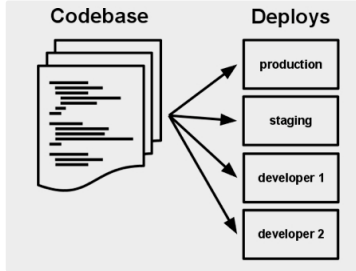
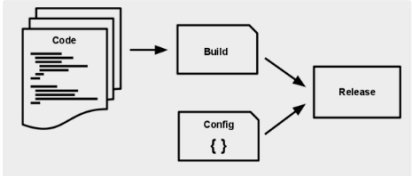
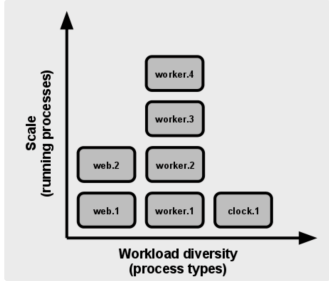


## Cloud Native Ready App - 12 Factor App

No.	Name	Descriptions
1.	Codebase	<ul style="list-style-type: none"><li>• "One codebase tracked in revision control, many deploys"</li><li>• Gunakan 1 codebase untuk 1 service / aplikasi, yang di track oleh version control system seperti Git.</li><li>• Hanya ada 1 codebase untuk setiap aplikasi / service, tetapi bisa terdapat beberapa deployment.</li><li>• Codebase sama pada keseluruhan deployment environment tetapi mungkin memiliki versi yang berbeda pada setiap deployment.</li></ul> 
2.	Dependencies	<ul style="list-style-type: none"><li>• "Explicitly declare and isolate dependencies"</li><li>• Dependency aplikasi terhadap modul atau library lain harus di deklarasikan secara eksplisit dan terisolasi sehingga tidak mempengaruhi aplikasi yang lain.</li><li>• Contoh:<ul style="list-style-type: none"><li>◦ Pada Python dependency di deklarasikan pada file requirements.txt dan diinstall menggunakan pip pada environment yang terisolasi virtualenv sehingga tidak mempengaruhi aplikasi yang lain.</li><li>◦ Pada NodeJS List dependency di deklarasikan pada file package.json dan diinstall menggunakan NPM.</li></ul></li></ul>
3.	Config	<ul style="list-style-type: none"><li>• "Store config in the environment"</li><li>• Semua parameter konfigurasi harus disimpan pada environment variable, tidak ada lagi konfig parameter yang di hardcode pada kode aplikasinya.</li><li>• Hal ini memudahkan apabila terjadi perubahan konfig parameter tidak perlu dilakukan perubahan dari sisi kode cukup melakukan redeployment sesuai dengan environment variable yang digunakan.</li></ul>
4.	Backing Services	<ul style="list-style-type: none"><li>• "Treat backing services as attached resources"</li><li>• Perlakukan service pendukung sebagai resource tambahan yang terpisah dari sistem aplikasi.</li><li>• Contoh<ul style="list-style-type: none"><li>◦ Pada database ketika menggunakan database MySQL lokal, database tersebut harus dapat dengan mudah diganti menggunakan MySQL yang disediakan pihak ketiga</li></ul></li></ul>

		<p>seperti Amazon RDS atau Google Cloud SQL hanya dengan merubah konfigurasi parameternya saja.</p> <ul style="list-style-type: none"> <li>◦ Email service dapat dengan diubah menggunakan service dari smtp server manapun hanya dengan merubah konfigurasi parameternya saja.</li> <li>• Semua perubahan service pendukung dapat dilakukan tanpa melakukan perubahan code.</li> </ul>
5.	Build, release, run	<ul style="list-style-type: none"> <li>• “Strictly separate build and run stages”</li> <li>• Proses build dan run dilakukan terpisah sesuai dengan release version yang akan di jalankan.</li> <li>• Setiap perubahan yang akan direlease harus membuat release version yang unik dan baru. Contoh v1.0.1</li> <li>• Release version juga dapat digunakan untuk mengidentifikasi sistem sebelumnya ketika ingin melakukan rollback ketika terdapat kegagalan sistem pada release version yang terakhir.</li> <li>• Proses build dilakukan setiap code akan di deploy.</li> <li>• Eksekusi runtime berjalan secara otomatis menggunakan build release version aplikasi yang sesuai, sehingga saat server reboot atau scaling up eksekusi runtime dapat berjalan otomatis.</li> </ul> 
6.	Processes	<ul style="list-style-type: none"> <li>• “Execute the app as one or more stateless processes”</li> <li>• Aplikasi berjalan sebagai 1 atau lebih stateless proses, dengan adanya stateless proses, request dari user, dan request selanjutnya kemungkinan akan dikerjakan oleh proses aplikasi yang berbeda.</li> </ul>
7.	Port Binding	<ul style="list-style-type: none"> <li>• “Export services via port binding”</li> <li>• Aplikasi harus berjalan mandiri dapat diakses langsung melalui port tanpa memerlukan server web terpisah seperti Apache, Nginx, Tomcat untuk menjalankannya.</li> </ul>
8.	Concurrency	<ul style="list-style-type: none"> <li>• “Scale out via the process model”</li> <li>• Metode ini mendorong aplikasi menjalankan beberapa proses yang lebih kecil dan terisolasi daripada penggunaan thread atau worker dalam satu proses yang besar.</li> </ul> 
9.	Disposability	<ul style="list-style-type: none"> <li>• “Maximize robustness with fast startup and graceful shutdown”</li> <li>• Kemampuan aplikasi untuk dengan mudah memulai (startup) dan berhenti (shutdown) dengan cepat dan bersih.</li> <li>• <b>Fast Startup:</b> Proses dalam aplikasi harus dapat dimulai dengan cepat.</li> </ul>

