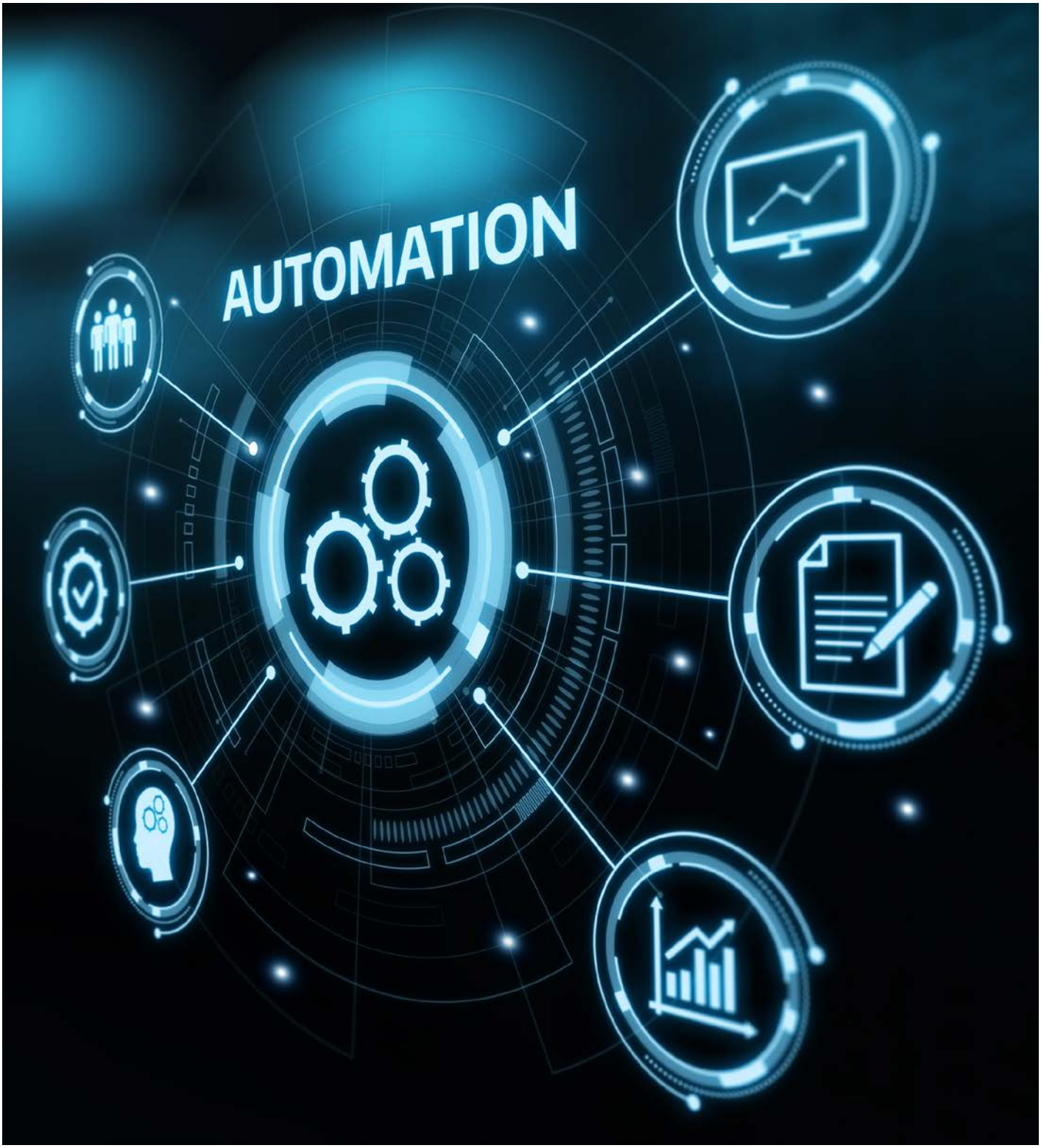
the cloud report



01–2020

Cloud-Native

Additional:

Energy Awareness - Agility - OpenStack Foundation

Room for

#CloudExcellence

EDITORIAL

Dear reader,
the cloud report wishes you a happy, successful, informative 2020!

We start the new year with a mixture of topics like automation, processes, microservices – in sum Cloud-Native.

Cloud-native is an approach to ensure that applications are designed and developed for cloud computing architecture. Cloud-native applications make consistent use of the services provided by the cloud and are based on microservices (cloudcomputing-insider.de). Christina Kraus gives more insights and answers the questions: Why do companies aim to implement cloud-native applications? What are the benefits of being cloud-native?

To be cloud ready you need a broad technical skillset and a special mindset. The first skill I think is needed is to know how to automate the basic functionalities to operate cloud services less erroneous than by hand and to provide bigger infrastructures quicker and reproducible. One approach to automate is shown by Sandra Parsick. Her article uses the provisioning tool Ansible to describe how automated quality assurance for provisioning scripts can look like.

As mentioned in the definition microservices are understood as one of the foundations of a cloud-native solution, since they allow faster development, technology abstraction and ease of deployment. Karsten Samaschke shows how to write a microservice, package it and deploy it onto a Kubernetes cluster.

Another aspect of being cloud-native is the processes. Not only the technical ones, but also the interpersonal processes and the mindset behind. Agility is one approach to review organizational processes, Jana Weiner and Anne- Sophie Schwindenhammer take a look into the daily challenges of changing organizations and give advices for a holistic communication.

Concerning mindset one of the big topics now and for the future is to keep environmental interests in mind. Cloud, IoT, 5G, streaming, digitization, ... for all these developments energy is needed! There are no global concepts for dealing with the "energy issue", and no single person or company can provide it. But everybody should have energy awareness! Sandro Kreten discusses some basic approaches to catch more attention for the important topic "energy".

And, again, we had the possibility to talk with interesting people! First with Mark Collier, COO and Co-Founder of the OpenStack Foundation, he gives very deep and personal insights of IaaS, cooperation, doubts and confidents. It is a very long interview, but so worth reading! The second interview is with Marco Görgmaier, he talks about the cloud strategy of the BMW Group and the evolution of big enterprises. He shows a concrete example in being ready for the future.

Have fun reading! Best wishes,
Friederike
Editor in chief

the cloud report

IMPRINT

Publisher	Cloudical Deutschland GmbH, Edisonstr. 63, 12459 Berlin
Managing director	Karsten Samaschke
Publishing director / Editor in chief	Friederike Zelke
Editorial office	Julia Hahn, Emelie Gustafsson
Online editorial	Stefan Klose
Artdirection and Layout	Anna Bakalovic
Production	Andreas Merkert
Editorial office contact	press@cloudical.io
Sales contact	sales@cloudical.io

the cloud report published by Cloudical Deutschland GmbH
Edisonstr. 63, 12459 Berlin
Managing director Karsten Samaschke
Mail: info@cloudical.io

the-report.cloud

ISSN 2626-1200

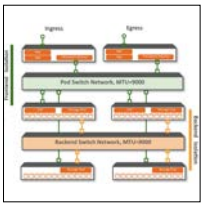
The Cloud Report is published quarterly at the beginning of January, April, July and October. The Cloud Report is available in two versions: the online edition which can be accessed online or via downloaded from the homepage, and the printed edition which can be subscribed at a price of 20 Euros a year via the personalized customer portal in which a personal account is set up. When you register, please enter the information to specify the execution in which you want to obtain the report. The subscription can be cancelled at any time via the personal access of the subscriber, by mail at the latest two weeks before the publication of the new edition via the e-mail address: sales@cloudical.io. We collect relevant personal customer data for the subscription. Individual issues can also be purchased without a subscription at a price of: 5 Euro, in this case too, relevant personal data will be used to fulfil the purchase contract. Further information can be found at: http://the-report.cloud/privacy-policy

the cloud report

TABLE OF CONTENT

EDITORIAL

FOCUS



INTERVIEW

PROJECT LEADERSHIP

CONFERENCES

TESTS

Why your organization has to be cloud-native

Implementing cloud-native processes for cloud governance

Cloud-native has become a paradigm that stands for highly scalable applications. While it often relates to the application architecture, microservices, the use of containers or declarative APIs, being cloud-native comes with organizational implications as well. But first things first: Why do companies aim to implement cloud-native applications? They want to release better software faster and they want to be able to react faster in case something goes wrong.

Cloud-native as a competitive advantage

Let's consider a rather traditional company like a bank. Banks have been using IT systems for years. Stability and availability of these systems is essential. Moving such applications that have been stable for years, do not evolve in terms of features and have no flexibility in their scale to the cloud does not necessarily bring any benefits. They will most probably not be cheaper nor more stable, when running in the cloud.

However, traditional industries like the financial services industry face strong competition, often from startups that aim to hit a market that has been stable for many years and has not really adopted to the technological development and resulting customer requirements or expectations.

In most cases these startups are cloud-native. They build digital applications in the cloud. With a strong customer focus, they are able to process customer feedback in fast iterations and quickly move towards a product their users like to use. In contrast to their established competitors, they have the great benefit to be able to move much faster, because they are not carrying a load full of legacy applications that they have to take care of in addition to their new product developments.

When building new customer-facing applications, having capabilities in cloud-native software development is a competitive advantage. New products can be built faster, continuously improved and adopted to customer needs. Instead of having new software releases yearly or twice per year, CI/CD allows us to integrate product updates on a daily basis. For this reason, a lot of traditional companies from different sectors, experience high pressure to digitize their business models and move to the cloud, in order to hold on to their position in the market and avoid customer retention

Fig. 1: Shadow IT is a great security risk and often a result of ineffective processes.
(Photo by Steve Halama on Unsplash)

Why cloud-native is relevant to the organization

Cloud-native is not only about the use of cloud platforms. Software architecture and additional tooling, like CI/CD or other automation tools play an important role, when leveraging the benefits of the cloud. In addition, in an enterprise context organizational processes are another challenge on the way to cloud-native success.

One of the main reasons developers love cloud platforms like AWS, Azure or Google Cloud Platform is that they can access a large variety of different cloud resources and services within minutes. They just need to create an account, pull out their credit card and are ready to get going. For enterprises, cloud providers offer organization accounts that group together accounts across the organization.

In many cases, developers are not allowed to create an account belonging to the organization in self-service. They have to go through some specified account provisioning process. For this purpose, enterprises often reuse existing tools or processes that originate from older times, e.g. from ordering on-premise IT hardware like monitors or phones. The risk is that the original cloud-native experience is being affected, as barriers are posed between the developer and the cloud.

Setting up new processes for the use of cloud is important for multiple reasons:

Loss of productivity
Slow processes for cloud access impact the time-to-market for new applications. Letting developers wait for weeks or months to gain access to their required resources is a waste and enemy of productivity.

Risk of shadow IT
Not only do slow processes slow down application delivery, they bear a great risk for shadow IT as developers may find other and faster ways to get to their cloud environment. It is not uncommon to find cloud costs in travel expenses for that reason. Apart from the uncontrollable cost, those accounts are completely under the radar of company regulations.

Limited scalability
Automated processes are scalable and bring benefits in terms of consistency, e.g. in security configurations. In contrast, manual routine-processes are error-prone and can hardly be controlled. Furthermore, they mostly don't keep up with the rising demand of IT resources, leading to bottlenecks and even longer waiting periods.

Negative impact on employer attractiveness
In times of skill shortage, being attractive as an employer plays an important role when it comes to hiring IT talent. Organizational processes reflect the agility of the organization as a whole. If a company is not progressive in their organization, it's likely that they are neither in their visions and the use of technology.

If you do it for one cloud, you should do it for all

We often face situations in which different clouds, such as Azure, AWS or GCP are treated as completely distinct silos. Processes (e.g. for access, tenant creation or cost management) are reinvented for each of those platforms, missing an integrated and comprehensive concept of how cloud is going to be used across the organization. This does not only require more resources for the implementation, it results in missing transparency from a holistic and strategic perspective.

- How do I proceed in my cloud transformation as a whole?
- How many applications have I migrated to the cloud?
- How does my cloud spend evolve and behave across platforms?
- How does the performance of different teams differ in terms of cloud migration?
- How do I ensure a consistent level of governance across different cloud platforms?
- How do I find a balance between individual requirements and services that should be standardized throughout the organization.

Having cloud silos, the information that is necessary to answer the above questions is missing or has to be assembled with a lot of manual effort, which again requires costly resources and much worse, is prone to error.

The spirit of cloud-native principles

In software development, the term cloud-native is related to certain design principles[1]:

- Designed as loosely coupled microservices
- Best-of-breed languages and frameworks
- Centered around APIs for interaction and collaboration
- Stateless and massively scalable
- Resiliency at the core of the architecture
- Packaged as lightweight containers and orchestrated
- Agile DevOps & Automation Using CI/CD

While there are variations on single items, depending on the source, the core of the principles stays the same. Scalability is one of the main aspects of cloud-native applications and often the shortage when it comes to organizational processes.

This is natural, because the use of cloud usually starts small in single departments and therefore does not initially require a comprehensive and scalable organizational strategy. Processes can be kept simple and pragmatic to incentivize the use and experimentation with new technologies and reduce administrative overhead at this stage. Therefore, early adopters often experience some kind of protection, as consistent rules still have to be defined and implemented.

At some point however, there is a need to collect the diverging cloud initiatives across the organization, as it otherwise becomes impossible to keep control of compliance and cost. Processes have to be well documented and auditable in order to be ready to run in production.

What the organization can learn from cloud-native development

Best-of-breed technologies
One of the cloud-native design principles is related to the use of best-of-breed languages and frameworks. Therefore, technological freedom is one reason to provide multiple cloud technologies and platforms to the DevOps teams.

Considering the market development of cloud-native tools and services, the past years have been very dynamic. And while the amount of cloud providers has been rather stable, the services they offer evolve and change continuously. Choosing a single specific technology to provide to DevOps teams bears a high risk. In order to stay flexible and be able to adapt to technological changes, processes should be independent of specific technologies. DevOps teams should experience freedom and be able to work with their preferred tools and it should be possible to integrate new upcoming technologies.

