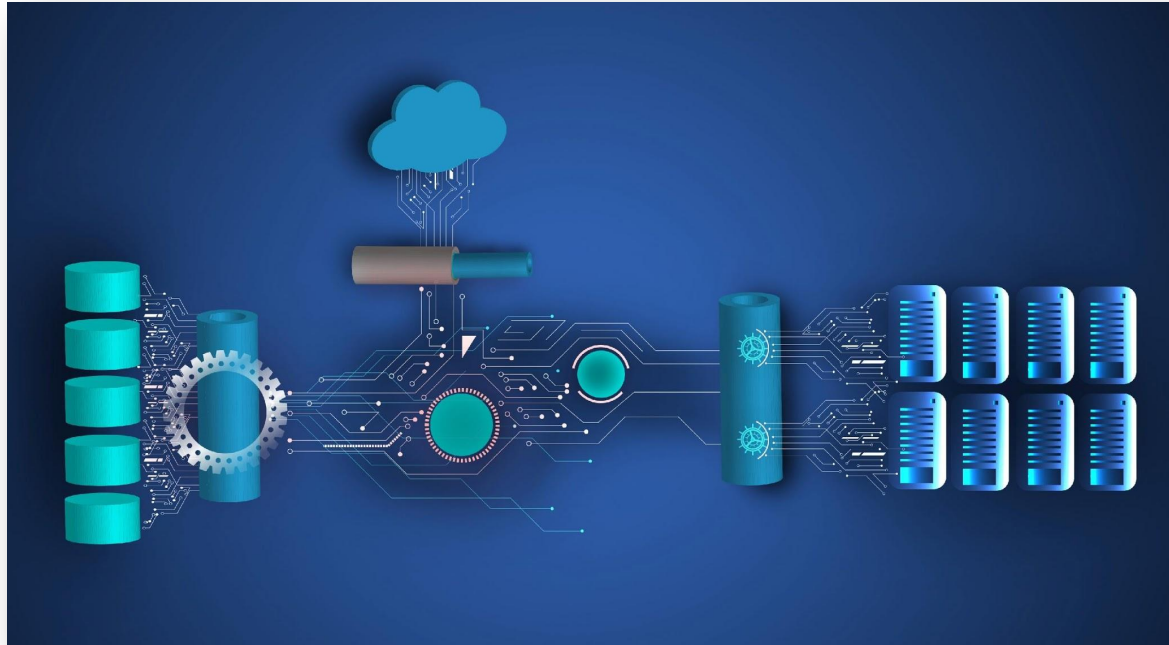


PRODUCTION-READINESS MICROSERVICES CHECKLIST



Microservices requires discipline, new toolsets and a change in team dynamics that envelopes all aspects of an organization but the adoption does improve the efficiency of an organization over time, enabling a faster delivery and reaction than any other systems.

In this guide, we will walk you through essential checklist which we will help you with consider the possible adoption of microservices. There are some common symptoms related to monolith applications which usually triggers the adoption of Microservices, read it thoroughly to see if you can locate some of them in your project.

This will be a checklist to run over all microservices—manually or in an automated way.

SAMPLE CHECKLIST

PRODUCTION-READY SERVICE IS STABLE AND RELIABLE

CHECKLIST 1

- It should have a standardized development cycle.
- The code should be thoroughly tested through lint, unit, integration, and end-to-end testing.
- The test, packaging, build, and release process should be completely automated.
- It should have a standardized deployment pipeline, containing staging, canary, and production phases.
- Their clients should be known.
- The dependencies should be known, and there are backups, alternatives, fallbacks, and caching in place in case of failures.
- It should have stable and reliable routing and discovery in place.
- A Production-Ready Service Is Scalable and Performant
- Its qualitative and quantitative growth scales are known.
- It should use hardware resources efficiently.
- Its resource bottlenecks and requirements should be identified.
- Capacity planning should be automated and performed on a scheduled basis.
- Its dependencies will scale with it.
- It should scale with its clients.
- Its traffic patterns should be understood.
- Traffic should be re-routed in case of failures.
- It should be written in a programming language that allows it to be scalable and performant.
- It should handle and process tasks in a performant manner.
- It should handle and store data in a scalable and performant way.

PRODUCTION-READY SERVICE IS FAULT TOLERANT AND PREPARED FOR ANY CATASTROPHE

CHECKLIST 2

- It should have no single point of failure.
- All failure scenarios and possible catastrophes should have been identified.
- It is tested for resiliency through code testing, load testing, and chaos testing.
- Failure detection and remediation should be automated.
- There are standardized incident and outage procedures in place within the microservice development team and across the organization.

PRODUCTION-READY SERVICE IS PROPERLY MONITORED

CHECKLIST 3

- Its key metrics should be identified and monitored at the host, infrastructure, and microservice levels.
- It should have appropriate logging that accurately reflects the past states of the microservice.
- Its dashboards should be easy to interpret and contain all key metrics.
- Its alerts should be actionable and are defined by signal-providing thresholds.
- There is a dedicated on-call rotation responsible for monitoring and responding to any incidents and outages.
- There is a clear, well-defined, and standardized on-call procedure in place for handling incidents and outages.

PRODUCTION-READY SERVICE IS DOCUMENTED AND UNDERSTOOD

CHECKLIST 4

- It should have comprehensive documentation.
- Its documentation should be updated regularly.
- Its documentation should contain a description of the microservice; an architecture diagram; contact and on-call information; links to important information; an onboarding and development guide; information about the service's request flow(s), endpoints, and dependencies; an on-call runbook; and answers to frequently asked questions.
- It should be well understood at the developer, team, and organizational levels.
- It should adhere to production-readiness standards and meets the associated requirements.
- Its architecture should be reviewed and audited frequently.