Foundations of Modern Cloud Operations

Exploring service integration, hybrid deployments, and enterprise outcomes

Mustafa Hussein, Farhad Karimov, Matteo Marchi and Tobias Zeier, the 8th of September 2025
University of Essex Online

1 Introduction

This report, developed as a collaborative team project, provides an analysis of cloud infrastructure and frameworks, using Microsoft Azure Resource Manager (ARM) as an illustrative case study.

2 Framework Overview and Core Features

Azure Resource Manager (ARM) is the control plane for all resources in Azure. It ensures that no matter how you interact with Azure (Portal, CLI, PowerShell, REST, SDKs), the request is processed in a consistent way. (Gupta, 2023). ARM's main jobs include: Authentication & Authorisation (via Azure AD + RBAC); Validation & Dependency Handling (ensuring resources are created in the right order); Routing requests to the right Resource Provider (e.g. Microsoft Compute for VMs, Microsoft Network for VNets)); Governance & Organisation (Policies, Locks, Tags, Cost visibility). (Gao, 2025a)

2.1 How it Works

All clients and tools communicate only through ARM. This guarantees consistency, auditing, and security for every deployment. (Gao, 2025a)

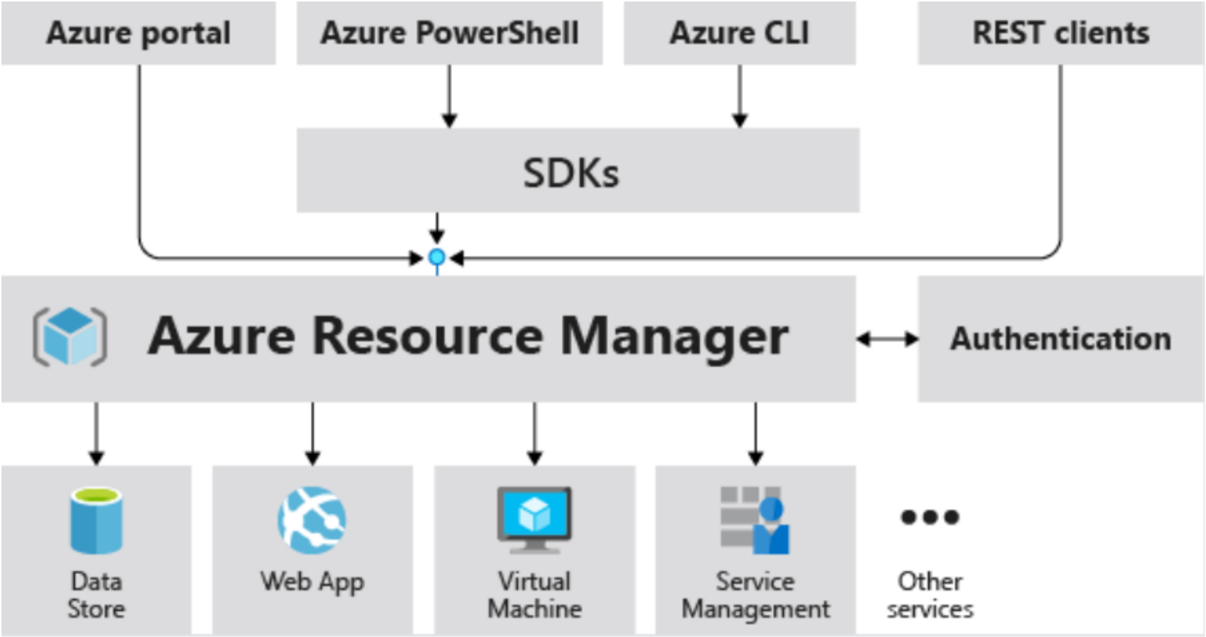

Figure 1: Azure Ressource Manager (Gao, 2025a)

Scope Hierarchy

ARM applies management and deployments at four levels:

1. Management Groups: Organise multiple subscriptions, apply governance globally;
2. Subscriptions: Billing and access boundary;
3. Resource Groups: Logical grouping of related resources;
4. Resources: Individual services such as VMs, databases, or storage accounts.