Fig. 2: When providing agility, it is important to implement some control and security mechanisms at the same time. (Photo by John Salvino on Unsplash)

Automation and APIs

Automation is one of the main drivers of cloud technologies. Therefore, the processes for using cloud shouldn't block the access to the native cloud APIs, e.g. by providing cloud resources via an additional cloud-service portal and not allowing access to the cloud-native portals. Apart from not being automation-friendly, such setups have the disadvantage that each service has to be integrated individually, which is very resource-intensive and leads to time lags regarding the availability of new services.

Stateless and massively scalable

In order to be horizontally scalable, applications should be stateless. meshcloud, a platform for enterprise-level cloud management and governance, has developed a cloud maturity model[2] that introduces the idea of "organizations as code". The idea behind that is following from "infrastructure as code", where you define the target state of your infrastructure. Similarly, you can define the target state of your organization, including information like the project that exist in each cloud platform, which users should have access to them and which policies should be applied. Using this paradigm for organizational processes has the benefit that it is very robust and scalable, and it is independent from specific technologies. Furthermore, it integrates essential documentation steps that can be helpful when it comes to audits.

Security

A final aspect that shouldn't be neglected is that such a bold move towards agility in the organization should on the other hand be balanced with security measures. In the past years, we have seen that most security incidents, e.g. Capital One's loss of personal data of one million people are due to faulty configurations[3], in this case a firewall misconfiguration. While it's impossible to stay 100% secure, a good approach is to provide a basic security level when handing out cloud accounts to DevOps teams to prevent the teams from basic mistakes. This can be done by so-called landing zones[4] that restrict the of the cloud account in accordance to security regulations.

Conclusion

Motives for using cloud-native technologies are scalability of applications as well as fast iterations in application development that allow to continuously integrate customer feedback and evolve along customer needs.

When it comes to provisioning cloud technologies across large organizations, there are different ways to design and implement processes around cloud governance. In order to be successful and leverage the benefits of the cloud, these processes should have a cloud-native spirit: They should provide autonomy and allow for technological freedom of the user, while keeping the balance with security measures and controls.

Sources:

- 1. https://medium.com/walmartlabs/cloud-native-application-architecture-a84ddf378f82
- 2. https://www.meshcloud.io/en/2019/02/04/the-path-to-proper-multi-cloud-management-a-maturity-model/
- 3. https://techcrunch.com/2019/07/29/capital-one-hacked-over-100-million-customers-affected/
- 4. https://www.meshcloud.io/en/compliance-and-security/

Christina Kraus
Co-founder meshcloud
hallo@meshcloud.io
www.meshcloud.io

meshcloud

Automated Quality Assurance for Ansible Playbooks

When server instances are provisioned in the cloud, they are rarely set up manually, but automated using provisioning scripts. The provisioning scripts describe how the server should be set up. This is normal code, as we know it from normal software development, except that this code is specialized on the domain infrastructure. In software development, static code analysis ("linting") and automated tests have established themselves as means for good maintainability of the production code. Why not apply this good practice to "Infrastructure as Code" as well? This article uses the provisioning tool Ansible to describe how automated quality assurance for provisioning scripts can look like.

Brief Introduction to Ansible

Ansible describes itself as a tool for configuration management, software distribution, and for executing ad hoc commands[1]. Ansible is written in Python. To use it, the developer does not necessarily need to have Python knowledge, since the Ansible scripts are written in YAML. YAML is an abstraction of JSON with the difference that it is more readable[2]. The ansible scripts are called "Playbooks".

In order to use Ansible, Python must be installed on the target machines and access to these machines must be possible via SSH. Ansible must only be installed on the machine running the Playbooks ("host machine"); it must be a Linux system. For target machines running Windows, Ansible provides rudimentary support. When the Playbooks written in YAML are executed, Ansible translates them into Python scripts and executes them on the target machines. They are then removed from the target machines. A typical Ansible Playbook looks like this (listing 1):

How a QA Pipeline Works for Ansible Playbooks

In software development, static code analysis and automated tests are performed continuously in a pipeline on a CI server. This concept is to be reused for Ansible Playbooks.

First, the developer stores the provisioning scripts in a version control system. Changes to the scripts that are checked in to the version control system trigger a pipeline. A pipeline for provisioning scripts runs through several steps (see also figure 1): First, a lint checks the code to see whether it is syntactically correct and whether it follows best practice. If the lint has nothing to complain about, a test environment is prepared. The provisioning scripts run against these test environments. If they have run without errors, tests are run to verify that everything has been configured as expected. At the end, the test environment is destroyed again.

```
- hosts: ansible-test-instance
  vars:
      tomcat_version: 9.0.27
      tomcat_base_name: apache-tomcat-{{ tomcat_version }}
      #catalina_opts: "-Dkey=value"

  tasks:
      - name: install java
      apt:
            name: openjdk-8-jdk state: present
      become: yes
      become_method: sudo
```

Listing 1: Ansible Playbook: For a deeper introduction to Ansible, we recommend the author's article "Ansible für Entwickler"[3].

Declarative: Checkout SCM	Lint Code	Prepare test environment	Run Playbooks	Run Tests	Declarative: Post Actions
112ms	1s	1min 16s	1min 27s	11s	2s
112ms	1s	1min 16s	1min 27s	11s	2s

Figure 1. All steps in a pipeline

Lint are tools that perform static code analysis.

In the following, the individual steps are presented in more detail and how they are then integrated into a pipeline. This is demonstrated using a git repository that includes an Ansible playbook, scripts to build the test environment, the tests, and a description of the pipeline. The Ansible playbook will install an OpenJDK 8 and a Tomcat instance. The complete project can be found on Github[4].

The starting point is the Git repository, which initially only receives the Ansible Playbook.

```
├─LICENSE
├─README.md
└─setup-tomcat.yml
```

Static code analysis with ansible-lint

First, a lint checks whether the Ansible Playbooks are syntactically correct and whether they are written according to best practice rules. For Ansible Playbooks there is the lint `ansible-lint`[5]. It is executed on the CLI (listing 2):

In this example, `ansible-lint` finds several rule violations. If single rules should not be checked, there are two ways to exclude them. The rule can be switched off either globally for the whole project, or for individual cases (so-called false-positive cases). For the global setting a `.ansible-lint` file is placed in the root directory of the project.

```
├─.ansible-lint
├─LICENSE
├─README.md
├─setup-tomcat.yml
```

```
$ ansible-lint setup-tomcat.yml
[502] All tasks should be named setup-tomcat.yml:30
Task/Handler: file name=/opt file =setup-tomcat.yml line =31 mode=511
owner=tomcat group=tomcat

[502] All tasks should be named setup-tomcat.yml:57
Task/Handler: find patterns=*.sh paths=/opt/{{ tomcat_base_name }}/bin
```

Listing 2

In this configuration file you maintain an `exclude` list:

```
skip_list:
  - skip_this_tag
  - and_this_one_too
  - skip_this_id
  - '401
```

In this configuration file further behavior can be configured, e.g. in which path they are stored. More information can be found on the project page[4].

If the developer wants to exclude false-positive cases from the check, she leaves a comment in the Ansible Playbook.

```
- file: # noqa 502
      name: /opt
      mode: 0777
      owner: tomcat
      group: tomcat
  become: yes
  become_method: sudo
```

If the developer lacks further rules, then she can define further rules herself with the help of Python scripts (see also documentation[5]). Listing 3.

Setting up and Destroying the Test Environment with Terraform

After the lint has successfully passed, the test environment for the Ansible Playbooks should be built. This is done with the help of Terraform[6] and the Hetzner Cloud[7]. Terraform helps developers to provision cloud infrastructure with code.

Before the developer can get started with the Terraform script, she must store a public SSH key in the Hetzner Cloud account and generate an API token. The Public SSH Key is later stored in the server instance to be created, so that Ansible can connect to this instance.

With the help of Terraform, the developer describes which server type she wants, which operating system, in which location the server instance is hosted and what is to be provisioned as the basic setup. For this purpose, the developer creates a `testinfrastrucuture.tf` file in the root directory of the project (listing 4).

```
├.ansible-lint
├LICENSE
├README.md
├setup-tomcat.yml
├terraform.tfvars
└testinfrastructure.tf
```

As a basic setup, the developer specifies which public SSH key should be stored on the server (`ssh_keys`) and that Python should be installed (`provisioner remote-exec`). The public SSH key and Python are needed so that later Ansible can execute its scripts on this server.

Since the test server instance is to be operated in the Hetzner Cloud, Terraform must install the required provider plug-in. The developer calls `terraform init` in the folder containing `testinfrastructure.tf`.

Then everything is ready to provision the server instance.

```
terraform apply -var="hcloud_token=..."
```

The developer must give the `apply` command the variable `hcloud_token` with the API token, which the developer generated before in the Hetzner Cloud console.

As soon as the server is available, the developer can execute the Ansible Playbooks and the tests against this server. Regardless of the success of the Playbooks or the tests, the server instance is destroyed with the help of Terraform. The `destroy` command also requires the API token.

```
terraform destroy -var="hcloud_token=..."
```

```
from ansiblelint import AnsibleLintRule

class DeprecatedVariableRule(AnsibleLintRule):

  id = 'ANSIBLE0001'
  shortdesc = 'Deprecated variable declarations'
  description = 'Check for lines that have old style ${var} ' + \
      'declarations'
  tags = { 'deprecated' }

  def match(self, file, line):
      return '${' in line
```

Listing 3

```
# testinfrastructure.tf
variable "hcloud_token" {}
variable "ssh_key" {}

# Configure the Hetzner Cloud Provider
provider "hcloud" {
  token = var.hcloud_token
}

# Create a server
resource "hcloud_server" "ansible-tests" {
  name = "ansible-tests"
  image = "ubuntu-18.04"
  server_type = "cx11"
  location = "nbg1"
  ssh_keys = ["ansible-test-infrastructure"]

  provisioner "remote-exec" {
    inline = [
    "while fuser /var/lib/apt/lists/lock >/dev/null 2>&1; do sleep 1; done",
    "apt-get -qq update -y",
    "apt-get -qq install python -y",
    ]

    connection {
      type = "ssh"
      user = "root"
      private_key = file(var.ssh_key)
      host = hcloud_server.ansible-tests.ipv4_address
    }
  }
}
```

Listing 4: testinfrastructure.tf

Running Ansible Playbooks

After the developer has created the server instance with Terraform, she executes the Ansible Playbook against this server instance. In a classical infrastructure, the server instance would have a fixed IP address and the developer would have entered this IP address in the Ansibles static inventory so that Ansible knows which server to connect to *static Inventory*.

```
[ansible-tests]
78.47.150.245
```

Since the servers in the cloud are assigned a new IP address each time they are deployed, the developer cannot use the static inventory in this case. Since the developer goes against the Hetzner Cloud, she uses the Ansible Inventory Plugin `hcloud`. Therefore an `inventory` folder with the file `test.hcloud.yml` has to be created.

```
├.ansible-lint
├ansible.cfg
├inventory
│ └test.hcloud.yml
├LICENSE
├README.md
├setup-tomcat.yml
├terraform.tfvars
└testinfrastructure.tf
```

The suffix `hcloud.yml` is important for the file name. *test.hcloud.yml*

```
plugin: hcloud
```

The developer then has to enter the correct server name in the Playbook under `hosts`, which she previously defined in the terraform script. *Ansible-Playbook Snippet*

```
- hosts: ansible-test-instance
```

When running the Playbooks the developer has to make sure that she gives the correct Private SSH key and that the API token is defined in the system environment variable `HCLOUD_TOKEN`.

```
$  export  HCLOUD_TOKEN=...
$ ansible-playbook --private-key=/home/
sparsick/.ssh/id_hetzner_ansible_test -i
inventory/test.hcloud.yml setup-tomcat.yml
```

The API token can also be defined by the developer in the inventory file `test.hcloud.yml`. However, since this file is stored in a version control system, it is not advisable to do this, since no credential should be stored in a VCS.

Functional Tests with Testinfra

After the test environment has been set up and the Playbooks have passed successfully, tests should be run to check if the openjdk package has been installed and if the file `/opt/tomcat/bin/catalina.sh` exists on the server.

There are some test frameworks for provisioning tools such as ServerSpec[8], Goss[9] and Testinfra[10]. The main difference between the test frameworks is the syntax in which the tests are written. For ServerSpec the tests are described in Ruby syntax, for Goss in YAML syntax and for testinfra in Python syntax.

The way the test frameworks work is the same. They connect to a server that was previously provisioned with provisioning tools and check whether the server provisioning corresponds to the test descriptions.

Here, the tests are written with Testinfra. To do this, the developer creates the `tests` folder in the root directory of the project. This is where the tests are stored.

```
├.ansible-lint
├ansible.cfg
├inventory
│ └test.hcloud.yml
├LICENSE
├README.md
├setup-tomcat.yml
├terraform.tfvars
├testinfrastructure.tf
├tests
│ └test_tomcat.py
```

The developer writes the tests in `test_tomcat.py`. (listing 5)

Testinfra comes with ready-made modules that simplify the test description. In these tests the developer uses e.g. `host.package` to query the Package Manager or `host.file` to test the existence of a certain file.

Testinfra supports several ways to connect to the server. One way is to reuse Ansible's connection configuration. Since Ansible uses a dynamic inventory here, and Testinfra cannot read all the information from that dynamic inventory, the developer must explicitly enter some configurations in the Ansible configuration file `ansible.cfg`. Listing 6.

