Protera

White paper

Azure Landing Zones

Part 1

Author:

Eric MacDonald

VP of Cloud Transformation

TABLE OF CONTENTS

Protera

Resource organization with subscriptions and management groups

(Platform-vs-Application model in depth)

When I was in the us army back in the mid 2000s as a cavalry scout(19D), every mission brief opened with two map sheets: one showing the grid for the mission, the other laying out the radio plan to keep us tied into troop HQ. Azure landing zones follow the same rhythm. First, stake out the ground each team owns – your subscriptions. Second, chart the path for every order – your management groups. Once that framework is locked, the rest – identity, networking, security, and cost – falls neatly into place.

This white paper is a quick-reference guide. Each topic here can easily fill its own workshop, but I have pulled together the key points you need to keep top of mind as you build and run your landing zones

Why resource organization comes first

Microsoft's Cloud Adoption Framework (CAF) puts Resource Organization at the front of its eight design areas because every control plane inherits from it. A rushed hierarchy means re-wiring hundreds of role assignments and policies later. CAF guidance, updated March 2025, also clarifies the split between platform landing zones and application landing zones, each backed by one or more subscriptions. Microsoft Learn

Subscriptions — the operational units

Key traits

Area	Details
Owner	Billing, service quotas, Azure Policy scope and RBAC; resources cross a subscription line only through deliberate links such as VNet peering or Private Link
Lifecycle	Tie the life of the subscription to the life of the workload. Deleting the sub guarantees full clean-up.
Tenant link	One subscription belongs to one Entra tenant. Moving it later is possible but slow and audited.
Soft limits	vCPU, route tables, resource-group count and Azure Monitor rules. New limits began rolling out July 2025. Plan for three-year growth rather than short spikes. Microsoft Learn

Platform-vs-Application view

Key traits

Class	Purpose	Typical services	Guardrails
Platform	Shared foundation used by every workload	Hub VNet, Azure Firewall, Log Analytics, Key Vault, Defender plans	Locked-down RBAC, global diagnostics, cost exports to Finance
Application	Isolated workload space aligned to a single product or cost center	AKS clusters, App Service, Cosmos DB, App Gateway	Deny public IP, team Contributor role, workload budget

Decision levers

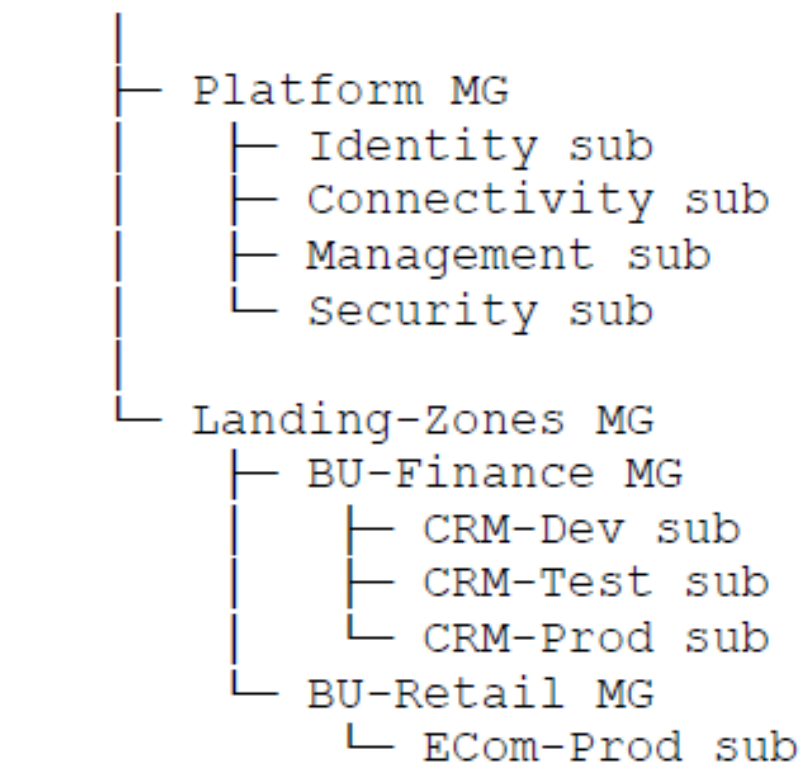
Driver	Questions to ask	Typical triggers
Quota runway	Will the workload hit vCPU or storage caps within three years?	Analytics clusters, rapid CI-CD sprawl
Autonomy	How much self-service does the product team need?	DevOps culture, SaaS teams
Data sovereignty	Must data stay in a specific geography?	EU GDPR, US FedRAMP
Legal mobility	Could the business unit be divested or merged?	M and A plans
Shared-service ties	Does the workload depend on the hub VNet or shared Key Vault?	All landing-zone patterns

