CHECKLIST

AWS Virtualization Security Best Practices for Isolated Workloads Checklist

A practical checklist to validate network segmentation, IAM, host hardening, logging, metadata access, and change control for isolated AWS virtualized workloads.

AWS Virtualization Security Best Practices for Isolated Workloads

Checklist

Use this checklist to validate that an isolated AWS virtualized workload is actually protected by layered controls, not just placed in a private subnet or separate account.

---

1) Define the trust boundary

- [ ] Document the workload?s purpose and sensitivity level.
- [ ] Identify the assets that must remain isolated from general-purpose networks and users.
- [ ] Define the approved inbound sources, outbound destinations, and administrative paths.
- [ ] Record which controls enforce each boundary.
- [ ] Confirm the boundary is reviewed after every major change.

2) Restrict inbound access

- [ ] Allow inbound traffic only from explicitly approved sources.
- [ ] Avoid broad CIDR ranges unless there is a documented business requirement.
- [ ] Tie every inbound rule to a real consumer such as a management subnet, jump host, or upstream service.
- [ ] Remove any temporary access rules after use.
- [ ] Verify that unused ports are closed.

3) Restrict outbound access

- [ ] Define the minimum outbound destinations required for patching, logging, monitoring, and application dependencies.
- [ ] Block unnecessary internet access.
- [ ] Prevent unrestricted egress from workloads that handle sensitive data or administrative functions.
- [ ] Confirm the workload cannot reach internal services it does not need.
- [ ] Review outbound rules whenever new dependencies are added.

4) Validate route and subnet isolation

- [ ] Check that route tables do not create unintended paths to sensitive networks.
- [ ] Confirm the subnet design matches the intended level of isolation.
- [ ] Verify that network ACLs, where used, support the intended segmentation model.
- [ ] Ensure security groups remain the primary source of rule clarity and ownership.
- [ ] Test reachability from both approved and unapproved paths.

5) Apply least-privilege IAM

- [ ] Use temporary credentials or instance roles instead of static long-lived keys.
- [ ] Grant only the permissions required to boot, operate, patch, log, and recover the workload.
- [ ] Remove broad read access to secrets, storage, and service APIs unless specifically needed.
- [ ] Separate administrative access from application access.
- [ ] Review role permissions after every application, platform, or automation change.

6) Protect secrets and credentials

- [ ] Store secrets only in approved secret management mechanisms.
- [ ] Do not embed secrets in images, user data, configuration files, or scripts.
- [ ] Rotate secrets based on exposure risk and operational need.
- [ ] Verify that secrets are accessible only to the processes that require them.
- [ ] Confirm temporary credentials expire as expected.

7) Harden the guest image and operating system

- [ ] Start from a minimal approved image baseline.
- [ ] Remove unnecessary packages, services, and startup tasks.
- [ ] Disable or restrict remote access methods that are not required.
- [ ] Apply patches on a defined cadence.
- [ ] Enforce local privilege controls and limit interactive admin access.
- [ ] Rebuild from a known-good image rather than relying on manual drift corrections.

8) Lock down metadata access

- [ ] Verify metadata access is restricted to trusted workload processes only.
- [ ] Confirm the workload does not rely on metadata exposure to untrusted applications or scripts.
- [ ] Ensure user data and bootstrap scripts do not leak credentials.
- [ ] Review how agents, sidecars, and helper processes access the instance metadata service.
- [ ] Test that metadata-based credentials cannot be obtained by unauthorized local processes.

9) Secure logging and monitoring

- [ ] Send infrastructure and guest logs to a centralized logging platform.
- [ ] Capture authentication, authorization, network, and administrative events.
- [ ] Alert on unusual access attempts, policy changes, and configuration drift.
- [ ] Retain logs long enough to support investigation and compliance needs.
- [ ] Verify logs remain available even if the workload is compromised.

10) Control administrative access

- [ ] Limit human access to break-glass or approved administration workflows.
- [ ] Require strong approval and logging for elevated actions.
- [ ] Use dedicated admin paths rather than general user access channels.
- [ ] Review who can assume administrative roles.
- [ ] Remove stale access and unused exception paths.

11) Validate container or sidecar components, if used

- [ ] Minimize container capabilities.
- [ ] Limit writable paths and shared volumes.
- [ ] Use approved images only.
- [ ] Check that runtime permissions do not exceed the host?s isolation model.
- [ ] Review any additional agents, sidecars, or helper services for unnecessary privileges.

12) Check change management and drift

- [ ] Re-validate the workload after every image, network, policy, or role change.
- [ ] Compare the deployed state to the approved baseline.
- [ ] Remove emergency changes after the incident or maintenance window ends.
- [ ] Re-test access paths after any exception request.
- [ ] Maintain a simple owner for each control so gaps are not missed.

Quick validation test

If you can answer yes to each of the following, your isolation model is much stronger:

- [ ] I know exactly who can reach the workload.
- [ ] I know exactly what the workload can reach.
- [ ] I know what damage is limited if the workload is compromised.
- [ ] I can prove the controls still work after change.
- [ ] I have centralized logs and alerting for suspicious activity.

Production readiness questions

Before declaring the workload ready, confirm:

- [ ] Are inbound and outbound rules documented and reviewed?
- [ ] Are IAM permissions scoped to the minimum necessary set?
- [ ] Are secrets handled through approved mechanisms only?
- [ ] Is the image hardened, patched, and reproducible?
- [ ] Is metadata access constrained and validated?
- [ ] Are logs centralized and retained?
- [ ] Is the change process strong enough to prevent drift?
- [ ] Have access paths been tested from both approved and unapproved locations?

Rule of thumb

If a control cannot be clearly explained, owned, and rechecked after change, it is not strong enough for an isolated workload.

Treat isolation as a layered system: network, identity, host, secrets, telemetry, and operational discipline all have to work together.