**CORE SERVICE DESIGN:**

**Azure App Service**

atabricks

|  |  |
| --- | --- |
| IT Owner Details | |
| **Department** | DTS |
| **Contact Name** | Dominic Panzera |
| **Address** | 375 Manningham Road, Doncaster, Victoria 3108 |

|  |  |
| --- | --- |
| Document Control | |
| Title | Ambulance Victoria – Azure App Service Core Service Design |
| **File Name** | Ambulance Victoria – Azure App Service Core Service Design v2.0.docx |
| **Version** | 2.0 |
| **Status** | Released |
| **Release Date** | 7/03/2024 |

|  |  |  |  |
| --- | --- | --- | --- |
| Preparation |  |  |  |
| **Prepared** | Daniela Nikolic |  |  |
| **Authorised** | Dileep Pradeep |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version Tracking | | | |  |
| Version | Remarks | Change Requested | Pages Affected | Release Date |
| 1.0 | Initial Release | N/A | All | 29/02/2024 |
| 2.0 | Updated | Minor Feedback | 7, 16, 17, 19, 20, 24-28 | 7/03/2024 |

Table of Contents

[1. Overview 5](#_Toc159931485)

[1.1 Purpose and Audience 5](#_Toc159931486)

[1.2 Scope and Key Deliverables 5](#_Toc159931487)

[1.3 Glossary and Definitions 6](#_Toc159931488)

[2. Executive Summary 7](#_Toc159931489)

[3. Resource Cost 8](#_Toc159931490)

[4. WAF and Security Control Alignment 9](#_Toc159931491)

[4.1 Reliability 9](#_Toc159931492)

[4.1.1 Overview 9](#_Toc159931493)

[4.1.2 Azure App Service Reliability Checklist 9](#_Toc159931494)

[4.2 Cost Optimisation 12](#_Toc159931495)

[4.2.1 Overview 12](#_Toc159931496)

[4.2.2 Azure App Service Cost Optimisation Checklist 12](#_Toc159931497)

[4.3 Operational Excellence 13](#_Toc159931498)

[4.3.1 Overview 13](#_Toc159931499)

[4.3.2 Azure App Service Operational Excellence Checklist 13](#_Toc159931500)

[4.4 Performance Efficiency 14](#_Toc159931501)

[4.4.1 Overview 14](#_Toc159931502)

[4.4.2 Azure App Service Performance Efficiency Checklist 14](#_Toc159931503)

[4.5 Security 15](#_Toc159931504)

[4.5.1 Overview 15](#_Toc159931505)

[5. Architecture Summary 16](#_Toc159931506)

[5.1 Resource Overview 16](#_Toc159931507)

[5.1.1 App Service Plan 16](#_Toc159931508)

[5.1.2 App Service Environment 16](#_Toc159931509)

[5.2 RBAC 16](#_Toc159931510)

[5.3 Design Decisions and Justifications 17](#_Toc159931511)

[5.3.1 Tier 17](#_Toc159931512)

[5.3.2 Redundancy 17](#_Toc159931513)

[5.3.3 ARR Affinity 17](#_Toc159931514)

[5.3.4 Always On 18](#_Toc159931515)

[5.3.5 Backup and Restore 18](#_Toc159931516)

[5.3.6 High Availability 18](#_Toc159931517)

[5.3.7 Networking Requirements 19](#_Toc159931518)

[5.3.8 Network Connectivity 19](#_Toc159931519)

[5.3.9 Network Protection 19](#_Toc159931520)

[5.3.10 Upgrade Preference 20](#_Toc159931521)

[5.3.11 Scaling 20](#_Toc159931522)

[5.3.12 Health Check 21](#_Toc159931523)

[5.3.13 Logging and Insights 21](#_Toc159931524)

[5.3.14 Threat Detection 21](#_Toc159931525)

[5.3.15 Managed Identity 21](#_Toc159931526)

[5.3.16 Credentials, Secrets, and Certificates 22](#_Toc159931527)

[5.3.17 Encryption 22](#_Toc159931528)

[5.3.18 Secure Configurations 22](#_Toc159931529)

[5.3.19 DevOps Deployments 22](#_Toc159931530)

[6. Azure Policies 23](#_Toc159931531)

[7. Configuration Templates 24](#_Toc159931532)

[7.1 Primary Region Platinum/Gold/Silver – Premium Tier Settings 24](#_Toc159931533)

[7.2 Secondary Region Platinum/Gold/Silver – Premium Tier Settings 25](#_Toc159931534)

[7.3 Primary Region Platinum/Gold/Silver – Isolated Tier Settings 26](#_Toc159931535)

[7.4 Primary Region Bronze – Basic Tier Settings 27](#_Toc159931536)

[8. Acceptance 29](#_Toc159931537)

# Overview

This document covers the baseline design for the Azure App Service core service. The intention of this document is to define the overall resource design in isolation from a specific application. It is aimed to highlight the general process and requirements for building a Azure App Service in a repeatable fashion with consistent configurations. Design decisions and justifications have been included in the Architecture section, and this document can be used as a reference for new builds that require a Azure App Service.

This design caters to a Level 2 design which covers both Microsoft’s WAF (Well Architected Framework)[[1]](#footnote-2) and the Department of Health Control list.