This file is stored in the root directory of the project. The developer can then run the tests. (listing 7)

```
def test_openjdk_is_installed(host):
  openjdk = host.package("openjdk-8-jdk")
  assert openjdk.is_installed

def test_tomcat_catalina_script_exist(host):
  assert host.file("/opt/tomcat/bin/catalina.sh").exists
```

Listing 5: test_tomcat.py

```
[defaults]
remote_user=root
private_key_file = /home/sparsick/.ssh/id_hetzner_ansible_test
```

Listing 6: ansible.cfg

```
$ py.test --connection=ansible --ansible-inventory=inventory/test.hcloud.yml --force
-ansible -v tests/*.py
=================================================================================
=
====================== test session starts
=================================================================================
=
======================
platform linux2 -- Python 2.7.15+, pytest-3.6.3, py-1.5.4, pluggy-0.13.0 --
/usr/bin/python cachedir: .pytest_cache
rootdir: /home/sparsick/dev/workspace/ansible-testing-article, inifile: plugins:
testinfra-3.2.0
collected 2 items

tests/test_tomcat.py::test_openjdk_is_installed[ansible://ansible-test-instance]
PASSED
[50%]
tests/test_tomcat.py::test_tomcat_catalina_script_exist[ansible://ansible-test- in-
stance] PASSED
[100%]

=================================================================================
=
=================== 2 passed in 11.52 seconds
=================================================================================
=
===================
```

Listing 7

```
#Jenkinsfile

pipeline {
  agent any
  environment {
    HCLOUD_TOKEN = credentials('hcloud-token')
  }

  stages {
    stage('Lint Code') {
      steps {
      sh 'ansible-lint setup-tomcat.yml'
    }
  }

    stage('Prepare test environment') {
      steps {
      sh 'terraform init'
      sh 'terraform apply -auto-approve -var="hcloud_token=${HCLOUD_
TOKEN}"'
      }
    }

    stage('Run Playbooks') {
      steps {
      sh 'ansible-playbook -i inventory/test.hcloud.yml setup-tomcat.
yml'
      }
    }

    stage('Run Tests') {
      steps {
      sh 'py.test --connection=ansible --ansible
-inventory=inventory/test.hcloud.yml --force-ansible -v tests/*.py'
      }
    }
  }

  post {
    always {
      sh 'terraform destroy -auto-approve -var="hcloud_token=${HCLOUD_
TOKEN}"'
    }
  }
}
```

Listing 8: Jenkinsfile

Pipeline Integration

After each step in the pipeline has been considered individually, the steps are to be merged into a pipeline in CI-Server Jenkins.

The pipeline is described by the developer in a Jenkins file. This Jenkins file describes four stages and one post-action. One stage each for the code check, build test environment and execute playbooks and the last stage for the execution. In the post-action, the test environment is dismantled, regardless of whether errors occurred in the stages (listing 8).

In order for this pipeline to work, the developer must deposit the API token of the Hetzner Cloud in the Jenkins. So that the token is not stored in plain text, the developer stores it in the `Credential` area as `Secret Text` and assigns an ID, which he can then retrieve using the `credential` method in the Jenkins file (here: `hcloud-token`).

Simplification for Ansible Role

For Ansible Role, a structuring option in Ansible to reuse Playbooks across multiple projects, there is a simplification for this presented pipeline. With Ansible Role, the developer can use Molecule to configure the complete pipeline and the presented tools in one go. She then only needs one command (`molecule test`) to execute the complete pipeline. A very good introduction to Molecule is given in the blog post "Test-driven infrastructure development with Ansible & Molecule"[11,12] by Jonas Hecht.

Conclusion

This article provides an overview of how to build a quality assurance pipeline for Infrastructure as Code.

Sources:

- 1. https://docs.ansible.com/
- 2. https://en.wikipedia.org/wiki/YAML
- 3. https://www.sandra-parsick.de/publication/ansible-fuer-dev/ (German)
- 4. https://github.com/sparsick/ansible-testing-article/tree/cloudreport19
- 5. https://github.com/ansible/ansible-lint
- 6. https://www.terraform.io/
- 7. https://www.hetzner.com/cloud
- 8. https://serverspec.org/
- 9. https://github.com/aelsabbahy/goss
- 10. https://github.com/philpep/testinfra
- 11. https://github.com/ansible/molecule
- 12. https://blog.codecentric.de/en/2018/12/test-driven-infrastructure-ansible-molecule/

Sandra Parsick

Sandra Parsick works as a freelance software developer and consultant in the Java environment.
Since 2008 she is working with agile software development in different roles. Her focus is on Java enterprise applications, agile methods, software craftsmanship and the automation of software development processes.
She enjoys writing articles and speaking at conferences.
In her spare time Sandra Parsick is involved in the Softwerkskammer Ruhrgebiet, a regional group of the Software Craftmanship Community in the German speaking area. Since 2019 she is a member of the Oracle Groundbreaker Ambassador Program.

E-Mail: mail@sandra-parsick.de
Twitter: @SandraParsick
Homepage: https://www.sandra-parsick.de

Rook: New Winds in the v1.1 release

Rook is a storage orchestrator for Kubernetes using the operator pattern. That means that Rook can run, e.g., Ceph, EdgeFS, Yugabyte and other persistence providers in a Kubernetes cluster. This allows applications to create, e.g., a YBCluster object to get a YugabyteDB cluster to provide persistence for your applications in Kubernetes. Be sure to check out Rook's website Rook.io for more information.

Numbers

The Rook project is continuously growing, more contributors, more Twitter followers and the growing Slack member count. The following are the numbers from the official Rook v1.1 release blog post:

- 5K to 6K+ Github stars 150 to 196 Contributors
- 40M to 74M+ Container downloads 3050 to 3700+ Twitter followers
- 1610 to 2200+ Slack members

Those numbers are awesome for the storage backends and Rook itself!

It is good to see the project grow in numbers, but also mature further in regards to governance, project charter and CNCF project graduation in the future. With each release Rook is getting more boring. This is good, as especially newly introduced features get faster stable because of that.

New Features and Improvements

Let's dive into some of the most outstanding features and improvements of the latest Rook 1.1 release.

New Storage Provider: Yugabyte DB

In the latest Rook release version 1.1, Yugabyte DB joined the list of storage providers in the Rook project. Bringing the total of storage providers to 7.

Below list contains all current storage providers which are integrated in Rook:

- Cassandra
- Ceph CockroachDB EdgeFS Minio
- NFS
- Yugabyte DB

To show how easy it know is to create a Yugabyte DB cluster in Kubernetes, just look at the code snippet:

Yep, just a few lines of code and the Rook Yugabyte DB operator will take care of creating everything needed for a Yugabyte DB cluster. That is the magic that the Operator Pattern in Kubernetes brings. Applications like that are destined to have an operator with CustomResourceDefinitions to make life easier for the Development and Operations teams in Kubernetes.

One more thing to note is that you can technically run a Rook Ceph or EdgeFS Cluster in your Kubernetes and run,

```
apiVersion: yugabytedb.rook.io/v1alpha1
kind: YBCluster
metadata:
  name: hello-ybdb-cluster
  namespace: rook-yugabytedb
spec:
  master:
    replicas: 3
    volumeClaimTemplate:
      [...]
  tserver:
    replicas: 3
    network:
      ports:
        - name: yb-tserver-ui
          port: 9000
        - name: yb-tserver-rpc
          port: 9100
        - name: ycql
          port: 9042
        - name: yedis
          port: 6379
        - name: ysql
          port: 5433
    volumeClaimTemplate:
      [...]
```

e.g., Yugabyte DB, Minio, and so on, on top of that storage provider.

If you want to know more about Yugabyte DB Rook integration, checkout their blog post Yugabyte DB Blog - Announcing the New Rook Operator for Yugabyte DB.

Ceph

CSI the new default for Provisioning and Mounting Storage

In previous releases the default was to use the Rook Flexvolume driver for provisioning and mounting of Ceph storage. In comparision with the now default Ceph CSI driver, the Flexvolume is lacking features, like dynamic provisioning of PersistentVolumeClaim for CephFS (filesystem storage).

The Ceph CSI driver brings many improvements and the mentioned dynamic provisioning feature for CephFS PersistentVolumeClaims. There is a lesser used features that is not implemented in the Ceph CSI driver yet, mounting erasure-coded block storage, but the Ceph CSI team is working on implement this feature to bring it on the same level as the Flexvolume.

For the impatient people that want to get the CephFS dynamic provisioning for their cluster and / or get started with Rook in general, checkout the Rook v1.1 Documentation.

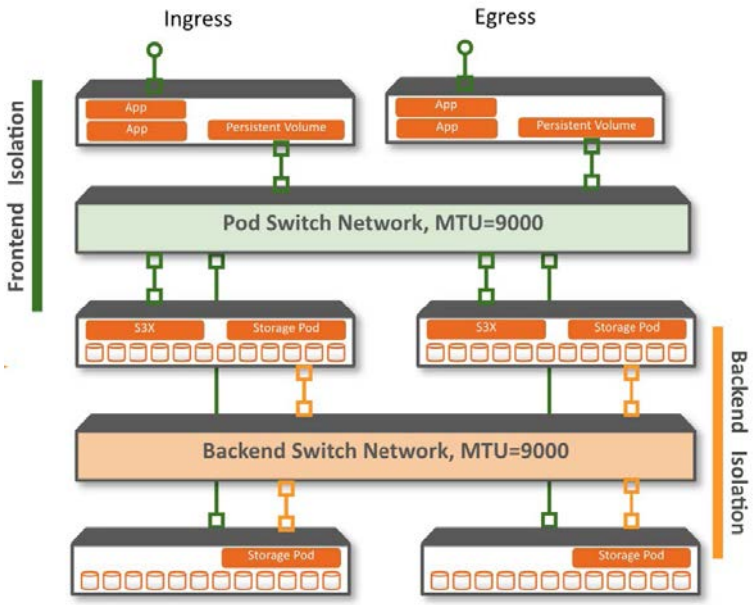
OSDs can now run on PersistentVolumeClaims

Yep, you read that right, Ceph OSDs (data stores) can now run on PersistentVolumeClaims. This finally gives many people an option to reliably run in cloud environments (e.g., Azure, AWS, Google Cloud) with their Rook Ceph cluster setups. Fun fact: This technically enables one to run a Rook Ceph Cluster on PersistentVolumes from another Rook Ceph cluster.

Let's continue on that fun fact in Rook Ceph in Rook Ceph in Rook Ceph ....

Rook Ceph in Rook Ceph in Rook Ceph ...

I will soon be doing an experiment, in which I'll be running a Rook Ceph Cluster on top of another Rook Ceph Cluster. "Wrapping" a Rook Ceph Cluster once in another Rook Ceph Cluster is boring though, so the experiment will be to wrap a Rook Ceph Cluster on top of as many other warpped Rook Ceph Cluster as possible.



EdgeFS

Has accomplished a milestone in the Rook project by being the second storage provider to release their CRDs as stable v1. That is awesome to see, but not the only big accomplishment for the v1.1 release.

For a general roundup of all that is new in EdgeFS, checkout Kubernetes Rook EdgeFS 1.1 Released - EdgeFS.

Multi homed network

EdgeFS is the first to implement multi (homed) network. Their initial integration efforts are done with Intel's project Multus CNI.

This is thanks to the Google Summer of Code (GSoC) participant @giovanism!

In their v1.1 blog post, from which the diagram was taken, contains a performance benchmarks that shows that multi homed network can improve the performance of the storage. Besides the performance improvements, depending on the network setup this can even allow the storage to be more resilient as it would be, e.g., unaffected by application network outages.

This is a huge thing to happen in the Rook project and hopefully soon to be seen to be implemented in the Rook Ceph Cluster CRD as well.

Summary

The Rook project is getting boring. There are still many important things to further work and improve on. One burning topic is disaster recovery, which is pretty much a complex manual process right now. That process just "screams" to be "automated" to a certain aspect, to remove the human factor as much as possible.

Join the in-cluster storage revolution, Rook on!

How to get involved

If you are interested in Rook, don't hesitate to connect with the Rook community and project using the below ways.

- Twitter
- Slack
- Contribute to Rook
- Website
- Mailing List
- Community Meetings

Other articles about Rook

If you are interested in reading other articles about the Rook project, be sure to checkout the following links:

- http://the-report.cloud/rook-more-than-ceph
- http://the-report.cloud/rook-v1-0-adds-support-for-ceph-nautilus-edgefs-and-nfs-operator
- https://blog.rook.io/rook-v1-1-accelerating-storage-providers-5b9e8d5901d8
- https://medium.com/edgefs/kubernetes-rook-edgefs-1-1-released-452799283fce https://blog.yugabyte.com/rook-operator-announcement/

Alexander Trost
Rook Maintainer and DevOps Engineer
alexander.trost@cloudical.io

CLOUDICAL

CLOUDICAL

Delivering CloudExcellence

Rook Engineer

We are applying to Rook Engineers (all genders)

Wanna work with Alexander Trost on the Rook project? Alexander is a DevOps Engineer, Developer, Gamer with passion and Rook maintainer. We are proud of the Rook project and to have him.

We are looking for people instead of positions. We want you to be who you are. Authentic. You strive for knowledge, excellence and understanding. You share your knowledge with us, our customers and partners. You see yourself as a visionary and are characterized by openness and tolerance, both interpersonal and technological. You have a CloudNative Mindset, you can easily adapt to new situations and you have a generous willingness to travel. We are wherever our customers are.

As a learning organization we work together on your further (cloud) development. We provide the appropriate times for this purpose. We love to spend up to 80% of our working hours remotely. We know flexible working time models, open-ended contracts or company holidays at Christmas and between the years.

Are you ready to join us on our way from an information society to a knowledge society?

YOUR FURTHER RESPONSIBILITIES

- Development of Rook.io
- Participation in community meetings and events
- Conducting technical trainings
- Technical consulting and competent contact person for our customers
- "Hands on" Troubleshooting
- Monitoring and operation of customer infrastructures
- Sustainable and consistent automation of recurring tasks

WHAT YOU SHOULD BRING WITH YOU

- Experience in the development and / or operation of applications
- Advanced knowledge in at least one of the areas:
- Software defined Storage (Ceph, Rook, EdgeFS, LizzardFS)
- Container (Docker, Kubernetes, Containerd, CRI-O)
- Continuous Integration/Delivery (Jenkins, Gitlab CI, DroneCI)
- Source Code Management (Git, Github, Gitlab, Gerrit)
- Programming languages (Golang, Python, C++)

WHO WE ARE?

We've got Room for #CloudExcellence

We are Cloudical!
Come to us!

