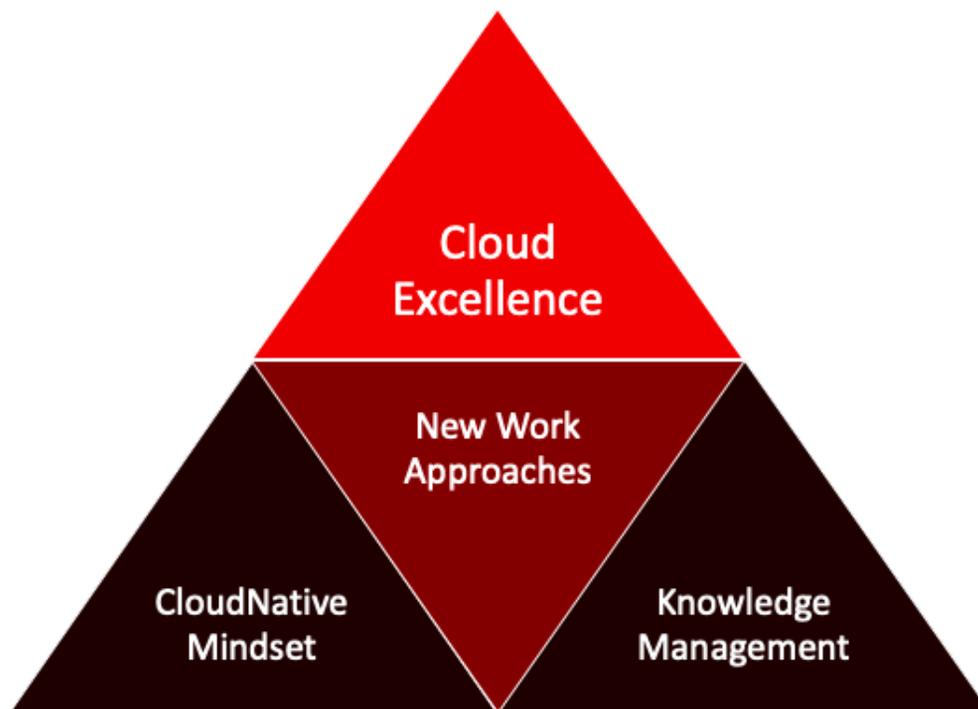


How to approach Cloud properly

The idea of relying onto Lift & Shift-approaches when considering the move into cloud environments is simple. And it is plain wrong and (financially) risky. And it will most likely be a disappointment in regard to scaling, availability and performance. Instead, one must think completely different of Cloud to actually gain an advantage. Let me explain and let me show you the pillars required for a successful cloud experience.



Kontakt

Karsten Samaschke
karsten.samaschke@cloudical.io
www.cloudical.io

CONTENTS

- 3 Lift & Shift into a cloud environment
- 3 #1: CloudNative. By Design.
- 5 #2: CloudNative. By Approach.
- 5 Automation and Versioning
- 6 Agile Teams, DevOps and Iteration
- 7 #3: CloudNative. By mindset.
- 8 #4: New Work
- 8 #5: Knowledge Management
- 9 The sum of things: CloudExcellence

Cloud is often understood as a way of thinking of an infrastructure: automated, operated by a cloud provider, easy to set up. While this is true and while it could definitely be the right decision to move to public or private cloud providers, it gives the wrong impression: if you move into a cloud environment only by executing Lift & Shift approaches, you might perhaps save some money on the infrastructural side and perhaps some time on the provisioning side – but you simply exchange one datacenter operator (yourself or your current one) by another, very generic, one (Microsoft, Amazon, Google, Digital Ocean, etc.).

Lift & Shift into a cloud environment



Logos of Hyperscalers

In fact, quite often you do not even save money, since the overwhelming number of offerings and the reduced amount of customization can cause lacks of transparency and might even lead to higher operational costs, as the cloud environments are typically operated on an infrastructural level only, leaving management and operations in your hands.

