CHECKLIST

Citrix ADC SSL Offload and Health Checks Readiness Checklist

A practical vendor-neutral checklist for validating SSL offload and application health checks in a Citrix ADC load balancing design before production use.

Citrix ADC SSL Offload and Health Checks Readiness Checklist

Use this checklist to validate a load balancing design that terminates TLS at the edge and uses health checks to protect production traffic. Mark each item complete before go-live.

1) Architecture and ownership

- [ ] The user-facing endpoint and VIP are clearly defined.
- [ ] The TLS termination point is documented and agreed by network, security, and application owners.
- [ ] The backend connection type is documented: HTTP, HTTPS, or re-encrypted traffic.
- [ ] The purpose of the load balancer is clear: traffic routing, TLS termination, health monitoring, or all three.
- [ ] Ownership of certificates, health checks, and rollback steps is assigned.
- [ ] The application team confirms the backend can operate correctly after TLS offload.

2) TLS offload validation

- [ ] The client-facing certificate matches the intended public hostname.

- [ ] The full certificate chain is present and trusted by clients.
- [ ] The private key is stored and handled securely.
- [ ] The configured TLS policy matches current security requirements.
- [ ] Legacy protocols and weak ciphers are disabled where appropriate.
- [ ] The load balancer presents the correct certificate during handshake.
- [ ] Certificate renewal and expiration monitoring are in place.
- [ ] A rollback plan exists if certificate deployment fails.

3) Backend request handling after offload

- [ ] The backend application knows whether requests arrive as HTTP or HTTPS.
- [ ] The correct forwarded protocol header is preserved or inserted.
- [ ] The original host name is preserved if the application depends on it.
- [ ] Client IP handling is documented and tested.
- [ ] Session cookies, redirects, and absolute links behave correctly after offload.
- [ ] Any required trust relationship between the load balancer and backend is configured.
- [ ] Re-encryption is used if internal traffic must remain encrypted.

4) Health check design

- [ ] The health check is more meaningful than a simple port open test.
- [ ] The check validates application readiness, not only network reachability.
- [ ] The probe path reflects a real user-facing dependency where possible.
- [ ] The probe returns a clear success and failure signal.
- [ ] The check does not depend on an overly complex chain of services.
- [ ] The health threshold and interval are tuned to avoid false positives and false negatives.
- [ ] The check is documented so operators know what it proves.

5) Recommended health check levels

Use the smallest check that still represents real user readiness.

- [ ] Transport check: confirms the backend listener is reachable.
- [ ] HTTP check: confirms a valid response code is returned.
- [ ] Content check: confirms a required string or response pattern is present.
- [ ] Application readiness check: confirms the service can actually process requests.
- [ ] Dependency-aware check: confirms critical dependencies such as authentication or database access are available, if appropriate.

6) Failure behavior and failover testing

- [ ] A single backend can be removed from service without impacting healthy users.
- [ ] Active sessions are preserved or drained according to design.
- [ ] A failed backend is removed from rotation quickly enough to protect users.
- [ ] Recovery occurs cleanly when the backend becomes healthy again.
- [ ] The team has tested a backend outage, a slow-response condition, and a health-check failure.
- [ ] Logs confirm the expected pool member state changes during failover.
- [ ] Metrics confirm traffic shifts to healthy nodes only.

7) Security and trust boundaries

- [ ] The trust boundary created by TLS termination is documented.
- [ ] Internal network exposure created by offload is understood and approved.
- [ ] Segmentation and access control are appropriate for backend traffic.
- [ ] Re-encrypted backend traffic is used where risk requires it.

- [ ] Security teams have reviewed certificate handling and key protection.
- [ ] Compliance requirements for TLS inspection or termination are satisfied.
- [ ] Any mutual TLS or end-to-end TLS requirement has been explicitly evaluated.

8) Operational monitoring

- [ ] VIP availability is monitored independently of backend health.
- [ ] Backend pool member status is visible to operators.
- [ ] Response times and error rates are tracked after offload.
- [ ] Health-check failures are alerting with enough context to act.
- [ ] Certificate expiry alerts are configured.
- [ ] Logs capture enough detail to troubleshoot certificate, header, and routing issues.
- [ ] Baseline performance is recorded before production rollout.

9) Production readiness review

- [ ] The design has been reviewed by network, application, and security stakeholders.
- [ ] The application works correctly with TLS terminated at the load balancer.
- [ ] The health check accurately reflects user readiness.
- [ ] Failure testing has been completed and documented.
- [ ] Certificate ownership, renewal, and rollback steps are documented.
- [ ] The rollout window and rollback window are approved.
- [ ] A go-live decision has been made based on evidence, not assumption.

10) Go-live signoff

Record the final validation results below.

Item	Result	Notes
TLS certificate validation		
Backend request handling		
Health check accuracy		
Failover test		
Monitoring and alerting		
Security review		
Rollback readiness		

Signoff

- [ ] Network owner approves
- [ ] Application owner approves
- [ ] Security owner approves
- [ ] Operations owner approves

Quick decision rule

Proceed to production only if all of the following are true:

- [ ] TLS terminates where the design intends.
- [ ] The backend application behaves correctly after offload.
- [ ] The health check detects real readiness, not just an open port.
- [ ] Failure testing shows unhealthy nodes are removed as expected.
- [ ] Monitoring, certificate management, and rollback are ready.

Notes

- Keep the check simple enough to avoid false alarms, but strong enough to detect real application failure.
- Revisit the design whenever the certificate model, backend application behavior, or security requirement changes.
- If the backend requires direct visibility of client TLS properties, reconsider whether offload is still the correct pattern.