Any deviations required to the standards defined in this document will require separate exemption and approval from the Cloud Governance Forum if they are required for any reason for a specific build.

## Purpose and Audience

This document will outline the standard design and configuration of this Azure service in Ambulance Victoria’s Azure tenancy as a baseline for any application infrastructure deployments.

This design is intended to:

* Meet Microsoft WAF standards.
* Meet the controls stipulated by the Department of Health.
* Define the baseline required for the deployment of the resource.

The audience for this document is those involved in the planning, designing, and implementing of the Application/Data infrastructure. This includes:

* + Ambulance Victoria IT staff

It is assumed that the reader knows and is familiar with Azure Cloud concepts and related topics.

## Scope and Key Deliverables

The scope of this core service design is to define the baseline deployment requirements and standards for the Azure App Service core service.

The key deliverables for this are:

* This design to outline the service definition Level 2 baseline standards.
* A technical configuration document that defines the deployment of this resource for each of the Service Tiers, or for any other logical standard such as size
* IaC templates for repeatable deployment of this core service

## Glossary and Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **ARR** | Affinity Request Routing |
| **AV** | Ambulance Victoria |
| **WAF** | Well Architected Framework |
| **CAF** | Cloud Adoption Framework |
| **Level 1** | Refers to a resource that has been designed to a CAF standard |
| **Level 2** | Refers to a resource that has been designed to a WAF standard with Department of Health controls overlayed |
| **AZ 2** | Refers to Ambulance Victoria’s legacy Azure Landing Zone still in use in some regards |
| **AZ 3** | Refers to Ambulance Victoria’s current Azure Landing Zone, also referred to as the Enterprise landing zone. This is the target state for migrations. |
| **SLA** | Service Level Agreement as defined by Microsoft |
| **DH** | Department of Health |
| **IaC** | Infrastructure as Code |
| **NSG** | Network Security Groups |

Table 1: Glossary and definitions

# Executive Summary

This design covers the baseline standards for the Azure App Service Core Service. This service has been assessed against the five pillars of WAF as well as the Department of Health Security Controls.

This section contains a summary of the major design decisions that have been made for defining the baseline of this resource as an outcome of the WAF and Security analysis detailed throughout this document.

Of the five WAF Pillars, it was found that Reliability, Operational Excellence, Cost Optimisation and Security were relevant.

For this service the main baseline configurations include:

* Public access will be disabled by default, and connectivity will take place using Private Endpoints.
* Defender for Web Apps will be enabled.
* ARR (Application Request Routing) will be disabled by default.
* Always On will be enabled by default.
* Remote debugging will be disabled.
* Azure DevOps will be used for application deployments.
* Upgrade preferences will be set to “None” so updates are pushed manually to avoid outages. A notification will be sent when the upgrade is available.

There are some notable differences across the service tier configurations for this service:

* The Isolated plan will be used for Production, and the Basic plan for Non-Production.
* Continuous deployment will not be enabled for Production slots.
* Non-production will not use scaling.

# Resource Cost

The following are the pricing constructs for the Basic and Premium plans. There is also a Free and Shared tier but this has no SLA so is not recommended for general use[[2]](#footnote-3):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Basic Service Plan | Cores | RAM | Storage | Pay as you go |
| **B1** | 1 | 1.75 GB | 10 GB | **$0.143**/hour |
| **B2** | 2 | 3.50 GB | 10 GB | **$0.286**/hour |
| **B3** | 4 | 7 GB | 10 GB | **$0.571**/hour |

Table 2: Pricing construct for the Basic tier

For the Premium plan, the current version is Premium v3:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Premium v3 Service Plan | Cores | RAM | Storage | Pay as you go |
| P0v3 | 1 | 4 GB | 250 GB | **$0.327**/hour |
| P1v3 | 2 | 8 GB | 250 GB | **$0.513**/hour |
| P1mv3 | 2 | 16 GB | 250 GB | **$0.567**/hour |
| P2v3 | 4 | 16 GB | 250 GB | **$1.026**/hour |
| P2mv3 | 4 | 32 GB | 250 GB | **$1.134**/hour |
| P3v3 | 8 | 32 GB | 250 GB | **$2.051**/hour |
| P3mv3 | 8 | 64 GB | 250 GB | **$2.267**/hour |
| P4mv3 | 16 | 128 GB | 250 GB | **$4.534**/hour |
| P5mv3 | 32 | 256 GB | 250 GB | **$9.068**/hour |

Table 3: Pricing construct for the Premium tier

# WAF and Security Control Alignment

The following are the five pillars of the Microsoft Well Architected Framework:

* [Reliability](https://learn.microsoft.com/en-us/azure/well-architected/#reliability)
* [Cost optimization](https://learn.microsoft.com/en-us/azure/well-architected/#cost-optimization)
* [Operational excellence](https://learn.microsoft.com/en-us/azure/well-architected/#operational-excellence)
* [Performance efficiency](https://learn.microsoft.com/en-us/azure/well-architected/#performance-efficiency)
* [Security](https://learn.microsoft.com/en-us/azure/well-architected/#security)

For this design, the security section will also cover the Department of Health Controls in addition with any Microsoft Security Best Practices. Each of these sections will detail relevant controls or baseline requirements for this core service that will be put in place.

