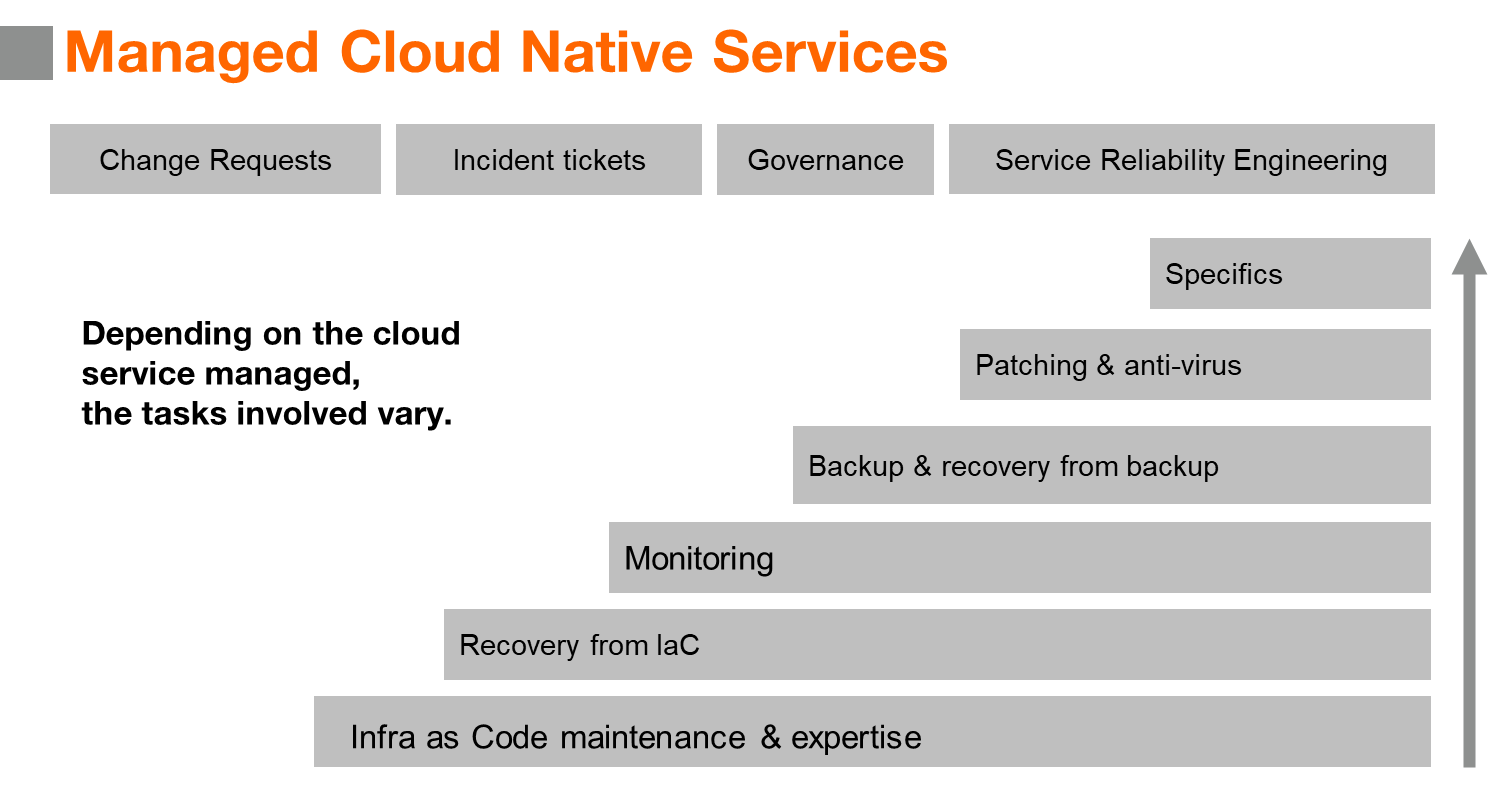
|  |  |  |  |
| --- | --- | --- | --- |
| User plane services | Data services | Other services | |
| **Compute**   App Service   App Service (Linux)   Azure Functions   Container Instances   Dedicated Host   Kubernetes Service   Service Fabric   Virtual Machines   VM Scale Sets  **Networking**   Application Gateway   Azure Bastion   Azure DNS   Azure Firewall   Azure Front Door   Express Route   Load Balancer   Network Watcher   Private Link   Traffic Manager   Virtual Network   VPN Gateway  **Integration**   API Management   Logic Apps   Notification Hubs   Service Bus  **Media**   Azure CDN   Media Services  **Automation**   Automation   Site Recovery | **Storage**   Azure Storage   Managed Disks   StorSimple  **Databases**   Cosmos DB   Database for MariaDB   Database for MySQL   Database for PostgreSQL   Redis Cache   SQL Database   SQL Server Stretch  **Identity & Security**   Azure Active Directory   Azure AD B2C   Azure AD DS   Azure Key Vault   Azure Lighthouse | **Management &**  **Governance**   Azure Advisor   Azure Arc   Azure Backup   Azure Batch   Azure Blueprints   Azure Monitor   Azure Policy   Azure Portal   Cloud Shell   Container Registry   Cost Management   Scheduler  **Security management**   Azure Sentinel   Security Center  **Integration**   Event Grid | **Development**   App configuration   Azure DevOps   DevTest Labs   Lab Services   Visual Studio App Center  **Migration**   Azure Migrate   Data box   DB Migration Service |

**Azure Cloud Native services by category**

### Tasks involved Cloud Native service management

The tasks involved for the management of a cloud native service depends on the service. They consist in:

* Configuring and deploying the service: Infrastructure as Code is leveraged in order to configure the service, the observability, the backup. Level 3 expertise on the service is leveraged for proper implementation thanks to the scope of work (refer to detailed description of build and SRE services)
* Applying the security group and access control policy defined by the customer.
* Service recovery thanks to Infrastructure as Code: in case of failure, most of the services requires to be recovered thanks to a redeployment. Re-configuring the service manually from scratch is not an efficient option: it takes time and is error prone. This is why recovery / redeployment from Infrastructure as Code is preferred.
* Supervision and remedial consists in watching for alarms raised on the service during the monitoring range (typically: 8x5 or 24x7). When an alarm occurs, an incident ticket is raised, a priority is assigned, the customer is notified. Then remedial action is taken thanks to the procedures made available to Level 2 / 1 by the Level 3. The remedial on a cloud native service may be necessary to restore the service of the business application. Would the procedure not remedy to the incident, then the incident is escaladed to the Level 3. Would the root cause be the CSP itself, then the incident is raised to the CSP by the Level 3.
* Backup and restore: depending on the service (if the service has persistence), it is necessary to backup the service data. The management service consists in configuring the backup solution and monitoring the proper run of it. Note: the backup solution has to be subscribed separately e.g Azure backup. Restoring the service on incident may involve restoring the data from a backup.
* OS patching and anti-virus: keeping OS up to date and virus free is a managed service for Managed Virtual Machine / Managed OS. Please refer to the detailed description.
* Specifics: some cloud native services may have specific configuration or management tasks.
* Business application specifics: by default, standard alerts are watched. The configuration of alerts, logs on a cloud native service which are specific to a business application is subject to a specific scope of work.



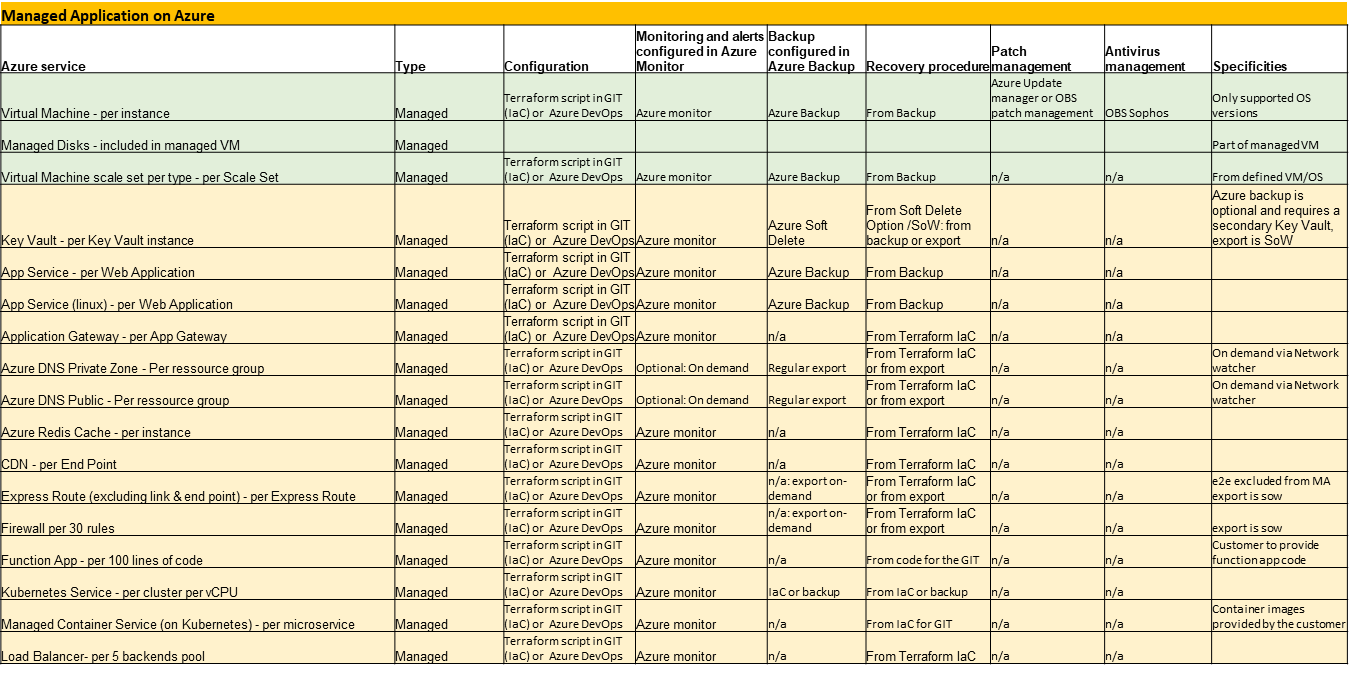
**Tasks involved in managed services for cloud native service**

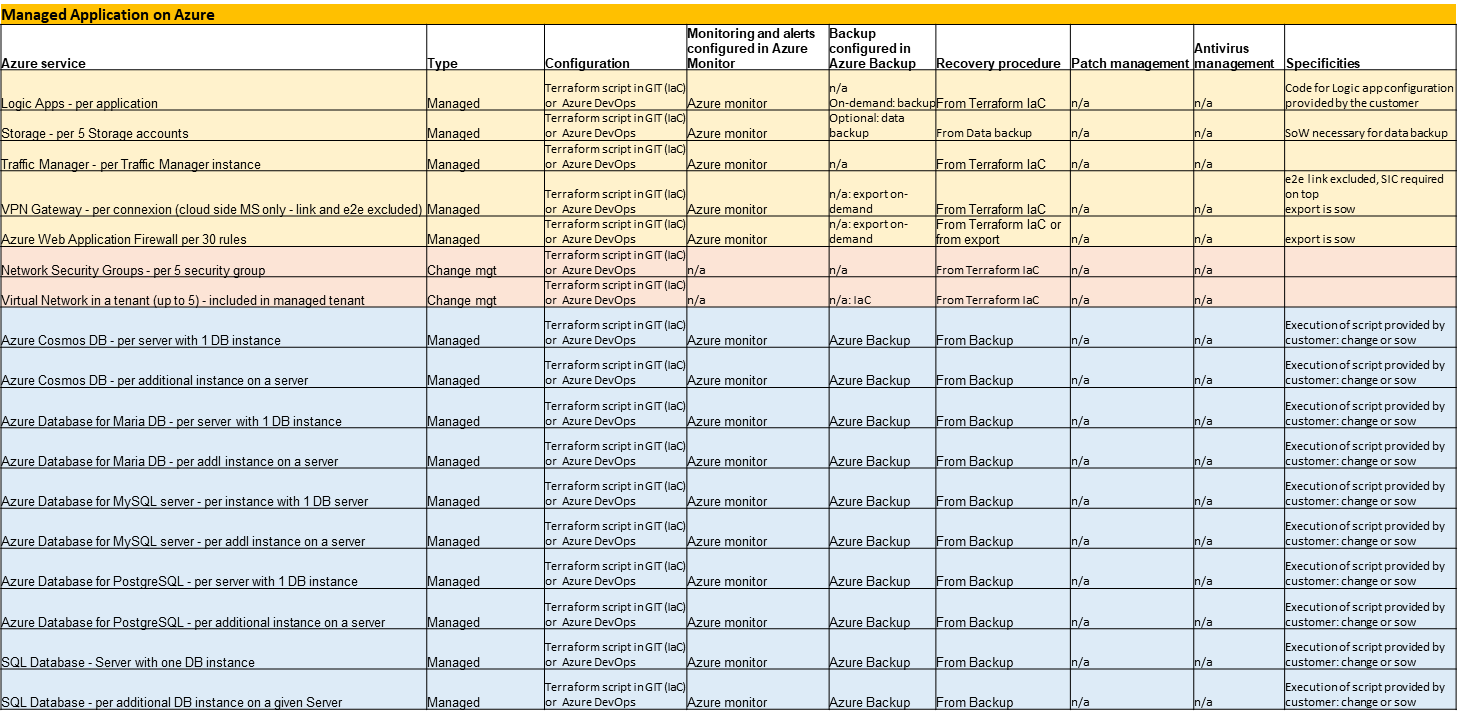
Depending on the cloud native service managed, more or less management tasks are necessary and included in the managed service. This drives the complexity of the managed service.

