anchor@anchor-cyber:~$ cat cloud-security-hardening.md

ANCHOR CYBER SECURITY LLC

Cloud Security Hardening Reference

AWS · Azure · GCP — side-by-side hardening controls for IAM, logging, network segmentation, and encryption. Mapped to CIS Benchmarks and platform Well-Architected equivalents.

$ cloud-harden --iam

Identity & Access Management

IAM misconfigurations are the leading cause of cloud breaches. Apply least privilege, eliminate standing privilege where possible, and enforce MFA for all human access.

AWS

- Enable MFA on root account; do not use root for day-to-day work
- No long-lived access keys for service-to-service calls — use IAM roles
- Use IAM Identity Center (SSO) for human access; avoid per-user IAM users
- Apply SCPs via AWS Organizations to deny services not in use
- Use `aws:PrincipalTag` or ABAC for scalable least privilege
- Audit IAM Access Analyzer findings and unused permissions quarterly
- Rotate access keys ≤ 90 days; alert on keys older than threshold

AZURE

- Require MFA via Conditional Access for all users; block legacy auth
- Use Entra ID PIM for just-in-time privileged role activation
- Apply built-in RBAC roles at resource group scope; avoid subscription-level Owner
- Enable Entra ID Identity Protection for risky sign-in detection
- Block service principal secret-based auth; prefer managed identities
- Require compliant or hybrid-joined devices in Conditional Access
- Audit role assignments and guest access monthly via Access Reviews

GCP

- Use Workload Identity Federation — avoid user-managed service account keys
- Bind service accounts to specific resources; don't share across workloads
- Apply Org Policy constraints to block public resource sharing by default
- Use IAM recommender to right-size overpermissive roles
- Require phishing-resistant MFA (security keys) for privileged accounts
- Avoid roles/owner at project level; prefer predefined or custom roles
- Audit service account key creation with Cloud Audit Logs

$ cloud-harden --logging

Logging & Monitoring

Logs are only useful if collected completely, retained long enough, and alerted on. Management-plane activity is the minimum floor; data-plane logging adds cost but is often required for compliance.

AWS

- Enable CloudTrail in all regions; write to a separate, hardened S3 bucket
- Enable S3 server access logging for buckets containing sensitive data
- Enable VPC Flow Logs on all VPCs; route to CloudWatch or S3
- Use AWS Config to detect configuration drift on key resources
- Route CloudTrail to CloudWatch Logs; alert on root login, IAM changes, SG changes
- Enable GuardDuty in all accounts and regions; integrate findings to SIEM
- Retain CloudTrail logs ≥ 1 year (7 years for regulated workloads)

AZURE

- Enable Diagnostic Settings for all resources; route to Log Analytics workspace
- Enable Azure Activity Log and send to Log Analytics or Event Hub
- Enable Microsoft Defender for Cloud on all subscriptions
- Turn on Entra ID Sign-In Logs and Audit Logs; retain ≥ 30 days (P1/P2) or archive to storage
- Use Network Security Group Flow Logs v2 for traffic visibility
- Configure Azure Monitor Alerts for subscription-level policy changes
- Use Azure Policy to enforce diagnostic settings on new resources

GCP

- Enable Admin Activity audit logs (free, always on) for all services
- Enable Data Access audit logs for storage, BigQuery, and KMS
- Route Cloud Audit Logs to Cloud Storage or BigQuery for long-term retention
- Enable VPC Flow Logs for subnets handling sensitive traffic
- Use Security Command Center (SCC) Premium for threat detection
- Create log-based metrics and alerts for IAM policy changes, firewall rule changes
- Retain audit logs ≥ 400 days; use Org Policy to enforce log retention

```
$ cloud-harden --network
```

Network Segmentation & Boundary Protection

Default-deny inbound, private subnets for compute and data, and private connectivity to cloud services are the baseline. Avoid public IPs on resources that don't need them.

AWS

- No security group with `0.0.0.0/0` inbound except 80/443 on load balancers
- Place compute in private subnets; use NAT Gateway for outbound-only internet
- Use VPC Endpoints (Gateway and Interface) for S3, DynamoDB, and key services
- Enable Network Firewall or AWS WAF in front of public workloads
- Use NACLs as a secondary deny layer for known-bad CIDR ranges
- Avoid EC2-Classic; audit for publicly associated Elastic IPs quarterly
- Use AWS PrivateLink for cross-account service access

AZURE

- Apply NSG deny-all inbound as default; add specific allow rules per workload
- Use Private Endpoints for Azure PaaS services (Storage, SQL, Key Vault)
- Deploy Azure Firewall or a third-party NVA for centralized egress inspection
- Disable public network access on storage accounts and SQL by default
- Use Service Endpoints as a fallback where Private Endpoints aren't feasible
- Enable Azure DDoS Protection Standard on public-facing load balancers
- Audit public-facing resources monthly; remove unused public IPs

GCP

- Default VPC firewall rules allow all egress and deny all ingress — keep it that way
- Use Shared VPC to centralize network management across projects
- Enable Private Google Access on subnets so VMs can reach GCP APIs without public IPs
- Use VPC Service Controls to create security perimeters around sensitive data
- Apply hierarchical firewall policies at the org/folder level for baseline denies
- Use Cloud Armor for DDoS and WAF protection on external load balancers
- Audit resources with public IPs using Cloud Asset Inventory monthly

$ cloud-harden --encryption

Encryption at Rest & in Transit

Platform-managed encryption (default) is acceptable for most workloads. Use customer-managed keys (CMEK/CMK) for sensitive data, regulated workloads, or where key separation is required.

AWS

- Enable default S3 bucket encryption (SSE-S3 minimum; SSE-KMS for sensitive data)
- Enable EBS encryption by default at the account level
- Enforce RDS and Aurora encryption at rest; cannot be enabled post-creation
- Use ACM for TLS certificates; enforce HTTPS on all ALBs and CloudFront
- Set bucket policy to deny `aws:SecureTransport: false` for S3
- Use KMS key policies to restrict who can use or manage CMKs
- Enable key rotation (annually is automatic for AWS-managed keys)

AZURE

- Azure Storage is encrypted at rest by default using platform-managed keys
- Use Customer-Managed Keys in Key Vault for storage, SQL, Disk, and Backup
- Enforce HTTPS-only on Storage accounts and App Service via Azure Policy
- Use Azure Disk Encryption (ADE) for VM disks containing sensitive data
- Enable double encryption for extra-sensitive data (Platform + CMK)
- Use Key Vault Managed HSM for regulated workloads requiring FIPS 140-2 Level 3
- Set Key Vault soft-delete and purge protection on all key vaults

GCP

- All data at rest is encrypted by default with Google-managed keys (AES-256)
- Use CMEK via Cloud KMS for GCS, BigQuery, GKE, Cloud SQL, and Compute
- Enforce TLS 1.2+ on all HTTPS load balancers via SSL policy
- Use Cloud HSM for FIPS 140-2 Level 3 key storage requirements
- Use Org Policy `constraints/gcp.restrictPu` to block public Cloud SQL
- Enable key rotation schedules in Cloud KMS; alert on key version age
- Use VPC Service Controls to prevent CMEK keys from being used outside the perimeter

Sources: CIS Benchmarks (AWS v2, Azure v2, GCP v3), AWS/Azure/GCP Well-Architected Frameworks, NIST SP 800-53 Rev 5. Anchor Cyber Security LLC.

exit 0