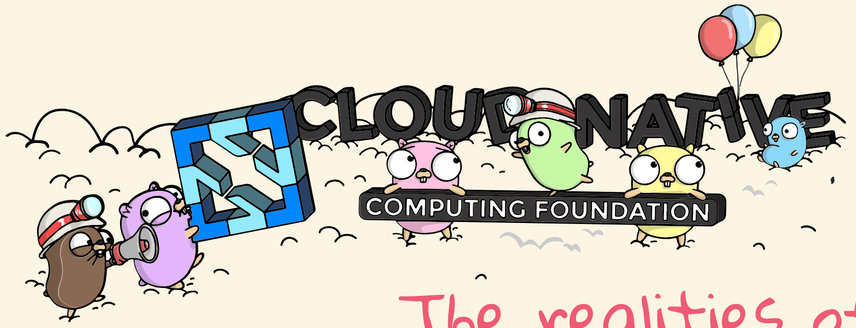


DevOps Tooling Hell (Avoiding it)

&

The realities of the Cloud-Native Landscape for Newbies

The Big Picture of DevOps

Delivering software faster, save cost while doing it, accelerate the delivery of business value ahead of competition, grab larger market share, responding quicker to customer demands, unpredictability in the market & the regulatory environment.

Speed

+

Money

A very important aspect of speed, is to make sure you're racing in the right direction (You must make sure, the applications been delivered to customers are resilient to failure, always available & secure

DevOps is just a single pillar of a much bigger blueprint of building modern applications which is Cloud Native

