CLOUDSWITCHED

FREE RESOURCE — GOVERNANCE GUIDE

Azure Resource Naming Convention Guide

Standardise resource naming in Microsoft Azure for governance, cost tracking and operational clarity. Establish a consistent pattern from day one — avoid the chaos that creeps in once a tenant has more than fifty resources.

5	12	7	Policy
NAME COMPONENTS	RESOURCE PREFIXES	MANDATORY TAGS	ENFORCED VIA AZURE POLICY

NAMING PATTERN · RESOURCE PREFIXES · TAGGING STRATEGY · GOVERNANCE

PREPARED FOR	PREPARED BY	VERSION	FORMAT
Cloudswitched Knowledge Library	Cloudswitched Ltd.	2026 Edition	Reference Guide

00

Why a naming convention matters

Without a naming convention, an Azure environment becomes unmanageable inside a single quarter. Inconsistent names make it impossible to identify resource owners, group resources for cost reporting, write Azure Policy that scales, or hand a tenant over to a new engineer without a verbal walkthrough. **Establish your convention before any resources are created.** Renaming after the fact is painful — some resource types (Storage Accounts among them) cannot be renamed at all.

WHAT THIS GUIDE GIVES YOU

A standard naming pattern, a resource-type prefix table for the dozen Azure resources every UK SME tenant carries, the seven mandatory tags we ship on every Cloudswitched engagement, and the four Azure-native enforcement levers that keep the convention alive after launch — Azure Policy, Management Groups, Resource Locks and tag-driven Cost Management.

THE COST OF GETTING THIS WRONG

An Azure tenant we audited in 2026 carried 412 resources across 31 resource groups — named `vm1, server-old, test-rebuild, NEW-DC, prod_VM` and 408 other variants. Identifying the owner of any given resource took an average of **18 minutes**. Cost-allocating to a department was structurally impossible. The remediation engagement — rename what we could, redeploy what we could not, retrofit Azure Policy — came in north of £42k.

How this guide is organised

- **01 Naming Convention Structure** — the five-component pattern, an example name decomposed, and Azure's length and character restrictions.
- **02 Resource Type Prefixes** — the 12 most common Azure resources with their standard prefix and a worked example name.
- **03 Tagging Strategy** — the seven mandatory tags every resource should carry, with purpose and example values.
- **04 Governance & Enforcement** — using Azure Policy, Management Groups, Resource Locks and Cost Management to keep names and tags clean.
- **Worksheet** — the page where you write your organisation's actual prefixes, tag values and abbreviations down.

USE THE WORKSHEET ON THE LAST PAGE

The pattern in this guide is a sensible default, but every organisation lands on a slightly different version — different workload abbreviations, different cost-centre codes, different region tags. Capture *your* answers on the worksheet and pin the resulting page somewhere your IT team and your Azure engineers can both find it.

01

Naming Convention Structure

Use a single, consistent format across every Azure resource. The five-component pattern below scales from a 20-resource tenant to a 2000-resource estate without losing readability or breaking sort order. Keep everything lowercase, separate components with a single hyphen, and never use spaces.

RECOMMENDED FORMAT

`{resource-type}-{workload}-{environment}-{region}-{instance}`

Example: **`vm-erp-prod-uksouth-001`** — a Virtual Machine running the ERP workload, in Production, in UK South, first instance.

The five components

COMPONENT	ALLOWED VALUES	EXAMPLE
Resource type prefix	`vm, rg, vnet, nsg, st, sql, app, kv, pip, lb, snet, afw`	vm- (Virtual Machine)
Workload / application	`erp, web, dc, file, backup, monitoring, hub`	erp (ERP system)
Environment	`prod, stag, dev, test, dr`	prod (Production)
Region	`uksouth, ukwest, westeu, northeu`	uksouth (UK South)
Instance number	`001, 002, 003`	001 (first instance)

AZURE NAMING RESTRICTIONS TO KNOW UP FRONT

Different resource types enforce very different rules. **Resource Groups** allow 1–90 characters. **Storage Accounts** allow only 3–24 characters, lowercase alphanumerics, *no hyphens* — which is why a storage account is named `sterpproduks001` not `st-erp-prod-uksouth-001`. **Virtual Machines** allow 1–64 characters but the Windows computer name limit is 15. Always check the Microsoft docs for the specific resource type before you commit. Standardise on lowercase to keep DNS, CLI and Terraform output predictable.

02

Resource Type Prefixes

A prefix is the first thing your eye reads when scanning a resource list. Use this table as the canonical set — aligned to the Cloud Adoption Framework abbreviations — for every Azure resource you create. Pin the table to your engineering wiki and reference it in your Terraform / Bicep modules.

RESOURCE TYPE	PREFIX	EXAMPLE NAME
Resource Group	`rg-`	`rg-erp-prod-uksouth`
Virtual Network	`vnet-`	`vnet-hub-prod-uksouth`
Subnet	`snet-`	`snet-servers-prod`
Network Security Group	`nsg-`	`nsg-web-prod-uksouth`
Virtual Machine	`vm-`	`vm-dc01-prod-uksouth`
Storage Account	`st`	`sterpproduks001` *(no hyphens)*
SQL Database	`sql-`	`sql-erp-prod-uksouth`
App Service	`app-`	`app-portal-prod-uksouth`
Key Vault	`kv-`	`kv-erp-prod-uksouth`
Public IP	`pip-`	`pip-fw-prod-uksouth`
Load Balancer	`lb-`	`lb-web-prod-uksouth`
Azure Firewall	`afw-`	`afw-hub-prod-uksouth`

STORAGE ACCOUNTS — THE SPECIAL CASE

Storage Account names must be globally unique across all of Azure, 3–24 characters, lowercase alphanumerics with no hyphens. They are also the resource type Microsoft **will not let you rename** — if you get it wrong you redeploy and re-migrate the data. Adopt a compact concatenated style (`sterpproduks001`) and pre-flight your candidate name with `az storage account check-name` before deploying.

