Bilko Azure IaC — Terraform azurerm (rg-bilko-demo)

Bilko Azure IaC — Terraform azurerm (rg-bilko-demo)

MC: #103720 (child of #103715)
Status: Converged (terraform plan = no changes)
Branch: feat/103715-azure-terraform-iac → PR #380
Date: 2026-06-16

Overview

Context: GCP project billing exhausted → Bilko migrated to Azure. Azure side had ZERO IaC (14 resources hand-created via az CLI). This created drift risk, manual errors, and no audit trail.

Solution: Entire `rg-bilko-demo` resource group (~23 resources) now under Terraform azurerm provider. Infrastructure is declarative, version-controlled, and reproducible.

Current state: `terraform plan` returns "No changes. Your infrastructure matches the configuration." (fully converged).

Architecture

Azure Topology

- **Subscription:** 5b0b4d9b-e677-464e-abf0-5170cbce3b8e
- **Resource Group:** rg-bilko-demo
- **Region:** swedencentral

Resource Inventory

Resource	Name	Type/SKU	Notes
Resource Group	rg-bilko-demo	-	Parent container
ACA Environment	bilko-demo-env	Consumption	Shared environment for all container apps
Container Registry	bilkodemo	ACR	Private image registry
PostgreSQL	bilko-demo-pg	Flexible, B_Standard_B1ms, PG16, zone 1	Main database
Key Vault	kv-bilko-demo2	-	2 access policies: managed_identity + terraform_user
Managed Identity	mi-bilko-demo	-	For ACA → Key Vault
App Insights	appi-bilko	-	+ action group appi-bilko-alerts
Availability Alert	-	-	→ alem@alai.no
5xx Metric Alert	-	-	→ alem@alai.no
Container App	bilko-api-demo	ACA	Adopted (ignore_changes)
Container App	bilko-web-demo	ACA	Adopted (ignore_changes)
Container App	bilko-unleash	ACA	Adopted (public Docker Hub image)
Container App	bilko-api-stage	ACA	Adopted (ignore_changes)
Container App	bilko-web-stage	ACA	Adopted (ignore_changes)
Firewall Rule	-	Postgres	FORGE runner 10.0.0.2/32

Module Map

Repo: `infrastructure/azure/terraform/`

```
graph LR
    ENV[envs/demo] --> RG[module: resource-group]
    ENV --> LA[module: log-analytics]
    ENV --> ACR[module: acr]
    ENV --> MI[module: managed-identity]
    ENV --> KV[module: keyvault]
    ENV --> PG[module: postgres]
    ENV --> ACAENV[module: aca-environment]
    ENV --> ACA1[module: aca-app bilko-api-demo]
    ENV --> ACA2[module: aca-app bilko-web-demo]
    ENV --> ACA3[module: aca-app bilko-unleash]
    ENV --> ACA4[module: aca-app bilko-api-stage]
    ENV --> ACA5[module: aca-app bilko-web-stage]
    ENV --> AI[module: app-insights]
```

```
    ACAENV --> LA
    ACA1 --> ACAENV
    ACA2 --> ACAENV
    ACA3 --> ACAENV
    ACA4 --> ACAENV
    ACA5 --> ACAENV
```

9 modules:

1. resource-group
2. log-analytics
3. acr
4. managed-identity
5. keyvault
6. postgres
7. aca-environment
8. aca-app (reusable, 5 instances)
9. app-insights

State Backend

Provider: azurerm (NOT GCS — gcloud out of the loop)

Backend config:

```
backend "azurerm" {
  storage_account_name = "stbilkotfstate"
  container_name       = "tfstate"
  key                  = "demo.terraform.tfstate"
}
```

Ops Access

To run terraform plan/apply manually:

```
export ARM_ACCESS_KEY=$(az storage account keys list -g rg-bilko-demo -n stbilkotfstate --
query "[0].value" -o tsv)
cd infrastructure/azure/terraform/envs/demo
terraform plan
```

CI/CD

Workflow: azure-infra.yml

New workflow (added in this PR):

- **Trigger:** PR with paths `infrastructure/azure/**` → runs `terraform plan`
- **Apply:** ONLY via manual `workflow_dispatch` + `confirm="APPLY"` input (never auto-apply — ZAKON PI2, live customer demo)
- **Runner:** self-hosted FORGE
- **Auth:** AZURE_CREDENTIALS SP (alai-cli-deployer f2a3b94b, Contributor role)
- **Backend auth:** ARM_ACCESS_KEY from storage account key

Boundary: Infra vs. App Rollout

CRITICAL: Infrastructure = Terraform; APP ROLLOUT stays imperative.

