Cloud Native Development

1. Loosely Coupled Microservices
2. Polyglot Tools & Technology
3. Agile with Devops
4. APIs at the core
5. Powered by Opensource
6. Container adoption
7. Separation of Stateless & Stateful
8. No OS dependencies
9. Highly automated
10. Serverless architecture

Cloud native computing is an approach in software development that utilizes cloud computing to its fullest due to its use of an open source software stack to deploy applications as microservices. [wiki]

TRUST	Uptime, recovery, scale independtly
FREEDOM	Run any where without vendor lock-in
SPEED	Work on app fast , not on infra

- Deployability
- Testability
- Disposability
- Observability
- Reliability
- Agility

Cloud Native Adoption

NETFLIX	UBER	WECHAT
600+ service in production	1000+ service in production	300+ service in production
Deployment 100 times a day	Deployment 1000 times a week	Deployment 1000 times a day

Candidates for Cloud Native

Source: Microsoft

1. A large, strategic enterprise system that needs to constantly evolve business capabilities/features
2. An application that requires a high release velocity - with high confidence
3. A system where single feature must release *without* a full redeployment of the entire system
4. An application developed by teams with expertise in different technology stacks
5. An application with components that must scale independently

BACKING SERVICES

- Monitoring
- Security
- Storage Services
- Streaming Services
- Relational Databases
- NoSQL
- Caching
- Analytics

CLOUD NATIVE COMPUTING FOUNDATION

Principles

- Fast is better than slow
- Open
- Fair
- Strong Technical Identity
- Clear Boundaries
- Scalable
- Platform Agnostic

With draw.io

Dzone

thoughts2share

By @simplymanas