🚩 Subscription anti-patterns to avoid

- One giant “all-in” subscription
- Region-only subscription splits
- Dozens of empty “just in case” subs
- Lifting a VM into Azure while its database stays on-prem inside the same sub
- These add governance drag and dilute the blast-radius benefit. Microsoft Learn

Management groups - Governance headquarters

CAF reference hierarchy (platform-vs-application)



Policy cascade

Level	Typical policies
Root MG	Allowed locations, mandatory tag inheritance
Platform MG	Diagnostic settings, Defender enablement, cost exports
Landing-Zones MG	Security baseline, deny unsupported SKUs
Application sub	Workload-specific rules such as deny public IP

RBAC good practice

- Owner – automation identities only
- Contributor – platform engineers on platform subs, workload owners on their subs
- Reader or Cost Management – finance and audit at MG scope
- Once your organization matures more, start working towards least privilege with custom RBAC roles where they're needed. Some of the azure built in roles may be overly permissive

Extra patterns

- Hub-and-Spoke governance – mirrors network hub-spoke topologies.
- Value-stream alignment – map MGs to agile value streams to survive org reshuffles.
- Canary MG branch – parallel branch for policy testing before promoting to production. Microsoft Learn

Protera

Naming that lasts

Azure resources rarely allow rename operations. Microsoft's Naming and Tagging guidance (June 2025) recommends:

```
php-template
CopyEdit
<Org>-<Env>-<ResourceAbbrev>-<Instance>
```

Example: `pro-prd-kv-core01`.

Add-ons

- Follow the official abbreviation list to stay inside length limits. Microsoft Learn
- Capture changing details in tags, not names.
- Enforce via an Azure Policy initiative at Platform MG.
- Microsoft publishes the Azure Naming Tool to generate names that match the rule. Microsoft Learn

Blueprints for platform subscriptions

Subscription	Primary services	Recommended policies	Example name
Identity	Entra Connect, PIM, break-glass accounts	Global diagnostics, deny public IP	pro-plt-id-core
Connectivity	Hub VNet, VPN, ExpressRoute, Azure	NSG flow logs required, deny wildcard rules	pro-plt-net-hub
Management	Log Analytics, Update Manager, Backup Vault	All diagnostics to central LA workspace	pro-plt-mgmt
Security	Defender plans, SIEM integration	Auto-provision Defender extensions	pro-plt-sec

Blueprints for application landing-zone subscriptions

Standard environment trio

Environment	Focus	Policy set	Example name
Dev	Rapid spin-up and tear-down	Audit only	crm-dev
Test	Integration and performance	Audit plus soft denies	crm-test
Prod	Customer traffic	Enforced denies, Defender plans, budget	crm-prod

Compliance overlay

Create additional prod subs such as pci-prod or hipaa-prod and attach stricter initiatives for encryption and data residency.

