The app executes as one or more stateless processes (e.g., master/ workers) that share nothing. Any necessary state is externalized to backing services (cache, object store, etc.).

The correct answer is: Processes

Continuous delivery and deployment are enabled by keeping development, staging, and production environments as similar as possible.

The correct answer is: Dev/prod parity

Each deployable app is tracked as one codebase tracked in revision control. It may have many deployed instances across multiple environments.

The correct answer is: Codebase

The app is self-contained and exports any/all services via port binding (including HTTP).

The correct answer is: Port binding

Configuration, or anything that is likely to differ between deployment environments (e.g., development, staging, production) is injected via operating system-level environment variables.

The correct answer is: Config

Robustness is maximized via processes that start up quickly and shut down gracefully. These aspects allow for rapid elastic scaling, deployment of changes, and recovery from crashes.

The correct answer is: Disposability

The stages of building a deployable app artifact, combining that artifact with configuration, and starting one or more processes from that artifact/configuration combination, are strictly separated.

The correct answer is: Build, release, run

An app explicitly declares and isolates dependencies via appropriate tooling (e.g., Maven, Bundler, NPM) rather than depending on implicitly realized dependencies in its deployment environment.

The correct answer is: Dependencies

Concurrency is usually accomplished by scaling out app processes horizontally (though processes may also multiplex work via internally managed threads if desired).

The correct answer is: Concurrency

Backing services, such as databases or message brokers, are treated as attached resources and consumed identically across all environments.

The correct answer is: Backing services