## Reliability

### Overview

The term reliability refers to the availability of the system and its ability to recover from failure[[3]](#footnote-4). Resiliency strategies must be built into each element of the architecture. The pillars of reliability include:

* Design for business requirements
* Design for failure
* Observe application health
* Drive Automation

### Azure App Service Reliability Checklist

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Checklist Item | Applicable to AV | Built Into Design | Enforcement Option | Applicability |
| **R1** | Consider disabling ARR Affinity for your App Service. | Yes | Yes | IaC | At deployment |
| **R2** | Use a different store for session state. | Yes | No | Governance | Operational – during application design and deployment |
| **R3** | Enable Always On to ensure Web Jobs run reliably. | Yes | Yes | IaC | At deployment |
| **R4** | Access the on-prem database using private connections like Azure VPN or Express Route. | Yes | Yes | Already in place | Already in place |
| **R5** | Set up backup and restore. | Yes | Yes | N/A | Default enabled on deployment |
| **R6** | Understand IP Address deprecation impact. | Yes | No | Governance | Operational |
| **R7** | Deploy in highly available configuration across Availability Zones. | No | No | Not Available in Australia Southeast | N/A |
| **R8** | Configure ASE Network correctly. | Yes | Yes | IaC | At deployment |
| **R9** | Configure Upgrade preference if multiple environments are used. | Yes | Yes | IaC | At deployment |
| **R10** | Scale out the ASE cluster. | Yes | Yes | IaC | At deployment |
| **R11** | Use Deployment slots for resilient code deployments. | Yes | Yes | No | Operational – during application design and deployment |
| **R12** | Avoid unnecessary worker restarts. | Yes | Yes | No | Operational – during application design and deployment |
| **R13** | Run From Package to avoid deployment conflicts | Yes | Yes | No | Operational – during application design and deployment |
| **R14** | Use Basic or higher plans with two or more worker instances for high availability. | Yes | Yes | IaC | At deployment |
| **R15** | Evaluate the use of TCP and SNAT ports. | Yes | No | Governance | Operational – review monthly |
| **R16** | Enable Health check to identify non-responsive workers. | Yes | Yes | IaC | At deployment |
| **R17** | Enable Autoscale to ensure adequate resources are available to service requests. | Yes | Yes | IaC | At deployment |
| **R18** | Enable Local\_Cache to reduce dependencies on cluster file servers. | Yes | Yes | Governance | Operational – during design and deployment |
| **R19** | Enable Diagnostic Logging to provide insight into application behaviour. | Yes | Yes | IaC – Application Insights IaC | At deployment |
| **R20** | Enable Application Insights alerts to make you aware of fault conditions. | Yes | Yes | IaC – Application Insights IaC | At deployment |
| **R21** | Review Azure App Service diagnostics to ensure common problems are addressed. | Yes | Yes | Governance | Operational – review monthly |
| **R22** | Evaluate per-app scaling for high density hosting on Azure App Service. | Yes | Yes | Governance | Operational – during design and deployment |

Table 4: WAF Reliability checklist summary

## Cost Optimisation

### Overview

The cost optimisation pillar is structured to support creating cost-effective workloads in the cloud[[4]](#footnote-5). It looks at removal of unnecessary spend and improving operational efficiency. The principles of cost optimisation revolve around:

* Choosing the correct resources
* Setting up budgets and maintaining cost constraints
* Dynamically allocate and deallocate resources
* Optimising workloads whilst aiming for scalable costs
* Continuously monitoring and cost managing

### Azure App Service Cost Optimisation Checklist

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Checklist Item | Applicable to AV | Built Into Template | Enforcement Option | Applicability |
| **CO1** | Ensure the ASE subnet is appropriately sized. | Yes | Yes | IaC | At deployment |
| **CO2** | Use App Service Premium v3 plan over the Premium v2 plan | Yes | Yes | IaC | At deployment |
| **CO3** | Use a scale-out and scale-in rule combination | Yes | Yes | IaC | At deployment |
| **CO4** | Understand the behaviour of multiple scaling rules in a profile. | Yes | No | Governance | Operational – during application deployment |
| **CO5** | Consider Basic or Free tier for non-production usage. | Yes | Yes | IaC | At deployment |

Table 5: WAF Cost Optimisation checklist summary

## Operational Excellence

### Overview

Operational Excellence aims to ensure that once the architecture is built, the ongoing operations are flawless. This includes repeatable and reliable deployments, automating to eliminate human error. To do this the following must be considered:

* Optimise the build and release process (including CI/CD and IaC)
* Understand Operational Health
* Test recovery and failure
* Focus on continuous improvement
* Use loosely coupled architecture

### Azure App Service Operational Excellence Checklist

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Checklist Item | Applicable to AV | Built Into Template | Enforcement Option | Applicability |
| **OE1** | Create a deployment plan because redeploying the app service can reset the scaled units. | Yes | No | Governance | Operational – during deployment |
| **OE2** | Review the App Service Advisor recommendations. | Yes | No | Governance | Operational – review monthly |
| **OE3** | Ensure you configure the App Service Environments (ASE) Network correctly. | Yes | Yes | IaC | At deployment |

Table 6: WAF Operational Excellence checklist summary

## Performance Efficiency

### Overview

Performance Efficiency refers to the ability of your systems and applications to meet user demands without breaking or creating a negative user experience[[5]](#footnote-6). This covers capacity and scalability:

* Design for horizontal scaling
* Run stress and performance tests
* Continuously monitor performances, particularly in Production systems

### Azure App Service Performance Efficiency Checklist

There is no guidance for Azure App Service under the Performance Efficiency pillar.