Contact: office@cloudical.io
www.cloudical.io

From Idea to Deployment: Writing a Service and running it on Kubernetes

Microservices are the next big thing. They are understood as one of the foundations of a cloud-native solution, since they allow for faster development, technology abstraction and ease of deployment. In this article, we will look into how to write a microservice, package it and deploy it onto a Kubernetes cluster – but before doing this, we should make ourselves understand, what microservices are and in which way they differ from traditional approaches to software development.

What is a microservice?

First of all: A microservice is still a program. But, to earn this name, a microservice needs to comply to some principles:

A microservice is a functional domain

This principle is the most important one, since it depicts a change in how to think of components of an application. Traditionally, applications are divided into technical components - and switching to a microservices world therefore appears to be straight-forward: Just wrap the technical components (such as a database layer) into microservices, and off you go. But, this approach is misleading, since it does not help in deployment- or update-scenarios, since the technical layer is usually used by multiple other services, thus creating a strong dependency between them.

The better - and recommended – approach to microservices is to create functional domains. This means, a microservice handles one functional task, such as the management of a booklist, completely end-to-end: It exposes a RESTful API, it stores, retrieves, manipulates and returns the data. No other service in a microservices application will be implementing this functionality, all services relying on this functionality will utilize this specific microservice.

Writing a microservice this way, it could scale better and is independent from other functional services, the dependency from other services is a functional one, not a technical. Whenever functionality changes, only this specific kind of service needs to be redeployed, leaving all other services untouched.

Example 1

```
package io.cloudical.cd;

import org.springframework.http.MediaType;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class RestEndpoint {

  @RequestMapping(path="/", produces=MediaType.APPLICATION_JSON_VALUE)
  public ResponseEntity<?> sayHello(@RequestParam("name") String name) {

    final String response = "Hi, %s!";

    return ResponseEntity.ok("{ \"response\" : \"" + String.format(response,
name) + "\" }");
  }
}
```

Code for exposing a simple REST-endpoint (RestEndpoint.java)

A microservice is exposed via a webserver

Usually, a microservice is deployed in a Webserver as Application. It listens to a specific port (most often 8080 or 8000), it either renders a webpage or exposes a RESTful Service API.

Since HTTP or HTTPS is used as communication technology, microservices are better abstracted from each other. They communicate over the network, no direct memory-access between components is possible. This allows for a greater variety of technologies used within a microservice-based application, since the common denominator will be the network-protocol, not the programming language being used. Scenarios, where NodeJS-based services for simple workloads, Java-based services for a deep integration with middleware and Microsoft-R-based services for statistical computations run side-by-side, are not unrealistic and allow for a "Best-tool-for-the-Job"-approach, since all these services would have to have in common, will be a RESTful-API being exposed to each other.

The key to this, is to write a microservices-based application in a way it can be packaged inside a Webserver.

A microservice needs to be self-contained

A microservice is bundled in one package. There must no external server needed to run the service. The term "package" is a bit misleading, though, since it might imply the service to package all in itself. Instead, "package" refers to a deployment package, which needs to be self-contained.

Typically, microservices are packaged as container images. Therefore, a self-contained microservice package

refers to a webserver being part of the container image, with the service being deployed as application inside the webserver. There is no specific technology required for this - any modern webserver being able to run the microservice is suitable.

The only hard constrain is related to the packaging: A microservice needs to be packaged into one deployable and completely self-contained unit. All dependencies need to be part of the package, no external dependencies are allowed. If the service depends on external configuration information, this information needs to be retrieved at runtime by reading environmental variables or config maps.

The main idea is to have a service which just needs to be deployed and started.

A microservice needs to be small enough

The size of a microservice is always something to discuss upon: To some, it needs to be as small as possible, others tend to write something like „makroservices". The truth lies inbetween the two extremes: A microservice needs to have a reasonable size, it should contain all functional logic, should be maintainable by a small team and should have a small overall footprint.

If you are not satisfied with the size of a microservice, you can always split it into two or more smaller services. This is usually easy to accomplish, as all requests to the service are executed using the HTTP-protocol and therefore can be proxied to one of the smaller services, allowing other services to utilize the functionality without knowing anything of the actual structure of the called component. If you run the services within a Kubernetes environment, you have additional capabilities of rerouting requests without having to change a single line of code in the calling components and without having to write a proxy yourself – the open-source service-mesh ISTIO [1] is a viable component to consider.

Writing a simple microservice

As mentioned above, a microservice can be written in any programming language which allows to be deployed on and to be accessed via a webserver. Typically, it is done using Java, Go or NodeJS, since these programming languages either have a long lasting tradition within enterprises (Java) or are easy to learn and very efficient (NodeJS, Go). To simplify packaging, programming languages should be preferred that allow to run completely self-contained, meaning: They should be able to start a webserver and expose the application via this server, avoiding the need to download and maintain an external server by hand.

For this simple example (Example 1), we choose Java with Spring Boot for creating a simple microservice which exposes a single REST-endpoint [2]:

As we chose the Spring Boot framework, a bootstrapping class is required to start the webserver, deploy the application and make the service available to the outside world: Example 2.

This code is able to run locally on a developer's machine for testing purposes, for any other purpose it needs to be packaged into a container image.

Example 2

```
package io.cloudical.cd;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class MicroserviceApplication {

  public static void main(String[] args) {
    SpringApplication.run(MicroserviceApplication.class, args);
  }
}
```

Code to start the microservice-application within a Spring Boot provided webserver (Microservice Application.java)

Example 3

```
# Start with a base image containing Java runtime
FROM openjdk:14-jdk

# Add Maintainer Info
LABEL maintainer="karsten.samaschke@cloudical.io"

# Add a volume pointing to /tmp
VOLUME /tmp

# Make port 80 available to the world outside this container
EXPOSE 80

# The application's jar file
ARG JAR_FILE=target/CGPMicroservice-0.0.1-SNAPSHOT.jar

# Add the application's jar to the container
ADD ${JAR_FILE} service.jar

# Run the jar file
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/service.
jar"]
```

Dockerfile for creating a deployable image (Dockerfile)

Example 4

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    run: simpleservice
  name: simpleservice
spec:
  replicas: 2
  selector:
    matchLabels:
      run: simpleservice
  template:
    metadata:
      labels:
        run: simpleservice
    spec:
      containers:
      - image: cloudicalio/simpleser-
vice:latest
        name: simpleservice
        ports:
        - containerPort: 80
```

A deployment script for running the microservice in two instances (deployment.yaml)

Example 5

```
kind: Service
apiVersion: v1
metadata:
  name: simpleservice
spec:
  selector:
     run: simpleservice
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: LoadBalancer
```

A service exposing the microservice to the world (service.yaml)

```
Administrator: Windows PowerShell
PS C:\Users\KarstenSamaschke\google-cloud-build-sample> docker build -t ksamaschke/simpleservice:latest .
Sending build context to Docker daemon  17.03MB
Step 1/7 : FROM openjdk:14-jdk
 ---> 2a704ad35a86
Step 2/7 : LABEL maintainer="karsten.samaschke@cloudical.io"
 ---> Using cache
 ---> bb975a9e3165
Step 3/7 : VOLUME /tmp
 ---> Using cache
 ---> 33311c163f0b
Step 4/7 : EXPOSE 80
 ---> Using cache
 ---> c53eb6c81696
Step 5/7 : ARG JAR_FILE=target/CGPMicroservice-0.0.1-SNAPSHOT.jar
 ---> Using cache
 ---> d19ee78a938f
Step 6/7 : ADD ${JAR_FILE} service.jar
 ---> Using cache
 ---> 24334e1b372e
Step 7/7 : ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/service.jar"]
 ---> Using cache
 ---> 70fc8b5a56e6
Successfully built 70fc8b5a56e6
Successfully tagged ksamaschke/simpleservice:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and director
ies added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions f
or sensitive files and directories.
PS C:\Users\KarstenSamaschke\google-cloud-build-sample>
```

Fig. 1: Building and tagging a Container image

Packaging the microservice

For packaging a microservice, containers have evolved to be the tool of choice. They abstract the microservice from any environment, providing a well-known infrastructure from the services point of view. Since a container is self-contained, it is simple to run from a orchestrator's perspective - usually, the process is as simple as pulling it from a registry, starting and later on stopping it.

There are several container engines available for Kubernetes, with Docker being the most widely used one. Creating a Docker container is very straightforward: The container is described using a Dockerfile and then the image is built and tagged using Docker's build command. Finally, it is uploaded and therefore stored in a repository, where it can be retrieved by an orchestrator later on based on the tag it was given during the build-process.

This is executed only once, since it is always the same image to be deployed to any environment. Only parameters and configurations differ, the image being used is always the same.

For a Java-based microservice, a Dockerfile might look like this: Example 3.

The Docker build command needs to be run to create the image. It is advised to do the tagging during this stage as well (fig. 1):

```
docker build -t cloudicalio/simpleservice:
  latest.
```

Once the image has been built and tagged, it needs to be pushed to a registry:

```
docker push cloudicalio/simpleservice:
  latest
```

Now, the service is made available for deployment.

Deploying a microservice

Having Kubernetes in place, a microservice stored in a container registry can be deployed on the environment and made available to the public using YAML-files.

First, let's have a look at the deployment for the service: Example 4.

This script defines the container image named cloudicalio/simpleservice:latest to be deployed to Kubernetes. Using the replica-tag, it is defined to run in two instances. The deployment (called a "pod" in a Kubernetes context) will be named simpleservice and has the label run: simpleservice attached to it. The label is very handy to identify the pod for making it available to the public.

```
Administrator: Windows PowerShell
PS C:\Users\KarstenSamaschke\google-cloud-build-sample> kubectl get pods
     NAME                                READY   STATUS    RESTARTS   AGE
simpleservice-668f8c7657-7dk8w   1/1     Running   0          16s
simpleservice-668f8c7657-l2gxc   1/1     Running   0          13s
PS C:\Users\KarstenSamaschke\google-cloud-build-sample>
```

Fig. 2: Checking for a deployed Pod

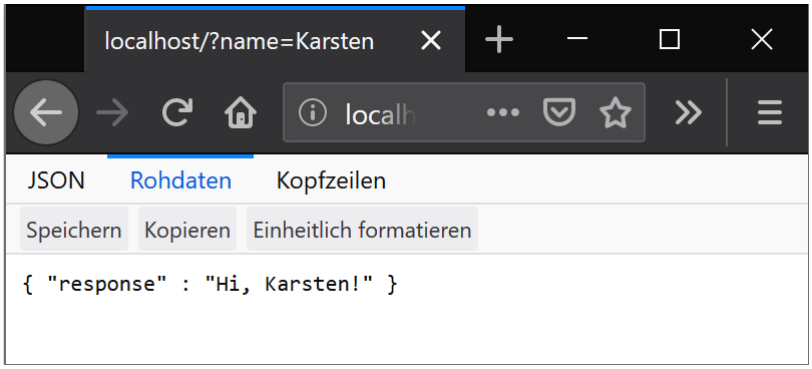


Fig. 3: The service can be accessed using any web browser

To actually deploy the service, the YAML-file needs to be transferred to the Kubernetes cluster. The kubectl-command line interface is used for this:

```
kubectl apply -f deployment.yaml
```

Within some seconds, the deployment will be transferred to the cluster and made available (fig. 2).

Unfortunately, the service is not accessible to the public yet - a service definition needs to be established, allowing the microservice to be exposed: Emample 5.

Kubernetes provides different kinds of services for exposing a deployment. In this scenario, we created a service of type LoadBalancer, which is exposed on the cluster and is assigned an external IP-address. Whenever a request hits the IP-address, an instance of a pod labelled with run: simpleservice (as defined above) is invoked and is required to handle the request. The Load-Balancer listens to the HTTP-port 80, the pod being involved - or more specifically: The microservice in the pod - is expected to listen on this port as well.

The file needs to be transferred onto the Kubernetes cluster similarly to the previous one:

```
kubectl apply -f service.yaml
```

After a few seconds, the service can be accessed and used (fig. 3):

Don't do it this way!

It is very easy and straightforward to write and deploy a microservice, isn't it?

To be open: When you do it the way it has been shown to you in this article and in many tutorials, you will run into a lot of problems, since it is one of the worst approaches to writing and deploying microservices. The reason for this: There is no automation in place and the process is very error-prone doing it this way.

The next step - after having it done exactly once this way for learning purposes - needs to be the establishment of a CI/CD-pipeline, which will handle build, tagging, upload and deployment for you. An approach to this will be shown in the next issue of this magazine.

Sources:

- 1. ISTIO-framework: https://www.istio.io
- 2. Sources for this project: https://github.com/cloudical-io/simpleservice-sample

Karsten Samaschke
CEO of Cloudical
karsten.samaschke@cloudical.io
cloudical.io / cloudexcellence.io

CLOUDICAL

OpenStack – everything for Infrastructure as a Service

Mark Collier talks about the future of OpenStack

At the German OpenStack Days – DOST – The Cloud Report had the opportunity to talk with Mark Collier, COO of the OpenStack Foundation and one of the founders of OpenStack.

Hi Mark, nice to meet you! Please, tell us about you and the OpenStack Foundation.
Hi, I'm Mark Collier with OpenStack Foundation. Originally, I was one of the founders of OpenStack, the project before we had the foundation. We started this project when I was working for a company called Rackspace, which is a large hosting cloud provider in the US and UK. As a cloud provider we were trying to figure out how to build a competitive cloud and how to go up against very large tech companies. I knew that we were going to need a lot of engineering, a lot of help, and a lot of developers. And we were trying to hire a lot of developers at the time. We already had a cloud. Rackspace had one of the first clouds along with Amazon, it was actually before Microsoft or Google entered the business. But we knew that in order to get access to technology we needed to have partners and really build an ecosystem around it. So, the decision was to embrace open source fully and that's what became OpenStack. We launched that nine years ago and 2010 we partnered with NASA, the space agency, and 25 other companies. And the idea was that for infrastructure as a service we wanted to build a large community with many companies and many individuals contributing and about two years after we started the project, we created the foundation, so, that became the OpenStack Foundation. And the reason for that was really just to continue the growth, the project needed an independent home, nonprofit, not tide to one commercial entity. So, we could be a neutral third party and really help to facilitate the communities´ growth and bring in much bigger players like IBM, Cisco and Red Hat that we wanted to see in the foundation in order to make sure that OpenStack had a long term future. That kind of what brought us to the foun-

dation, and I am the COO of the foundation and since then it has really been a wild ride of trying to just manage the growth.

We have tremendous number of users in the technology landscape, it changes very quickly. We had a lot of companies come into the market to build products and a lot of developers come in to contribute to the code. In the most recent OpenStack release, Train, that was made available in October of 2019, 25,500 changes were committed by more than 1,100 developers from over 50 countries and 165 organizations. This pace of development makes OpenStack one of the three most active open source projects in the world. A lot of people don't know that because we actually don't use GitHub which is where a lot of the stats people are looking for. We actually outgrew GitHub pretty early on, so we use Git technology which is an open source tool for code collaboration and code repositories, so we host our own Git. We use Git but we don't use GitHub, the commercial product. And because we use our own Git a lot of data doesn't show up when people just do a quick study. In recent years there has been this miss that "everything happening in open source is happening in GitHub". Linux Kernel, Chromium and OpenStack, the three most active open source projects in the world and none of them use GitHub because... GitHub is great, there is nothing wrong with GitHub, but it has come along more recently and it got a different collaboration model. So, we use a combination of our own Git, we use a tool called Gerrit for code review which is an open source project that google released several years ago, so we are active in that and we use a tool called Zuul for our code testing and that was developed within the OpenStack community. So, between Git, Gerrit and Zuul we manage our collaboration and flow of code. We have hundred thousand members now in 180 different countries, so, managing that size community with all the developer activity we needed

a lot of tools for ourselves. But now we are starting to see other companies adapt those tools who want to write open source software in a similar fashion to how OpenStack was developed and some of those projects are actually now part of the OpenStack Foundation. That is one of the updates that we wanted to share while we are here, the OpenStack Foundation itself have now evolved to help host other open source projects.

The **idea of OpenStack** was that for infrastructure as a service we wanted to build a **large community** with many companies and many individuals contributing.

Kubernetes is more than a trend. The CNCF builds the state of the art ecosystem for containerized workloads. How does this effect the present and future of OpenStack?
OpenStack is still the third biggest open source project in the world, but there are now four other open source projects that are part of the OpenStack Foundation and I can give you a quick rundown of them: there is Airship StarlingX, Zuul and Kata Containers. They all have some container related tiles, because containers are very useful. Since containers and docker really came along that had a very positive influence on OpenStack, so, we have used containers, they have been really useful making OpenStack more reliable because we can containerize the control plane and we can quickly spin up and manage and upgrade OpenStack itself, because the OpenStack servers are in containers. OpenStack is in some ways an application, a Linux application. It is an application that provides infrastructure as a service. So, we´ve used containers for several years to make the control plane easier to upgrade, the same reason any application developer love container. So, we´ve used it in that way. Kubernetes and OpenStack have proven to be an extremely powerful combination. More than half of our users in our user survey tell us they are using Kubernetes on top of OpenStack. And so, I think a lot of people have a misunderstanding thinking somehow, they are competitive, or one replaces the other. In reality, Kubernetes runs on top of programmatic infrastructure whether that is AWS, Azure, Google, or an OpenStack Cloud. In addition to thousands of private clouds, OpenStack powers dozens of public clouds around the world. This includes a large footprint in Europe with providers like City Network, OVH, and Open Telekom Cloud. Recently a new project in Europe was announced, Gaia-X, to build even more clouds for Europe driven by data sovereignity requirements which are especially strong in Germany. This will be a huge opportunity for OpenStack public clouds who are already supporting this requirement throughout Europe. It is really playing that infrastructure as a service layer role to make the bare metal infrastructure programmatic. With the last several releases of OpenStack bare metal provisioning becoming a key part of that, so you can even if you want to run Kubernetes on bare metal without virtual machines which isn't usually a good idea for most use cases in sense of security, but you can if you have really tightly secure environment where you control or you are running only app and is not multitenant... You can do that, and you are still running OpenStack, so, sometimes people conflate OpenStack with virtual machines which is not really accurate.

So, OpenStack starts of bare metal, has an occasion with keystones and has whole series of components that make up how you control access to resources, that are programmatic, and then virtual machines become kind of an optional piece of it. Still, most people who run it, run virtual machines with OpenStack, but you can run your containerized Kubernetes cluster in virtual machines which is really how most people do containers in production the run in VMs because of the security or if you really feel bold you can run it on bare metal and you can still use OpenStack for the bare metal piece . So, containers have been a really good thing for OpenStack and I think there are a lot of people that think it is the opposite for some reason but it is just the nature of when something new comes along people think that whatever was there at the time must be in jeopardy but, the reality is that all these things build on each other. Linux didn't go away when OpenStack showed up, Open Stack builds on Linux. This is something that we try to clear up. We have seen some of our largest users are running Kubernetes with OpenStack. With the new projects as the Foundation has evolved and

we know how we are now hosting and helping communities around additional projects outside of OpenStack, they all have some sort of container tile. So Kata Containers specifically was the first pilot project we did that wasn't part of OpenStack, but it was part of the OpenStack Foundation. It is a little bit confusing since they both have the word OpenStack in them. We are not trying to add hundreds of projects, we´ve only added four, so, we are taking it slow and deliberated and only hosting new projects, which really makes sense for our community.

OpenStack is one of the most complex open source software solutions at the moment. The so called „core projects" are stable and enterprise ready enough to build a robust infrastructure as a service platform. But the Foundation has so many small additional projects for OpenStack like Sahara to be an anything as a service platform. How do you see the future of these projects? Is OpenStack a good choice for anything as a service or should it better be used for infrastructure as a service?

I think that the primary focus is going to be infrastructure as a service, and that one of the things that happened somewhere in the history of Open Stack is that there was a period of time where we made the mistake of thinking that if it needs to go on a cloud it should be open source. That was not a mistake, but the second piece was therefore should be OpenStack. That was the mistake. What we realized was we want everything you need to build a modern cloud to be open source, but it doesn't mean it all needs to go in the OpenStack and that was kind of a revelation two or three years ago. Like: wait a minute, we should just focus OpenStack on what it's pieces. We should work with the other communities, we should help establish new communities that are building and solving problems, that work with OpenStack. But it doesn't need to go into OpenStack. So, if you see in our Foundation again like KataContainer, Airship, StarlingX, they are all things that our community is helping with but we are not putting them into OpenStack, they are their own communities in a sense and they are building their own deliverables and releases on their owns schedules and have their own governances. Trying to shove all that into OpenStack was probably the wrong way to go. Now we´ve taken more realistic views. If you look in any OpenStack cloud, it is never just OpenStack and it never really was. There is Linux and KVM as a Hypervisor, you have RabbitMQ and things written in Python, so there has always been a combination of different open source

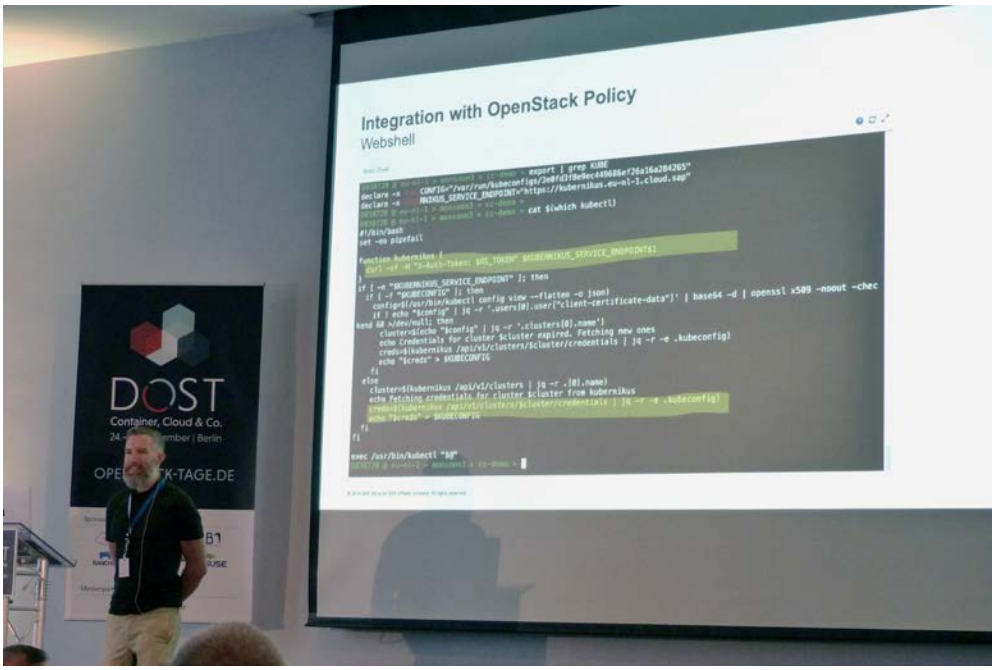
things in practice when you actually take it in production. But we are just trying to take that bigger view and go. There is so much open source now, it is a wonderful thing, but it's also creating a new set of problems for our users. So, what can we as a Foundation do to help? A lot of that involves testings, it involves working at cross communities and sort of saying not everything should go into OpenStack , not everything should go into Kubernetes, not everything should go into Linux and if you look into AI machine learning you have things like TensorFlow which isn't at the Foundation and that is totally fine. The Foundation side is sort of irrelevant, what really matters is how can we make all of this stuff work together? So, we are doing much more with testing, we are doing tests today within the Kubernetes community, every new Kubernetes release will not come out if it breaks on top of OpenStack, because OpenStack is part of their testing, just like they are testing against Amazon, Google, Microsoft. In the same way a new release of OpenStack won't come out if it breaks on Kubernetes. So, where these pieces need to fit together, we are trying to serve the market better by collaborating, cooperating and testing, testing, testing, and then when there are gaps, we will create new projects. But our focus is on infrastructure.

There are projects that sort of cross the boundaries but for most part they are most like installers so like Magnum is a Kubernetes project within Open Stack, the important thing it does is that it installs Kubernetes. It doesn't replace it, it installs it in an automated way. In France CERN has a 300,000-core OpenStack Cloud and they run hundreds of clusters of Kubernetes on top and they use Magnum to do that. So, there are integration projects, but they are not trying to replace other functions outside of an infrastructure service.

Kubernetes and OpenStack have proven to be an **extremely powerful** combination.

Some voices in the industry say that the time of OpenStack is over, what is your statement about the general future of OpenStack as IaaS platform?

I think news like that definitely get clicks or people wouldn't put it on their LinkedIn. They have some reasons doing that, but all we do is to talk to our users, our developers and our ecosystem every day and try to help them out. While we have



Michael Schmidt from SAP talks about the newest inventions of OpenStack at the DOST 2019 in Berlin.