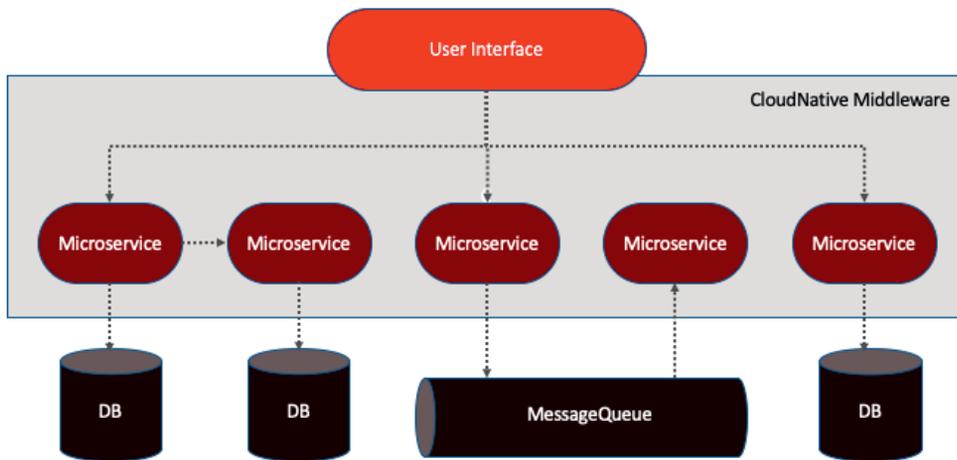
If you only execute a Lift & Shift approach, your software and middleware will not substantially benefit from what cloud actually has to offer: automated scaling, fail-over-functionalities, zero-downtime deployments, and so on. You might mimic these functionalities by bringing in more infrastructure – but at which costs?

Or you would use proprietary offerings from these cloud providers. Which will tie you to them and trap you inside their ecosystem. This kind of vendor lock-in is to be considered a major risk for any enterprise and project and should therefore be avoided.

So, there must be a better way.

To actually and substantially save money and utilize the advantages of cloud environments, one must change the way, software and middleware are set up and integrated with each other. That means: only software being designed for cloud environments is running in such environments – i.e. microservice-based solutions or solutions that understand and utilize the volatile nature of cloud environments.

#1: CloudNative. By Design.



CloudNative Architecture

Such solutions provide fail-over capabilities, autoscaling functionalities, etc. They will try to avoid single-point-of-failure scenarios and vendor lock-ins and embrace stateless approaches. They will usually be set up as containers run by cloud-native middleware instead of being installed by hand and run inside VMs.

A cloud-native middleware will try to integrate deeply into the software it runs (let's better call them "workloads" from now), it will monitor the workload's health and state automatically. But: it will never be invasive, it will simply utilize interfaces and data being made available by the workloads. The typical cloud-native middleware in 2019 is Kubernetes and solutions built around it, such as RedHat OpenShift or SUSE CAAS.

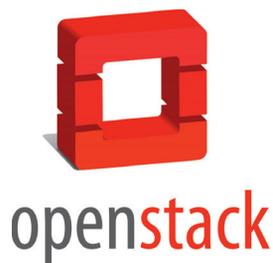


OPENSIFT

Logos of CloudNative Middleware products and projects

They allow for running containerized workloads, for integrating into authorization backends and into already existing infrastructure – and they are completely Open Source (at least Kubernetes and SUSE CAAS are, OpenShift brings in a lot of proprietary aspects). There are vendors providing commercial support for these solutions, such as SUSE or RedHat or third-parties such as my company, Cloudical.

This software stack runs on top of VMs or OpenStack infrastructures, ensuring dynamic provisioning of infrastructures and abstraction from the underlying environment. When planned properly, your software and your middleware could literally run anywhere in public or private clouds, without utilizing proprietary functionalities.



Logos of IaaS-Vendor and -Middleware products

And despite one could technically run old, not cloud-native software stacks on top of this kind of middleware, it would not solve the problem of running a software not being designed for cloud environments in such an environment.