## Security

### Overview

Security refers to the ability of the environment to resist and manage threats.

This section covers both Microsoft Best Practices as well as relevant security controls provided by the Department of Health. With respect to the Microsoft WAF, Security is underpinned by the following[[6]](#footnote-7):

* Plan resources and how to harden them
* Automate and use least privilege
* Classify and encrypt data
* Monitor system security, plan incident response
* Identify and protect endpoints
* Protect against code-level vulnerabilities
* Model and test against potential threats

In addition to the Microsoft controls, the Department of Health has mandated security posture to Ambulance Victoria. Note there may be duplication between the Microsoft Security Best Practices and the Department of Health controls.

The following Microsoft Security Benchmark controls are applicable:

* NS-1: Establish network segmentation boundaries
* NS-2: Secure cloud services with network controls
* NS-5: Deploy DDOS protection
* NS-6: Deploy web application firewall
* IM-3: Manage application identities securely and automatically
* IM-8: Restrict the exposure of credential and secrets
* DP-3: Encrypt sensitive data in transit
* DP-4: Enable data at rest encryption by default
* DP-7: Use a secure certificate management process
* LT-1: Enable threat detection capabilities
* LT-4: Enable logging for security investigation
* BR-1: Ensure regular automated backups
* PV-2: Audit and enforce secure configurations
* DS-6: Enforce security of workload throughout DevOps lifecycle

# Architecture Summary

## Resource Overview

App Services in Azure are comprised of several components:

* **The App Service –** which can be a Web App, a Static Web App, a Web App + Database, or WordPress on App Service
* **The App Service plan** – which is the underlying compute package supporting the App Service
* **App Service Environment** – a completely isolated network that is available when using the Isolated tier of App Service Plan.

The Azure App service is a hosting service for web applications, REST APIs, and mobile backends. It has compatibility with major developer languages including .NET, Java, and Python among others. It is a PaaS service, so the underlying patching and OS of the run stacks are managed by Microsoft[[7]](#footnote-8).