		<ul style="list-style-type: none"> <li>• <b>Graceful Shutdown:</b> Proses harus mendukung shutdown yang bersih (graceful shutdown). Proses harus bisa menyelesaikan tugas yang sedang berjalan dan melepaskan sumber daya (seperti koneksi database) dengan benar sebelum berhenti.</li> <li>• <b>Handling SIGTERM and SIGINT:</b> Proses harus mengelola sinyal penghentian seperti SIGTERM dan SIGINT dengan benar, memastikan mereka menyelesaikan pekerjaan yang sedang berjalan dan membersihkan sumber daya sebelum berhenti.</li> <li>• <b>Stateless Processes:</b> Menghindari penyimpanan status dalam memori proses, status dapat disimpan pada database atau cache yang terdistribusi.</li> </ul>
10	Dev / prod parity	<ul style="list-style-type: none"> <li>• “Keep development, staging, and production as similar as possible”</li> <li>• Sebisa mungkin environment dev, staging, production dibuat seidentik mungkin tanpa terlalu banyak perbedaan.</li> <li>• Semua environment harus menggunakan stack yang sama, dan juga versi library yang sama.</li> <li>• Dengan menggunakan Kontainer seperti docker menjaga aplikasi berjalan dengan konsisten pada baik di laptop developer maupun server production.</li> <li>• Secara teratur menguji aplikasi di environment yang mendekati production seperti environment staging, untuk memastikan aplikasi akan berfungsi baik ketika di deploy.</li> <li>• Dengan kesamaan environment, dapat membantu mendeteksi masalah sejak dini selama pengembangan maupun staging.</li> </ul>
11.	Logs	<ul style="list-style-type: none"> <li>• Treat logs as event streams</li> <li>• Logs adalah catatan dari berbagai event yang terjadi di dalam aplikasi selama waktu tertentu.</li> <li>• Logs harus terus mengalir sebagai aliran event yang berkelanjutan.</li> <li>• Logs harus informatif, memberikan informasi yang akurat terkait event yang terjadi.</li> <li>• <b>Log</b> Aplikasi dialirkan ke <b>stdout</b> tanpa disimpan sebagai file yang perlu pengelolaan untuk rotasi file dan pembersihan file</li> <li>• Tools yang menangani pengumpulan dan penyimpanan log dapat menggunakan <b>Logstash</b>, atau logging yang dikelola cloud seperti AWS cloudwatch, Azure monitor, Google cloud logging.</li> <li>• Aplikasi hanya bertugas menghasilkan log terpisah dari log management.</li> </ul>
12.	Admin Processes	<ul style="list-style-type: none"> <li>• “Run admin/management tasks as one-off processes”</li> <li>• <b>Admin Processes</b> adalah tugas-tugas administratif atau manajemen yang biasanya dilakukan sekali atau hanya sesekali, seperti migrasi basis data, menjalankan skrip perbaikan, atau tugas administratif lainnya.</li> </ul>

- |  |  |
|--|--|
|  | <ul style="list-style-type: none"><li>• Proses admin harus dijalankan sebagai proses satu kali (one-off processes). Proses ini seharusnya tidak menjadi bagian dari siklus hidup aplikasi yang berjalan terus-menerus.</li><li>• Menjalankan database migrations dapat menggunakan migrations script atau menjalankan one-time scripts. Script dapat dijalankan secara manual atau dalam pipeline CI/CD.</li></ul> |
|--|--|

Reference:

1. [The Twelve-Factor App \(12factor.net\)](https://12factor.net/)