been here at this event, we talked to BMW who's growing their footprint this year by 30 or 40 %. BMW is a massive OpenStack user and they are growing, they actually can't grow fast enough. They also use Zuul which is a CI/CD tool as a top-level project. We talk to customers or users like that and they are just growing their footprint and we have new users that come online and talk about how they are solving problems with it.

Honestly, I've been doing this for nine years and I don't think there has been a single year where there wasn't some headline like "Open Stack is dead". And I say "well, I've heard this before" and then I go back to work hard to make it better.

I would say that one of the challenges is that the technology world is very obsessed with what's new and hot and young and the new thing and what's next. But if the world is running your software production you are not really next you are right now. Once you get adoption you are not in the future. It doesn't mean you won't be there in the future, it just doesn't fit the narrative of what a lot of people are getting excited about and want to read and tweet about, because they are just quietly solving problems. I mean Linux is now widely adopted then ever but probably getting less press than ever. But Linux is everywhere. And OpenStack is similarly like that. The largest power grid in the world, state grid of China runs OpenStack. So, if you're in China and you're getting electricity it is coming through an OpenStack cloud. China railways, the largest rail system in the world, they run OpenStack. The billion tickets a year, they sell on their trains, run on an OpenStack platform. If you are in in the US and you have AT&T as your carrier of horizon your 4G and 5G calls are going through OpenStack. If you look at ecommerce there are a lot of automobile companies like BMW. So, OpenStack has never been more pervasive, it more becomes of the fabric of the infrastructure you use every day. Less people may tweet about it, but our goal was always to build a useful piece of technology and get it widely adopted, so in that regard I would say we are very happy with where we are nine years into the project.

We see the same here. Three years ago, all the companies in Germany were really interested in OpenStack, like BMW and other big vendors, and then we had less requests last year, mostly the customer are talking about Kubernetes, Open Shift. And at the moment we see the second wave of this. We have so many requests for OpenStack and so many companies are building private clouds at the moment based on OpenStack.
Great! Tell them to call me! The funny thing about open source and why we love coming to these events is we always meet new users. Because there is nothing in the software that phones home and tells us, so often time I meet a user and I say "these are the reasons you should use OpenStack" and they say "we have been running it for three years ". We literally don't know who runs it unless they come and reach out to us or we meet them at one of these events. You are giving me good news.

Some users tell us that "OpenStack is great to use, but it is hard to operate". It is one of the most complex open source technologies at the moment. What do you recommend for these users? Use Vanilla OpenStack? Use vendor OpenStack like Redhat, Suse, Mirantes or Canonical?
Everything depends on the individual user, but I think that one of the things we were very deliberate about when we set up the OpenStack project and the Foundation was that we wanted there to be a strong ecosystem. Because that is part of what keeps it going a long term, as companies can help users adapt the software and those companies also employ most of the developers that write the software. So, it is a virtuous cycle. We've never been anti vendor by any means, we have actually been very much about being that one of the three forces that makes a platform successful as a strong ecosystem. So, most users, I think, really should work with a vendor. It doesn't mean you can't do it yourself and be successful, there are many examples of

We should work with the other communities, we should help **establish new communities** that are building and **solving problems**, that work with OpenStack. But it doesn't need to go into OpenStack.

people that have. But I usually tell people that it doesn't hurt to start talking to the vendors. There are many options out there, there are a lot of major players. Every Linux distribution has an OpenStack product like Mirantis and other OpenStack distributions. A good place to start.

Another thing that I would point to is that both Airship and StarlingX are new open source projects within the OpenStack Foundation, but they both essentially contain OpenStack. They both take the best of OpenStack and Kubernetes and bring them together for specific use cases. So, Airship use case is sort of unique in that they have hundreds of sites, they are running hundreds of OpenStack clouds and they want to run thousands, because it is pushing out to the edge of their network. They want to be able to do zero touch provisioning, upgrades in the field. These are challenges that we see in the operation of OpenStack. AT&T reached a scale where they just didn't have a choice, just doing it in a hard way which is not working at all, they HAD to do it completely automated so that's why they created Airship. As it turns out, not everyone wants to run a thousand clouds at the edge, Airship is actually applicable to people even though they only have one cloud. Because they create a repeatable upgradable lifecycle management system for the whole infrastructure from the bare metal up and it allows to mix and match Kubernetes and OpenStack depending on the workload and connect it all together. So, even though there is telco and they did it for 5G is not really specific to that, it is really just about automated deployment, upgrades infrastructure including OpenStack and Kubernetes. Airship is a much newer project, so people have to evaluate it of course. It is not nine years old like OpenStack, but it is relying on a lot of proven technologies.

We just talked to a company last week. One of the largest coffee chains in the world is looking at Airship as a way to get OpenStack into the edge of all their coffee shops. So Airship is really interesting.

I've been doing this for **nine years** and I don't think there has been a single year where there wasn't some headline like "**Open Stack is dead**". And I say "well, I've heard this before" and then I go back to **work hard to make it better**.

StarlingX is also combining OpenStack and Kubernetes. It is very much designed for both edge and industrial IOT use cases. So, those two projects are somehow similar, but they are both kind of run independently, there are common components between them. StarlingX is also relatively new, but it has been originated by Wind River and Intel and a few other companies, and that was originally a commercial product that Wind River built based on OpenStack. But then they decided to open source it and came to OpenStack Foundation "We have this commercial product, we want to build a community around it, we want it to be open source. It is based on OpenStack, so bring it into your Foundation makes a lot of sense." So it is kind of OpenStack and a bunch of other stuff that optimize it for small use cases. Two servers type deployment where you have it at edge. Because it started like a commercial product it is already running in production with some big industrial players for example: China Union Pay that crazy big China financial services company. They do five hundred million dollars a year in transactions and it is fifty thousand transactions a day. So, they are in mobile payment in China and dominant everywhere you go in China as payment form. For five years they have been running OpenStack, but now they are interested in trying out StarlingX, because they have been using OpenStack more for the data center side, taking all the data coming in from all those transactions and crunching the numbers and storing it and everything. But what they want to do is actually push OpenStack all the way to the edge into the corner of sale where cards are being swiped. This is a radical use case for putting something out really far to the edge, so they are looking at Starling X for that, because it allows you to slim down to really just some sub components of OpenStack and/or Kubernetes in different combinations to meet the edge user case. These new infrastructure use cases are one of the main reasons we have new project at the Foundation, because OpenStack alone can't do it all, but as a Foundation we want to help all these people adopt open source. Our mission is to help people build and operate open

infrastructure, wherever infrastructure is going and growing we want to be there, we want open source to be a viable alternative to proprietary solutions and often times that means collaboration, writing software and testing. So, that is kind of where we are going as a Foundation with our community. OpenStack is at the heart of that, but there are other cool stuff going on making sure open source is viable in all these new environments.

The foundation has cancelled the COA exam in March this year, is there a new exam planned?

When we talked about winding down the COA, a lot of people came out of nowhere to say "We love it! We need it and we want it to continue." The outcry for support and demand for it to continue took us pleasantly by surprise. So, we said, "let's find another option" and we did. So, Mirantis approached us about helping to continue the administration of it and keep it going and update it and upgrade it for the new versions. Due to the heightened community demand, the OpenStack Foundation partnered with Mirantis to continue delivering the COA exam starting in late 2019 (www.openstack.org/coa).

Do you want to add something?

For us it is fascinating to hear the feedback and to hear what the perceptions are out there. We have a constant struggle for setting the record straight as far as "Is OpenStack dying?" It has been going on for nine years, it is a bit repetitive but we want to know where people are confused and where we can do a better job of educating the market, because for the problem that is solves there is nothing else out there that is proven at scale. We have ten million cores of compute managed by OpenStack today. And it is probably much higher than that. That is just the people who tells us they are running it and we add up the numbers so it could be many times higher. But we have a user survey we put out every year and we are going to publish a report around that later this year. We just closed it for 2019. The twentieth release of OpenStack got Train that is on October 16. Those are some of the upcoming things happening. Our next Summit will be in Shanghai, first time in mainland China. That should be exciting. It is an exciting time to do business in China.

It is a crazy world, but open source can bring us together. And one of the things that I love about it is that we have members from 187 countries that are part of our Foundation which is almost every country in the world. So, we try to work without regards for borders, corporate borders or national borders. Just with the OpenStack and not getting caught up in the political drama. It is a good thing we can work on common goals and not having to worry about our politicians.

These are very nice words at the end. Thank you for your time!

The Interview was conducted by
Kim-Norman Sahm and Friederike Zelke

CLOUDICAL

Strategy for the Future

Marco Görgmaier gives insights into the BMW Group's cloud approach

The Cloud Report had the opportunity to talk with Marco Görgmaier, Head of DevOps Platform and Cloud Technologies at BMW Group, about cloud strategy, technology, implementation and the future.

What cloud strategy does the BMW Group pursue?
At the BMW Group, we pursue a multi cloud strategy and take a very strategic approach when selecting our cloud partners. Today we have three large public cloud providers – Hyperscalers - with whom we work: AWS, Microsoft Azure and the Google Cloud Platform. On-Premises we rely on Openshift from RedHat and OpenStack for Infrastructure-as-a-Service. By using multiple vendors, we avoid a single vendor lock-in. The risk here is the increased complexity. To manage this complexity, the various providers are used according to their strengths in the various domains: customer, vehicle and factory (production & logistics). Thus, the domains have clear degrees of freedom in the use of the different providers. For example, one domain works with AWS, another with Azure. Between domains with functional and technical dependencies, however, we ensure uniform use and stringency.

We also see **Kubernetes** as a central building block for switching between different cloud providers.

How is the cloud strategy implemented?
We provide our teams with very fast onboarding for all building blocks and cloud providers on our platform. Access and infrastructure are provided to the teams within an hour. For example, if a team chooses AWS, they can start developing immediately. We provide building blocks or templates that the teams can reuse. With our DevOps platform we follow a very integrated DevOps approach, i.e. it is important for us to optimize the value flow in software development and operation. The focus is clearly on enabling agile in-house software development. This is exactly the goal we have been pursuing for several years. This is an approach that we want to massively strengthen and further expand. The DevOps platform with the corresponding cloud-native technologies is the basis for this, which we provide to all teams worldwide. The platform team currently employs 80 people, while the BMW Group IT employs a total of 5,500 people in domains and products.

Which technologies does the BMW Group rely on and what should be considered?
Where possible, we rely on cloud-native open source technologies and solutions. A very central technology in the cloud environment for us is Kubernetes. Kubernetes is used to orchestrate container systems. The open source solution automates the setup, operation and scaling of containerized applications.

We also see Kubernetes as a central building block for switching between different cloud providers, which gives us a lot of flexibility. Especially when we build complex and large applications, we have to consider very carefully what