So, software and middleware need to be cloud-native to utilize all technical advantages of cloud environments without being trapped in vendor lock-in scenarios or being confronted with unfulfilled hopes and expectations.

Setting up cloud-native infrastructures and deploying cloud-native applications will solve all problems and challenges, right? Unfortunately, no. In fact, having infrastructure and software in place which allow to scale, will most likely scale your expenses and costs as well.

#2: CloudNative. By Approach.

Why is that?

As a matter of fact, cloud-native infrastructure and cloud-native software is way more complex than their traditional counterparts. Additionally, ops teams will not only have to monitor and run some virtual machines, instead there will be hundreds or even thousands of services to be monitored and executed upon. Also, cloud-native middleware will move workloads around the underlying infrastructure, making error analysis and resolving incredibly complex, especially when considering the volatile nature of containers which are understood as units of work, to be thrown away and discarded when not needed or when being faulty.

To handle a way more complex infrastructure and a way more unpredictable execution model, different approaches are required: automation and agile DevOps teams.

Automation and Versioning are essential to run a cloud environment properly. The classical approach of manual interactions with infrastructure needs to be replaced by a scripted, versioned one. If something is not working properly, environment and / or software is reverted back to the previously working version without manual intervention or SSHing to containers or VMs.

Automation and Versioning

- Every kind of infrastructure is versioned and rolled out automatically.
- Every build of a software is versioned and tested automatically.

- Every deployment of a software is versioned and rolled out automatically.

- Every configurational change is versioned and rolled out automatically.

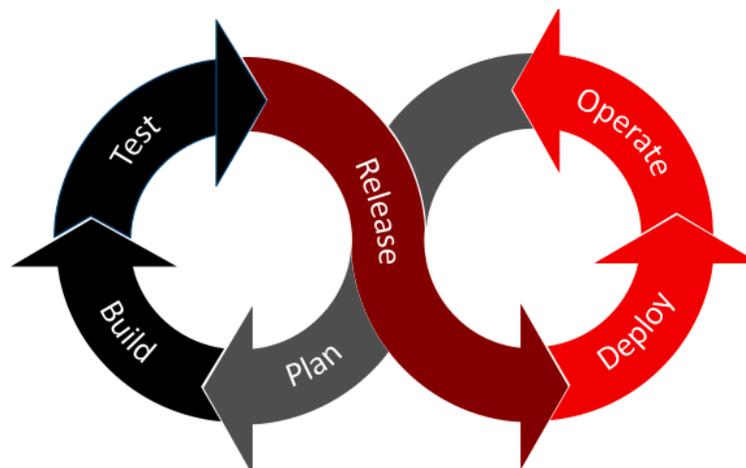
This allows faster iterations, better controlled environments and an implicit documentation of everything. It is executed up to the point, where SSH keys are created automatically inside the environment, deployed onto machines and infrastructures and are never known to any Ops team member.



Logos of DevOps-Infrastructure products

To handle the complexities of cloud environments and the software running inside them, a joint development and operations team needs to be established, acting way more agile than in the past by incorporating Scrum and Kanban principles in their working model. This DevOps team forms the core, but to establish a truly cross-functional approach, other stakeholders such as business units and departments as well as legal and governance, need to be involved as well on a regular basis. Openness and transparency are fundamental, strong communication and collaboration skills are required. Such a team will foresee a lot of problems. It will iterate and produce results. It will share knowledge between team members, thus allowing to have a smooth transition from development to operations. It will also involve external vendors as required and will execute in a self-organized way. The team will define and coordinate SLAs with stakeholders. It will be responsible for its own quality assurance and it will prevent from vendor lock-ins.

Agile Teams, DevOps and Iteration



Agile and iterative DevOps approach

If set up properly and executed consequently, such a team will lower operational costs and efforts required to run and operate workloads in a cloud environment, since it will develop matching processes and automate everything.