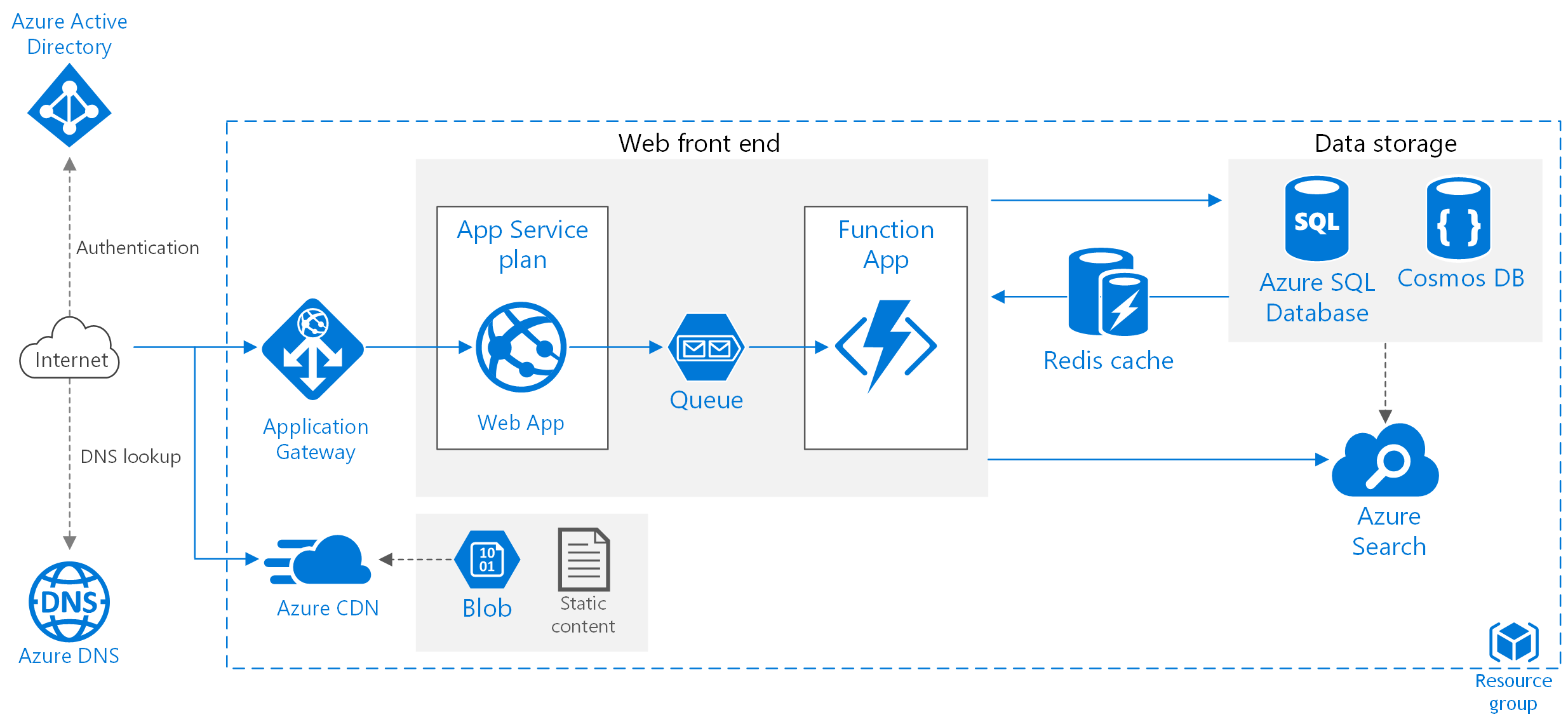


Figure : Example App Service flow

### App Service Plan

The App Service Plan is the underlying compute package that supports the Web App Itself, though it can also be used to support Function Apps[[8]](#footnote-9). The Azure App Service plans are available in several tiers[[9]](#footnote-10):

* **Shared compute:** the **Free** and **Shared** tiers run your app on a shared VM with other App Service apps which may be the same as other customers.
* **Dedicated compute:** The **Basic, Standard, Premium, PremiumV2, and PremiumV3** tiers run apps on dedicated Azure VMs. Only apps in the same App Service plan share the same compute resources. Higher tiers offer more VM instances for scale-out.
* **Isolated:** The **Isolated**and**IsolatedV2** tiers run dedicated Azure VMs on dedicated Azure Virtual Networks. It provides network isolation on top of compute isolation to your apps. It provides the maximum scale-out capabilities.

### App Service Environment

The App Service Environment is a feature of Azure App Service that provides isolated and dedicated environments for applications to run. It is compatible to host both Linux and Windows web apps.

The App Service Environment is a completely isolated instance, so must be used with **Isolated or IsolatedV2** tiers, and it configures Virtual Network integration inherently. For the Premium tier, network integration must be configured as part of the deployment.

## RBAC

There are no specific RBAC roles applicable to this service.

## Design Decisions and Justifications

This section covers the design decisions and justifications that reflect the findings of the WAF and Security alignment. This will form the baseline requirements for the Azure App Service core service and will be captured in the accompanying Configuration Template with a set of pre-approved deployment settings for this resource. Any changes, modifications or removals to the pre-approved deployments must have specific approval from the Cloud Governance Forum prior to deployment.

### Tier

**Design Reference:** Table 4 – [R14](#_Azure_App_Service), Table 5 – [CO2, CO5](#_Azure_App_Service_2)

**Design Decision**: The Isolated tier will be used for Production. Basic can be used for Non-Production workloads. It is not recommended to use the Free or Shared plans unless for very specific Dev/Test requirements.

**Design Justification**: The Isolated tier offers the most secure form of deployment for Azure App Service, and is reliability and scalable. The latest version should always be used where possible, and will be deployed with an App Service Environment.

The Basic tier can be used for Non-Production or simple applications as it is unable to manage high traffic requirements, and cannot perform auto-scale or traffic management features.

### Redundancy

**Design Reference:** N/A

**Design Decision:** Redundancy cannot be enabled in the service natively as Availability Zones are not available in Australia Southeast. Geo-redundancy must be architected across the two regions for each application.

**Design Justification:** Azure Availability Zones are not currently available in Australia Southeast which is being used as the primary region for Azure. As such Zonal redundancy is not available. Additionally, there is no geo-redundancy that can be enabled within the Azure App Service, so it must be architected for the application during the application design. This would mean instances of Azure App Service in both regions with a failover mechanism to be determined during the application design.

### ARR Affinity

**Design Reference:** Table 4 – [R1](#_Azure_App_Service)

**Design Decision:** ARR (Application Request Routing) will be disabled by default.

**Design Justification:** ARR has some benefits, but it can cause issues depending on how the users interact with the application. Some users may not close browsers and remain connected to the application for extended periods, which means that the user may unnecessarily remain attached to the affinity cookie provided to them. ARR will be disabled as the default which will result in better load balancing to the application. It can be enabled if it is required by a specific application architecture.

### Always On

**Design Reference:** Table 4 - [R3](#_Azure_App_Service)

**Design Decision:** Always On will be enabled by default.

**Design Justification:** Always On keeps the application loaded even when no traffic is running. This essentially keeps the site responsive, whereas with Always On disabled it will have to reload fully and can lead to performance issues with the application.

### Backup and Restore

**Design Reference:** Table 4 - [R5](#_Azure_App_Service), Microsoft Security Benchmark [BR-1](#_Overview)

**Design Decision:** The native automatic backup service for Azure App Service will be used.

**Design Justification:** Azure App service has an inbuilt capability to perform backups. Automatic backups will take place by default if the pricing tier is supported (which it is for Basic, Standard, and Premium tiers). The automated backups take place hourly, and have the following retention pattern inbuilt:

* 30 days total retention, not configurable.
  + Days 1-3: hourly backups retained.
  + Days 4-14: every 3 hourly backup retained.
  + Days 15-30: every 6 hourly backup retained.