should be used natively by a vendor like AWS or Azure, although you might risk a vendor lock-in and what should be built agnostically to remain changeable. For new and smaller applications, it can be crucial to focus on the speed-to-market. In this case, we deliberately use the native services of a cloud provider, even though we risk a lock-in. In the end, this is always a trade-off between switching costs and development costs in order to remain agnostic. Which way is the more entrepreneurial and financially sensible is therefore always a matter of weighing up from application case to application case.

The topic of skills, training and certification of our teams is therefore also decisive. In order to be able to decide which solution makes the most sense for a certain application and which can be best built architecturally, appropriate know-how is required. This is why we send our employees to the training and certification processes of the major cloud partners with whom we work. In addition, the BMW Group also has a targeted internal development of skills for all employees. In addition to a wide variety of training courses, this also takes place via a very active in-house community, which above all also ensures "sharing of best practice" between the teams.

Are there in-house solutions? How do employees deal with them?

Yes, we bundle cloud services in our BMW Group DevOps platform. The platform also offers a secure foundation and integrated security tools. The platform ranges from our cloud solution in the BMW Group data center to public cloud providers. This helps the development teams enormously because they can focus on

Marco Görgmaier

- With the BMW Group since 2012
- Previously at Audi
- Lastly Head of Strategic Planning and Innovation Management at BMW Group IT - in this role responsible for the IT strategy and the orientation of IT towards a BizDevOps setup with End2End setup in the product portfolio.
- Since January he is responsible for the domain "DevOps Platform and Cloud Technologies" with the corresponding IT products. The international team currently comprises almost 80 employees.
- The BMW Group IT DevOps Platform is the platform for the development teams of the BMW Group on which they can develop modern microservice applications in the cloud and operate them with higher quality and stability.

Increasing our internal software engineering **competencies** and the ability to design and develop first-class software solutions in-house are an **integral part** of the BMW Group IT strategy.

their core task, namely developing and operating MicroService applications of the highest quality. So far, this has been extremely well received. As already mentioned, we also benefit a lot here from the development of our active in-house community and our numerous training courses - the right skills are a core requirement for BizDevOps maturity. This year alone we have already enabled more than 500 developers through our trainings.

Increasing our internal software engineering competencies and the ability to design and develop first-class software solutions ourselves are an integral part of the BMW Group IT strategy. The Back2Code initiative has been shaping this internal competence development for two years and provides a framework for it. Through a variety of measures - from Meet-ups to information events to regular Coding Dojos - we ensure the indispensable exchange of experience. With the so-called Back2Code Campus, we have set up an innovative three-month program this year to ensure the best possible qualification of our software engineers. In accordance with our understanding of the role of a software engineer, we address the various tasks from requirements management to design and development as well as the operation of the software solutions. The participants will be released from their usual work for the duration of the program and trained in the latest technologies, tools and working methods by experienced software developers in groups of maximum 15 people. Upon completion of the program, participants return to their internal feature teams and apply what they have learned directly.

Critical TechWorks, which was founded last year as a joint venture, also plays a key role in in-house software development and platform usage. At the locations in Porto and Lisbon, competencies in the areas of premium mobility, automotive software engineering and the development of solutions for onboard and offboard applications are bundled. Of course, in this case, too, the approach applies: Cloud First. It is a central concern of ours that the teams that we have on site and that are also very much in contact with the teams at the other locations share insights on the use of the cloud at the BMW Group and the experience gained.

Fig. 1: BMW Group High Performance D3 platform

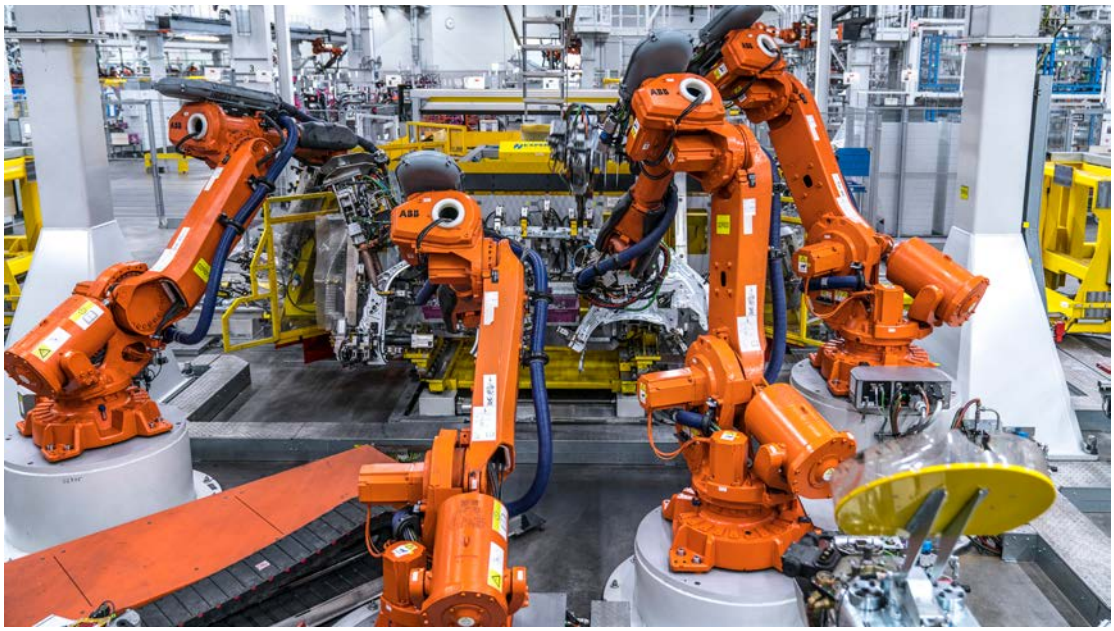


Fig. 2: Smart Data Analytics in Production System

Are there partners with whom the BMW Group cooperates?

Yes, there are, of course. For example, we see the large cloud providers as partners with whom we work at eye level. In April 2019, we introduced the Open Manufacturing Platform together with Microsoft. This is based on Microsoft Azure and will enable us to work with other companies to quickly deliver innovative software solutions to the manufacturing industry. This cooperation is not specifically intended for the automotive industry, but the goal is to establish an open technology platform for Smart-Factory solutions across all industries. The relationship is clearly shifting from "customer service providers" to strategic allies working together on solutions. The focus here is on learning from each other in a spirit of partnership, no longer just on selling licenses.

And last but not least: What does the future look like?

The future is, of course, difficult to predict. We have extremely short development cycles and enormous investments in new technologies in the cloud environment, especially with public cloud providers. I think this will also lead to a boost in innovation for us as a result of rapid technology development. If we succeed in the partnership with the public cloud providers already mentioned, this will generate an enormous transfer of innovation and, in the automotive sector, the decisive development of skills in cloud-native technologies. Especially for intelligent security solutions, there is enormous potential for the BMW Group.

The multi cloud strategy naturally involves complexity and we will have to invest more time in managing the interfaces and integration options. The central question will continue to be in the future: How do we deal with this complexity between the individual cloud providers we use? I think that this will be a crucial challenge for all large and internationally networked companies in the future. Networked services will also have to scale internationally. Due to regulatory requirements in individual countries alone, this requires the use of different providers.

All in all, I see a clear shift towards the public cloud to cover our enormous demand for networked services in production, logistics and, of course, above all in the services for our customers in our vehicles.

The interview was conducted by Julia Hahn

CLOUDICAL

Energy Awareness in Cloud Technologies

The trend away from the traditional data center towards the Hyperscale Cloud Data Center, which has started to gain momentum in the last decade, has also had an impact on the annual power consumption of ICT. The following article describes how the power consumption of data centers has changed since then, how it will change and how the cloud contributes to it. Furthermore, current research on energy efficiency in the cloud is presented and possible problem solutions are discussed.

When I talk to administrators and developers about the cloud, we often think of the same positive aspects: Better utilization of existing hardware, high availability, flexible scaling. But I'm not just a developer, I'm a scientist, too, and I'm concerned with the energy consumption of software and hardware. A topic that fits perfectly into our time, in which students go out on the streets and demonstrate each and every day, for a better awareness of global warming and environmental pollution. As Jeff Barr, Chief Evangelist for AWS, described in his blog „Cloud Computing, Server Utilization, & the Environment", the way to the cloud can reduce energy consumption by up to 84%. This results from "[...] 77% fewer servers required (i.e. cloud requires only 23% of the number of servers required for the same workloads) by 71% more efficient servers [...]"[1]. So, can we see the cloud as the savior of energy consumption in modern data centers? Well, that would be going too far. Although Jeff Barr's statement in 2015 is absolutely correct, other factors play a role in the change of the data center landscape. In the following I will briefly explain which changes will occur and why we should deal with them more intensively, on the one hand, of course, when thinking of our children and future generations and on the other hand in the sense of meeting the requirements and challenges of administrators or developers.

Development of the cloud and consequences

In November 2018, Cisco released its Cloud Global Index[2]. An overview of the changes in data centers over the last

Photo by taner ardali on Unsplash

few years, including a forecast up to 2021, showing that the number of Hyperscale data centers will grow by 13% annually between 2016 and 2021 (see Figure 1).

In contrast, the number of classic data centers will decline slightly. Accordingly, the overall energy consumption of data centers should also be reduced, based on the facts already learned. However, this is a misconception, because the already mentioned features such as increased accessibility or flexibility also increase the demand for cloud services. Data in the cloud, check, apps in the cloud, check. A life without the cloud is no longer imaginable and dependence continues to rise. As a fact, already between 2016 and 2018 the traffic to cloud data centers increased annually by 27%.

So we're in a vicious circle. With every improvement of the cloud (and also its energy consumption), the demand for cloud services increases immensely, which leads to additional data center infrastructure and so on. One major reason for the increasing data flows is online video streaming, as the Shift Project shows in its study „Unsustainable Use of Online Video". According to this study, 60% of the total online data flow is generated by online video streaming. Further triggers are increasing computing-intensive applications in the field of artificial intelligence and the confirmation of the proof of work of Bitcoin and similar blockchain systems. All these factors result in an annual increase in worldwide electricity consumption, which has an impact on the climate on the one hand, and on administrators and developers in terms of price on the other. More and more researchers are becoming aware of this problem, especially from a climate point of view, as they try to find or

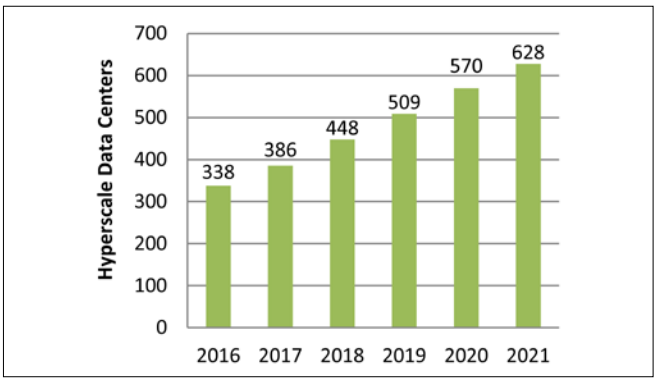


Figure 1 Global hyperscale data center growth[2]

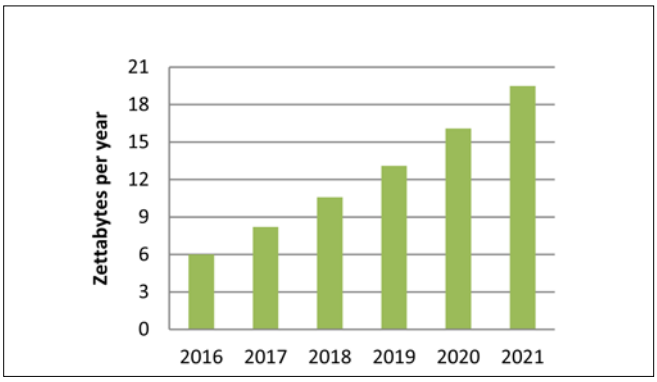


Figure 2 Cloud data center traffic growth[2]

at least draw attention to solutions in the field of Green ICT (information and communication technology). For example, Anders Andrae of Huawei Technologies (Sweden R&D Center) discussed the global power consumption of data centers in his lecture „Predictions on the way to 2030 of internet's electricity use"[3]. In 2018, this amounted to around 211 TWh, representing 1% of global electricity consumption. In October 2017, he predicted that by 2025, data centers would use 1200 TWh, and in the worst case even more than 3000 TWh of energy[4]. This prediction was slightly adjusted over time and extended to 2030 but remained in many people's minds. In this respect Ralph Hinteman et al[5] are concerned with a comparison of different prediction models, in order to be finally able to set up their own model (see figure 3).

Current measures and achievements

Even if these models differ from each other, they all have one thing in common: the power consumption of data centers will continue to rise. The Borderstep Institute is currently working on the study „Energy-efficient Cloud Computing Technologies and Policies for an Eco-friendly Cloud Market"[6], which focuses on sensitization and solution finding. Workshops and expert interviews have already taken place for this purpose. There are also projects that deal intensively with the efficiency standards of computer systems. ISO Standard 23544 (Information Technology - Data Centres - Application Platform Energy Effectiveness (APEE)) is currently under development[7]. In addition, ETSI - EN 303 470 has been an active standard since March 2019 and deals with „Energy Efficiency measurement methodology and metrics for servers"[8].

Furthermore, research projects are dedicated to topics that directly affect data centers, for example, to the energy efficiency of software, which implicitly influences the energy consumption of computer systems. As shown in figure 4, IT as a whole accounts for approximately 45% of data center power consumption.

Current research with focus on software

The load driver is the software running on the machines. In order to be able to draw more attention to this subject area, the awarding of a blue angel for energy-efficient software is currently under development in Germany[10].

All in all it can be said, that the demand for energy efficiency in the ICT sector is increasing. There are an increasing number of projects that are put in the public's focus. I was able to experience this myself, because when you are dealing with a practical topic such as the energy efficiency of container systems, for example, you are often rejected at

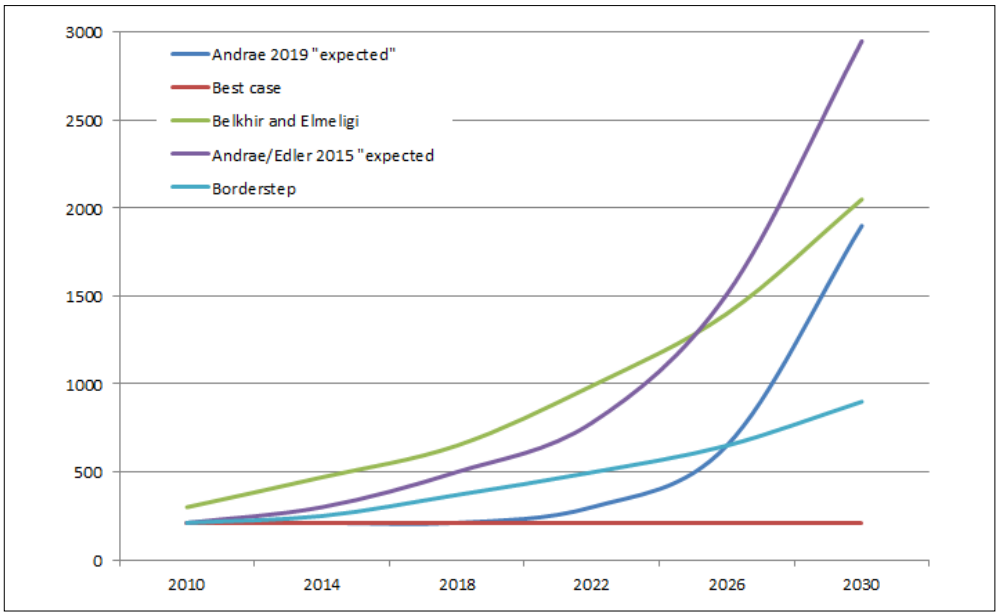


Figure 3 Energy consumption of servers and data centers worldwide – forecasts to 2030[5]