Concern	Tool	Location
Resource creation/config	Terraform	azure-infra.yml
App image rollout	az containerapp update --image	azure-stage.yml / azure-deploy.yml

Do NOT move rollout to Terraform. The `aca-app` module uses `lifecycle { ignore_changes }` on container image to preserve imperative rollout.

Adopt-vs-Managed Pattern

The `aca-app` module has **TWO modes**:

1. Managed (greenfield)

Full Terraform control of env vars, secrets, image, traffic weight.

```
ignore_env_secrets = false
```

2. Adopted (existing apps)

Terraform imports existing resource but ignores runtime config (env/secrets/image/revision_mode/custom_domain/traffic_weight). Used for the 5 hand-built apps adopted as-is.

```
ignore_env_secrets = true


lifecycle {
  ignore_changes = [
    template[0].container[0].image,
    template[0].container[0].env,
    secret,
    ingress[0].custom_domain,
    ingress[0].traffic_weight,
    template[0].revision_suffix
  ]
}
```

All 5 current apps use adopted mode:

- bilko-api-demo
- bilko-web-demo
- bilko-unleash
- bilko-api-stage
- bilko-web-stage

Gotchas & Lessons

1. Adopted ACA updates trigger NEW REVISION

Issue: Even with `ignore_changes`, any Terraform change to an adopted container_app triggers a new revision (graceful zero-downtime rolling restart) — NOT a silent no-op.

Mitigation: Minimize unnecessary Terraform changes to adopted apps. Review plan carefully before apply.

2. ACA environment force-replacement bug

Issue: azurerm 3.x tries to force-replace ACA environment on unchanged `log_analytics_workspace_id`.

Fix: Added `ignore_changes = [log_analytics_workspace_id]` to aca-environment module.

3. Postgres zone must be pinned

Issue: Azure blocks zone changes on existing Postgres Flexible servers.

Fix: Hardcode `zone = "1"` + `ignore_changes = [zone]`.

4. Public-image apps (unleash) must NOT get ACR registry block

Issue: Unleash pulls from Docker Hub, not ACR. If module tries to set ACR registry, plan fails.

Fix: Dynamic registry block gated on `registry_username != null`:

```
dynamic "registry" {
  for_each = var.registry_username != null ? [1] : []
  content { ... }
}
```

5. workload_profile_name drift

Issue: Imported apps have `workload_profile_name = "Consumption"`. If not set in Terraform, drifts to `null`.

Fix: Explicitly set `workload_profile_name = "Consumption"` for adopted apps.

6. NEVER commit .terraform/ or local tfstate

Issue: `.terraform/` contains 273MB provider binary. Local tfstate can leak secrets.

Fix: Added to `.gitignore`.

Runbook: Safe Plan/Apply

Local Development

```
# 1. Authenticate
az login
export ARM_ACCESS_KEY=$(az storage account keys list -g rg-bilko-demo -n stbilkotfstate --
```

```
query "[0].value" -o tsv)

# 2. Navigate
cd infrastructure/azure/terraform/envs/demo

# 3. Plan
terraform init  # first time only
terraform plan

# 4. Apply (if safe)
terraform apply

# 5. Verify
az containerapp list -g rg-bilko-demo --query "[].{name:name,
status:properties.provisioningState}" -o table
```

CI Apply (Production)

1. Open PR with infrastructure changes
2. Review `terraform plan` output in PR checks
3. Merge PR to main
4. Go to Actions → azure-infra.yml → Run workflow
5. Set `confirm` input to `APPLY`
6. Monitor run
7. Verify resources in Azure Portal

ZAKON PI2: Never auto-apply. Always manual approval for live customer demo environment.

Open Follow-ups

MC	Priority	Description
#103745	M	Migrate ACA secrets → Key Vault (live ACA secrets are write-only/unreadable; adopted apps still use manual secrets)
TBD	L	Narrow azure-stage.yml paths filter (coordinate with MC #103579)
TBD	M	Rotate live Unleash DB credentials (weak cred still active on running app)

MC #103720 (child of #103715) — ZAKON-PLAN mandatory documentation task

Revision #1
Created 2026-06-16 19:36:41 UTC by John
Updated 2026-06-16 19:36:41 UTC by John