CHECKLIST

AWS Virtualization Isolation Validation Checklist

A practical checklist for validating workload isolation, trust boundaries, and production readiness in AWS virtualization designs.

AWS Virtualization Isolation Validation Checklist

Use this checklist to evaluate whether an AWS virtualization design provides the level of workload isolation your system requires.

1) Define the workload risk profile

- [ ] Classify the workload by sensitivity: low / moderate / high / regulated
- [ ] Identify whether the workload processes confidential data, secrets, or tenant data
- [ ] Determine whether untrusted code, third-party tooling, or shared automation will run on or near the workload
- [ ] Document business impact if the workload is compromised
- [ ] Define the acceptable blast radius for a failure or breach
- [ ] Confirm whether compliance or contractual obligations require stronger isolation

2) Confirm the trust boundary

- [ ] Identify the primary isolation boundary in use: guest, host, network, account, or combination
- [ ] Verify that the chosen boundary matches the workload risk profile

- [ ] Confirm which components are shared with other workloads
- [ ] Document what adjacent workloads, operators, and tools are trusted
- [ ] Identify any implicit trust in shared images, base AMIs, golden templates, or automation pipelines
- [ ] Review whether the design depends on assumptions that have not been explicitly validated

3) Validate account and administrative separation

- [ ] Place workloads with different trust levels in separate accounts when appropriate
- [ ] Separate production, development, and testing administrative scopes
- [ ] Confirm IAM roles follow least privilege
- [ ] Review all cross-account access paths and justify each one
- [ ] Verify that break-glass access is controlled, logged, and periodically reviewed
- [ ] Confirm no shared long-lived credentials are used across environments
- [ ] Ensure infrastructure automation has only the permissions it truly needs

4) Validate network isolation

- [ ] Confirm workloads are placed in the correct VPCs and subnets
- [ ] Restrict east-west traffic to only required flows
- [ ] Review security groups and network ACLs for overly broad rules
- [ ] Remove unnecessary public exposure
- [ ] Confirm routing does not create unintended paths between trust zones
- [ ] Validate that administrative access uses approved, controlled entry points
- [ ] Review service-to-service communication for implicit trust or wildcard access

5) Validate guest and instance hardening

- [ ] Use a hardened guest image with minimal packages and services
- [ ] Patch the operating system and instance agent components regularly
- [ ] Disable unnecessary local services, ports, and protocols
- [ ] Verify host firewall or guest firewall policy is enforced where applicable
- [ ] Use secure boot or equivalent integrity protections if supported and required
- [ ] Confirm instance metadata access is restricted and protected from misuse
- [ ] Review user accounts, SSH keys, and local administrator access

6) Validate data isolation and key management

- [ ] Separate encryption keys by environment, tenant, or sensitivity class as needed
- [ ] Restrict who can use, administer, or rotate keys
- [ ] Confirm storage snapshots and backups follow the same access controls as live data
- [ ] Verify that sensitive data is not copied into shared logs or shared analytics systems
- [ ] Review secret storage and rotation practices
- [ ] Confirm temporary files, caches, and artifacts are handled securely

7) Validate observability without overexposure

- [ ] Ensure logs do not leak secrets, tokens, or customer data
- [ ] Limit access to monitoring dashboards, log archives, and tracing systems
- [ ] Confirm telemetry collection does not broaden the trust boundary unnecessarily
- [ ] Review whether agents require host-level privileges and whether that is acceptable
- [ ] Separate operational visibility by environment or tenant where needed
- [ ] Confirm alerting is in place for unusual access, configuration drift, and privilege escalation

8) Validate recovery and failure behavior

- [ ] Test how the isolation design behaves during patching and maintenance
- [ ] Test recovery from instance replacement, host changes, and automation failures
- [ ] Verify backup restore paths do not bypass access controls
- [ ] Confirm failover does not move workloads into a weaker trust boundary
- [ ] Validate that scaling and recovery do not reuse insecure templates or stale permissions
- [ ] Review whether incident response can contain the workload without impacting adjacent systems

9) Validate operational controls

- [ ] Confirm change management is in place for network, IAM, and image changes
- [ ] Review deployment pipelines for permission scope and artifact integrity
- [ ] Require peer review for changes affecting trust boundaries
- [ ] Track configuration drift against a known-good baseline
- [ ] Periodically recertify access for administrators and operators
- [ ] Document ownership for every isolation control

10) Decide whether the design is production-ready

Answer these questions before approving production use:

- [ ] Can a compromise of one workload meaningfully affect another workload without crossing an accepted boundary?
- [ ] Are identity, network, guest, and data controls aligned with the intended isolation model?
- [ ] Have failure, maintenance, and recovery scenarios been tested?
- [ ] Are shared services and shared admins limited to what is strictly necessary?
- [ ] Is the residual risk documented and accepted by the right owner?

Red flags that usually mean the design is too weak

- [ ] Multiple sensitivity levels share the same administrative domain without compensating controls
- [ ] The workload depends on broad security group rules or open routing paths
- [ ] Secrets, keys, or snapshots are accessible by more systems than intended
- [ ] A shared monitoring or automation stack has privileges across trust zones
- [ ] The design has not been tested during host replacement, rollback, or incident response
- [ ] The team cannot clearly explain the accepted blast radius

Quick approval checklist

Approve the design only if all of the following are true:

- [ ] The isolation boundary is explicit and documented
- [ ] The account, network, guest, and data controls match the workload risk
- [ ] Shared dependencies have been reviewed and minimized
- [ ] Recovery and maintenance procedures preserve the intended isolation
- [ ] Access is logged, reviewed, and tightly scoped
- [ ] Residual risk is understood and formally accepted

Notes section

Use this space to record implementation decisions, exceptions, and review outcomes.

- Workload name:
- Owner:
- Risk classification:
- Primary isolation boundary:

- Shared dependencies:
- Approved exceptions:
- Review date:
- Approver:

Tip: If you cannot explain the workload?s trust boundary in one sentence, the design is not ready for production.