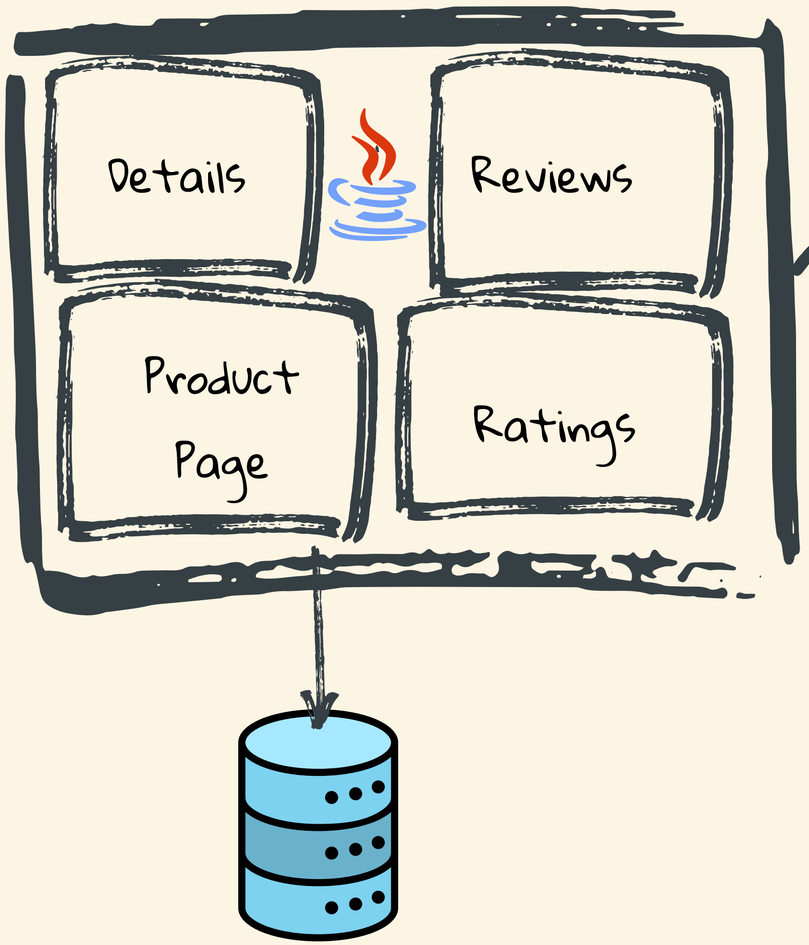
github.com/fiduh
@_as_code

Cloud-Native Landscape

One of the core pillars of Cloud Native is Application Architecture

CLOUD NATIVE LANDSCAPE

Monolithic Application



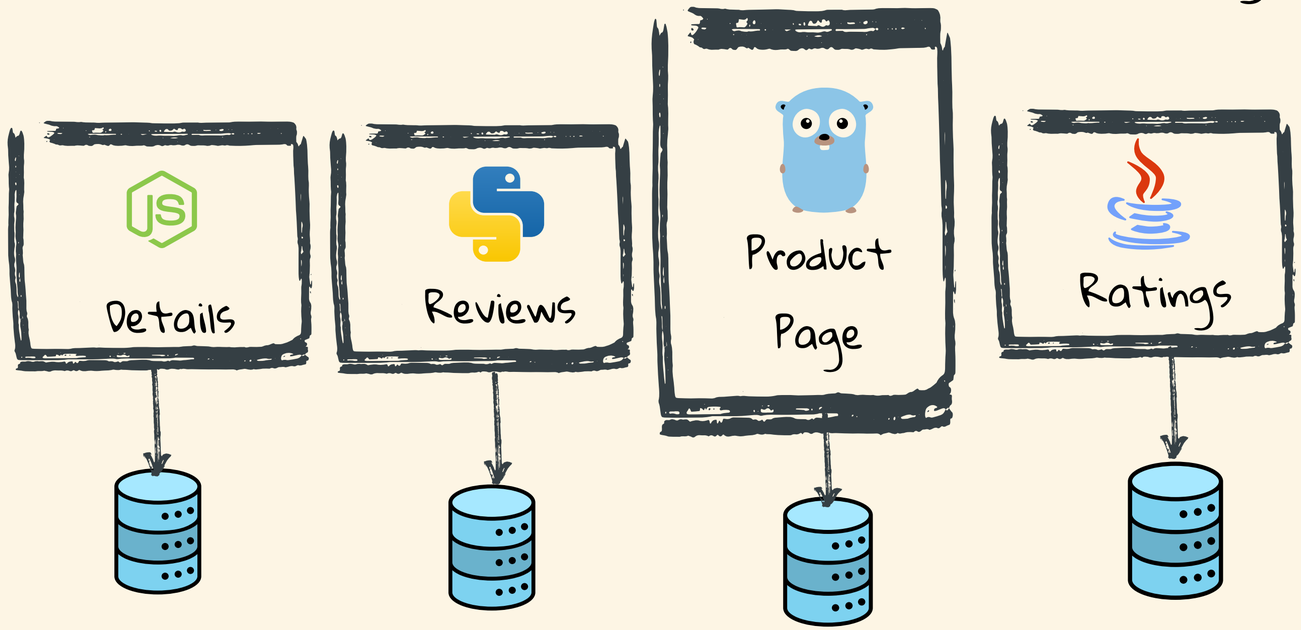
Traditional way of building Applications

You have Different modules as part of a whole application, Mostly running on a single process & a database.

This has it's downsides: Bugs in one module could easily cascade into the entire application, it is difficult to scale only a single module of the application, teams have to use the same language.

Shift to Micro-services

Functionalities of a large application broken down as smaller independent services, they can be written in different languages, scaled & deployed independently.



Micro-services Application

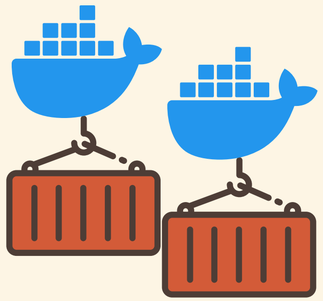
This shift to Micro-services gave rise to a whole new level of complexity been added to building applications. Most Cloud Native tools are mainly trying to solve these complexities.

Note: Emphasis should be on the problems these tools are solving & not necessarily the tools themselves. These tools are open source & Cloud agnostic (Implementation on any cloud platform are similar).