Governance rules set at a higher-level flow down to lower levels. (Gao, 2025a)

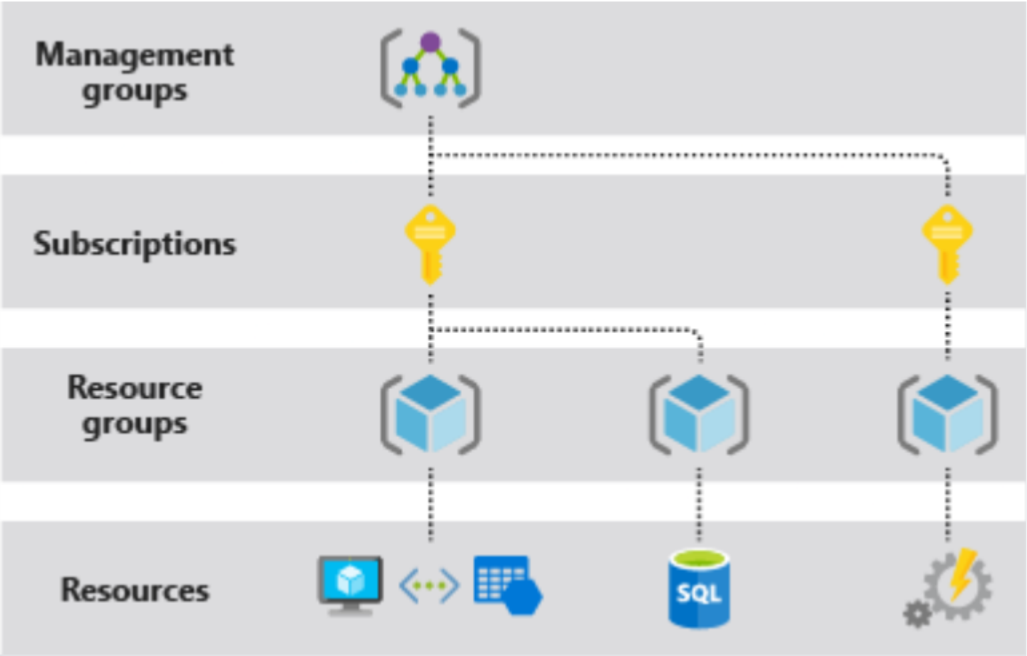

Figure 2: Levels of Management Scope (Gao, 2025a)

Key Capabilities

- Infrastructure as Code with Bicep or ARM JSON templates (Gao, 2025a).
- Automation with idempotent deployments (Gao, 2025a).
- What-if deployment previews (Gao, 2025b).
- Governance: RBAC, Policy, Locks, Tags.
- Deployment Stacks to manage groups of resources.
- Integration with CI/CD tools (Gao, 2025a).

Quick Comparison: ARM vs. AWS CloudFormation

Feature	**Azure Resource Manager (ARM)**	**AWS CloudFormation**
IaC Format	Bicep (preferred), ARM JSON	YAML / JSON
Change Preview	What-if	Change Sets (AWS, 2025)
Grouping Mechanism	Resource Groups, Deployment Stacks	Stacks, StackSets (AWS, 2025)
Governance Integration	RBAC, Policy, Tags built-in	IAM + Config + Tags
Scope Hierarchy	MG → Subscription → RG → Resource	Account → Stack → Resource

3 Service Model Analysis

ARM serves as the orchestration layer for Microsoft Azure to deploy and govern resources through its deployment mechanism. The system uses declarative JSON or Bicep templates to provide consistent management across IaaS and PaaS and SaaS resources (Gao, 2023).

3.1 IaaS Support

ARM automates provisioning of Virtual Machines (VMs), networking (VNets, firewalls), and storage. The Infrastructure-as-Code (IaC) approach enables organisations to deploy resources securely and consistently (Morris, 2021). The governance system becomes stronger through Role-Based Access Control (RBAC) and Azure Policy which decrease human mistakes while maintaining compliance standards (Gao, 2023).

3.2 PaaS Support

ARM enables straightforward deployment of Azure App Services together with SQL Database and Azure Kubernetes Service (AKS). A single template allows users to create entire application stacks which enables CI/CD integration and operational efficiency (Gupta, 2024). The deep integration system enables developers to concentrate on applications because the infrastructure operates automatically.

3.3 SaaS Support

ARM does not provide SaaS directly, but it controls identity management and compliance for Microsoft 365 and Dynamics 365 and all integrated SaaS solutions. ARM enables organisations to enforce Azure AD rules which restrict Teams and SharePoint access to compliant devices thus enhancing the security posture (Mell and Grance, 2011; Gao, 2023).

