PLATFORM · TEMPLATE

Building a Multi-Account AWS *Landing* Zone

KAANSYSTEMS.COM/LIBRARY/MULTI-ACCOUNT-AWS-LANDING-ZONE · AUGUST 8, 2026

ABOUT THIS TEMPLATE

Account topology, a starter SCP set, and an honest Control Tower vs. Terraform call for regulated SMBs building an AWS landing zone.

THE TEMPLATE

The OU map, the two SCPs not shown above, and the Terraform repo layout.

```
Root
├── Management account        billing + org admin; runs nothing; root creds in a safe
├── Security OU
│   ├── log-archive           org CloudTrail + Config → S3, Object Lock (compliance mode)
│   └── security-tooling      GuardDuty / Security Hub / Access Analyzer delegated admin
├── Infrastructure OU
│   ├── network               peering or TGW, IPAM, (later) central egress
│   └── shared-services       CI runners, ECR, artifact storage, internal tools
├── Workloads OU
│   ├── prod                  1h admin sessions, tightest SCPs
│   └── staging               mirrors prod topology; SCP canary target
└── Sandbox OU
    └── sandbox-<engineer>    $200 budget alarm, no peering, aws-nuke nightly
```

Region restriction, attached at the root (adjust the region list and the exempt global services to taste):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyOutsideAllowedRegions",
      "Effect": "Deny",
      "NotAction": [
        "iam:*", "organizations:*", "sts:*", "cloudfront:*",
        "route53:*", "support:*", "budgets:*", "health:*"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:RequestedRegion": ["us-east-1", "us-west-2"]
        }
      }
    }
  ]
}
```

Log-archive protection, attached to the Security OU. The pipeline role is exempted; otherwise this SCP would also block the Terraform stack that manages the bucket's versioning, lock, and lifecycle configuration, and day-2 applies would fail:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ProtectLogArchiveBucket",
      "Effect": "Deny",
      "Action": [
        "s3:DeleteBucket",
        "s3:DeleteBucketPolicy",
        "s3:PutBucketVersioning",
        "s3:PutBucketObjectLockConfiguration",
        "s3:PutLifecycleConfiguration"
      ],
      "Resource": "arn:aws:s3:::org-log-archive-*",
      "Condition": {
        "ArnNotLike": {
          "aws:PrincipalArn": "arn:aws:iam::*:role/terraform-pipeline"
        }
      }
    }
  ]
}
```

Terraform repo layout, one state per stack, applied by a pipeline role:

```
terraform/
├── backend.hcl
├── org/                      # runs against the management account
│   ├── main.tf               # organization, OUs, member accounts
│   ├── scps.tf               # policy documents + OU attachments
│   └── identity.tf           # Identity Center permission sets + assignments
├── security/
│   ├── log-archive/          # bucket, Object Lock, org CloudTrail
│   └── tooling/              # GuardDuty / Security Hub delegated admin
├── infrastructure/
│   └── network/              # VPCs, peering or TGW, IPAM pools
└── modules/
    ├── organizations/        # OU + account factory (tags, close-on-delete)
    ├── scp/                  # policy rendering + attachment
    ├── log-archive/          # bucket with lock, replication, lifecycle
    ├── identity/             # permission set + group assignment pattern
    └── network/              # standard VPC shape, flow logs on by default
```

HOW TO USE IT

The account count wants to grow. Let it, slowly.

A new account is justified by a new blast radius worth isolating: a new compliance boundary, a workload with its own on-call, a data classification that must not share IAM space with the rest. It is not justified by a new microservice or a new team lead's preference. Ten to fifteen accounts is the healthy range for this size of company. If you are at forty, you have rebuilt the governance problem you started with, except now it has a monthly baseline bill in every account — Config, GuardDuty, and Security Hub all meter per account — and nobody remembers what `prod-svc-legacy-2` does.

SCPs are production changes

An SCP typo can lock every engineer out of every workload account simultaneously, which makes the SCP repo the highest-blast-radius code you own. Treat it accordingly: PR review with plan output, rollout through the staging OU with a soak period before it touches Workloads, and a CloudWatch alarm on `organizations:*` mutations from any principal that is not the pipeline role. The upside in a regulated shop is that this discipline is self-documenting. Your SCP git history, with reviews and timestamps, is change-control evidence an auditor accepts without a follow-up meeting.

Retrofitting under a live workload

The landing zone is designed so production never has to move. Invite the existing account into the org and place it under the Workloads OU as prod, exactly where it belongs. Build the new accounts around it: log-

archive first (start capturing the org trail immediately, since evidence accrues from day one), then Identity Center, then shared-services, and migrate CI runners and staging out of the old account at leisure. Apply SCPs to the staging OU, watch for a week, then extend to prod. The anti-pattern is the migration weekend: a planned big-bang move of production into a "clean" account, which converts a governance project into an availability incident. Prod stays put. The org grows around it.