The tasks involved typically depends on the category of the cloud native service, whether user plane, data plane on which the business application depends, or other services upon which the business application does not depend.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Charging model | User plane services | Data plane services | Other services |
| Purpose |  | Used to support customer application | Used to support customer application | Used to operate user plane or data plane |
| Build | One-time charge based on SoW | IaC in Git, pushed via CI / CD | IaC in Git, pushed via CI / CD | IaC in Git, pushed via CI / CD |
| Maintaining IaC without changes | Monthly recurring charge | Yes | Yes | Yes |
| Monitoring & alerts | Monthly recurring charge | Yes | Yes |  |
| Configuration restore on incident | Included in MRC | Yes, from IaC or export | Yes, from IaC  or backup | Yes, from IaC when applicable |
| Data backup and restore on incident | Included in MRC |  | Yes |  |
| Network and Security Management | Based on SoW | Optional: Based on SoW | Optional: Based on SoW |  |
| Service Desk | Per incident ticket or percentage | Yes | Yes | Yes |
| Change Management | Per change, in Tokens vs complexity | Via IaC in Git, pushed via CI / CD. | Via IaC in Git, pushed via CI / CD. | Via IaC in Git, pushed via CI / CD |
| Disaster recovery | Specific design and quote | Optional: Based on SoW | Optional: Based on SoW |  |

### Table of tasks involved in the management a Cloud Native service





**Table of tasks involved in the management of cloud services (extract of services)**

### Tooling used for cloud native managed services

Azure tooling and OBS backend operations tooling are leveraged to deliver the managed services. Would the customer require the use of a different tooling, the feasibility shall be confirmed with OBS and the RACI and work-units may be revised.

|  |  |
| --- | --- |
| **Process** | **Tool used by OBS MA delivery** |
|  | |
| Configuration of the infrastructure | Terraform script  Azure Devops  GIT referential  CI / CD |
| Supervision solution | Azure Monitor with connector to OBS supervision |
| Backup | Azure Backup (incl snapshots) |
| OS patching solution | Azure Update Manager  OBS MA patching tool (BRAC)  OBS OS factory |
| Antivirus solution | OBS MA Sophos tool |
| Logging solution | Azure Insight (on demand based on Scope of Work)  Azure Log Analytics (on demand based on Scope of Work) |
| Recovery | From backup when it exists  From Terraform script in GIT when it exists  Ideally from up-to-date Infra as code with CI/CD |
| Admin connectivity | VPN to OBS CASA Zone |
| Portal for access to MA contract, incident & change ITSM | OBS Cloudstore |

### General pre-requisites to the run of managed services

The following pre-requisites are necessary to all managed services:

* The Customer shall have defined a valid architecture. (OBS can optionally provide Professional Services for architecture definition).
* The Customer shall have **a valid subscription to Azure including subscription to Azure Support plan and procure the Azure resources and Azure support plan. OBS can optionally supply this subscription inclusive of Azure support (ref to Multi-Cloud Ready offer for Azure), however, the subscription, the IaaS resources, the Azure support are not part of the Managed Services.** The Managed Services will leverage this support contract to escalades incident to Azure CSP.
* Azure platform for the Customer shall be urbanized alongside best practices of Azure’s landing zone or shall offer comparable services.
* OBS proposes a default RACI depending on the class of transition and the resource managed. As a pre-requisite to the project, OBS and the Customer shall agree on the RACI.
* Agreement on the tooling used for GIT, CI / CD chain, Monitoring, Logging and Alerting solution.
* Additional pre-requisites are required when transition is not the entire responsibility of OBS (e.g required for partial build like “Operations Build” or “Backend Build” models, refer to chapter 8 of the document: Build Scope of Work)

In the case of Fully Managed service, OBS is using its own Git, CI / CD chain, Monitoring, Logging and Alerting solution.

In the case of a Co-managed service, OBS and the Customer agree on the Git, CI / CD chain, Monitoring, Logging and Alerting solution to be used. By default, the tooling is

* Either based on Azure tools i.e Azure DevOps, Azure Monitoring
* Or based on generic multi-cloud tooling proposed by OBS e.g CaasCad (Prometheus, Grafana,…)

This tooling not included in the Managed Applications work units and can be purchased separately as part of Azure Subscription or as a multi-cloud tooling proposal made by OBS.

### Criteria for the run of a managed cloud native service component

Criteria shall be met with an approval by Level 2 before turning a cloud native component to an active manage service (i.e Run) by the Level 2 / Level 1 operations. The owner of the Build and of the Level 3 support owns the responsibility of making sure that the criteria are met:

* The architecture and deployment of the service shall be defined.
* The service shall be deployed thanks to Infrastructure-as-Code and tested prior to transitioning to the run team. Typically, successful testing in pre-production, with a pre-production environment iso-production. Note: IaC is necessary to recover the services in case of major failure.
* The use of the service shall be explained to the operation team
* The security policies and access control shall have been configured.
* The access shall have been configured allowing OBS Level 2 teams access.
* The service shall export the necessary metrics towards Azure Monitor.
* The data backup shall be configured in Azure Backup when backup is applicable.
* The disaster recovery shall be configured when applicable.
* The troubleshooting and service restoration procedures shall be provided to Level 2.
* Whereas a procedure requires logs or dashboard those shall have been developed and deployed prior to transferring to run phase.
* A remedial procedure on incident shall not last more than 15 minutes. Beyond, that time amount, the effort would be charged on time base.

# The build of services & managed services on Azure

## The build and deployment scope of work

The **Full Build** and deployment of a business applications, services and managed services on Azure involve:

* **The build of the cloud infrastructure including:**
  + The Landing Zone transversal for multiple applications
  + The cloud infrastructure specific to each application
* **The deployment of the business application software on the infrastructure**
* **The build of the operations layer including:**
  + The selection and configuration of tooling used for operations, including Azure tooling services
  + The configuration and deployment of exporters of observability metrics, logs, agents, backup and other operational parameters necessary to the operations
* **The integration into OBS administration backend including:**
  + Provision and connectivity for administrative access to the cloud platform
  + Integration of Azure monitoring alerting towards OBS Level 1 / 2 & 3 supervision tooling
  + Integration of troubleshooting procedures into operations Knowledge Database
  + Configuration and provision of OBS ITSM tooling for Incident Tickets and Change Requests handling.
  + Configuration and provision of OBS portal for access to contractual documents and billing.

**At the end of the build, all pre-requisites shall be available and ready to meet the criteria for the run (ref chapter 7.1.3 Criteria for the run of a managed cloud native service component).**

Depending on the project status and on customer’s request, **the remaining Scope of Work for the build** **to meet the criteria for the run may be the full build or a partial build.** The scope of work of the build is discussed during the pre-sale phase.

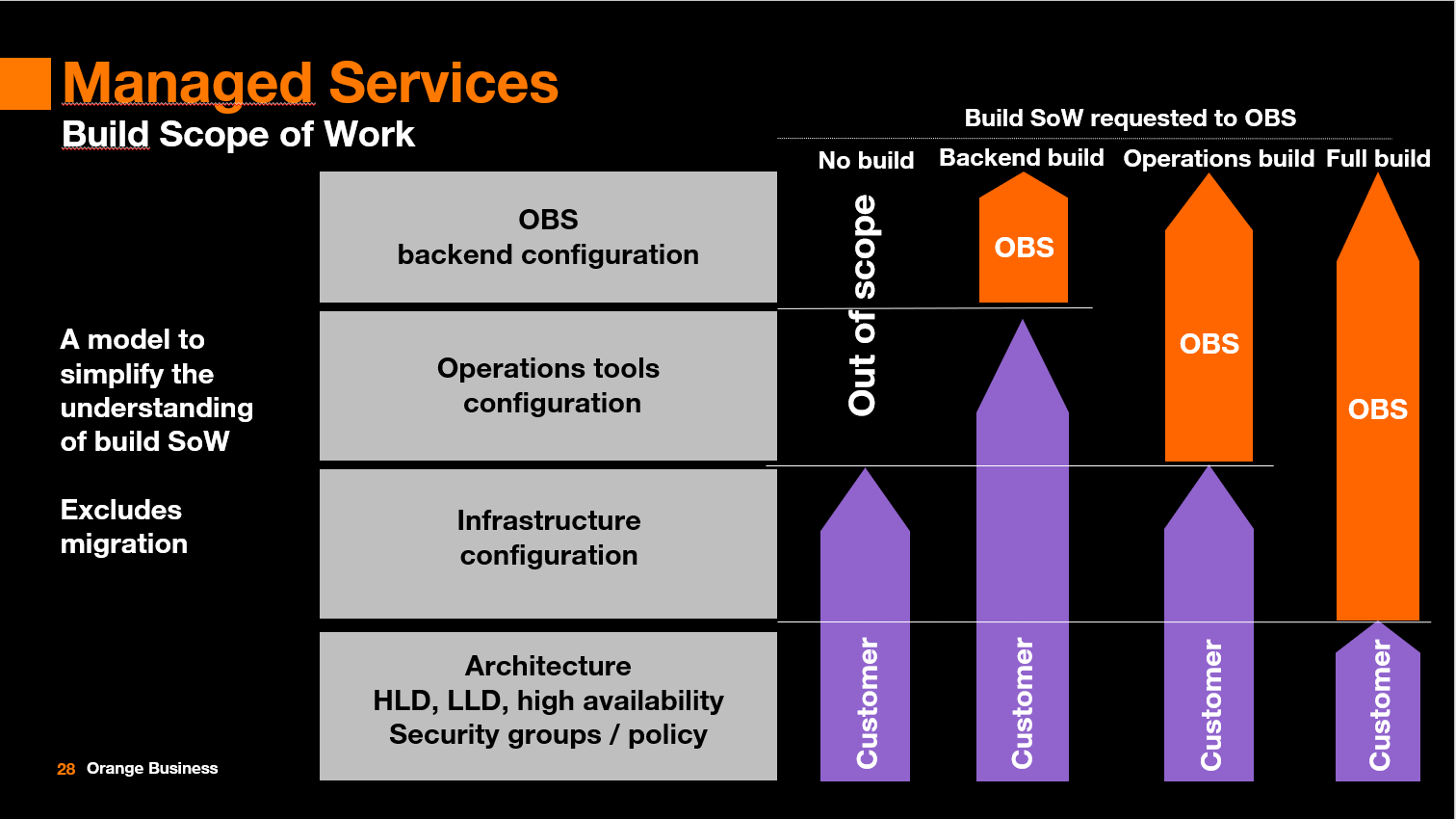
The customer **may retain or delegate part of the Build responsibilities to OBS:**

* The Build of the Cloud Infrastructure layer (IaC)
* The Build of the Operations layer
* The integration into OBS administration backend tooling

**The Build effort estimation is custom and depends on the Scope of Work of the project.** Upon customer’s request, OBS can provide **Cloud Expert resources** and **Service Reliability Engineer** prestation to join and strengthen customer’s development team.

While the build effort and its estimate can be custom, for sake of simplification and budgetary anticipation by the customer, **OBS has pre-defined 4 models of build** for a managed cloud native resource:

* **Model “No build”:** no build requested from OBS – resource is not managed.
* **Model “Backend build” a.k.a Class 2 build:** OBS is only requested to integrate in its administration backend. The customer takes care of all other aspects of the build
* **Model “Operations build” a.k.a Class 4 build:** OBS is requested the build of the operations layer and the integration in its administration backend. The customer takes care of the build of the infrastructure and software deployment
* **Model “Full build” a.k.a Class 5 build:** OBS is requested to build the infrastructure with IaC, to build the operations layer and to integrate in its administration backend. The customer takes care of the Software.



**Main models of build Scope of Work requested to OBS**

As far as the Scope of Work is concerned, building the first cloud native resource of a given type usually involves a larger effort than building a subsequent resource of same type, as the Infrastructure as Code might be mostly reused. This possibility depends on the cloud native resource considered.

As result, **each build model distinguishes:**

* **The effort and price of build for the first resource of a given type**
* **The effort and price of build for the subsequent resource of the same type**

For the Scope of Work differ from the models, a custom scope of work shall be estimated.

## Project management and coordination during the build phase

During the build phase, a Scope of Work is defined for Project Management and Service Implementation Coordination to ensure efficient execution of the project between the customer and OBS.

Optionally, the project may require Cloud Expert Services and Service Reliability Engineering.

Those services are charged based on time and material.

## Cloud Expert Services for migration

The project may require migration of the workloads to Azure. OBS will propose Cloud Expert Services for migration to Azure based on the specific migration Scope of Work in addition to the Managed Services.

## Criteria for assessing the model of build for a resource

When the build effort is uncertain from pre-sales documentation, an assessment is proposed at the beginning of the build project by OBS Cloud Expert Services. During this assessment, the following tasks are performed:

* Collection of the architecture diagrams with dependences, HLD, LLD of applications, and infrastructure to be managed and any other useful information.
* Check of the inventory of resources to be deployed and managed.
* Review for each of the dependence the remaining work requested to OBS for completing the build to reach readiness for the run. Review the criteria for a resource build to qualify to a given model of build. Hence determining for each resource which build model applies: No build, Backend build, Operations Build or Full Build.
* Confirmation that the pre-required tools for operations are in place (or alternatively agreeing on a specific scope of work for different tooling if agreeable).
* Establishing requested responsibilities defined between the customer and OBS (RACI) for build and for the run.
* Identifying potential limitations on the managed application service if criteria are not met.

