# Health Checks

On both Cloud Foundry and Kubernetes, it is possible to provide two separate endpoints for liveness checks ("are you alive?") and readiness checks ("are you ready for more requests?"). A failure on the former leads to a restart, whereas a failure on the latter temporarily takes the app instance out of the request dispatching rotation until a subsequent readiness probe is successful.

↳ *Learn more about health checks on Cloud Foundry.*

↳ *Learn more about health checks on Kubernetes.*

> **Limited support for readiness checks on CF**
>
> Although supported by the Cloud Foundry core, readiness checks are not yet supported by the Cloud Foundtry CLI as well as the Cloud MTA Build Tool (MBT).

The Node.js runtime provides an out-of-the-box endpoint for liveness and readiness checks at `/health`. Requests that reach this public endpoint are answered with the status code 200 and the body `{ status: 'UP' }`.

For CAP Java, the `cds add` facets `mta`, `cf-manifest`, `helm`, and `helm-unified-runtime` add the necessary dependencies and configuration for the publicly available probe endpoints `/actuator/health/liveness` and `/actuator/health/readiness`. For more details on adding Spring Boot health checks to your CAP Java app, please see Spring Boot Health Checks.

For deployment to Kyma/ Kubernetes, `@sap/cds-dk` adds the necessary configurations to Helm charts to make the setup complete. Once supported, `@sap/cds-dk` will also add the necessary configurations to MTA deployment descriptors.

> **TIP**

> Don't forget to adjust the values in your deployment descriptors in case you add different Spring Boot health check endpoints or use a *fully custom* `server.js` .

↳ *Learn more about availability checks in Node.js.*

↳ *Learn more about availability checks in Java.*

Edit this page

Last updated: 25/02/2025, 10:05

Previous page
Custom Builds

Next page
Multitenancy

Was this page helpful?

👍 👎