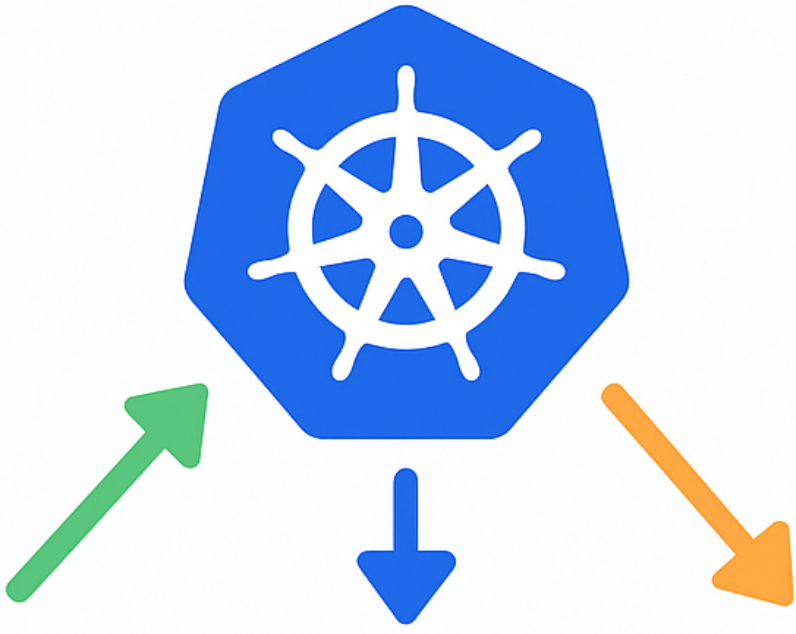
Kubernetes Container Health Checks

Master Guide for Production-Ready Applications

Startup Liveness Readiness

Swipe for advanced insights →

What Are Container Health Probes?



Automatic Recovery

Traffic Management

Self-Healing Systems

99.9% uptime achievable with proper health checks

Three Types of Probes

STARTUP PROBE

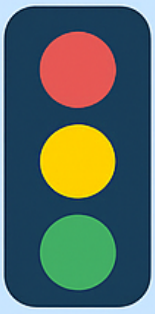
Detects when app has initialized

During startup phase only

LIVENESS PROBE

Checks if container is running

Throughout lifecycle



READINESS PROBE

Ready to accept traffic

Parallel with liveness

Probe Mechanisms Deep Dive

HTTP GET

`httpGet:` {path: "/healthz, port 8080}

Success criteria 200-399 status cdes

TCP Socket

`tcpSocket:` {port: 3306}

Connection established

exec Command

`exec:` {`command:` {"cat", '/tmp/healthy"}

Exit code 0

Formula: Max startup time = failureThreshold × periodSeconds

Master These 5 Parameters

Parameter	Purpose
initialDelaySeconds	Startup buffer
periodSeconds 5–10s	Check frequency 5–10s
timeoutSeconds 3–5s	Response timeout 3–5s
failureThreshold 3–5	Tolerance level 3–5
successThreshold 1–2	Recovery needed 1–2

💡 Formula: Max startup time = failureThreshold x periodSeconds

Production-Grade Best Practices

Lightweight Checks

Keep probes resource-efficient

Appropriate Thresholds

Balance speed vs stability

All Three Types

Implement complete coverage

Test Under Load

Verify performance impact

Separate Concerns

Liveness ≠ Readiness logic

Monitor & Alert

Track probe success rates

⚠️ **Avoid:** Dependency checks in liveness probes

Production Web App Configuration

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: web-application
spec:
  -contatυpProbe
    name: web-app
    image: nginx:latest
    startupProbe:
      httpGet:
       path: /startup
      failureThreshold: 30
      periodSeconds: 2
    livenessProbe
      httpGet:
       path: /healthz
      initialDelaySeconds:10
      periodSeconds: 5
    readinessProbe
      httpGet:
       path: /ready
      initialDelaySeds: 5
```

Startup Probe

Liveness Probe

Readiness Probe

Debug Like a Pro

- ✅ kubectl describe pod <name>
  – Check probe status
- ✅ kubectl logs <pod>
  – Review application logs
- ✅ Test endpoints manually in container
- ✅ Verify probe timing vs app startup
- ✅ Check resource constraints
- ✅ Validate network connectivity

🔥 **Top 3 Issues:**

- Timeout too low
- Missing startup probe
- Wrong endpoint

Quick fix: Increase failureThreshold first!

Advanced Health Check Patterns

spec readinessGates:

redisensGates

Custom external conditions

```
spec:
readinessG
Reos:
 httpGet:
  path:
  port:8080
```

Readiness Gates

Custom external conditions

```
spec:
  readinessGates:
  - conditionType: "mg/feature"
```

Multi-Container Pods

Independent probe strategies

```
livenessProbe:
  httpGet:
    path: /healtz
    port:8080
```

Service Mesh Integration

Envoy sidecar health

Monitor probe overhead – target <1% CPU usage

LEVEL UP YOU K8s GAME!

Save this post for reference

Share with your DevOps team

Comment your biggest health probe challenge

Follow for more K8s deep dives

78% fewer production incidents with proper health checks

Build resilient systems that heal themselves 🚀