### Criteria for qualifying as “backend build” model a.k.a class 2 SoW for a resource:

The “backend build” scope of work model for a resource is used for:

* a resource/service in scope for managed service for which the infrastructure is already built and deployed by the customer leveraging Infrastructure-as-Code.
* And, for which Azure tooling is fully configured and operational prior to transition under customer’s responsibility. The tooling used shall be:
  + Azure Monitor for supervision with proper alerts defined
  + Azure Backup properly configured and functional
  + Update Manager configured for VM patching
  + Remedial and troubleshooting procedures on known incident are defined and provided
  + Recovery procedures to be used are defined and provided by the customer
* And, customer provides documentation i.e schema, HLD and DAT/LLD, architecture explaining how availability & HA, monitoring, security policies and access control, backup, disaster recovery, baseline security, SLA are achieved.

The build effort provided by OBS in the “backend build” includes integrating the alarms from Azure Monitoring to the OBS backend systems, capturing the procedural guides provided by the customer into the OBS knowledge repository of operations, and operations readiness. It includes as well getting the administrative backend, the OBS ITSM, the portal and billing readiness for operations.

### Criteria for qualifying as “operations build” model a.k.a class 4 SoW for a resource:

The “operations build” scope of work model for a resource is used for:

* a resource/service in scope for managed service for which the infrastructure is already built and deployed by the customer leveraging Infrastructure-as-Code.
* And, customer provides documentation i.e schema, HLD and DAT/LLD, architecture explaining how availability & HA, monitoring, backup, disaster recovery, baseline security, SLA are achieved.
* And, agreement reached between the customer and OBS to use the Azure and OBS backend tooling.

The build effort provided by OBS in the “operations build” includes that of the “backend build” plus the configuration and deployment of Azure tooling thanks to Infrastructure as Code and of OBS backend i.e:

* + Azure Monitor for supervision with alerts
  + Azure Backup configuration and deployment
  + Update Manager configuration for VM patching
  + Anti-virus configuration for VM
  + Use of standard remedial and troubleshooting procedures on known incident for the cloud native service.
  + Use of standard recovery procedures for the cloud native service.

For further details on the operations per service, please refer to **Chapter 9: detailed description per cloud service.**

### Criteria for qualifying as “full build” model a.k.a class 5 SoW for a resource:

The “full build” scope of work model for a resource is used for:

* a resource/service in scope for managed service not yet built and deployed.
* And, customer provides documentation i.e schema, HLD and DAT/LLD, architecture explaining how availability & HA, monitoring, backup, disaster recovery, baseline security, SLA are achieved.
* And, agreement reached between the customer and OBS to use the Azure and OBS backend tooling.

The build effort provided by OBS in the “full build” includes that of the “backend build” plus that of the “operational build” plus

* + The configuration of the Landing Zone and the infrastructure of the resource leveraging Infrastructure as Code.

For further details on the operations per service, please refer to **Chapter 9: detailed description per cloud service.**

For further details of Infrastructure as Code for full build model, **please refer to chapter Infrastructure as code methodology.**

### Mitigation in case of pre-requisites or criteria not met:

The assessment may reveal that criteria are not met for qualifying to a given build model. Then 3 options are possible:

* the scope of work shall be revisited with a more appropriate build model. This may affect the duration of the project, efforts, quote and price.
* the customer may remedy to the missing criteria. This may affect the duration of the project and project management and coordination efforts.
* the customer and OBS may agree to live with some limitations in the management capabilities and responsibilities due to the missing criteria.

Would the project be delayed and would resources effort be overspent by OBS as result of pre-requisites and criteria under customer’s responsibility not being met, then OBS would be entitled to charge the overspent effort based on time and material.

### Charging model for build

|  |  |
| --- | --- |
| Service | Work Unit |
| Project management | Time and material |
| Service Implementation Coordination | Time and material |
| Service Reliability Engineer | Time and material |
| Technical Architect | Time and material (when necessary for documentation) |
| Full build model - 1st Resource Unit\* | One Time Charge per resource |
| Full build model - subsequent Resource Unit of same type\* | OTC per resource |
| Operations build model - 1st Resource Unit\* | OTC per resource |
| Operations build model - subsequent Resource Unit same type\* | OTC per resource |
| Backend build model - 1st Resource Unit \* | OTC per resource |
| Backend build model - subsequent Resource Unit same type\* | OTC per resource |

**Resource unit\*: please refer to Chapter 9: detailed description per cloud service for the definition of the Resource Unit per cloud native service.**

# Detailed responsibilities and accountabilities

The following tables describe the standard default responsibilities between OBS and the customer depending **on the build model**.

The following tables describe the standard default responsibilities between OBS and the customer depending on classes of service. Those may be amended with mutuel consent depending on project.

* R stands for responsible
* A stands for Accountable
* C stands for Contributor
* I stands for Informed

##### RACI for Managed OS

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Service Implementation** | **OBS** | **Customer** | **OBS** | **Customer** | **OBS** | **Customer** |
| OS Server infrastructure implementation Full build | | | Operations build | | Backend build | |
| Deployment of the infrastructure | R, A | I | I | R, A | I | R, A |
| Deployment of LAN components | R, A | I | I | R, A | I | R, A |
| Deployment of DNS and NTP services | R, A | I | R, A | I | I | R, A |
| Backup tools for operations (Azure backup & Azure Snapshots) | R, A | I | R, A | I | I | R, A |
| Deployment of the OS patching solution (Azure Update Mgt) | R, A | I | R, A | I | I | R, A |
| Deployment of the Antivirus solution | R, A | I | R, A | I | SoW | SoW |
| Deployment of the supervision solution (Azure Monitor) | R, A | I | R, A | I | I | R, A |
| Deployment of the logging solution (Azure Insight) | R, A | I | R, A | I | I | R, A |
| Deployment of security groups and firewall rules | R, A | I | SoW | SoW | I | R, A |
| Recovery procedure (Infra as Code, restore, other…) | R, A | I | I | R, A | I | R, A |
| Testing and validation of infrastructure implementation | R | A | I | R, A | I | R, A |
| Testing and validation of Azure tooling implementation and lifecycle management | R | A | R | A | I | R, A |
| OS Server Implementation | | |  |  |  |  |
| Evaluation or deployment of the operating system | R, A | I | R, A | I | I | R, A |
| Deployment of new packages | R, A | I | R, A | I | R, A | I |
| Test and validation of operating system implementation for new packages | R, A | I | R, A | I | R, A | I |
| Service implementation documentation | | |  |  |  |  |
| Conception, architecture and low-level design for infrastructure | I | R, A | I | R, A | I | R, A |
| Implementation and operation documentation for infrastructure | R, A | I | I | R, A | I | R, A |
| Conception and low-level design for tooling (Azure) | R, A | I | R, A | I | I | R, A |
| Implementation & operation documentation for tooling (Azure) | R, A | I | R, A | I | I | R, A |

##### RACI for Database as a Service

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Service Implementation** | **OBS** | **Customer** | **OBS** | **Customer** | **OBS** | **Customer** |
| Database aaS services conception and implementation Full build | | | Operations build | | Backend build | |
| Maintenance of Infrastructure architecture referential | R, A | I | I | R,A | I | R,A |
| Maintenance of tooling configuration referential | R, A | I | R, A | I | I | R,A |
| Deployment of the infrastructure | R, A | I | I | R,A | I | R,A |
| Deployment of the supervision solution (Azure Monitor) | R, A | I | R, A | I | I | R,A |
| Deployment of the logging solution (Azure Insight) (optional) | R, A | I | R, A | I | I | R,A |
| Deployment of the backup solution (Azure Backup, Snapshot) | R, A | C, I | R, A | C, I | I | R,A |
| Recovery procedure for infrastructure from referential (Infra as code, restore from backup, other…) | R, A | C, I | I | R, A | I | R,A |
| Recovery procedure for tooling from referential (Infra as code, restore, other…) | R, A | C, I | R, A | C, I | I | R,A |
| Testing and validation of infrastructure implementation | R, A | I | I | R,A | I | R,A |
| Testing and validation of tooling implementation and lifecycle management | R, A | I | R, A | C, I | I | R,A |
| Customer provided script execution on DB instance | R | A, I | R | A, I | R | A, I |
| OBS script execution on DB instance | R, A | C, I | R, A | C, I | R, A | C, I |
| Service implementation documentation | | |  |  |  |  |
| Conception, architecture and low-level design for infrastructure | C, I | R, A | I | R,A | I | R,A |
| Implementation and operation documentation for infra | R, A | C, I | I | R,A | I | R,A |
| Conception and low-level design for tooling (Azure) | R, A | C, I | R, A | C, I | I | R,A |
| Implementation & operation documentation for tooling (Azure) | R, A | C, I | R, A | C, I | I | R,A |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Service Operation** | **OBS** | **Customer** | **OBS** | **Customer** | **OBS** | **Customer** |
| Database aaS services operations Full build | | | Operations build | | Backend build | |
| Monitoring through Azure Monitor | R | I | R | I | R\* | I |
| Investigation through Azure Insights | R, A | C,I | R, A | C,I | R\* | A |
| Restore from Infra as Code and backup | R, A | C,I | R, A | C,I | R\* | A |
| Changing capacity of database instance | R, A | C,I | C, I | R, A | C, I | R, A |
| ITSM operations | | |  |  |  |  |
| Change Management | R | A | R | A | R | A |
| Incident Management | R, A | R\*\*,I | R, A | R\*\*,I | R, A | R\*\*,I |
| Event management | R, A | I | R, A | I | R, A | I |
| Baseline security management | R | A | SoW | SoW | SoW | SoW |
| Configuration management | R, A | C, I | R | A | R | A |
| Report management via SDM service | R, A | C, I | R, A | C, I | R, A | C, I |
| Invoicing management | R, A | I | R, A | I | R, A | I |

R\*: within the limitations of tooling provided by the Customer

R\*\*: in co-management model, customer may have joint responsibilities related to the activity & incident

##### RACI for other Native Services managed

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Service Implementation** | **OBS** | **Customer** | **OBS** | **Customer** | **OBS** | **Customer** |
| Native service infrastructure implementation Full build | | | Operations build | | Backend build | |
| Deployment of the infrastructure | R, A | I | I | R, A | I | R, A |
| Backup tools for operations (Azure backup )(1) | R, A | I | R, A | I | I | R, A |
| Deployment of the supervision solution (Azure Monitor)(1) | R, A | I | R, A | I | I | R, A |
| Deployment of the logging solution (Azure Insight) optional (1) | R, A | I | R, A | I | I | R, A |
| Deployment of security groups and firewall rules | R, A | I | SoW | SoW | I | R, A |
| Recovery procedure (Infra as Code, restore, other…) | R, A | I | I | R, A | I | R, A |
| Testing and validation of infrastructure implementation | R | A | I | R, A | I | R, A |
| Testing and validation of Azure tooling implementation | R | A | R | A | I | R, A |
| Packages | | |  |  |  |  |
| Deployment of new packages (1) | I | R, A | I | R, A | I | R, A |
| Service implementation documentation | | |  |  |  |  |
| Conception, architecture and low-level design for infrastructure | C, I | R, A | I | R,A | I | R,A |
| Implementation and operation documentation for infrastructure | R,A | I | I | R,A | I | R,A |
| Conception and low-level design for tooling (Azure) | R,A | I | R,A | I | I | R,A |
| Implementation & operation documentation for tooling (Azure) | R,A | I | R,A | I | I | R,A |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Service Operation** | **OBS** | **Customer** | **OBS** | **Customer** | **OBS** | **Customer** |
| Native service operations Full Build | | | Operations build | | Backend build | |
| Monitoring (1) | R, A | I | R, A | I | R\* | A |
| Backup (1) | R | A | R | A | R\* | A |
| Restore from Infra as Code and backup (1) | R, A | C,I | R, A | C,I | R\* | A |
| Security groups, Firewall rules setting | R | A | SoW | SoW | I | R,A |
| ITSM operations | | |  |  |  |  |
| Change Management | R | A | R | A | R\* | A |
| Incident Management | R, A | R\*\*,I | R, A | R\*\*,I | R\*, A | R\*\*,I |
| Event management | R, A | I | R, A | I | R\* | A |
| Baseline security management | R | A | SoW | SoW | SoW | SoW |
| Report management via SDM service | R, A | I | R, A | I | R, A | I |
| Invoicing management | R, A | I | R, A | I | R, A | I |

R\*: within the limitations of tooling provided by the Customer

R\*\*: in co-management model, customer may have joint responsibilities related to the activity & incident

(1) When applicable as per detailed description per service

# Detailed build methodology

The following chapter describes the build methodology followed by OBS for the Full Build model. It is also the method recommend to the customer when building his cloud infrastructure and operations layer.

The quality of the build will determine the resilience, the maintainability, the recovery of the workloads and the efficiency of the run operations.

The methodology varies from one project to another and as such the quote is specific and depends on the scope of work.