GOING BEYOND THIS TABLE

For services not listed here — AKS clusters, Cosmos DB accounts, Service Bus namespaces, Container Registries, Function Apps — follow the Microsoft Cloud Adoption Framework abbreviations page (aka.ms/CAFresourceabbreviations) and add the resulting prefixes to your own copy of this table. Keep one canonical list per tenant; do not let teams invent their own.

03

Tagging Strategy

Tags add metadata that a name alone cannot carry — cost allocation, ownership, lifecycle and criticality. Apply the seven mandatory tags below to **every** Azure resource. Enforce required tags with Azure Policy at the Management Group level so a deployment that omits them is rejected before it ever lands.

TAG NAME	PURPOSE	EXAMPLE VALUES
`Environment`	Cost separation, access control, deployment guard-rails.	`Production, Staging, Development, DR`
`CostCentre`	Financial reporting, chargeback to department or project.	`IT, Finance, Marketing, Operations`
`Owner`	Accountability — who do you ring when the resource breaks.	`john.smith@company.com`
`Application`	Group related resources into one logical workload.	`ERP, Email, Website, Backup`
`CreatedDate`	Lifecycle management — spot abandoned resources.	`2026-01-15`
`ReviewDate`	Forced check-in — is the resource still needed.	`2026-07-15`
`Criticality`	Prioritise DR, backup tier and incident response.	`High, Medium, Low`

TAG INHERITANCE AND ENFORCEMENT

Tags do **not** automatically inherit from a resource group to its resources. Use the Azure Policy initiative *"Inherit a tag from the resource group if missing"* for each mandatory tag — that closes the gap without forcing every Bicep / Terraform author to remember the full list. Pair it with a separate *"Require a tag and its value"* policy in audit mode to catch what slips through.

MAKE COSTCENTRE WORK FOR FINANCE

Once **CostCentre** is on every resource, you can pivot the Azure Cost Management dashboard by cost centre instead of by resource group — which is the report Finance actually wants. Standardise the allowed values up front (a fixed list, not free text) and align them with the codes your accounts package already uses.

04

Governance & Enforcement

A naming convention written on a wiki page nobody reads is no convention at all. Azure gives you four native levers to make the convention self-enforcing — Azure Policy, Management Groups, Resource Locks and Cost Management. Turn them on, point them at the rules in this guide, and the tenant stays tidy without willpower.

- **Azure Policy.** Create deny policies that reject resource creation if names do not match your convention pattern — use the built-in *"Allowed locations"*, *"Require like pattern"* and custom regex match definitions. Run in audit mode for a week, then flip to deny.
- **Required tags.** Use the *"Require a tag and its value on resources"* Azure Policy for each of the seven mandatory tags — layered as an initiative at the Management Group root. Deployments without the tags fail before they consume any spend.
- **Management Groups.** Organise subscriptions into a Management Group hierarchy *(Tenant Root → Platform / Workloads → Prod / Non-prod)*. Policies and RBAC inheritance flow downward consistently — a new subscription inherits the right rules the moment it joins the hierarchy.
- **Resource locks.** Apply `CanNotDelete` locks to critical production resource groups — production databases, key vaults, DR storage. A misclick in the portal stops being a tenant-ending event.
- **Cost Management.** Use the tag-driven views in Azure Cost Management to publish budgets, alerts and chargebacks per *CostCentre*, *Application* or *Environment*. The same tags that satisfy governance are the ones finance reports on.
- **Regular audits.** Schedule a monthly compliance report — Azure Resource Graph query or a Power BI dashboard — that lists every resource failing the naming pattern or missing a mandatory tag. Owner, count, gap. Send it to the platform team and treat it as a backlog.

START EARLY, ENFORCE CONSISTENTLY

It is dramatically easier to enforce a naming convention from day one than to retrofit it across a tenant with 500 misnamed resources. Some Azure resources *cannot be renamed* after creation (Storage Accounts are the headline example) — so the rename plan ends up being a redeploy plan. Establish your convention, document it, layer in the Azure Policy initiative **before** the first production workload is provisioned.

THE FOUR-POLICY STARTER PACK

If you only ship four policies on day one, ship these: **(1)** require resource name to match the agreed regex pattern; **(2)** require the seven mandatory tags with a fixed allowed-values list for Environment and Criticality; **(3)** restrict deployments to *uksouth* and *ukwest*; **(4)** require *CanNotDelete* lock on every Production resource group. Everything else can be added once these four are clean.

Your conventions worksheet

Capture your organisation's actual values below. Print this page, fill it in, and pin it somewhere both your IT team and your Azure engineers can reach. These values feed straight into the Azure Policy initiative on the previous page.

Naming pattern

PATTERN Your agreed format	
SEPARATOR & CASE e.g. hyphen, lowercase	

Environment & region abbreviations

PRODUCTION e.g. prod	
STAGING / DEV / TEST e.g. stag / dev / test	
DR ENVIRONMENT e.g. dr	
PRIMARY & DR REGION e.g. uksouth / ukwest	

Mandatory tag values

COSTCENTRE VALUES fixed list	
CRITICALITY VALUES e.g. High / Medium / Low	
DEFAULT OWNER FORMAT e.g. firstname.surname@	

Sign-off

CONVENTION AUTHOR	DATE AGREED	NEXT REVIEW

CLOUDSWITCHED

AZURE GOVERNANCE & CLOUD SERVICES

info@cloudswitched.com

New London House, 6 London St
London EC3R 7LP · United Kingdom