3.4 Strengths and Limitations

The primary advantages of ARM include its unified governance system together with its scalability features and its ability to integrate with various ecosystems (Gupta, 2024). The platform delivers one of the most seamless IaaS, PaaS and SaaS combinations. The main difficulties with ARM stem from its complex JSON templates which Bicep helps address and its restricted ability to work across multiple clouds when compared to Terraform (Morris, 2021). The platform provides robust SaaS support for Microsoft solutions but its support for third-party platforms remains limited.

4 Deployment Types

Azure Resource Manager (ARM) provides a declarative, infrastructure as code approach for modelling, deploying, and governing resources across public, private, and hybrid environments. In practice, templates (ARM/Bicep), role-based access control (RBAC), tagging, and policy-based governance standardise provisioning and operations, reducing configuration drift and improving repeatability (Gao, 2025a; Rahman, Mahdavi-Hezaveh and Williams, 2019).

4.1 Public cloud (Azure).

In Azure regions, ARM is the native control plane. RBAC enforces least privilege access, while Azure Policy constrains locations, SKUs, and configuration baselines. Advantages include elastic scale, global reach, and CI/CD-friendly automation. Trade-offs include the shared responsibility model, management of egress costs, and designing for multi-region resilience. Capacity is effectively cloud scale, but deployments are governed by subscription and regional quotas (Gao, 2025a).

4.2 Private cloud (Azure Stack Hub).

Where data sovereignty, locality, or edge latency are decisive, an Azure consistent platform on premises allows continued use of ARM constructs—templates, resource groups, and RBAC—inside organisational boundaries. This preserves residency control and aligns on-premises operations with cloud practices. Trade-offs include hardware-bound capacity and a narrower control surface than public Azure, increasing reliance on process and monitoring. Infrastructure as code remains valuable for auditability and consistent builds across environments (Rahman, Mahdavi-Hezaveh and Williams, 2019; Gao, 2025a).

4.3 Hybrid (Azure Arc + ARM).

Hybrid scenarios project non-Azure resources (for example, servers and Kubernetes clusters) into a single Azure control plane so tagging, RBAC, and policy can be applied alongside native workloads. This reduces configuration drift and provides a unified view for assurance. Compliance must address cross-jurisdictional obligations such as GDPR Article 17, with auditable procedures for locating and erasing personal data across cloud and on-premises stores, including backups and edge locations (Kelly, Furey and Curran, 2021).

4.4 Organisational considerations.

Public Azure maximises agility and policy breadth; private Azure prioritises sovereignty and predictable locality; hybrid centralises governance while accommodating placement and latency constraints. The choice should weigh demonstrable erasure and audit processes (Kelly, Furey and Curran, 2021), the need for repeatable infrastructure as code pipelines and drift control (Rahman, Mahdavi-Hezaveh and Williams, 2019), and the benefits of elastic capacity versus hardware-bound scale (Gao, 2025a).

5 Infrastructure Design and Case Study

Designing a basic infrastructure with Microsoft Azure Resource Manager (ARM) follows a declarative approach, enabling resources to be defined, provisioned, and managed consistently. The process typically involves the following steps:

1. **Define scope and governance.** Infrastructure design begins by creating a resource group, which acts as the logical container for resources. Policies, naming conventions, and role-based access controls (RBAC) should be applied

at the resource group or subscription level to ensure compliance and security (Artiom, 2025).

2. **Configure networking.** A virtual network *(VNet)* is provisioned to provide secure communication across resources. Subnets, network security groups, and routing rules establish segmentation and access control. Private endpoints and firewalls can be incorporated to safeguard external access (Microsoft, 2023a).
3. **Provision compute.** Compute services, such as Virtual Machines (VMs), are defined with parameters including size, image, and availability options. Extensions may be deployed to configure security agents or monitoring tools automatically (Patni, Banerjee and Tiwari, 2020).
4. **Attach storage.** Storage *Accounts* provide access to blob, table, and file services. Choices around redundancy (e.g., locally redundant storage or geo-redundant storage) and encryption must be defined according to business continuity requirements (Patni, Banerjee and Tiwari, 2020).
5. **Parameterise and automate.** Templates (ARM or Bicep) are used to express dependencies between resources, enabling repeatable and idempotent deployments across environments such as development, test, and production (Patni, Banerjee and Tiwari, 2020).