## Inputs to the build

For proper accurate quote of the build and kick-off of a build task the following specifications are pre-requisites:

* Architecture diagram of the application and its layout of deployment on Azure services
* Description of the environments required (Dev, Pre-Prod, Prod)
* Security policies and access control
* Environments topology
* Type and inventory of applications, middleware, IaaS/PaaS services used
* Scope and RACI

Should the Customer provide the information during the pre-sales, OBS would quote the build accurately.

Alternatively, should the Customer not have such information during the pre-sales phase, then generic hypothesis would be taken for the build estimation. The build specification and quote would be updated during the initial phase of the project after an audit or after the information is provided.

Nevertheless, please find here-below OBS default reference approach for building Managed Services on Azure. The scope of work will vary depending on customers’ projects.

## Initialization of the project: building the landing zone, IaC and pipelines on Azure

At the initiation of the project, we build a so-called **Landing Zone**.

The deployment of the infrastructure is modeled as **Infrastructure as Code** (IaC) for quality, replicability, disaster recovery. The Azure services are deployed using this IaC as opposed to the use of Human Machine Interface on Azure. As opposed to the use of Human Machine Interface:

* The deployment code can be tested and validated on non-production environment (e.g. dev, integration, staging, preprod) before going to production.
* Can be replicated and evolved
* Can be used to rebuild the service from a disaster
* Can be versioned
* Can associate deployment of resources as well as operations configuration (monitoring, logging, backup…)
* Optional: when available in a multi-cloud tooling e.g Terraform, CaasCad, the Iac could be leveraged for other clouds with minimal changes.

By default, OBS uses **Terraform and ARM** (Azure Resource Manager) as scripting language for the IaC

* Terraform is more generic and usable across multiple clouds with limited adaptations
* ARM is specific to Azure.

The code base IaC is managed with a tool chain:

* By default, the preference would be to use **Azure Devops**. In such case, usage is part of Azure subscription as pre-requisite to Managed Application service.
* Optionally, multi-cloud platforms like **CaasCad** can be proposed based on cloud agnostic **OpenSource** like Concourse, Gitea / Gitlab, Terraform and Rancher, Quay for containers. In such case, the tooling is a specific quote separate from Azure subscription.
* Use of customer’s specific software factory needs feasibility assessment and scoping as it impacts the build and run processes.

3 main parts of **Azure Devops** are leveraged

* The **Repository** based on GIT where the IaC code base is stored and versioned
* The **Pipelines for Build**
* The **Pipelines for Releases**
* Note that the Project Management tools from Azure Devops are not used by default. Those includes tasks, Projects, backlogs (Agile type of).

Based on the specifications, OBS IaC developer assigned to the build, will code the IaC and prepare the **pipelines for build** based on Terraform plans.

Example the code can be structured in a variety of build pipelines and Terraform plans for:

- subscription,

- management

- identity

- VM

- Database

- etc…

The IaC developer will test the quality of the code

Example for a build

* + He makes a pull request on the master branch
  + He runs automatically a Terraform Format, a Terraform validate to validate the syntax
  + Launches a deployment on Azure to validate the proper deployment.

OBS IaC libraries help gain time in IaC development, nevertheless, projects are often specific and need specific adaptations and developments.

Then OBS developer create **the release pipelines**

* The release pipeline is the way one deploys the IaC on each Azure environment
* The release pipeline chains deployment to dev platform, then integration/staging, then pre-prod, then production as example would the Customer have such environments.
* The release pipeline is a code base which is moving through environment
* For each environment, OBS IaC developers sets **environment variables** to adapt consumptions according to the platform: eg tiny VM on dev, large on prod.

**Custom development** might be necessary depending on Customer’s SoW based on specific quote. E.g implementation of a DNS forwarder for the platform connectivity.

OBS developer enriches the IaC build pipelines with the **tooling for operations i.e** monitoring, logging, soft delete, backups. By default, Azure Monitoring, Azure Backup will be used. Then connection to OBS Managed Application central supervision system is set-up to alert Level 1 on incident.

Would the customer subscribe to Managed Services for **Application Layer,** this one would be added to the pipelines leveraging Ansible or Jenkins. The pipe can be decoupled or combined with the infrastructure layer.

The application layer pipelines can deploy the applications on a variety of services whether IaaS, Kubernetes, DBaaS or PaaS leveraging less or more decoupling with the underlying infrastructure and less or more agility and segregation of duty between the application management and the infrastructure management. This architecture is a key factor in providing agility to the application developers thanks to PaaS automation of the underlying infrastructure layers. It also drives the RACI between OBS and the Customer’s developers.

Multiple RACI can be thought through between the Customer and OBS depending on level of delegation desired or depending on environment platforms.

Example: there can be a Software Factory for the application code under the responsibility of the Customer deploying on an infrastructure managed by OBS thanks to a separate Software Factory of the Infrastructure.

Example 2: developers can test the alarms by themselves in the development, integration and staging environments and then, thanks to SoW and procedures, such alarms can be used by OBS for the monitoring of the Production Environment.

## Change management methodology

The principle for changes is typically to edit the IaC in order to deploy additional resources or modification, and then leverage the release pipeline to test and deploy the change.

Large changes can involve coding evolution of the IaC or modifying the pipelines and therefore are quoted based on Scope of Work and Impact on the services.

Changes for redeployment consist in leveraging the release pipeline to redeploy.

**Direct changes through the Azure user interface are avoided when possible** as those changes could not be reproduced automatically in case of disaster or need.

Changes can consist in restoring a backup version of data plane service.

Depending on the organization and quality of code, changes maybe complex or less.

## Transition from build to run

OBS Managed Application promotes the principle by which the team in charge of the run develops the IaC. Nevertheless, the developer of the IaC needs to explain the use of it to the whole team involved in the run including how to edit and release.

## Security policies and access control

The customer shall define the Security Groups and Security Policies as well as Firewalling and Web Application Firewalling rules to be applied. Those policies will be implemented by OBS in the Infrastructure as Code and Firewall as a Service configurations.

The customer shall define the Virtual Private Networking topology and IP address schema that will have to be implemented by OBS.

The customer shall define the Access Control and Credentials that will have to be used by OBS administrators and implemented.

Security recommendations can be part of an optional security Scope of Work based on customer specific case and request. **By default, the MRC work units for the RUN do not cover advanced security recommendations.**

# Detailed description of the run tasks per cloud service (Extract)

## API Management

### Description

Azure API management allows the secured publication of APIs at scale to developers, partners and employees.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the API management.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Metrics supported for API Management service:

* BackendDuration
* Capacity
* ConnectionAttempts
* Duration
* EventHubDroppedEvents
* EventHubRejectedEvents
* EventHubSuccessfulEvents
* EventHubThrottledEvents
* EventHubTimedoutEvents
* EventHubTotalBytesSent
* EventHubTotalEvents
* EventHubTotalFailedEvents
* FailedRequests
* NetworkConnectivity
* OtherRequests
* Requests
* SuccessfulRequests
* TotalRequests
* UnauthorizedRequests
* WebSocketMessages

Alerts observed

* FailedRequests
* UnauthorizedRequests

##### Backup and restore for the Site Recovery configuration

Service restore: The Continuous Deployment chain is used to redeploy the same configuration of the Site Recovery from the reference Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

The service can be deployed in multi-region by design.

##### Limitations & pre-requisite

Whenever the API is customized, there should be procedures provided by the customer describing how to monitor and troubleshoot the API.

### Charging model

|  |
| --- |
| Work Unit |
| Per API |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Modify API behavior | On quote |
| Other changes | Estimation in tokens based on time spent |

## Application Gateway

### Description

Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications..

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the CDN.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Metrics supported by Application Gateway V2 SKU:

* Backend connect time
* Backend first byte response time
* Backend last byte response time
* Application gateway total time
* Client RTT
* Bytes received
* Bytes sent
* Client TLS protocol
* Current capacity units
* Current compute units
* Current connections
* Estimated Billed Capacity units
* Failed Requests
* Fixed Billable Capacity Units
* New connections per second
* Response Status
* Throughput
* Total Requests
* Backend response status
* Healthy host count
* Unhealthy host count
* Requests per minute per Healthy Host

Metrics supported by Application Gateway V1 SKU

* CPU Utilization
* Current connections
* Failed Requests
* Response Status
* Throughput
* Total Requests
* Healthy host count
* Unhealthy host count

Alerts observed

* Backend connect time (V2)
* Backend response status (V2)
* Application Gateway Total Time (V2)
* Throughput (V1, V2)
* Client RTT (V2)
* Failed Requests (V2)
* Custom: %age of failed request (Failed Requests / Total Requests) (V2)
* Unhealthy Host Count (V2)
* CPU Utilization (V1)
* Failed Requests (V1)
* Response Status (V1)

##### Backup and restore

Data backup and restore

Can be exported from CI/CD Pipeline.

Service restore

The Continuous Deployment chain is used to redeploy the Application Gateway from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

For Application Gateway V2, the service HA is managed by Microsoft.

The DR can be customized by design.

### Charging model

|  |
| --- |
| Work Unit |
| Per Instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Add/modify Backend | 1 Token |
| Certificate Installation | 1 Token |
| Other changes | Estimation in tokens based on time spent |

## Application Insights – basic monitoring with class 2 transition

### Description

Application Insights, a feature of [Azure Monitor](https://docs.microsoft.com/en-us/azure/azure-monitor/overview), is an extensible Application Performance Management (APM) service for developers and DevOps professionals. Use it to monitor your live applications. It will automatically detect performance anomalies, and includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability.

The basic monitoring excludes the middleware and application management as well as remedial actions.

### Build to run service included in the OTC

##### Build to run service pre-requisite

The pre-requisite to Application Insights basic monitoring with class 2 transition is that Application Insights has been configured by the Customer including

* Resources monitored
* SDK deployed on the resources when applicable
* Metrics and alerts forwarded to Azure Monitor
* Performance dashboards

##### Build to run service

For Application Insight basic monitoring with class 2 transition, the build to run service included in the OTC consists in integrating the alerts from Azure Monitor configured in Application Insights into OBS supervision backend.

### RUN services included in the MRC

##### Run service pre-requisite

* The resource monitored is in the inventory Scope of Work of managed service : infrastructure resource, middleware resource, application resource, database resource, Kubernetes cluster resource, microservice resource, etc…
* A referential file exists in the Git including the reference configuration of Application Insights.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

Alerts observed

* Alerts defined in Application Insights and forwarded to Azure Monitor for resources in the Scope of Work of managed services.

##### Monitoring service

As part of the Application Insights basic monitoring service, OBS operations will monitor the alerts, raise tickets and inform the Customer on incident. The basic service excludes remedial of incident.

##### Backup and restore

Backup and restore of Application Insights: N/A

Service restore of Application Insights: The configuration of Azure Application Insight can be recovered from Infrastructure-as-code if its configuration has been done through infrastructure as code.

Backup and restore of resources monitored by Application Insights: N/A

Restore from IaC for resources monitored by Application Insights: N/A

##### Limitations & pre-requisite

the Application Insights basic monitoring service is monitoring only.

### Charging model

|  |
| --- |
| Work Unit |
| Per managed resource |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Other changes | Estimation in tokens based on time spent |

## Application service

### Description

Azure App Service is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, be it .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications run and scale with ease on both Windows and Linux-based environments.

App Service not only adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. You can also take advantage of its DevOps capabilities, such as continuous deployment from Azure DevOps, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and TLS/SSL certificates.

With App Service, you pay for the Azure compute resources you use. The compute resources you use are determined by the App Service plan that you run your apps on

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### Co-manage option

To be defined

##### KPI & alerts

Monitoring

Yes

KPI monitored

Response Time   
Average memory working set  
CPU Time   
Data In   
Data Out   
Health Check Status   
Requests   
Thread Count   
Other azure metrics on demand

Alerts observed

Health Check Status   
Others will be identified with the customer

##### Backup and restore

Data backup and restore

Provided by Azure Backup depending on customer’s design and build.

Service restore

On-demand from Azure Backup.

##### Azure SLA High Availability

HA and non HA are provided by Azure depending on the design and service parameter configuration

##### Recovery for region failure

Optional with charge: based on regular snapshot and recovery from this snapshot.

### Charging model

|  |
| --- |
| Work Unit |
| Per Web Application |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Deploy a new version of an existing webapp | 1 token |
| Add a custom domain on an Azure App Service | 2 tokens |
| Configure a connection string to access another resource | 1 token |
| Add CORS functionality | Estimation in tokens based on time spent |
| Add a SSL certificate | 1 token |
| Enable authentication for front-end application | 1 token |
| Enable authorization for front-end application | Estimation in tokens based on time spent |
| Move an App Service in another region | 1 token |
| Other changes | Estimation in tokens based on time spent |

## Azure DNS

### Description

Azure DNS host your Domain Name System (DNS) domains in Azure.

Azure DNS Private Zones provides a simple, reliable, secure DNS service to manage and resolve names in a VNET without the need for you to create and manage custom DNS solution. This capability allows you to use your own domain names, rather than the Azure-provided names available today. It provides name registration in VNet and also resolution for VNets that does not need registration.

Additionally, you can configure zones names with a split-horizon view allowing a private and a public DNS zone to share the same name.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git used by OBS which includes the reference configuration of the DNS.
* This file can be executed with a CI/CD used by OBS and the execution has been tested successfully.

##### Co-manage option

For the Public part, OBS work with the customer for the publics domain naming context.

For the private Part, a RACI must be done.