If additional or specific scheduling is required, it can be customised, but will require a separate storage account. It is recommended to use the automated backup service unless specifically required by the application. The backups will natively backup the file contents and application configurations. If there are any linked databases that require backup, they will need to be backed up separately or through the custom backup settings.

### High Availability

**Design Reference:** Table 4 **–** [R7](#_Azure_App_Service)

**Design Decision:** Availability zones cannot be used in Australia Southeast as they are not supported yet. They should be reviewed and deployed should they become available in future.

**Design Justification:** Australia Southeast does not support Availability Zones at present so this feature cannot be leveraged in the Primary region. When it does become available it should be reviewed and considered for Azure App Service deployments.

### Networking Requirements

**Design Reference:** Table 4 - [R8](#_Azure_App_Service), Table 5 – [CO1](#_Azure_App_Service_2), Table 6 – [OE3](#_Azure_App_Service_1), Microsoft Security Benchmark [NS-1](#_Overview)

**Design Decision:** When integrating with a virtual network,there is a minimum subnet requirement of /27. A /24 is recommended for Production workloads. Ensure that NSGs associated with this network meets the minimal requirements for its operation. Virtual Network integration is configured by default when using the App Service Environment with the Isolated tier.

**Design Justification:** Though a minimum of /27 is required, providing a /24 where possible for App Service environments allows sufficient room for scaling. During platform upgrades, free IP addresses are also required to ensure that upgrades do not impact outbound traffic. A subnet of this size ensures that it is future proof for growth.

To function properly, the subnet also requires NSGs rules to be configured. The minimum is to allow Inbound traffic from AzureLoadBalancer on port 80 to the subnet. There will also be other rules required to allow application traffic. NSGs by default have an implicit deny rule so explicit allow rules are required to be added for application functionality.

### Network Connectivity

**Design Reference:** Microsoft Security Benchmark [NS-2](#_Overview)

**Design Decision:** Private endpoints will be used where possible with public connectivity disabled by default. If public connectivity is required, traffic should be filtered through a Firewall.

**Design Justification:** Private endpoints are the most secure form of connectivity. If they cannot be used for a specific application, the any public connectivity should be filtered through a Web Application Firewall.

### Network Protection

**Design Reference:** Microsoft Security Benchmark [NS-5, NS-6](#_Overview)

**Design Decision:** Application traffic will be filtered through a Firewall. DDoS Protection Basic is enabled by default everywhere on the platform, and the Standard DDoS Protection should be enabled anywhere with public access enabled on the platform.

**Design Justification:** All traffic should be filtered through a Firewall. The Azure Firewall will most likely handle any traffic that enters through private connections, including private endpoint connectivity. If the application is designed so that public traffic is required, this traffic should be filtered through a Web Application Firewall. At this stage any external traffic entering the platform should come through the Perimeter Services entry point which is fronted by an Azure Application Gateway.

### Upgrade Preference

**Design Reference:** Table 4 – [R9](#_Azure_App_Service)

**Design Decision:** Automatic upgrades will not be enabled. A notification will be sent when the upgrade is available through a Service Health alert.

**Design Justification:** Setting automatic upgrades can cause the application to break unintentionally. Non-Production environments should be upgraded first so that any issues with the upgrade can be identified through Non-Production. Once the upgrade is available there are 15 days before it is pushed onto all App Services. As such a Service Health alert will be set up for planned maintenance, so that an outage can be planned for the manual upgrades.

### Scaling

**Design Reference:** Table 4 – [R10, R17](#_Azure_App_Service) ,Table 5 – [CO3](#_Azure_App_Service_2)

**Design Decision:** Rules Based scale out and scale in should be configured for the app service.

**Design Justification:** Automatic scale-out is currently in public preview so is not suitable for use in Production but should be considered for review when it becomes Generally Available.

Until it does become available Rule Based scaling can be used to automatically scale out an application when a certain metric is met, CPU typically, but it can be set to any metric required by an application design.

If configuring Scale Out, a Scale In rule should also be applied to ensure that the scale units shrink when the condition triggering the scale out is no longer being met.

**Design Details:** More specific scale-out and scale-in rules should be configured during the application design and as a part of operational process improvements. As a starting point the following are recommended to be implemented as a baseline:

Scale out if:

* Greater than 80% CPU usage
* Duration - 10 minutes
* Increase count by – 1
* Cool down – 5 minutes
* Maximum – 3 instances

Scale in if:

* Less than 80% CPU usage
* Duration – 10 minutes
* Cool down – 5 minutes

### Health Check

**Design Reference:** Table 4 – [R16](#_Azure_App_Service)

**Design Decision:** Health Checks will be enabled.

**Design Justification:** Health Checks can be used to maintain application availability by identifying unhealthy instances and rerouting requests to healthy instances.

### Logging and Insights

**Design Reference:** Table 4 – [R19, R20](#_Azure_App_Service) Microsoft Security Benchmark [LT-4](#_Overview)

**Design Decision:** Diagnostics logs will be enabled with allLogs and AllMetrics sent to the central log analytics workspace for that region. Application Insights will also be leveraged for app performance management.