This structured methodology supports consistency, reduces configuration drift, and embeds governance as code, ensuring infrastructures are scalable, secure, and aligned to organisational requirements.

5.1 Case Study

A notable example of Microsoft Azure Resource Manager (ARM) adoption is the case of ASOS, a leading online fashion retailer. Facing the need to scale infrastructure for seasonal demand while maintaining cost efficiency, ASOS used ARM templates to automate deployments and enforce tagging policies (FitzMacken, 2025). This automation reduced manual errors, improved governance, and accelerated service rollouts by leveraging ARM's declarative, idempotent model (Gao, 2025). Through centralised management, ASOS integrated cost-control mechanisms into deployments and achieved annual savings of 15–20%, rising to 25–40% after sustained optimisation (Margetts, 2021). Beyond financial benefits, ARM enabled greater organisational agility, with new environments provisioned in hours instead of weeks. Similar DevOps-driven implementations highlight that ARM templates provide repeatability, traceability, and operational efficiency across multi-team environments (Lane, 2025). Thus, ARM not only enhanced ASOS's scalability but also embedded cost optimisation and governance directly into its cloud operations.

Word Count: 1,368

References:

AWS (2025) *Update CloudFormation stacks using change sets*. Available at: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks-changesets.html (Accessed: 21 August 2025).

Gao, J. (2025a) *What is Azure Resource Manager?*. Available at: https://learn.microsoft.com/en-us/azure/azure-resource-manager/management/overview (Accessed: 21 August 2025).

Gao, J. (2025b) *What is Bicep?*. Available at: https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/overview (Accessed: 21 August 2025).

Gao, J. (2023) *Azure Resource Manager overview. Microsoft Docs*. Available at: https://learn.microsoft.com/en-us/azure/azure-resource-manager/ (Accessed: 19 August 2025).

Gupta, D. (2024). *The Cloud Computing Journey: Design and Deploy Resilient and Secure Multi-Cloud Systems with Practical Guidance*. 1st edn. Birmingham: Packt Publishing, Limited.

Mell, P. and Grance, T. (2011) *The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology (Special*

Publication 800-145). National Institute of Standards and Technology. Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf. (Accessed 19 August 2025).

Morris, K. (2021) *Infrastructure as code: dynamic systems for the cloud age*. 2nd edn. Sebastopol, California: O'Reilly Media, Incorporated.

Kelly, M., Furey, E. and Curran, K. (2021) 'How to achieve compliance with GDPR Article 17 in a hybrid cloud environment', *Sci*, 3(1), p. 3. Available at: https://doi.org/10.3390/sci3010003

Rahman, A., Mahdavi-Hezaveh, R. and Williams, L. (2019) 'A systematic mapping study of infrastructure as code research', *Information and Software Technology*, 108, pp. 65-77. Available at: https://doi.org/10.1016/j.infsof.2018.12.004

Artiom, K. (2025). 'Cloud Migration Framework: Transitioning from On-Premises to Azure Cloud for Improved System Reliability and Scalability'. *The American Journal of Applied Sciences*, 7(2), pp. 5-11. Available at: https://doi.org/10.37547/tajas/volume07issue02-02

Patni, J.C., Banerjee, S. and Tiwari, D. (2020) 'Infrastructure as a Code (IaC) to Software Defined Infrastructure using Azure Resource Manager (ARM)', *2020 International Conference on Computational Performance Evaluation (ComPE)*, Shillong, India, 2-4 July. IEEE. Pp. 575-578. Available at: https://doi.org/10.1109/ComPE49325.2020.9200030

Lane, J. (2025) *Case Study: Data Driven DevOps with ADO*. Available at: https://techcommunity.microsoft.com/blog/appsonazureblog/case-study-data-driven-devops-with-ado/4128814 (Accessed: 21 August 2025).

Margetts, I. (2021) *ASOS implements cost optimization to fashion innovation for the future*. Available at: https://www.microsoft.com/en/customers/story/1375958406314056214-asos-retailers-azure (Accessed: 23 August 2025).

FitzMacken, T. (2025) *What are ARM templates?*. Available at: https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/overview (Accessed: 23 August 2025).

Gao, J. (2025) *Deploy ressources with ARM templates and Azure portal*. Available at: https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-portal (Accessed: 23 August 2025).