##### KPI & alerts

Monitoring

Yes, On demand by Network watcher

KPI monitored

Number of changes in the DNS database.

Alerts observed

Number of changes in the DNS rules

##### Backup and restore

Data backup and restore

Yes. Backup is proposed based on regular export.

Service restore

The CI/CD chain is used to redeploy the records from a backup zone into the native DNS service or from an export

##### Azure SLA High Availability and Disaster Recovery inter-region

Microsoft global network of name servers has the scale and redundancy to give you ultra-high availability for your domains.

### Charging model

|  |
| --- |
| Work Unit |
| Per resource group |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Create / update/ delete zone (one zone including reverse) | 1 token |
| Create / update/ delete record (up to 10 records) | 1 token |
| Zone delegation\* | 1 token |
| Configure Firewall DNS | 2 tokens |
| Other changes | Estimation in tokens based on time spent |

Zone Delegation\*: Specification should be received as a prerequisite.

## Azure Firewall

### Description

Azure Firewall is a managed, cloud-based network security service that protects your Azure Virtual Network resources. It's a fully stateful firewall as a service with built-in high availability and unrestricted cloud scalability.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git used by OBS which includes the reference configuration of the service.
* This file can be executed with a CI/CD used by OBS and the execution has been tested successfully.

##### Co-manage option

No, OBS manages the Firewall

##### KPI & alerts

Monitoring

Yes

KPI monitored

* Application rules hit count
* Network rules hit count
* Data processed
* Throughput
* Firewall health state
* SNAT port utilization

Alerts observed

**Default**

Firewall health state

**Optional**

Application rules hit count

Network rules hit count

Data processed

Throughput

SNAT port utilization

##### Backup and restore

Data backup and restore

On demand export of rules in JSON format file

Service restore

The Continuous Deployment chain is used to redeploy the Firewall from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

Azure Firewall can be configured during deployment to span multiple Availability Zones for increased Availability depending on design Scope of Work.

##### Network and security managed services

Additional Network and Security Managed services might be added optionally depending on Scope of Work.

### Charging model

|  |
| --- |
| Work Unit |
| Per pack of 30 rules in Azure Firewall |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Add / modify / delete rules or NAT (up to 5 rules) | 1 token |
| Other changes | Estimation in tokens based on time spent |

## Azure Function

### Description

Azure Function processes events with serverless code

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference code of the Function.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### Co-manage option

TBD

##### KPI

KPIs:

* Response Time
* Average memory working set
* Function execution count
* Function execution unit
* Data In
* Data Out
* Health Check Status
* Requests
* Thread Count
* Other azure metrics on demand

##### Alerts

By default no, customized alerting can be added as an option based on customer needs.

##### Backup and restore

Data backup and restore

Backup is not used by default.

Service restore

By default, the Function source code in the GIT is the referential and the Continuous Deployment chain workflow is used to deploy it. Shall a problem occur on a Function, the Continuous Deployment chain is used to redeploy the Function from the version of reference in the GIT.

##### Azure SLA High Availability

HA and non HA are provided by Azure depending on the design and service parameter configuration as per design Scope Of Work.

##### Disaster Recovery inter-region

In the design Scope Of Work, customer can request HA inter-region to be configured to protect against region loss.

### Charging model

|  |
| --- |
| Work Unit |
| Per package of 100 lines of Function code |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Activate / deactivate a function | 1 token |
| Connect a function app to a Virtual Network | 1 token |
| Configure a connection string to access another resource | 1 token |
| Add a customer domain on Function App | 2 tokens |
| Add CORS functionality | Estimation in tokens based on time spent |
| Add SSL certificate | 1 token |

## Azure storage

### Description

The description of Managed Services for storage exclude that Managed Services for Disks which is included in the Managed OS for Virtual Machines.

### Build to run service included in the OTC

##### Build service pre-requisite

* Please refer to generic description

##### Build to run service

* Build to run service for Storage are necessary. They encompass the parameters setting for the storage e.g Tiering. Optionally, if an optional recurring managed service has been requested, build to run task will include the selection of Kpis to be observed and alerts to be set up based on KPI thresholds, or external calls to test the availability of the storage. Please refer to generic build to run description.

### RUN services included in the MRC

Recurring run managed services for Azure Storage are optional. Depending on Customer’s interest in monitoring the storage KPIs, in alerting based on KPIs, in backup / restore, the Customer may request the service. By default, there is no recurring task proposed on storage, but on demand changes and on demand investigations.

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the storage and of the metrics and alerts observed for the storage.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### Co-manage option

Yes

##### KPI & alerts

Monitoring

Optional: Metrics and health probes

KPI monitored

* Work on-going

Alerts observed

* API not reachable
* Transactions failure rates
* Size threshold

##### Backup and restore

Data backup and restore

Optional: storage can be highly available. Whether the customer wants a versioning of backup for storage, it is provided has part of a recurring proposal

Service restore

Optional: subject to customer having ordered backup and restore for storage.

##### Azure SLA High Availability and Disaster Recovery inter-region

Multiple available options are proposed by Azure depending on the class of service.

### Charging model

|  |
| --- |
| Work Unit |
| Per 5 storage accounts |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Configure access policy for storage account | 1 token |
| Changes | Estimation in tokens based on time spent |

## Content Delivery Network (CDN)

### Description

Azure Content Delivery Network (CDN) is a global CDN solution for delivering high-bandwidth content. It can be hosted in Azure or any other location. With Azure CDN, you can cache static objects loaded from Azure Blob storage, a web application, or any publicly accessible web server, by using the closest point of presence (POP) server. Azure CDN can also accelerate dynamic content, which cannot be cached, by leveraging various network and routing optimizations.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the CDN.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### Co-manage option

Yes based on RACI determined during pre-sales or build.

##### KPI & alerts

Monitoring

Yes: Metrics and diagnostic logs

KPI monitored

* Byte Hit Ratio
* Request Count
* Response Size
* Total Latency
* Customized ping page per zone

Alerts observed

* Customized ping page per zone
* Latency

##### Backup and restore

Data backup and restore

Can be exported from CI/CD Pipeline.

Service restore

The Continuous Deployment chain is used to redeploy the CDN from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

Based on design SOW, the service can be built in multiple regions.

### Charging model

|  |
| --- |
| Work Unit |
| Per Endpoint |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Purge CDN | 1 Token |
| Add URL | 1 Token |
| Other changes | Estimation in tokens based on time spent |

## Event Hubs

### Description

Azure Event Hubs is a big data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters.

The following scenarios are some of the scenarios where you can use Event Hubs:

* Anomaly detection (fraud/outliers)
* Application logging
* Analytics pipelines, such as clickstreams
* Live dashboarding
* Archiving data
* Transaction processing
* User telemetry processing
* Device telemetry streaming.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the Event Hubs.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### Co-manage option

No, OBS manages the Load Balancer

##### KPI & alerts

Monitoring

Yes

KPI monitored

* Incoming Requests
* Successful Requests
* Throttled Requests

Alerts observed

* Throttled Requests

##### Backup and restore

Data backup and restore

Not applicable. Event Hubs does not store data persistently.

Datastore is excluded from the scope of work of the work unit. It is a separate work unit.

Note: as a chargeable separate service the datastore where the data has been injected can be backed up and restored.

Service restore

The Continuous Deployment chain is used to redeploy the Event Hubs from the configuration file of reference for production environment committed in the Git.

Restore of the datastore is a separate work Unit.

##### Azure SLA High Availability and Disaster Recovery inter-region

Azure ensures High Availability of the Event Hubs with standard SKU.

Maintaining a cross region Disaster Recovery requires specific design and subject to a specific additional charging.

### Charging model

|  |
| --- |
| Work Unit |
| Per source type |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Add a new source into Event Hubs | On quote |
| Other changes | Estimation in tokens based on time spent |

## Express Route

### Description

Express Route allow to extend on-premises networks into Microsoft Cloud (Azure, Microsoft 365) over a private connection. Express Route connections offer more reliability, faster speeds, consistent latencies and higher security than connections over Internet.

Managed service for express route only covers the Azure End point. It does not cover the distant end point nor the end-to-end link. Managing end to end networking can be proposed by OBS additionally, based on Scope Of Work and RACI.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including a partial configuration of the connectivity.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

* Availability
* Bandwidth

Alerts observed

* Availability
* Bandwidth
* Custom status metric

##### Backup and restore

Data backup and restore

On demand. Backup is proposed based on export template.

Service restore

The Continuous Deployment chain is used to redeploy the initial configuration or from an export.

##### Azure SLA High Availability and Disaster Recovery inter-region

Azure ensures High Availability of the Express Route and can be maximize by design.

Cross region Disaster Recovery based on WAN Architecture requirements.

##### Network and security managed services

No by default.

### Charging model

|  |
| --- |
| Work Unit |
| Per peering |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Modify bandwidth | 1 token |
| Other changes | Estimation in tokens based on time spent |

## Front door

### Description

Azure Front Door is a global, scalable entry-point that uses the

Microsoft global edge network to create fast, secure, and widely

scalable web applications. With Front Door, you can transform your

global consumer and enterprise applications into robust, high-performing

personalized modern applications with contents that reach a global

audience through Azure.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the Azure Front Door.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Metrics supported for Front Door service:

* RequestCount
* RequestSize
* ResponseSize
* TotalLatency
* BackendRequestCount
* BackendRequestLatency
* BackendHealthPercentage
* WebApplicationFirewallRequestCount

Alerts observed

* BackendRequestLatency(CDN)
* BackendHealthPercentage(CDN)
* WebApplicationFirewallRequestCount(WAF)

##### Backup and restore

Data backup and restore: N/A

On-demand export template

Service restore

The Continuous Deployment chain is used to redeploy the Front Door from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

The service is in high-availability pattern by default in Azure.

### Charging model

|  |
| --- |
| Work Unit |
| Per Instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Add / modify /delete a rule | 1 Token |
| Add / modify a backend | 2 Tokens |
| Add a new CDN configuration | 3 Tokens |
| Other changes | Estimation in tokens based on time spent |

## Key vault

### Description

Azure Key Vault is a cloud service for securely storing and accessing secrets. Key Vault has two service tiers: Standard, which encrypts with a software key, and a Premium tier, which includes HSM-protected keys.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the KeyVault.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

KPIs:

* Vault availability
* Vault saturation
* Service API Latency
* Total Service API Hits by Activity Type
* Total Service API Hits by HTTP Status Code

|  |  |  |  |
| --- | --- | --- | --- |
| ***Alert*** | ***Description*** | ***Severity*** | ***Source*** |
| **Overall Vault Availability** | Alert if vault is unavailable (less than 100%) | P1 | Metrics |
| **Overall Vault Saturation** | Alert if vault capacity is exceeded (greater than 75%) | P3 | Metrics |
| **Overall Service API Latency** | Alerts if average latency is above 500 ms | P3 | Metrics |
| **Count Total Service API Hits By Status Code** | Alert if the total of error code exceed the standard value for the customer context (dynamic value) | P1 | Metrics |
| **Vault Deleted** | Alert if key vault is deleted | P1 | Activity Log |

##### Backup and restore

Data backup and restore

By default OBS enables soft delete option on Azure KeyVault which preserves the data for 90 days.

Backup is an optional task based on scope of work as it requires either a secured storage or a secondary KeyVault as a target. By setting-up backup to a secondary KeyVault, one protects against disaster on the KeyVault, see below.

##### Azure SLA High Availability and Disaster Recovery inter-region

Supported by Microsoft. The Key Vault content of one region is automatically replicated in its paired region except in the case of the Brazil South region.

The rare times an entire Azure region is unavailable, the requests that you make of Azure Key Vault in that region are automatically routed (failed over) to a secondary region except in the case of the Brazil South region.

##### Security

Security recommendations can be part of an optional security scope of work based on customer request.

By default, the MRC does not cover security recommendations

### Information on Azure service

##### SKU

* Standard: Software encrypted keys
* Premium: Hardware encrypted keys (HSM-protected keys)

##### Service Limits

<https://docs.microsoft.com/en-us/azure/key-vault/general/service-limits>

### Charging model

|  |
| --- |
| Work Unit |
| Per Key Vault instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Add/remove key | 1 token |
| Configure access policy | 2 tokens |
| Add/Remove/Configure end of life for Certificate | 1 token |
| Configure Azure native services to use key vault | Estimation in tokens based on time spent |
| Other changes | Estimation in tokens based on time spent |

## Load Balancer

### Description

Azure Load Balancer operates at layer 4 of the Open Systems Interconnection (OSI) model. It's the single point of contact for clients. Load balancer distributes inbound flows that arrive at the load balancer's front end to backend pool instances. These flows are according to configured load-balancing rules and health probes. The backend pool instances can be Azure Virtual Machines or instances in a virtual machine scale set.

A [public load balancer](https://docs.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configurations) can provide outbound connections for virtual machines (VMs) inside your virtual network. These connections are accomplished by translating their private IP addresses to public IP addresses. Public Load Balancers are used to load balance internet traffic to your VMs.

An [internal (or private) load balancer](https://docs.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configurations) is used where private IPs are needed at the frontend only. Internal load balancers are used to load balance traffic inside a virtual network. A load balancer frontend can be accessed from an on-premises network in a hybrid scenario.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the load balancer.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### Co-manage option

No, OBS manages the Load Balancer

##### KPI & alerts

Monitoring

Yes: insights, Metrics and health probes

KPI monitored

* Data path availability
* Health probe status
* SYN (synchronize) count
* SNAT connection count
* Allocated SNAT ports
* Used SNAT ports
* Used SNAT ports
* Bytecount
* Packet count

Alerts observed

* Data path availability
* Health probestatus

##### Backup and restore

Data backup and restore

Not applicable. Load balancer does not store data persistently.