Starter guide – fifteen concrete steps

Standard environment trio

1. Lock Root MG to fewer than five Global Admins. Microsoft Learn
2. Create Platform and Landing-Zones MGs.
3. Deploy the four platform subscriptions.
4. Run az provider register for each sub through an IaC pipeline.
5. Apply naming and tagging initiatives. Microsoft Learn
6. Import CAF policy sets such as Allowed Locations and Tag Inheritance. Microsoft Learn
7. Enable Defender plans in the Security sub.
8. Use ALZ Bicep modules to deploy the first application environment trio. GitHub
9. Attach a deny-public-IP initiative at each application sub.
10. Configure budgets and alerts at every prod sub and roll-up views at Landing-Zones MG.
11. Centralize diagnostics in the Management sub's Log Analytics workspace.
12. Schedule quarterly Entra access reviews.
13. Build a sub-factory GitHub Action that takes metadata and provisions new subs automatically.
14. Test new guardrails in the Canary MG branch, then promote.
15. Store every role assignment and policy file in Git for traceable history.

Automation pipelines

Tool	Scope	Reference
ALZ Bicep modules	MG hierarchy, policy sets, resource-provider registration	GitHub repo
Terraform CAF Enterprise-Scale	Same targets using HCL	Terraform Registry GitHub
Sub-factory GitHub Action	Creates sub, places under MG, registers RPs	Sample workflow in ALZ automation folder

Pipeline flow

- ServiceNow request posts JSON to GitHub.
- GitHub Action validates and opens a pull request.
- Bicep or Terraform applies hierarchy and policies.
- Action returns the new subscription ID to ServiceNow.

Monitoring and cost governance

Monitoring and cost governance

1. Azure Policy compliance dashboard at Landing-Zones MG for executive scorecards.
2. Cost Management exports in CSV, sent to Storage and visualised in Power BI.
3. Defender for Cloud regulatory dashboard scoped at Landing-Zones MG. Microsoft Learn

Scale and re-org playbooks

1. Region expansion – add a child MG such as Landing-Zones-EU, inherit global guardrails, attach EU specific data-residency policy.
2. M and A divestiture – move an entire BU MG subtree to a new tenant, redeploy platform guardrails with the same IaC.
3. Multi-cloud hub – house shared ExpressRoute or CXP gateways in the Connectivity sub and peer to application subs only.

Worked example – one year timeline

Just as a recon troop can pivot into new terrain without rewriting the entire op order, a well-codified hierarchy adapts without rework.

Month	Event
0	Platform hierarchy deployed
1	CRM workload goes live in North America.
4	EU sales team launches; new Landing-Zones-EU MG inherits policies.
6	Security team tests stricter deny-rules in Canary MG, then promotes
12	Retail business unit sold; BU-Retail MG plus subs moved to new tenant with zero downtime.

Common pitfalls and quick fixes

Pitfall	Impact	Fix
Owner at Platform MG	Wide over-privilege	Use PIM and delegate only to pipelines
Portal policy edits	Drift from IaC	Enforce pull-request gates
Stale service principals	Hidden attack path	Quarterly access reviews

Pitfall	Impact	Fix
Region hierarchy	Duplicate policy	Keep region at resource-group layer
Half-moved hybrid apps	Broken blast radius	Create integration sub for connectors

Field kits and references

1. Azure Landing Zones overview – CAF Landing Zone doc
2. Management groups design – CAF resource-org MG guidance
3. Azure naming and tagging – CAF naming best practices
4. Resource abbreviations – Azure abbreviation list
5. Subscription limits – Azure quota reference
6. Policy guardrails – Azure Policy overview
7. ALZ Bicep repo – GitHub Azure/ALZ-Bicep
8. Enterprise-Scale Terraform – GitHub Azure/Enterprise-Scale

Next steps

The command structure is in place. Part 2 will map Azure policy, identity and access onto this hierarchy, covering tenants, Privileged Access Workstations, break-glass accounts and fine-grained RBAC. With the map locked, the patrol is ready to move.

Learn More

Protera.com