CHECKLIST

Checklist: Securing AWS Virtual Machines with Nitro-Based Isolation

A practical checklist for evaluating Nitro-based isolation on AWS VMs, validating required controls, and preparing for production use.

Checklist: Securing AWS Virtual Machines with Nitro-Based Isolation

Use this checklist to evaluate whether Nitro-based isolation is the right fit for your AWS virtual machine workload and to verify the controls that still remain your responsibility.

1) Confirm the workload justifies stronger isolation

- [ ] The workload handles regulated data, customer records, credentials, or other sensitive material.
- [ ] A weaker shared-host trust model would materially increase risk for this workload.
- [ ] The team can explain what threat the isolation model is intended to reduce.
- [ ] The workload has a clear blast radius if compromised.
- [ ] Stronger isolation provides measurable value beyond standard security groups and IAM controls.

2) Verify platform and instance suitability

- [ ] The selected AWS instance family supports the expected Nitro-based isolation model.

- [ ] The planned instance generation is documented and approved.
- [ ] The guest operating system is supported on the chosen instance type.
- [ ] Any required CPU, memory, storage, or device features are compatible with the workload.
- [ ] Region-specific availability has been checked.
- [ ] The deployment design avoids unsupported combinations of instance type, OS, or features.

3) Validate identity and access controls

- [ ] IAM roles and policies follow least privilege.
- [ ] No long-lived credentials are stored on the VM unless strictly required.
- [ ] Secrets are delivered through a managed secrets mechanism or equivalent controlled process.
- [ ] Administrative access is restricted to named operators or approved automation.
- [ ] MFA or equivalent controls are required for privileged human access.
- [ ] Instance profiles are scoped narrowly and reviewed regularly.

4) Confirm guest hardening is still in place

- [ ] The operating system baseline is hardened according to policy.
- [ ] Unneeded services, ports, and packages are removed or disabled.
- [ ] Security updates and patching are performed on a defined schedule.
- [ ] Host-based firewall rules are present where appropriate.
- [ ] Disk encryption is enabled where required.
- [ ] Time synchronization, audit settings, and logging are configured.
- [ ] Administrative accounts are controlled and monitored.

5) Check network segmentation and exposure

- [ ] Security groups allow only required inbound and outbound traffic.
- [ ] Network ACLs, routing, and subnet placement support the intended isolation.
- [ ] Public IP exposure is avoided unless explicitly needed.
- [ ] Egress access is restricted to required destinations.
- [ ] The workload is separated from less trusted systems by network boundaries.
- [ ] Remote administration paths are documented and approved.

6) Review storage, backup, and recovery behavior

- [ ] EBS or equivalent storage is encrypted according to policy.
- [ ] Snapshot permissions are limited to approved principals.
- [ ] Backup schedules and retention settings meet recovery requirements.
- [ ] Restoration from backup has been tested successfully.
- [ ] Recovery procedures preserve the same security posture as production.
- [ ] Data loss tolerance and recovery time objectives are documented.

7) Validate monitoring and logging

- [ ] Cloud audit logs are enabled and retained.
- [ ] VM-level logs are collected centrally.
- [ ] Security-relevant events can be correlated across identity, network, and workload logs.
- [ ] Alerts exist for unauthorized access, configuration drift, and suspicious outbound activity.
- [ ] Monitoring covers both the guest OS and the surrounding AWS control plane.
- [ ] Log retention meets compliance and investigation needs.

8) Confirm the boundary and threat model are understood

- [ ] The team understands that Nitro-based isolation reduces exposure to the shared host layer.
- [ ] The team recognizes that guest compromise is still possible.
- [ ] The team understands that application security, secrets, and IAM remain critical.
- [ ] The architecture document states what the isolation model does and does not protect.
- [ ] Assumptions about the host, hypervisor, and tenant separation are explicitly documented.
- [ ] Any claims about isolation are backed by vendor documentation and internal verification.

9) Check compliance and evidence requirements

- [ ] Control ownership is assigned for identity, patching, logging, encryption, and recovery.
- [ ] Evidence exists for the selected instance family and configuration.
- [ ] The security review records what was validated before production.
- [ ] Compliance mappings are updated to reflect the new trust model.
- [ ] Audit materials avoid overstating the protection provided by the platform.
- [ ] Exceptions, if any, are approved and time-bound.

10) Test operational readiness before production

- [ ] Boot behavior works as expected in a nonproduction environment.
- [ ] Application startup, health checks, and service dependencies function correctly.
- [ ] Monitoring, alerting, and log collection are verified end to end.
- [ ] Backup and restore procedures are exercised successfully.
- [ ] Failure handling and incident response steps are documented.
- [ ] The production rollout plan includes rollback criteria.

11) Make the final go/no-go decision

Approve production only if all of the following are true:

- [ ] Nitro-based isolation provides a meaningful reduction in risk for this workload.
- [ ] Required instance and operating system support have been confirmed.
- [ ] Identity, network, storage, logging, patching, and recovery controls are in place.
- [ ] The team has tested critical operational workflows in a nonproduction environment.
- [ ] Evidence is sufficient for security review and compliance needs.
- [ ] The team can explain the new trust boundary clearly to operators and auditors.

Quick decision summary

Use Nitro-based isolation when you need stronger VM separation and a smaller trusted host surface, but only after validating the controls that still matter:

- Identity and access management
- Guest hardening and patching
- Network segmentation
- Encryption and secret handling
- Monitoring, logging, and recovery
- Documented evidence for compliance

If those controls are incomplete, the isolation model may improve the architecture but still leave the workload under-protected in practice.