Service restore

The Continuous Deployment chain is used to redeploy the Load Balancer from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

Azure ensures High Availability of the Load Balancer with standard SKU.

Maintaining a cross region Disaster Recovery requires specific design and subject to a specific additional charging.

### Charging model

|  |
| --- |
| Work Unit |
| Per Load Balancer instance |

### Changes catalogue – in Tokens, per act

|  |  |  |
| --- | --- | --- |
| Changes examples | Effort | Impact on MRC |
| Setup / modify / delete URI | 1 token |  |
| Change health probes / Add new backend | 2 tokens |  |
| Other changes | Estimation in tokens based on time spent |  |

## Log Analytics – basic monitoring with class 2 transition

### Description

Log Analytics is a tool in the Azure portal to edit and run log queries from data collected by Azure Monitor Logs and interactively analyze their results. You can use Log Analytics queries to retrieve records that match particular criteria, identify trends, analyze patterns, and provide a variety of insights into your data.

The basic monitoring excludes the middleware and application management as well as remedial actions.

### Build to run service included in the OTC

##### Build to run service pre-requisite

The pre-requisite to Log Analytics basic monitoring with class 2 transition is that Log Analytics has been configured by the Customer including

* Log collection for the resources
* Metrics and alerts forwarded to Azure Monitor

##### Build to run service

For Log Analytics basic monitoring with class 2 transition, the build to run service included in the OTC consists in integrating the alerts from Azure Monitor configured in Log Analytics into OBS supervision backend.

### RUN services included in the MRC

##### Run service pre-requisite

* The resource configured is in the inventory Scope of Work of managed service: infrastructure resource, middleware resource, application resource, database resource, Kubernetes cluster resource, microservice resource, etc…
* A referential file exists in the Git including the reference configuration of Log Analytics.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

Alerts observed

* Alerts defined in Log Analytics and forwarded to Azure Monitor for resources in the Scope of Work of managed services.

##### Monitoring service

As part of the Log Analytics basic monitoring service, OBS operations will monitor the alerts, raise tickets and inform the Customer on incident. The basic service excludes troubleshooting or remedial of incident.

##### Backup and restore

Backup and restore of Log Analytics: N/A

Service restore of Log Analytics: The configuration of Azure Log Analytics can be recovered from Infrastructure-as-code if its configuration has been done through infrastructure as code.

Backup and restore of resources monitored by Log Analytics: N/A

Restore from IaC for resources monitored by Log Analytics: N/A

##### Limitations & pre-requisite

The Log Analytics basic monitoring service is monitoring only.

### Charging model

|  |
| --- |
| Work Unit |
| Per managed resource |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Filter and send logs for a specific resource | Estimation in tokens based on time spent |
| Other changes | Estimation in tokens based on time spent |

## Logic App

### Description

Automate the access and use of data across clouds without writing code

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### Co-manage option

Yes

##### KPI

* ActionLatency
* ActionsFailed
* ActionThrottledEvents
* RunFailurePercentage
* RunLatency
* RunsCancelled
* RunsCompleted
* RunsFailed
* RunsStarted
* RunsSucceeded

##### Alerts

Optional to be discussed with customer based on case by case.

##### Backup and restore

Data backup and restore

Not in place by default.

Service restore

The Continuous Deployment chain is used to redeploy the Logic App from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability

Depends on design Scope Of Work.

### Charging model

|  |
| --- |
| Work Unit |
| Per Application |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Changes on demand | Estimation in tokens based on time spent |

## Network Security Group - Network and Security management services

### Description

Azure network security group used to filter network traffic to\from Azure resources in an Azure virtual network. It contains security rules that allow or deny inbound and outbound network traffic.

At the basic level, managing Network Security group consists in building, deploying and maintaining the IaC for it and managing the changes.

The management of Network Security Groups is included as part of a larger bundle of Network and Security Managed services which provides network and security design, maintain, network watching, intrusion detection, troubleshooting depending on an agreed Scope of Work.

### Charging model

|  |  |
| --- | --- |
| Work Unit | OTC & MRC |
| Network and security management services | Custom, depending on agreed Scope of Work |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Add / modify / delete Security rules (up to 5 rules) excluding dependencies\* | 1 token |
| Add / modify / delete Security group (up to 5 rules) excluding dependencies\* | 1 token |
| Other changes | Estimation in tokens based on time spent |

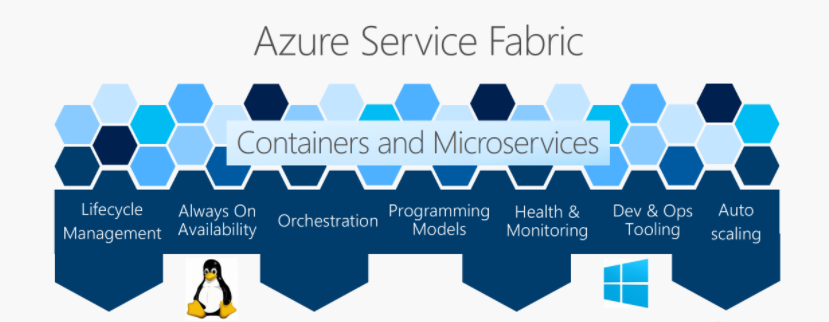
\*Dependencies include all triggered applications like Azure Sentinel, Log Analytics, Azure Firewall, Logic App Security, Azure DB services and other native services.

## Service Fabric

### Description

Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices and Azure Service Fabric is a distributed systems platform that makes it easy to package, deploy, and manage scalable and reliable microservices and containers. Service Fabric also addresses the significant challenges in developing and managing cloud native applications.

A key differentiator of Service Fabric is its strong focus on building stateful services. You can use the Service Fabric programming model or run containerized stateful services written in any language or code. You can create Service Fabric clusters anywhere, including Windows Server and Linux on premises and other public clouds, in addition to Azure.



Service Fabric powers many Microsoft services today, including Azure SQL Database, Azure Cosmos DB, Cortana, Microsoft Power BI, Microsoft Intune, Azure Event Hubs, Azure IoT Hub, Dynamics 365, Skype for Business, and many core Azure services.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the Service Fabric.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes – cluster level. KPI and Alerts for the micro-services layer is handled as part of the additional service called Managed Container.

KPI monitored

Metrics supported for Service Fabric:

* PrimaryCount
* ReplicaCount

Alerts observed

* Idem

##### Backup and restore for the Site Recovery configuration

Service backup and restore: The native Azure backup for Service Fabric is used.

##### Azure SLA High Availability and Disaster Recovery inter-region

A service fabric multi-node cluster delivers high-availability by design.

Deployed on multi-region, a multi-region availability can be achieved.

##### Limitations & pre-requisite

Managing the microservice layer is an additional managed service called Managed Container charged per microservices. Please refer to Managed Application main service description document.

### Charging model

|  |
| --- |
| Work Unit |
| Per cluster Node  Per microservice ? |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Scale application / services in a cluster | 1 Token |
| Deploy containers to Service Fabric cluster | 1 Token |
| Delete a cluster | 1 Token |
| Upgrade the runtime of a Service Fabric Cluster | 1 Token |
| Create a Service Fabric Cluster | On quote |
| Create a container image for Service Fabric (on quote) | Part of Managed Container ? |
| Deploy an Azure Service Fabric cluster across Availability Zones | On quote |
| Other changes | Estimation in tokens based on time spent |

## Site Recovery

### Description

Azure Site Recovery is delivering built-in disaster recovery service for Virtual Machines.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* List of Virtual Machines should be provided.
* Compatibility of workload Architecture with Site Recovery protection mechanism.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Metrics supported for Site Recovery service:

* Azure Site recovery replication stats
* Azure Site recovery replication dataupload rate
* Azure Site recovery replication recovery points
* Azure Site recovery replication replicated items

Alerts observed

* Azure Site recovery replication stats
* Azure Site recovery replication recovery points

##### Backup and restore for the Site Recovery configuration

Data backup and restore: N/A

Service restore: Restore will be recreating the Site Recovery Plan using the same configurations.

##### Testing the Site Recovery configuration (Failover/Failback Simulation)

Testing the site recovery can be handled as a complex change request. The time spent will be estimated in a number of Tokens.

##### Recovery of Virtual Machines with Site Recovery

The Virtual Machines protected by the Site Recovery can be recovered thanks to Azure mechanism. Implementing the recovery of Virtual Machines can be handled as a complex change request. The time spent will be estimated in a number of Tokens

##### Azure SLA High Availability and Disaster Recovery inter-region

The service purpose is to implement Disaster Recovery.

##### Limitations

Azure monitor is only used for replication monitoring within same region which might cause some limitations. A discussion is necessary for each customer case by case to discuss the monitoring.

### Charging model

|  |
| --- |
| Work Unit |
| Per Virtual Machine protected |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Adding a VM to an already defined Site Recovery | 2 Tokens |
| Testing the failover& failback Site Recovery | Estimation in tokens based on time spent |
| Failover for a site | Estimation in tokens based on time spent |
| Other changes | Estimation in tokens based on time spent |

## Traffic Manager

### Description

Azure Traffic Manager is a DNS-based traffic load balancer. This service allows you to distribute traffic to your public facing applications across the global Azure regions. Traffic Manager also provides your public endpoints with high availability and quick responsiveness.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the CDN.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

* Endpoint status by endpoint
* Queries by endpoint returned

Alerts observed

* Endpoint status by endpoint

##### Backup and restore

Data backup and restore

Can be exported from CI/CD Pipeline.

Service restore

The Continuous Deployment chain is used to redeploy the Traffic Manager from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

The service is globally managed by Microsoft

### Charging model

|  |
| --- |
| Work Unit |
| Per Profile |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Add/modify Backend | 1 Token |
| Other changes | Estimation in tokens based on time spent |

## Virtual Machines and OS

### Description

The Managed Service for Virtual Machines is called Managed OS. OBS manages both the OS and the Virtual Machine.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the Virtual Machines.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring is performed through configuration and activation of Azure Monitoring.

OBS backend supervision system is collecting alerts from Azure Monitoring.

**Alerts for VM with Linux OS:**

|  |
| --- |
| ***Alert*** |
| Filesystem Usage |
| Filesystem Inode Usage |
| Agent Status |
| Load Usage |
| Memory Usage |
| Network Status |
| Zombie Detected |
| Ping |

**Alerts for VM with Windows OS:**

|  |
| --- |
| ***Alert*** |
| Ping |
| Agent\_Status |
| CPU |
| Disks |
| Pagefile |
| Physical Memory |
| Windows Services |

##### OS patching

Azure Update Manager

For managed OS, OBS leverages Azure Update Manager for the patching of the Operating System (OS). It presents the advantage to provide real-time status of patching level, to be consistent with Azure security and visible through Azure Advisor.

Behavior: With Azure Update Manager, patches are decided by Microsoft and all patches are to be applied if mandatory for the Virtual Machine for Windows and Linux.

OBS Managed Application patching system

As an alternative when patches shall be chosen, OBS leverages its own central patching system whereby all patches have been validated and tested by OBS image factory. OBS patching system allows for central reporting to OBS operations teams of the proper patching level of each VM managed.

VPN connectivity to OBS CASA zone is a pre-requisite.

OBS shall not take responsibility of managed OS and its risks avoidance (security, defect) based on a Customer specific patching system.

##### Antivirus

For managed OS, OBS leverages its central anti-virus system based on Sophos. This requires the installation of the anti-virus agent on the VM OS for each VM as well as the VPN connectivity to OBS CASA zone. OBS systems allows for central reporting on Malware from its backend console system.

Would the Customer desire to keep its own Antivirus system, then OBS shall not be taken responsible for protection against viruses.

##### Backup and restore

Data backup and restore

By default, OBS leverages Azure Backup on the Virtual Machines for Managed OS. The configuration of Azure Backup pattern and well as retention period shall be agreed with the Customer prior to the RUN. As example: 1 x backup per week, 1x incremental backup per day per VM. The retention period depends on customer request.

Restore of VM are performed from the backup.

* In case of incident, latest version of backup can be restored
* Upon change request, a previous version of backup can be restored.

##### Azure SLA High Availability and Disaster Recovery inter-region

By default, a Virtual Machine is not highly available.

The Customer shall leverage Azure VM Availability Set to expect High availability for the Availability Set of VMs (design requirement)

The Customer shall leverage Azure Site Recovery to allow protection from disaster (optional).

##### Administration tasks tracing

Actions performed by OBS managed teams on the managed OS are done from OBS CASA zone through an access controlled by a CyberArk bastion. OBS CyberArk bastion protects the access and keep trace of the actions performed by the maintenance team allowing for audit.

The VPN connectivity to the CASA zone necessary for the management.

##### Login on to the Virtual Machine

For Windows OS based VM, access shall be granted by the Customer to OBS managed application operations staff through a domain account configured with proper privilege groups.

For Linux OS based VM, an encrypted key is created and provided to OBS managed application operations staff to log onto the VM. The key itself is stored in a safe i.e Azure KeyVault.

For Applications, in case of managed application: a secret stored in a safe.

