技術の概要

1. コンテナ開発（Docker）

インフラ、DevOps関連技術

コンテナ型の仮想環境を作成、配布、実行するプラットフォーム

・仮想マシンとの比較

　仮想マシンはホストマシン上でハイパーバイザを利用しゲストOSを操作させる。

　Dockerはホストマシンのカーネルを利用し、プロセスやユーザを隔離し別マシンが動作しているように見せかける技術。

　特徴：軽量、高速起動/停止

　※関連

　　オーケストレーションツール：Kubernetes

　　　Docker等のコンテナを複数管理するための機能を多く提供しているツール

　　CIツール：Jenkins、CircleCI

　　　継続的インテグレーション。ソフトウェアのビルド、テスト、デプロイを自動化。

　　環境テストツール：Serverspec（Ruby）

　　画面試験ツール：Selenium

　　上記のツールを用いて、開発・リリースサイクルを安定させることが目的となる。

1. パブリッククラウド（AWS、GCP）
2. Git

1. フロントエンドフレームワーク

TypeScript

　JavaScriptの強化版。静的型付け言語

※CoffeeScriptと選択

Angular.js

　　　Google社製。MVW（MVC拡張）の考え方。フルスタックフレームワーク

React.js

Facebook社製。コンポーネント指向。すべてのWebアプリケーションに導入できる。仮想DOMで高速動作。

[Next.js](https://nextjs.org/)

　Reactのフレームワーク。

Vue.js

　学習コストが低い。開発規模に影響を受けづらい。

Nuxt.js

　Vue.jsのフレームワークで、Vue.jsの弱点であるSSR（サーバサイドレンダリング）に対応したフレームワーク。

1. Go

処理が非常に高速。複雑なシステム構築に最適。並行処理が可能。

1. Java
2. Node.js

リアルタイム処理の実行効率性が高い。

1. Python
2. PHP
3. Ruby on Rails
4. C#.NET
5. VB.NET
6. Windows
7. Linux
8. MySQL
9. PostgreSQL
10. Azure
11. DynamoDB
12. Ansible

オーケストレーションツール。

1. Vagrant
2. SQLServer
3. Redis
4. Memcached
5. Oracle
6. Zabbix
7. VMware
8. Active Directory
9. Junit
10. Firebase

GCPの機能の一つ。モバイル向けアプリケーション開発において、バックエンドサービスとして、SNSアカウントログインやチャット機能がすでに構築されたものを利用することが出来る。

1. Asana

プロジェクト管理ツール。ガントチャート。

1. DevOps

比較されるのはアジャイル開発。

DevelopmentとOperationsを合わせた言葉。開発Gと運用Gが協力しあってより良い製品を開発することを目標としている。重要と感じる部分がそれぞれのグループで異なるため意見の対立が発生することが多いが、大枠でみてソフトウェア製品を提供するうえでの目標は同じはずなので、その協力する重要性と手法を説いた考え方。