Containers, Container Orchestration

(another core pillar of Cloud Native)

Docker



Micro-services are packaged in lightweight components that contain everything they need to run in any environment. Docker is a tool for this.

These Container images have to be saved somewhere, Container Registry like Harbor perform this task.

Control where Containers run

Detect failure & Repair

Balance load between microservices

As the number of containers grow, the process of controlling where they run, scaling & destroying them needs to be automated (Orchestration) & Kubernetes comes to the rescue.

DevOps (another core pillar of Cloud Native)

Repetitive tasks in application development & deployment needs to be automated, considering the numerous services that are now been developed, deployed & scaled independently. This is where DevOps comes in, this is a development practice that emphasizes collaboration, communication, & automation between development & operations teams. it's main component is CI/CD. Tools like GitLab CI, ArgoCD, gets the job done.

Continuous integration (CI): is the practice of regularly merging code changes into a shared repository & running automated tests to ensure that the code is working as expected.

Continues Delivery (CD): is the practice of automating the deployment of software to production environments, often through the use of automated deployment pipelines.

While Cloud-Native at its core is more concerned about building applications that can scale, that are highly available, reliable and secure. DevOps as one of it's core pillars is more concerned about how fast you can get these applications in the hands of customers in a reliable way. This is why there's so much emphasis on automating repetitive tasks & a need for seamless collaboration between development & operations teams.

Cloud Native Open Standards (another core pillar of Cloud Native)

In order for IT Organizations & engineers not to keep reinventing the wheel in solving these Micro-service architecture challenges, a bunch of Cloud-Native Open Standards emerged.

Service Mesh

Istio

A major challenge is how micro-services communicate with each other & doing so securely. Istio is an open source implementation of application networking (Retries, Timeouts, Circuit breaking, Client-side Load Balancing)

Observability

The main purpose of observability is to better understand the internal workings of your infrastructure/application, speed up troubleshooting, detect hard to catch problems, & monitor performance of an application

3 main pillars of observability

Logging Metrics Traces

fluentd

Prometheus

JAEGER

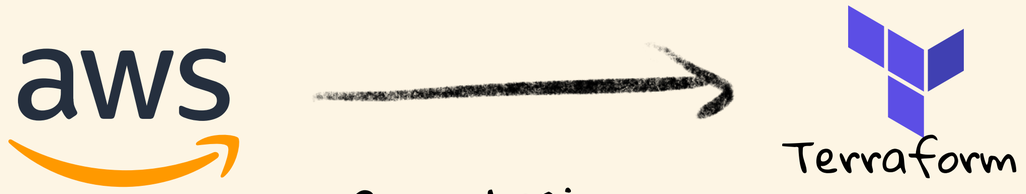
Traces allows you to follow individual request as it flows through the system, it gives much better understanding on how everything fits together in your application. Jaeger is the tool for this.

Logs are records of events that have occurred in your system. Elasticsearch, Fluentd, Kibana gets the job done

Metrics on the other hand provides information about the state of the system using numerical values. Prometheus is a tool for collecting metrics

Cloud Platform & Infrastructure as Code (IaC)

Applications & the various cloud native tools needs a place to run, the various cloud platforms (e.g. AWS, GCP, AZURE) provides hardware and software components which form the virtualized compute, storage, & networking infrastructure that we can provision via self-service APIs. Automating the process of provisioning resources in the cloud is also very important. Terraform solves this problem by implementing (IAC) Infrastructure as Code.



Conclusion

Note: While it still looks like these are lots of tools to learn, most of them are implemented in domain specific languages like YAML, HCL & JSON & they're mostly declarative. Tools like Packer, Terraform & Vault are configured using HCL. Kubernetes, Istio, etc are configured using YAML.

So picking up on the syntax is actually easy, the difficulty lies in understanding & integrating the various tools.

An easier way of approaching this, is by identifying the problem you want to solve, look out for CNCF graduated tools tailored to solve such problems. The more popular a tool is, the more resources you will find to learn and implement such a tool.