##### Logs

Log management is not included in the managed OS / managed virtual machine service.

Optionally it can be activated through Azure Log Analytics through Change Request process.

##### Security

By default, the MRC includes the use of security policies and groups as per customer’s configuration request.

The MRC does not cover security recommendations. Security recommendations can be part of an optional security scope of work based on customer request.

##### Limitations

Managed Application services is provided only for OS versions supported by the CSP vendor.

### Charging model

|  |
| --- |
| Work Unit |
| Per Virtual Machine instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Create Application Security Groups | 4 Tokens |
| Modify/delete Application Security Groups | 2 Tokens |
| Create Virtual Machines, Availability Sets, VMSS, disk and image | 8 Tokens |
| Modify Virtual Machines, Availability Sets | 4 Tokens |
| Modify VMSS, disk and image [4tk] | 4 Tokens |
| Delete Virtual Machines, Availability Sets | 4 Tokens |
| Delete VMSS, disk and image |  |
| Start/Stop/Restart Virtual Machines | 2 Tokens |
| Create/modify/delete Storage Accounts | 2 Tokens |

## VPN Gateway

### Description

A VPN gateway is a specific type of virtual network gateway that is used to send encrypted traffic between an Azure virtual network and an on-premises location over the public Internet or Microsoft backbone network.

Additional managed services ca n be added optionally based on Scope of Work, refer to network and security services.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git used by OBS which includes the reference configuration of the VPN Gateway.
* This file can be executed with a CI/CD used by OBS and the execution has been tested successfully.

##### Co-manage option

No, OBS manages the VPN Gateway

##### KPI & alerts

Monitoring

This service can be monitored by Azure Monitor using Alerts and Metrics

KPI monitored

* + - AverageBandwidth
    - P2SBandwidth
    - P2SConnectionCount
    - TunnelAverageBandwidth
    - TunnelEgressBytes
    - TunnelEgressPackets
    - TunnelEgressPacketDropTSMismatch
    - TunnelIngressBytes
    - TunnelIngressPackets
    - TunnelIngressPacketDropTSMismatch

Alerts observed

* + - AverageBandwidth
    - P2SBandwidth
    - P2SConnectionCount
    - TunnelAverageBandwidth
    - TunnelEgressBytes
    - TunnelEgressPackets
    - TunnelEgressPacketDropTSMismatch
    - TunnelIngressBytes
    - TunnelIngressPackets
    - TunnelIngressPacketDropTSMismatch

##### Backup and restore

Data backup and restore

The Backup is N/A for VPN Gateway, but the deployment template can be exported on-demand before any configuration change.

Service restore

The Continuous Deployment chain is used to redeploy the VPN Gateway from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

Azure proposed availability for the VPN Gateway:

* 99.9% availability for each Basic Gateway for VPN or Basic Gateway for ExpressRoute.
* 99.95% availability for all Gateway for VPN SKUs excluding Basic.
* 99.95% availability for all Gateway for ExpressRoute

Availability is ensured by Azure and depends on design.

##### Network and security managed services

Additional Network and Security Managed services might be added optionally depending on Scope of Work.

### Charging model

|  |
| --- |
| Work Unit |
| Per VPN Gateway |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Route modification | 1 token |
| Create IPSec site-to-site Tunnel | 2 tokens |
| Configure Network Gateway | Estimation in tokens based on time spent |
| Other changes | Estimation in tokens based on time spent |

## Web Application Firewall

### Description

Azure Web Application Firewall (WAF) provides centralized protection of your web applications from common exploits and vulnerabilities. Web applications are increasingly targeted by malicious attacks that exploit commonly known vulnerabilities. SQL injection and cross-site scripting are among the most common attacks.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### Co-manage option

No, OBS manages the WAF

OR, can be done with RACI determined during pre-sales or project build.

##### KPI & alerts

Monitoring

Yes

KPI monitored

* Data Processed
* Firewall Health
* SNAT Port Utilization
* Application Rule Hit
* Network Rule Hit

Alerts observed

* Firewall Health

##### Backup and restore

Data backup and restore

By default, N/A.

Service restore

The Continuous Deployment chain is used to redeploy the rules from the configuration file of reference for production environment committed in the Git.

##### Azure SLA High Availability and Disaster Recovery inter-region

Based on design Scope of Work, to be confirmed during presales phase.

##### Network and security managed services

Additional Network and Security Managed services might be added optionally depending on Scope of Work.

### Charging model

|  |
| --- |
| Work Unit |
| Per IP of protected asset |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Add already existing rule | 1 token |
| modify/delete rule/rules (up to 5) | 1 token |
| Create a simple rule | 1 token |
| Other changes | Estimation in tokens based on time spent |

* 1. **Azure Database for MySQL**
     1. **Description**

Azure Database for MySQL is a relational database service powered by the MySQL community edition. You can use either Single Server or Flexible Server to host a MySQL database in Azure. It's a fully managed database as a service offering that can handle mission-critical workloads with predictable performance and dynamic scalability.

* + 1. **Build to run service included in the OTC**
       1. ***Build service pre-requisite***
* Refer to generic description.
  + - 1. ***Build to run service***
* Refer to generic description.
  + 1. **RUN services included in the MRC**
       1. ***Run service pre-requisite***
* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.
  + - 1. ***KPI & alerts***

**Monitoring**

Yes

**KPI monitored**

Azure Monitor supported metrics for Azure Database for MySQL are available at: [**Azure Monitor supported metrics by resource type - Azure Monitor | Microsoft Docs**](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported)

**Alerts observed**

* active\_connections
* aborted\_connections
* total\_connections
* cpu\_credits\_consumed
* cpu\_credits\_remaining
* io\_consumption\_percent
* cpu\_percent
* memory\_percent
* storage\_percent
* network\_bytes\_egress
* replication\_lag
  + - 1. ***Backup and restore***
* **Data backup and restore**

Azure Database for MySQL servers are backed up periodically to enable Restore features. Using this feature, you may restore the server and all its databases to an earlier point-in-time, on a new server.

The backup retention period governs how far back in time a point-in-time restore can be retrieved, since it's based on backups available. It could be set between 7 and 35 days.

* **Service restore**

The Continuous Deployment chain is used to redeploy the rules from the configuration file of reference for production environment committed in the Git.

* + - 1. ***Azure SLA High Availability and Disaster Recovery inter-region***

Azure Database for MySQL provides fast restart capability of database servers, redundant storage, and efficient routing from the Gateway. For additional data protection, you can configure backups to be geo-replicated, and also deploy one or more read replicas in other regions. The estimation will be based on design Scope of Work, to be confirmed during presales phase.

* + - 1. ***Minor Version patching***

Azure Database for MySQL automatically patches servers with minor releases (within maintenance window).

* + - 1. ***Major Version patching***

Automatic in-place upgrades for major versions from 5.6 to 5.7 is supported.

Automatic in-place upgrades for major versions from 5.7 to 8.0 is not supported.

It could be done using either one of the following:

- Use mysqldump to move a database to a server created with the new engine version.

- Use Azure Database Migration service for doing online upgrades.

The estimation will be based on the database size.

* + 1. **Charging model**

|  |
| --- |
| Work Unit |
| Per Database Instance |

* + 1. **Changes catalogue – in Tokens, per act**

|  |  |
| --- | --- |
| Changes examples | Effort |
| Provision database | 2 tokens |
| Restart server | 1 token |
|  |  |
| Azure Database for MySQL failover | 1 token |
| Stop & start database | 1 token |
| Delete database | 1 token |
| Move an Azure Database for MySQL Flexible server from one Azure region to another using the Azure portal | 2 tokens |
| Create a clone | Estimation in tokens based on the database size |
| Restore a server to point-in-time and into a new copy of the server | Estimation in tokens based on the database size |
| Modify the service parameters configuration | 1 token |
| Major version upgrade in Azure Database for MySQL | Estimation in tokens based on time spent |
| Other changes | Estimation in tokens based on time spent |

## Azure Database for PostgreSQL

### Description

Azure Database for PostgreSQL is a relational database service based on the open-source Postgres database engine. It's a fully managed database-as-a-service that can handle mission-critical workloads with predictable performance, security, high availability, and dynamic scalability.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Azure Monitor supported metrics for Azure Database for PostgreSQL are available at:   
[Azure Monitor supported metrics by resource type - Azure Monitor | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported)

Alerts observed

* active\_connections
* connections\_failed
* connections\_succeeded
* cpu\_percent
* memory\_percent
* storage\_percent
* read\_throughput
* write\_throughput

##### Backup and restore

Data backup and restore

Azure Database for PostgreSQL takes backups of the data files and the transaction log. Depending on the supported maximum storage size, we either take full and differential backups (4-TB max storage servers) or snapshot backups (up to 16-TB max storage servers). These backups allow you to restore a server to any point-in-time within your configured backup retention period. The default backup retention period is seven days. You can optionally configure it up to 35 days. All backups are encrypted using AES 256-bit encryption.

Azure Database for PostgreSQL provides the flexibility to choose between locally redundant or geo-redundant backup storage in the General Purpose and Memory Optimized tiers.

Service restore

Recovery will be from Infra as Code.

##### Azure SLA High Availability and Disaster Recovery inter-region

Built on Azure architecture, the service has inherent high availability, redundancy, and resiliency capabilities to mitigate database downtime from planned and unplanned outages, without requiring you to configure any additional components.

##### Minor Version patching

Azure Database for PostgreSQL automatically patches servers with minor releases (within maintenance window).

##### Major Version patching

Automatic in-place upgrades for major versions are not supported. It could be done using either one of the following:

- Use pg\_dump and pg\_restore to move a database to a server created with the new engine version.

- Use Azure Database Migration service for doing online upgrades.

### Charging model

|  |
| --- |
| Work Unit |
| Per Database Instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Provision database | 2 Tokens |
| Restart instance | 1 Token |
| Delete instance | 1 Token |
| Modify compute/storage | 2 Tokens |
| Modify High availability | 1 Token |
| Modify Server parameters | 1 Token |
| Restore point-in-time to a new server | Estimation in tokens based on database size |
| Modify the server parameters | 1 Token |
| Other changes | Estimation in tokens based on time spent |

## Azure SQL Database

### Description

Azure SQL Database is a fully managed platform as a service (PaaS) database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring without user involvement.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Azure Monitor supported metrics for Azure SQL Database are available at:   
[Azure Monitor supported metrics by resource type - Azure Monitor | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported)

Alerts observed

* Metric alert
* Log alert
* Activity log alert

##### Backup and restore

Data backup and restore

Azure SQL Database creates:

* [Full backups](https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/full-database-backups-sql-server) every week.
* [Differential backups](https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/differential-backups-sql-server) every 12 or 24 hours.
* [Transaction log backups](https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/transaction-log-backups-sql-server) approximately every 10 minutes.

The exact frequency of transaction log backups is based on the compute size and the amount of database activity. When you restore a database, the service determines which full, differential, and transaction log backups need to be restored.

By default, Azure SQL Database stores data in geo-redundant storage blobs that are replicated to a paired region. Geo-redundancy helps protect against outages that affect backup storage in the primary region. It also allows you to restore your databases in a different region in the event of a regional outage.

This table summarizes the capabilities and features of [point-in-time restore (PITR)](https://docs.microsoft.com/en-us/azure/azure-sql/database/recovery-using-backups?view=azuresql#point-in-time-restore), [geo-restore](https://docs.microsoft.com/en-us/azure/azure-sql/database/recovery-using-backups?view=azuresql#geo-restore), and [long-term retention](https://docs.microsoft.com/en-us/azure/azure-sql/database/long-term-retention-overview?view=azuresql).

|  |  |  |  |
| --- | --- | --- | --- |
| Backup property | PITR | Geo-restore | LTR |
| Types of SQL backup | Full, differential, log. | Replicated copies of PITR backups. | Only the full backups. |
| Recovery point objective (RPO) | 10 minutes, based on compute size and amount of database activity. | Up to 1 hour, based on geo-replication. \* | One week (or user's policy). |
| Recovery time objective (RTO) | Restore usually takes less than 12 hours but could take longer, depending on size and activity. | Restore usually takes less than 12 hours but could take longer, depending on size and activity. | Restore usually takes less than 12 hours but could take longer, depending on size and activity. |
| Retention | 7 days by default, configurable up to 35 days. | Enabled by default, same as source. \*\* | Not enabled by default. Retention is up to 10 years. |
| Azure Storage | Geo-redundant by default. You can optionally configure zone-redundant or locally redundant storage. | Available when PITR backup storage redundancy is set to geo-redundant. Not available when PITR backup storage is zone-redundant or locally redundant. | Geo-redundant by default. You can configure zone-redundant or locally redundant storage. |

\* For business-critical applications that require large databases and must ensure business continuity, use [auto-failover groups](https://docs.microsoft.com/en-us/azure/azure-sql/database/auto-failover-group-sql-db?view=azuresql).

\*\* All PITR backups are stored on geo-redundant storage by default, so geo-restore is enabled by default.

Service restore

Recovery will be from Infra as Code.

##### Azure SLA High Availability and Disaster Recovery inter-region

Azure SQL Database and Azure SQL Managed Instance feature a built-in high availability solution, that is deeply integrated with the Azure platform. It is dependent on Service Fabric for failure detection and recovery, on Azure Blob storage for data protection, and on Availability Zones for higher fault tolerance (as mentioned earlier in document not applicable to Azure SQL Managed Instance yet). In addition, SQL Database and SQL Managed Instance use the Always On availability group technology from the SQL Server instance for replication and failover. The combination of these technologies enables applications to fully realize the benefits of a mixed storage model and support the most demanding SLAs.

