# APIM Best Practices

The Azure API Management landing zone accelerator offers an architectural approach and reference implementation for scalable API Management infrastructure. It aligns with the Cloud Adoption Framework's Azure landing zones and enterprise-scale design principles.

* It can be used a design guide for greenfield implementation and as an assessment guide for brownfield.
* The Infrastructure as a Code(IaaC) template can be customized to match naming conventions, utilize existing resources, and integrate with various backends.

## Discussed Topics and Reference Links

### Networking

* Use private endpoints when internal consumers want to access APIM endpoint(s). It’s because network traffic between a consumer on your private network and APIM traverses over the VNet and a private link on the Microsoft backbone network, eliminating exposure from the public internet. Ref. - [Set up inbound private endpoint for Azure API Management | Microsoft Learn](https://learn.microsoft.com/en-us/azure/api-management/private-endpoint)
* Deploy a Web Application Firewall (WAF) in front of API Management to protect against common web application exploits and vulnerabilities. Azure Application Gateway includes a built-in Web Application Firewall (WAF) feature. The WAF helps protect web applications from common web-based attacks and vulnerabilities. It provides an additional layer of security by inspecting incoming traffic and blocking malicious requests. The WAF feature in Azure Application Gateway offers customizable rule sets and provides protection against OWASP (Open Web Application Security Project) top 10 vulnerabilities, such as SQL injection, cross-site scripting (XSS), and more. Ref. <https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-integrate-internal-vnet-appgateway>
* In case of multi-region deployment, it’s recommended to use Azure Front door as the load balancing solution. Ref. [Azure Front Door](https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview).
* VNet peering supports high performance in a region but has a scalability limit of 500 networks. If you require more workloads to be connected, use a [hub spoke](https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/hybrid-networking/hub-spoke) architecture or [Private Endpoint](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview).

### Security

* Consider how you want to secure your frontend APIs beyond using [subscription keys](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-create-subscriptions). [OAuth 2.0](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-protect-backend-with-aad), OpenID Connect, and [mutual TLS](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-mutual-certificates-for-clients) are common options with built-in support.
* Use custom roles based on API Management [RBAC operations](https://learn.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftapimanagement) to set fine-grained access to API Management entities. Examples: API developers, backup operators, DevOps automation, etc.
* Manage constant string values and secrets across all API configurations and policies using Named Values. It’s a global name-value collection. Value can be plain string or can be a secret. Using key vault secrets is recommended because it helps improve API Management security. [How to use named values in Azure API Management policies | Microsoft Learn](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-properties?tabs=azure-portal)

### Management

* When you are thinking about implementing auto-scaling, make sure:
* Be aware of maximum throughput - <https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits#api-management-limits>
* Be aware of the maximum number of scale units per APIM service tier. <https://azure.microsoft.com/pricing/details/api-management/>
* Be aware of the time required to scale out/ deploy to another region or convert to a different service tier. APIM doesn't scale out automatically, you need additional configurations for that - <https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-autoscale>
* When you use Application Insights for logging and use policies to modify incoming request/ outbound response:
* Based on internal load tests, enabling the logging feature caused a 40%-50% reduction in throughput when request rate exceeded 1,000 requests per second. Ref. [Integrate Azure API Management with Azure Application Insights - Azure API Management | Microsoft Learn](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-app-insights)
* When multiple policies are applied, each policy adds processing overhead, which can affect the overall performance of the API Management instance. It is important to carefully evaluate and optimize the policies to strike a balance between functionality and performance. Reducing the number of policies or optimizing their execution can help improve the overall performance of Azure API Management.
* Use an [external cache](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-cache-external) for control and fastest performance.
* When designing your strategy for APIM Backup and BCDR(Business Continuity & Disaster Recovery), please make sure you have already determined RTO and RPO. Below mentioned links will help you get started with BCDR implementation:
  + This article shows how to automate backup and restore operations of your API Management instance using an external storage account. Please note Each backup expires after 30 days. If you attempt to restore a backup after the 30-day expiration period has expired, the restore will fail with a Cannot restore: backup expired message.
    - * [Backup and restore your Azure API Management instance for disaster recovery - Azure API Management | Microsoft Learn](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-disaster-recovery-backup-restore?tabs=powershell#calling-the-backup-and-restore-operations)
* What is not backed up? [Backup and restore your Azure API Management instance for disaster recovery - Azure API Management | Microsoft Learn](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-disaster-recovery-backup-restore?tabs=powershell#what-is-not-backed-up)
* In case of outage, you may want to consider the feasibility for deploying fresh instances or having a hot/ cold standby.
* Failover can be automated - Multizone is automatic whereas multi-region deployment required a DNS-based load balancer such as Traffic Manager.
  + - * <https://learn.microsoft.com/en-us/azure/api-management/zone-redundancy>
      * [Deploy Azure API Management instance to multiple Azure regions - Azure API Management | Microsoft Learn](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-deploy-multi-region)

### Governance

* It is recommended to use Azure Policies. Azure Policy helps to enforce organizational standards and to assess compliance at-scale. Through its compliance dashboard, it provides an aggregated view to evaluate the overall state of the environment, with the ability to drill down to the per-resource, per-policy granularity. There are in-built policies for API Management these are enforced when you create an APIM instance using the landing zone accelerator. Below reference will help you get started.
  + [Overview of Azure Policy - Azure Policy | Microsoft Learn](https://learn.microsoft.com/en-us/azure/governance/policy/overview)
  + In-built APIM policies - [Built-in policy definitions for Azure API Management | Microsoft Learn](https://learn.microsoft.com/en-us/azure/api-management/policy-reference)
    - API Management calls to API backends should be authenticated.
    - API Management direct management endpoint should not be enabled
    - API Management secret named values should be stored in Azure Key Vault
    - API Management subscriptions should not be scoped to all APIs

### Platform Automation and Governance

If you're keen on adopting an infrastructure-as-code approach to manage your APIM instance and harness the full potential of a version control system, ApiOPS (<https://github.com/azure/apiops>) can be utilized.

* + APIOps applies the concepts of GitOps and DevOps to API deployment. By using practices from these two methodologies,
  + APIOps can enable everyone involved in the lifecycle of API design, development, and deployment with self-service and automated tools to ensure the quality of the specifications and APIs that they’re building.

## References

* <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/app-platform/api-management/landing-zone-accelerator>
* https://azure.github.io/apiops/
* <https://github.com/Azure-Samples/app-templates-microservices-integration>