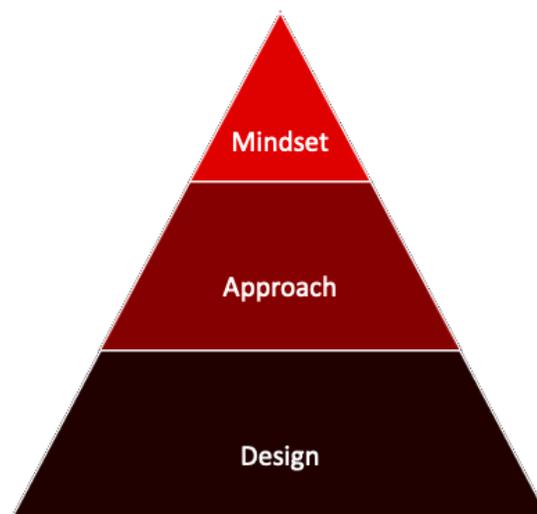
Going one step beyond establishing agile DevOps teams and enforcing automation and versioning, a common mindset needs to be established across all technical and non-technical units and stakeholders.

**#3: CloudNative.
By mindset.**

That mindset implies a different interaction with and understanding of cloud environments, as well as agile and knowledge-driven processes, completely automated operations and DevOps pipelines and an ever-improving ability to setup, deploy and operate environments and software running within them.

Such a mindset needs to be learned, it needs to be executed upon and it needs to be lived – way beyond the borders of “we do it since we are forced to”. It should be at the heart of every team working with and in cloud environments, as well as every stakeholder being involved. It should prevent from being scared of complexities and costs, it should allow to act and to iterate.

Ultimately, a cloud-native mindset is to be lived from top to bottom inside an organization. It should not be a grassroots movement alone, it needs to have support from management and C-level executives. IT and Cloud need to be considered to be one of the building blocks of what a company is doing. Without IT and modern cloud solutions, the race to success and the race to survival are lost even at the very beginning – with cloud environments and cloud-native approaches, it is way more a question of being fast and efficient instead of being slow and inflexible. And that question is answered by knowledge, approach and mindset.



Pillars of CloudNativeness

How to learn and execute the described levels of cloud-nativeness?
How to ensure technology, design, approaches and mindset are understood, executed upon and lived inside a company or an enterprise?

#4: New Work

New Work approaches can be an answer, since they allow for a better and more modern way of working and interacting with each other. Members of small, cross functional teams interact with each other without hierarchies, based only on merits, experience and knowledge.

These approaches change the way teams execute, even if they already work in an agile fashion. New Work implies more democracy without falling into anarchy. It needs to be learned, trained and lived properly – and when implemented successfully, it will bring a team, a department, a company onto a new level of performance and efficiency.



New Work

As cloud-nativeness and New Work approaches are strongly tied to knowledge gaining, training and collection of experience, it is necessary to manage them properly. This is where knowledge management comes into play, since it provides a path for structured learning and structured knowledge transfers. Without it, knowledge and learning are driven by coincidence. With knowledge management in place, knowledge is considered a strategical asset, a deliberately positioned aspect of a path towards CloudExcellence.

#5: Knowledge Management

All discussed components are required for reaching a state of CloudExcellence, as it covers technical and processual expertise, agile and iterative approaches as described above and implies a continuous improvement process and knowledge transfers between team members. If one aspect is missing, excellence and return of investment will not be reached.

**The sum of things:
CloudExcellence**

In short, CloudExcellence is the sum of cloud-native approaches, New Work aspects and knowledge transfers and management. It is independent of a specific cloud environment, it is independent of a specific middleware, it is independent of the kind of workload run – instead and most importantly: CloudExcellence is a mindset. So, the way to approach clouds properly, is to bring all the discussed pillars together: cloud-native design, cloud-native approaches, cloud-native mindset, New Work approaches and knowledge management, ultimately leading to Cloud Excellence.



What does CloudExcellence mean?