### Charging model

|  |
| --- |
| Work Unit |
| Per Database Instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| provision database | 2 tokens |
| delete database | 2 tokens |
| Restore a server to point-in-time | Estimation in tokens based on the database size |
| Modify the service parameters configuration | 1 token |
| Other changes | Estimation in tokens based on time spent |
| Changes examples | Effort |
| Other changes | Estimation in tokens based on time spent |

## Azure Cosmos DB

### Description

Azure Cosmos DB is a fully managed NoSQL database. Cosmos DB handles most of the database management functions with automatic management, updates and patching. It also handles capacity management with cost-effective serverless and automatic scaling options that respond to application needs to match capacity with demand.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Azure Monitor supported metrics for Azure Cosmos DB are available at:   
[Azure Monitor supported metrics by resource type - Azure Monitor | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported)

Alerts observed

* ServiceAvailability
* TotalRequests
* DataUsage
* IndexUsage
* ReplicationLatency
* ServerSideLatency
* CosmosDbRequests

##### Backup and restore

Data backup and restore

There are two backup modes:

* Continuous backup mode – This mode has two tiers. One tier includes 7-day retention and the second includes 30-day retention. Continuous backup allows you to restore to any point of time within either 7 or 30 days.
* Periodic backup mode - This mode is the default backup mode for all existing accounts. In this mode, you configure a backup interval and retention for your account. The maximum retention period extends to a month. The minimum backup interval can be one hour.

Data restore will be done from backup.

Service restore

Recovery will be from Infra as Code.

### Charging model

|  |
| --- |
| Work Unit |
| Per Database Instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| provision database | 2 tokens |
| delete database | 2 tokens |
| Restore a server to point-in-time | Estimation in tokens based on the database size |
| Modify the service parameters configuration | 1 token |
| Other changes | Estimation in tokens based on time spent |

## Azure Database for MariaDB

### Description

Azure Database for MariaDB is a managed service you can use to run, manage, and scale highly available MySQL databases in the cloud.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Azure Monitor supported metrics for Azure Database for Maria DB are available at:   
[Azure Monitor supported metrics by resource type - Azure Monitor | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported)

Alerts observed

* active\_connections
* connections\_failed
* cpu\_percent
* memory\_percent
* storage\_percent
* serverlog\_storage\_percent
* io\_consumption\_percent
* seconds\_behind\_master
* network\_bytes\_egress

##### Backup and restore

Data backup and restore

Azure Database for MariaDB takes backups of the data files and the transaction log. These backups allow you to restore a server to any point-in-time within your configured backup retention period. The default backup retention period is seven days. You can optionally configure it up to 35 days. All backups are encrypted using AES 256-bit encryption.

These backup files aren't user-exposed and can't be exported. These backups can only be used for restore operations in Azure Database for MariaDB.

Long-term retention of backups beyond 35 days is currently not natively supported by the service yet.

Azure Database for MariaDB provides the flexibility to choose between locally redundant or geo-redundant backup storage in the General Purpose and Memory Optimized tiers.

Data restore

In Azure Database for MariaDB, performing a restore creates a new server from the original server's backups and restores all databases contained in the server.

There are two types of restore available:

* Point-in-time restore is available with either backup redundancy option and creates a new server in the same region as your original server utilizing the combination of full and transaction log backups.
* Geo-restore is available only if you configured your server for geo-redundant storage and it allows you to restore your server to a different region utilizing the most recent backup taken.

Service restore

Recovery will be from Infra as Code.

##### Azure SLA High Availability and Disaster Recovery inter-region

Azure Database for MariaDB provides built-in high availability.

### Charging model

|  |
| --- |
| Work Unit |
| Per Database Instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Provision a database | 2 Token |
| Restart instance | 1 Token |
| Delete instance | 1 Token |
| Restore point-in-time to a new server | Estimation in tokens based on database size |
| Modify the server parameters | 1 Token |
| Other changes | Estimation in tokens based on time spent |

## Azure Managed Instance for Apache Cassandra

### Description

Azure Managed Instance for Apache Cassandra provides automated deployment and scaling operations for managed open-source Apache Cassandra datacenters.

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Azure Monitor supported metrics for Apache Cassandra are available at:   
[Azure Monitor supported metrics by resource type - Azure Monitor | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported)

Alerts observed

* cassandra\_cache\_hit\_rate
* cassandra\_cache\_size
* cassandra\_table\_row\_cache\_hit
* cassandra\_client\_request\_failures
* cassandra\_client\_request\_timeouts
* cassandra\_client\_request\_contention\_histogram
* cassandra\_table\_bloom\_filter\_false\_ratio
* cassandra\_table\_bloom\_filter\_false\_positives
* cassandra\_table\_bloom\_filter\_disk\_space\_used
* cassandra\_table\_write\_latency
* cassandra\_thread\_pools\_currently\_blocked\_tasks

##### Backup and restore

Data backup and restore

Snapshot backups are enabled by default and taken every 4 hours with Medusa. Backups are stored in an internal Azure Blob Storage account and are retained for up to 2 days (48 hours).

Data restore

Backups can be restored to the same VNet/subnet as your existing cluster, but they cannot be restored to the same cluster. Backups can only be restored to new clusters. Backups are intended for accidental deletion scenarios and are not geo-redundant. They are therefore not recommended for use as a disaster recovery (DR) strategy in case of a total regional outage. To safeguard against region-wide outages, we recommend a multi-region deployment.

Service restore

Recovery will be from Infra as Code.

##### Azure SLA High Availability and Disaster Recovery inter-region

### Charging model

|  |
| --- |
| Work Unit |
| Per Database Instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| provision cluster | 3 tokens |
| scale cluster | 3 tokens |
| delete cluster | 3 tokens |
| update Cassandra configuration | 1 token |
| Other changes | Estimation in tokens based on time spent |

## Azure Cache For Redis

### Description

Azure Cache for Redis provides an in-memory data store based on the Redis software. Redis improves the performance and scalability of an application that uses backend data stores heavily. It's able to process large volumes of application requests by keeping frequently accessed data in the server memory, which can be written to and read from quickly.

### Service Tiers

Azure Cache for Redis is available in these tiers:

|  |  |
| --- | --- |
| Tier | Description |
| Basic | An OSS Redis cache running on a single VM. This tier has no service-level agreement (SLA) and is ideal for development/test and non-critical workloads. |
| Standard | An OSS Redis cache running on two VMs in a replicated configuration. |
| Premium | High-performance OSS Redis caches. This tier offers higher throughput, lower latency, better availability, and more features. Premium caches are deployed on more powerful VMs compared to the VMs for Basic or Standard caches. |
| Enterprise | High-performance caches powered by Redis Inc.'s Redis Enterprise software. This tier supports Redis modules including RediSearch, RedisBloom, and RedisTimeSeries. Also, it offers even higher availability than the Premium tier. |
| Enterprise Flash | Cost-effective large caches powered by Redis Inc.'s Redis Enterprise software. This tier extends Redis data storage to non-volatile memory, which is cheaper than DRAM, on a VM. It reduces the overall per-GB memory cost. |

### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Azure Monitor supported metrics for Azure Cache for Redis are available at:   
[Azure Monitor supported metrics by resource type - Azure Monitor | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported)

Alerts observed

* Cache Latency (preview)
* Cache Misses
* Cache Miss Rate
* Cache Read
* Cache Write
* Connected Clients
* Connections Created Per Second
* Connections Closed Per Second
* CPU
* Errors
* Evicted Keys
* Expired Keys

##### Backup and restore

Data backup and restore

In Standard and Basic tiers, all data is stored in the memory of the service, meaning that data loss is possible if a failure occurs where Cache nodes are down. For the Premium tier, Redis persistence can be configured using either Redis Database (RDB) or Append Only File (AOF):

* RDB persistence - When you use RDB persistence, Azure Cache for Redis persists a snapshot of your cache in a binary format. The snapshot is saved in an Azure Storage account. The configurable backup frequency determines how often to persist the snapshot. If a catastrophic event occurs that disables both the primary and replica cache, the cache is reconstructed using the most recent snapshot.
* AOF persistence - When you use AOF persistence, Azure Cache for Redis saves every write operation to a log. The log is saved at least once per second into an Azure Storage account. If a catastrophic event occurs that disables both the primary and replica cache, the cache is reconstructed using the stored write operations.

If data persistence is enabled, geo-replication can't be enabled for the same cache.

Service restore

Service restore will be from Infra as Code.

##### Azure SLA High Availability and Disaster Recovery inter-region

Azure Cache for Redis provides built-in redundancy by hosting each cache on two dedicated virtual machines stored in separate update and fault domains. This applies to Standard, Premium and Enterprise tiers. To avoid datacenter level failures, zone redundancy is also supported for the Premium and Enterprise tiers and can be configured during the deployment process. With zone redundancy enabled, the cache runs on VMs spread across multiple availability zones, which provides higher resilience and availability with this configuration enabled, the data transfer between Azure Availability Zones will be charged at Microsoft’s standard bandwidth rates.

Additionally, the service supports geo-replication for Premium tier only. Geo-replication is designed as a disaster-recovery solution and links together two Premium Azure Cache for Redis instances as well as creates a data replication relationship. The two instances can be hosted in the same region or in two different regions, with one instance acting as primary and the other as secondary. The primary handles read and write requests and propagate changes to the secondary.

Automatic failover across Azure regions isn't supported for geo-replicated caches, meaning that a manual failover has to be performed during a disaster recovery scenario. To avoid performance issues, Microsoft recommends bringing up the entire application stack in a coordinated manner in the backup region.

Various high availability options are available in the Standard, Premium, and Enterprise tiers:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Option | Description | Availability | Standard | Premium | Enterprise |
| Standard replication | Dual-node replicated configuration in a single data center with automatic failover | 99.9% | ✔ | ✔ | ✔ |
| Zone redundancy | Multi-node replicated configuration across Availability Zones, with automatic failover | 99.9% in Premium; 99.99% in Enterprise | - | ✔ | ✔ |
| Geo-replication | Linked cache instances in two regions, with user-controlled failover | Premium; Enterprise | - | Passive | Active |
| Import/Export | Point-in-time snapshot of data in cache. | 99.9% | - | ✔ | ✔ |
| Persistence | Periodic data saving to storage account. | 99.9% | - | ✔ | Preview |

### Charging model

|  |
| --- |
| Work Unit |
| Per Redis cache |

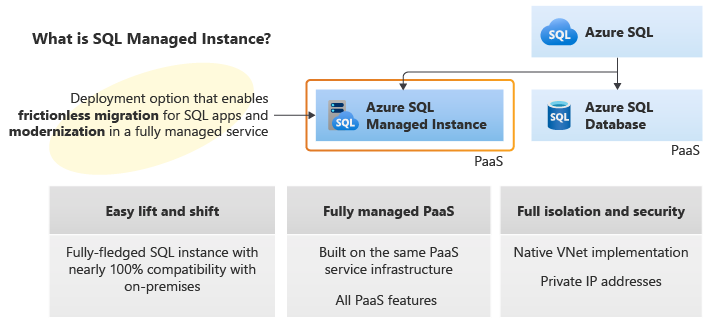
### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Reboot Redis services | I token (not available for enterprise tire) |
| Scaling the resource | 2 Token |
| Changing the service Model | 1 Token |
| Other changes | Estimation in tokens based on time spent |

## Azure SQL Managed Instance

### Description

Azure SQL Managed Instance is the intelligent, scalable cloud database service that combines the broadest SQL Server database engine compatibility with all the benefits of a fully managed and evergreen platform as a service. SQL Managed Instance has near 100% compatibility with the latest SQL Server (Enterprise Edition) database engine.



### Build to run service included in the OTC

##### Build service pre-requisite

* Refer to generic description.

##### Build to run service

* Refer to generic description.

### RUN services included in the MRC

##### Run service pre-requisite

* A referential file exists in the Git including the reference configuration of the service.
* This file can be executed with a CI/CD and the execution has been tested successfully.

##### KPI & alerts

Monitoring

Yes

KPI monitored

Azure Monitor supported metrics for Azure Database for MySQL are available at:   
[Azure Monitor supported metrics by resource type - Azure Monitor | Microsoft Docs](https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/metrics-supported)

Alerts observed

Built-in monitoring of basic MI telemetry (CPU, storage, IOPS).

##### Backup and restore

* Data backup and restore

Automated Backups: Full backups are taken every 7 days, differential 12 hours, and log backups every 5-10 min

* Service restore

Point-in-time Recover: It is possible to restore any database to an earlier point in time within its retention period.

##### Azure SLA High Availability and Disaster Recovery inter-region

The auto-failover groups feature allows you to manage the replication and failover of some or all databases on a logical server to another region.

### Charging model

|  |
| --- |
| Work Unit |
| Per Database Instance |

### Changes catalogue – in Tokens, per act

|  |  |
| --- | --- |
| Changes examples | Effort |
| Provision Managed Instance | 2 tokens |
| Instance property change (admin password, Azure AD login, Azure Hybrid Benefit flag | 1 token |
| Instance storage scaling up/down | 1 token |
| Instance compute (vCores) scaling up and down | 1 token |
| Other changes | Estimation in tokens based on time spent |

# End of the document