

Cloud Native Backend

Presentation by Yash Bhamare



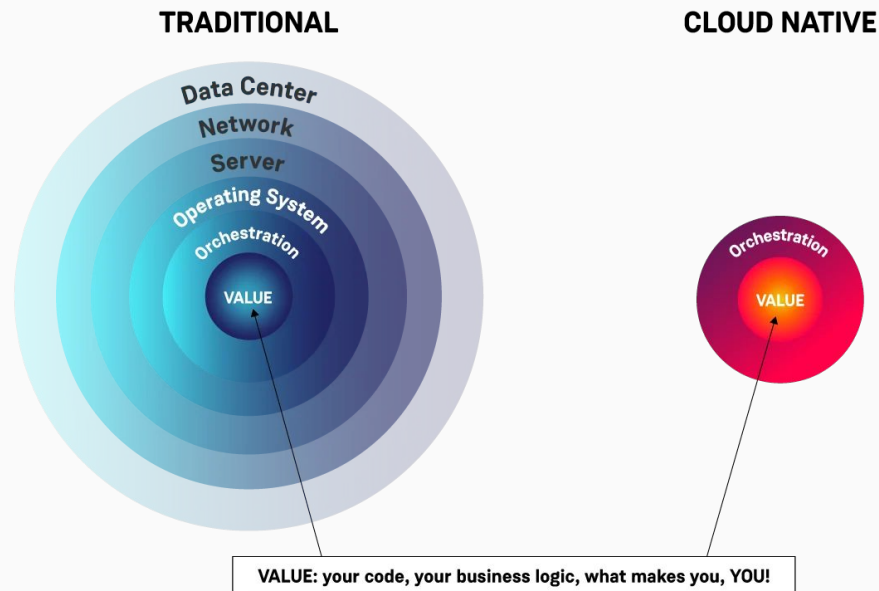
What is this topic?

Any general software system has 2 parts:

Front End - where the user or client interacts with the system

Back End - where the data is processed using CPUs, Network and Storage resources.

In cloud native architecture, the backend part is handled by the container orchestrator i.e. Kubernetes.



CPUs

Kubernetes uses vCPU.

vCPU : logical entity based on our physical entity.

How to calculate the number of vCPUs?

Formula: $(\text{Threads} \times \text{Cores}) \times \text{Physical CPU} = \text{Number vCPU}$

Example: we'll select Intel Xeon E-2288G as the underlying CPU. Key stats for the Intel Xeon E-2288G include 8 cores / 16 threads with a 3.7GHz base clock and 5.0GHz turbo boost. There is 16MB of onboard cache.

$(16 \text{ Threads} \times 8 \text{ Cores}) \times 1 \text{ CPU} = 128 \text{ vCPU}$

CPU's

Choice of cpu infra: **Bare-metal servers** vs **Vms given by Cloud Providers**

	PROs	CONs
Bare-metal servers:	Highly customisable	Highly complex and costly to maintain
VMs given by Cloud Service Providers:	Easy configuration and deployment	Vendor lock-in Less customisation

Storage

Kubernetes - universal deployment, highly portable, highly scalable!
but this can be **disadvantageous!**

Why? It is not stateful!

If an application is made stateful, then portability takes hit as the application needs a particular storage solution.

If a choice is made to go with particular solution, there are just too many options:

<https://landscape.cncf.io/card-mode?category=cloud-native-storage&grouping=category>

Storage

Solutions?

1. Lets use a database.
 - A. Pick out a database solution that fits your needs.
 - B. Containerize it to run on local disk,
 - C. Deploy it in your cluster as just another workload.

Issues: Spinning up and down of containers becomes slow due heavy disk usage.

Storage

Solutions?

2. Native Kubernetes support

Static provisioning

A. Persistent Volumes (PV) - are storage units independent of pods

B. Persistent Volume Claims(PVC) - are requests for the PVs for specific pods

Dynamic provisioning

C. Storage Classes - multiple profiles of storage are created. When a PVC is done, then the profile matching the requirement is assigned to the pod.

Storage

Solutions?

3. Open Source options:

A. Ceph

B. Rook

C. CSI – Container Storage Interface



QUESTIONS?