**Design Justification:** Diagnostic logs allow you to identify performance issues at the infrastructure layer and can be helpful in resolving configuration issues. Application Insights will support on the Application layer with identifying performance issues and providing remediation recommendations.

### Threat Detection

**Design Reference:** Microsoft Security Benchmark [LT-1](#_Overview)

**Design Decision:** Defender for App Service will be enabled.

**Design Justification:** Defender for App service is a natively integrated defence tool for Azure App Service and associated APIs. It will assess the resources covered by the App Service plan and create security recommendations to harden the environment. It will also prevent common threats from impacting the service.

### Managed Identity

**Design Reference:** Microsoft Security Benchmark [IM-3](#_Overview)

**Design Decision:** Managed Identities will be used instead of Service Principals for authentication and authorisation.

**Design Justification:** Managed Identities provide a more secure method for authentication against Azure services as Microsoft manage and rotate the keys, removing the need to remember and store credentials, thus reducing the risk of compromise.

### Credentials, Secrets, and Certificates

**Design Reference:** Microsoft Security Benchmark [IM-8, DP-7](#_Overview)

**Design Decision:** Credentials, Secrets, and Certificates related to the Azure App Service should be stored in Azure Key Vault and rotated in alignment with Key Vault best practices.

**Design Justification:** Azure Key Vault is a secure method of storing certificates instead of deploying them on the application directly. This also centralises the storage of these elements and the App Service can simply reference the latest versions updated in the Key Vault.

### Encryption

**Design Reference:** Microsoft Security Benchmark [DP-3, DP-4](#_Overview)

**Design Decision:** The minimum TLS will be set to 1.2. All HTTP requests will be directed to HTTPS. Azure Policies will be deployed to audit these settings on the service. Encryption at rest is enabled by default.

**Design Justification:** Setting the minimum TLS version to 1.2 and ensuring the use of HTTPS means that all traffic in transit will be encrypted securely. To ensure that App Services are deployed securely, Azure Policies will be deployed to audit, or enforce, correct configurations. These are defined in Section 6.

### Secure Configurations

**Design Reference:** Microsoft Security Benchmark [PV-2](#_Overview)

**Design Decision:** Additional Azure Policies will be applied to audit or disable insecure configurations on Azure App Service. Remote debugging will be turned off as a default.

**Design Justification:** Ensuring that applications have been deployed securely can be audited, or enforced, with Azure Policies that are defined in Section 6. Remote debugging should be turned off as this opens up inbound ports.

### DevOps Deployments

**Design Reference:** Microsoft Security Benchmark [DS-6](#_Overview)

**Design Decision:** Application code will be deployed using Azure DevOps pipelines. Continuous deployment will not be enabled on the Production slot.

**Design Justification:** To maintain source code control and versioning, Azure DevOps will be used for application deployments. This will additionally allow governance elements such as approval workflows to be built into the process to meet this control. Additionally, Continuous Deployment should not be enabled on a Production slot as this can lead to accidental deployments into the Production environment. The main branch should be deployed into a nonproduction slot and then swapped into production when ready.

# Azure Policies

The following policies will be applied for the Azure App Service. These are all built-in policies that can be enabled to audit, then enforce when ready:

|  |  |
| --- | --- |
| Policy Name | Description |
| App Service apps should have 'Client Certificates (Incoming client certificates)' enabled | Client certificates allow for the app to request a certificate for incoming requests. Only clients that have a valid certificate will be able to reach the app. |
| App Service apps should have remote debugging turned off | Remote debugging requires inbound ports to be opened on an App Service app. Remote debugging should be turned off. |
| App Service apps should not have CORS configured to allow every resource to access your apps | Cross-Origin Resource Sharing (CORS) should not allow all domains to access your app. Allow only required domains to interact with your app. |
| App Service apps should only be accessible over HTTPS | Use of HTTPS ensures server/service authentication and protects data in transit from network layer eavesdropping attacks. |
| App Service apps should require FTPS only | Enable FTPS enforcement for enhanced security. |
| App Service apps should use the latest TLS version | Periodically, newer versions are released for TLS either due to security flaws, include additional functionality, and enhance speed. Upgrade to the latest TLS version for App Service apps to take advantage of security fixes, if any, and/or new functionalities of the latest version. |

Table 8: Additional Azure Policies to be deployed for this service

# Configuration Templates

## Primary Region Platinum/Gold/Silver – Isolated Tier Settings

|  |  |
| --- | --- |
| Configuration Item | Configuration Settings |
| App Name | [app\_url].azurewebsites.net |
| Azure App Plan Name | asp-prd-ause-[appname]-01 |
| Subscription | AV ALZ [Application Name] |
| Publish | Code/Docker Container/Static Web App |
| Runtime Stack | If using Code to publish - Select stack based on application requirements – Linux not supported |
| Operating System | If using Docker Container to publish – select Linux or Windows |
| Tier | Isolated V2 |
| ASE Environment Name | [ase\_url].appserviceenvironment.net |
| Virtual IP Type | Internal/External |
| Continuous Deployment | Disable |
| Enable Application Insights | Yes |
| App Insights Name | api-prd-ause-[appname]-01 |
| ***Hosting Settings*** |  |
| Physical Hardware Isolation | Enabled/Disabled |
| ***App Service Environment Settings*** |  |
| Virtual Network | vnet-prd-ause-[appname]-01 |
| Subnet | snet-prd-ause-[appname]-[workload]-01 |
| DNS | Manual |
| Inbound IP Address | Automatic |
| **Scaling Rules** |  |
| Scale out | * Greater than 80% CPU usage * Duration - 10 minutes * Increase count by – 1 * Cool down – 5 minutes * Maximum – 3 instances |
| Scale in | * Less than 80% CPU usage * Duration – 10 minutes * Cool down – 5 minutes |

## Secondary Region – Isolated Tier Settings

|  |  |
| --- | --- |
| Configuration Item | Configuration Settings |
| App Name | [app\_url].azurewebsites.net |
| Azure App Plan Name | asp-pdr-auea-[appname]-01 |
| Subscription | AV ALZ [Application Name] |
| Publish | Code/Docker Container/Static Web App |
| Runtime Stack | If using Code to publish - Select stack based on application requirements – Linux not supported |
| Operating System | If using Docker Container to publish – select Linux or Windows |
| Tier | Isolated V2 |
| ASE Environment Name | [ase\_url].appserviceenvironment.net |
| Virtual IP Type | Internal/External |
| Continuous Deployment | Disable |
| Enable Application Insights | Yes |
| App Insights Name | api-dr-auea-[appname]-01 |
| ***Hosting Settings*** |  |
| Physical Hardware Isolation | Enabled/Disabled |
| ***App Service Environment Settings*** |  |
| Virtual Network | vnet-dr-auea-[appname]-01 |
| Subnet | snet-dr-auea-[appname]-[workload]-01 |
| DNS | Manual |
| Inbound IP Address | Automatic |
| **Scaling Rules** |  |
| Scale out | * Greater than 80% CPU usage * Duration - 10 minutes * Increase count by – 1 * Cool down – 5 minutes * Maximum – 3 instances |
| Scale in | * Less than 80% CPU usage * Duration – 10 minutes * Cool down – 5 minutes |

## Primary Region Bronze – Basic Tier Settings

|  |  |
| --- | --- |
| Configuration Item | Configuration Settings |
| App Name | [app\_url].azurewebsites.net |
| Azure App Plan Name | asp-[env]-ause-[appname]-01 |
| Subscription | AV ALZ [Application Name] |
| Publish | Code/Docker Container/Static Web App |
| Runtime Stack | If using Code to publish - Select stack based on application requirements – Linux not supported |
| Operating System | If using Docker Container to publish – select Linux or Windows |
| Tier | Basic B1 |
| Database (optional) | Create if required by application |
| Database engine | SQLAzure(recommended)/PostgreSQL Flexible/MySQL Flexible/Cosmos DB API for MongoDB |
| Continuous Deployment | Disable |
| Enabled Public Access | Off |
| Enable Network Injection | On |
| Virtual Network | vnet-[env]-ause-[appname]-01 |
| Enable Private Endpoints | On |
| Outbound access | On/Off |
| Enable Application Insights | Yes |
| App Insights Name | api-prd-ause-[appname]-01 |

# Acceptance

Signature of this page by appropriately delegated representatives of ​Ambulance Victoria​ signifies acceptance of this design document.

Logicalis will commence build and implementation work once it receives a signed copy of this design document.

Signature of this page by appropriately delegated representatives of Ambulance Victoria signifies acceptance of this design document.

Logicalis will commence build and implementation work once it receives a signed copy of this design document.

|  |  |
| --- | --- |
| Project | Core Services |
| Document Version | 1.0 |

**Signed on behalf of Ambulance Victoria**

|  |  |
| --- | --- |
| Name | Dan Howarth |
| Position |  |
| Signature |  |
| Date signed |  |

**Signed on behalf of Logicalis Australia**

|  |  |
| --- | --- |
| Name | Daniela Nikolic |
| Position | Senior Cloud Engineer |
| Signature |  |
| Date signed |  |

1. https://learn.microsoft.com/en-us/azure/well-architected/ [↑](#footnote-ref-2)
2. https://azure.microsoft.com/en-us/pricing/details/app-service/windows/ [↑](#footnote-ref-3)
3. https://learn.microsoft.com/en-us/azure/well-architected/resiliency/overview [↑](#footnote-ref-4)
4. https://learn.microsoft.com/en-us/azure/well-architected/cost/overview [↑](#footnote-ref-5)
5. https://learn.microsoft.com/en-us/azure/well-architected/scalability/overview [↑](#footnote-ref-6)
6. https://learn.microsoft.com/en-us/azure/well-architected/security/security-principles [↑](#footnote-ref-7)
7. https://learn.microsoft.com/en-us/azure/app-service/overview [↑](#footnote-ref-8)
8. https://learn.microsoft.com/en-us/azure/app-service/overview-hosting-plans [↑](#footnote-ref-9)
9. https://learn.microsoft.com/en-us/azure/app-service/overview-hosting-plans [↑](#footnote-ref-10)