conferences either because you are working too practically in terms of a research topic or because you are working on a too exotic topic for practitioners. Actually, it should be clear that if I design my containers to be energy efficient, the huge mass of containers can also save a lot of energy. Then why isn't this interesting for administrators and developers? The reason is that end users, for example a user of SAAS, FAAS or PAAS, do not notice the increasing energy consumption of the data centers, because with an increasing number of servers, more and more users are added to the cloud. The costs remain evenly distributed and the user continues to pay according to the duration of use of his services. But services are not the same and have to be considered in different ways, because there are acute fluctuations here. A small website in a container, with few accesses per day, does not consume as much power as a REST-API including a database with millions of entries accessed every minute. In my opinion, we have to rethink here. If we know how much energy our services consume and if we optimize them accordingly, the overall power consumption of data centers will decrease on one hand, and price models in the cloud area can be adjusted with even more flexibility on the other. A calculation based on CPU and RAM is not enough, because other factors such as network IO also play a role. It's not that we can't control these things. But both admins and developers should be encouraged to be energy efficient. I'm currently working on recommendations for energy-efficient container handling. The administration and orchestration as well as the development of applications with containers play a role. The goal is to save energy every time a container is started and stopped, every time it is scaled, and even beyond that over the entire lifetime of the container. Especially in scaling, the consideration of the power consumption of the services and not only their CPU utilization offers various possibilities to save energy. As I mentioned before, this results from the sheer mass of containers used in cloud data centers. As can be seen in table 1, the number of workloads and compute instances per server, such as VMs and containers, is rising with an annual rate of 22%.

Conclusion

As already mentioned, even the slightest effect on these calculation instances can make a big difference.

In conclusion, the cloud has made the lives of administrators, developers and end users easier and more adaptable than ever. As development continues in the PAAS, SAAS, FAAS and similar areas, there will be more and more data centers around the world that will consume more and more power. In terms of a sustainable and environmentally conscious way of life, administrators and developers

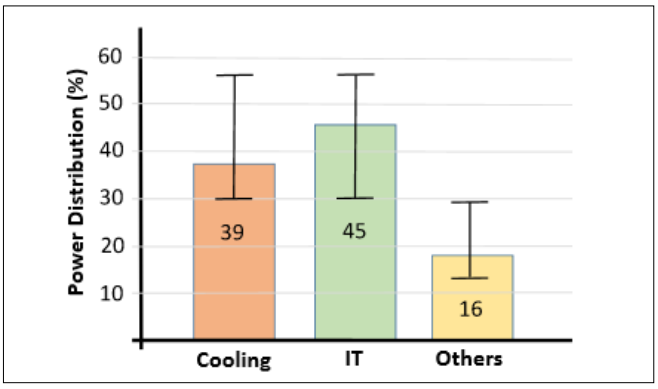


Figure 4 Power distribution in data centers[9]

Table 1 Workload and compute instances in cloud data centers[2]

2016	2017	2018	2019	2020	2021	Annual growing rate
199	262	331	393	459	533	22%

also have to face new problems and have to take responsibility and thus, solutions have to be found accordingly. Current research clearly points out the problems and offers first solutions, which give administrators and developers not only a clear conscience, but also the opportunity to offer new and more flexible pricing models in the cloud.

If you are interested in supporting me in my research in the field of energy efficient use of containers, I would be glad if you could participate in the following short survey on your usage of containers and, in particular, of Docker: http://www.incubator-solutions.com/survey

Sources:

- 1. Jeff Barr, Cloud Computing, Server Utilization, & the Environment, AWS News Blog, June 2015, https://aws.amazon.com/de/blogs/aws/cloud-computing-server-utilization-the-environment/, last accessed 11/09/2019
- 2. Cisco Global Cloud Index: Forecast and Methodology, 2016–2021 White Paper, November 2018, https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html, last accessed 11/09/2019
- 3. Anders Andrae, Predictions on the way to 2030 of internet's electricity use, The Lost Decade? Planning the Future, University Copenhagen, February 2019
- 4. Anders Andrae, Total Consumer Power Consumption Forecast, Nordic Digital Business Summit, October 2017
- 5. Ralph Hintemann, Simon Hinterholzer, Energy Consumption of Data Centers Worldwide – How will the Internet become Green?, ICT4S, Lappeenranta, Finland, 2019
- 6. European Commission, Study on energy-efficient cloud computing technologies and policies for an eco-friendly cloud market, 2019, https://ec.europa.eu/digital-single-market/en/news/study-energy-efficient-cloud-computing-technologies-and-policies-eco-friendly-cloud-market, last accessed 11/09/2019
- 7. ISO/IEC CD 23544, Information Technology — Data Centres — Application Platform Energy Effectiveness (APEE), 2019, https://www.iso.org/standard/76000.html, last accessed 11/09/2019
- 8. ETSI - EN 303 470, Environmental Engineering (EE); Energy Efficiency measurement methodology and metrics for servers, https://standards.globalspec.com/std/13240725/EN%20303%20470, last accessed 11/09/2019
- 9. G. Smpokos et al, On the Energy Consumption Forecasting of Data Centers Based on Weather Conditions: Remote Sensing and Machine Learning Approach, April 2018, arXiv:1804.01754
- 10. Green Software Engineering, Set of criteria for sustainable software, https://green-software-engineering.de/en/kriterienkatalog-v01/introduction.html, last accessed 11/09/2019

Sandro Kreten

He is a researcher at Trier University of Applied Sciences, Environmental Campus Birkenfeld and a Ph.D. student at Trier University in the field of energy efficiency of cloud technologies. He works as lecturer at the environmental campus Birkenfeld and is head of technology at the venture capital donor capacura. He graduated in applied computer sciences and mathematics (M.Ed.) in 2015.

Contact: s.kreten@umwelt-campus.de

Separation pains in everyday business

Taking a look at corporate communication holistically

The continuous shift towards agile work environments confronts enterprises with new challenges. Classically formed sections lose touch with more agile units within the organisation.

This circumstance is due to the different nature in organisational structure. Within the departments, communication and processes are fundamentally clashing. To reunite the areas, there are varyingly radical solutions, which can be realised with diverse levels of effort. On the smaller end, the focus is on promoting communication between departments, while large-scale solutions establish integrated and complete exchanges across the enterprise.

Different Structures

With which structures and concepts do the two separate business branches work?

Management, HR, controlling, finance, marketing, purchasing, and sales are usually organised in a classical way. Depending on scale, there is a numerous amount of roles across various levels of hierarchy.

If the field is large enough to warrant a department, it is led by a department head. This is followed by team leads and the single members of each team. The chain of command runs top-down.

Often, each role is supported by individuals, who act as stand-ins when necessary. The supervisor handles the distribution of tasks outside of daily operational duties, as well as their review after completion. These tasks occur ad-hoc and are processed accordingly. Clear rules on communication are usually set aside and the information flows primarily within the levels of hierarchy. Reports from teams are forwarded bottom-up.

Departments organised using agile principles, such as Scrum, work in an iterative system. Regular meetings support this process. The Scrum Master role moderates these meetings and supports and safeguards the team. The foundation for the planning of these iterative periods, called Sprints, are To-Dos, which are collected in the Product Backlog, sorted by priority. The prioritisation is handled by a further role – the Product Owner – working closely with the relevant stakeholders.

During the Sprint Planning, teams plan their capacities for the upcoming two to four weeks, depending on the chosen Sprint length, into a separate Sprint Backlog, and formulate the Sprint Goal. This goal states what the minimum required outcome to be achieved during this period is, and assists the team to guarantee, that a new deliverable result can be handed to the stakeholders.

To keep the team internally aligned, the Scrum Master introduces a brief meeting at the start of the day, the Daily Stand-up. Once the Sprint is completed, there is another meeting – the Sprint Review – which, next to the team, also includes the Scrum Master, Product Owner, and all relevant stakeholders. The attendees determine, whether or not the Sprint Goal has been achieved, and the Product Backlog is adjusted, if necessary. Additionally, any non-completed tasks are discussed and how to proceed further is defined. The Review, which considers technical factors, is followed by the Sprint Retrospective, during which there is room for feedback on the social aspects of cooperation. Topics include the communication within the team or how to avoid impediments.

Different Rhythm

The very different approaches and workflows present companies, in which both organisational structures are used, with challenges.

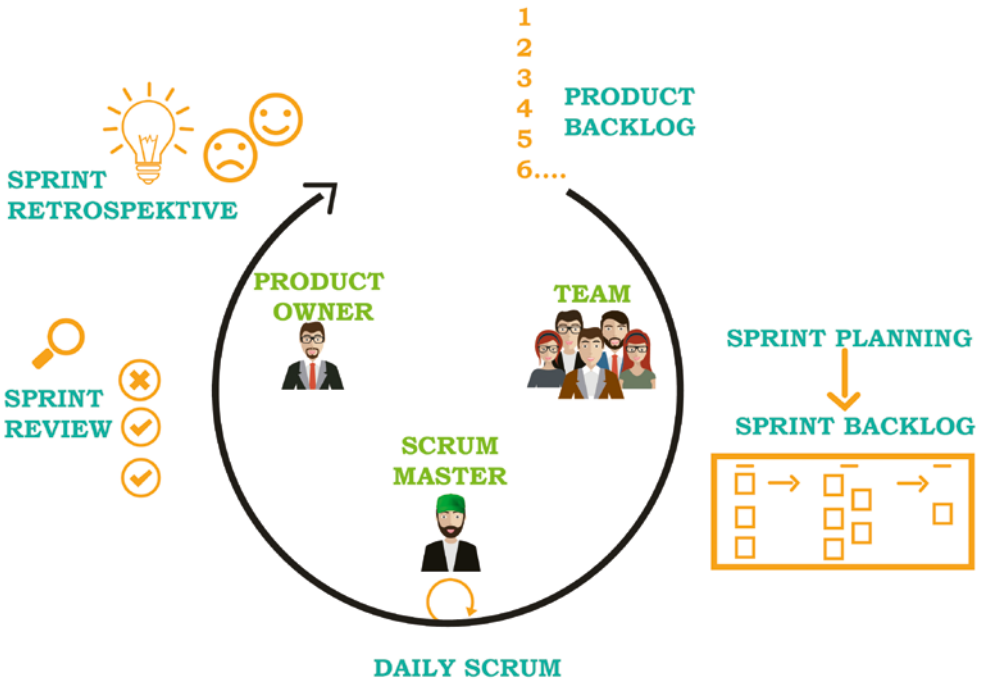


Figure 1: Scrum Flow

Take the following setting: An IT department, working according to Scrum, and a classically organised HR department. The HR department wants to publish its newest job posting on the website and requires assistance from IT as soon as possible. It is an urgent vacancy and time is a factor.

The requirement is however not communicated during the transition between Sprints, but instead while a regular work cycle has already started. The Sprint Backlog is already filled with selected tasks, which the team is currently working on. There is no extra time, or resources, planned. This causes both departments to be dissatisfied; IT is overloaded, and the other department becomes impatient. There is a different communication and work rhythm.

Time for Feedback?

In classic departments, feedback often happens incidentally and takes place, when a problem occurs, or can sometimes also be given in regular departmental meetings. This differs from the Scrum methodology, in which the Review and Retrospective offer fixed slots for feedback, and every event is treated as a potential opportunity to inspect and adapt.

Back to the example. Both departments are displeased and HR reports its dissatisfaction directly to IT. There, neither the feedback nor the actual requirement can be processed and integrated without causing unplanned overhead and disruptions in the standard Sprint procedure.

We understand **agile methods** – such as Scrum – as a kit, wich is **tailored to the needs** of the company or the team and **not as a dogmatic** set of rules.

Creating Time

It is optimal for HR if IT addresses the request promptly. This is possible if a time slot for such short notice tasks is integrated into every Sprint Backlog. This way, resources are set aside to handle various foreseeable urgent requests from other departments without impacting overall planning stability.

If all tasks are completed before the end of a Sprint and no or not many planned tasks from other departments remain, new backlog items can be taken into the current Sprint. The order of these potential tasks should already be discussed during Sprint Planning to avoid unnecessary idle periods. Feedback and discussions with other departments can also deplete this time slot.

Part of the challenge is thereby resolved. The case, that the department organised in agile methods is approached by multiple other departments, however remains open. Then, further differentiation by urgency and priority is important. In this case it is

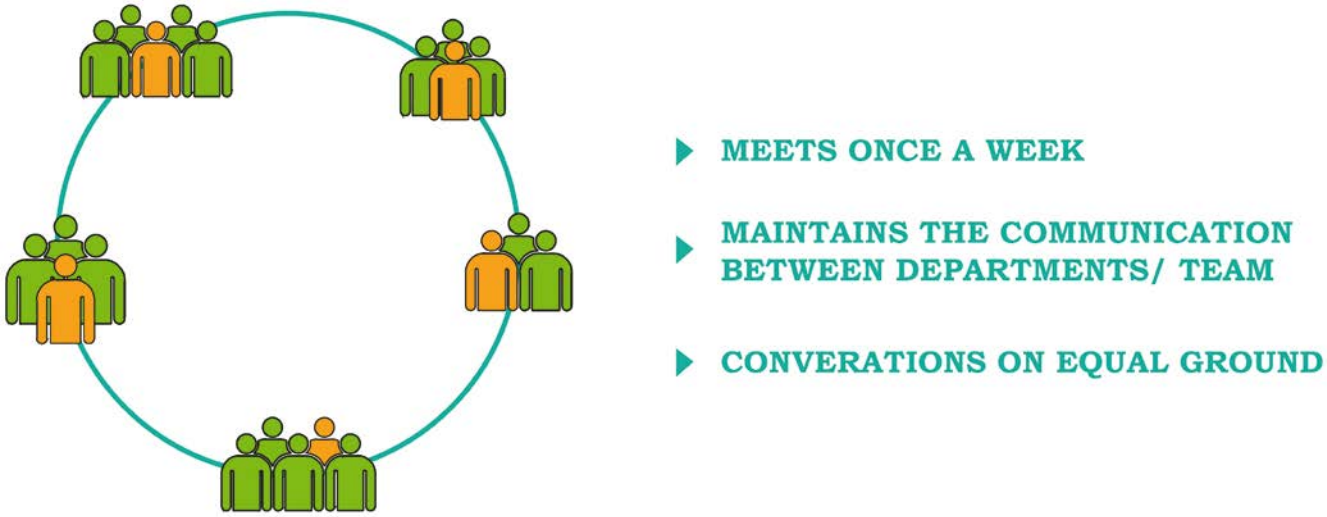


Figure 2: The Para-Team

critical that not only bilateral communication but also an exchange between multiple parties is guaranteed.

Holistic communication

To restore communication between all parts of a company, there is no need for large discussion groups with all employees and grass-root democratic ballots. It is entirely sufficient for each part to offer someone to take charge of inter-departmental communication. Within the agile context, there are dedicated Communities of Practice, where there is a place for such exchanges. This method of communication can be expanded onto the entire enterprise. In the face of this challenge the term Para-Team arose. "Para", stemming from Greek, can be translated to "over". It is a team, which communicates across departments, with the goal of accommodating other parts of the company. Para-Teams thus function as a link and a continuous dialogue is re-established.

This approach only works, if each member of the Para-Team is allocated a time budget for these activities and doesn't incur overload as a result of their participation. In our example, this Para-Team would include a dedicated member of HR and IT, along with every other relevant department. Clear communication via this channel could alleviate potential planning problems and inter-departmental dissatisfaction.

In our example, clear communication through a Para-Team, including a member of HR and IT, as well as every other department, could alleviate potential planning problems and inter-departmental dissatisfaction.

The next step would be to partially model other classically organised areas along agile principles. For daily tasks, this only conditionally makes sense. Nevertheless, there are smaller projects in each department, to which agile methodologies can be applied. This can promote mutual understanding and open new points of view.

The enterprise appears rounder on the out- and inside.

Sources:

- Scrum Guide
- https://www.duden.de/rechtschreibung/para_

Jana Weiner
Manager HR & Finance, Co-Founder
jana@yuna-visions.de

Anne- Sophie Schwindenhammer
Consultant
anne-sophie@yuna-visions.de

Working at Yuna Visions GmbH,
an agile IT & HR Consultancy in Darmstadt.

The "international city of peace and justice" hosting the Cloud Foundry Summit Europe 2019

The Hague, a city that is said to be the European Capital of peace and justice, human rights and equality. Could there have been a better place for an IT conference, where developers from all over Europe came to meet and learn from their peers, work jointly together and conduct a constructive dialogue with each other than The Hague? Although with 700 attending visitors, fewer participants than in the previous year, it was a lively conference.

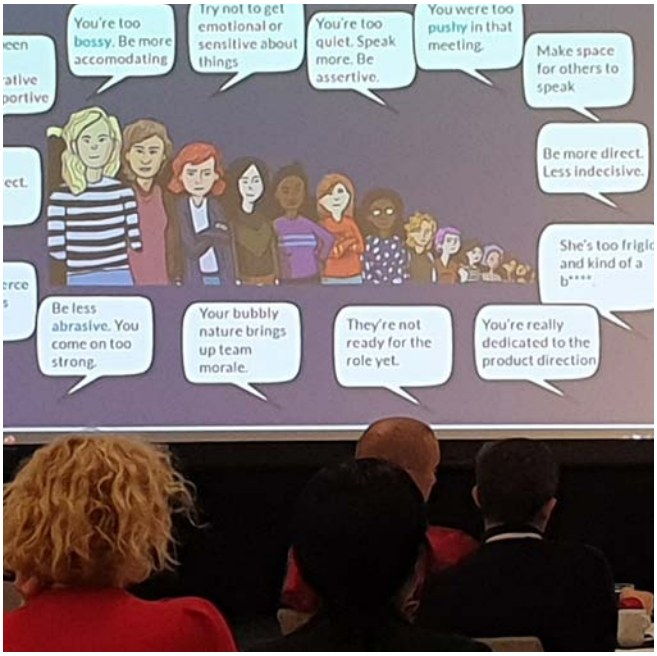
It seemed, that the European Cloud Foundry Event in The Hague was more of an exchange platform for the community this time:

"Cloud Foundry is committed to ensuring developers can focus on what matters to them: building applications and writing code. Summit puts developers front and center to share their stories with our European community. Nearly 40 percent of our users report that application development takes less than one day with Cloud Foundry. Developers from across Europe are here to meet with and learn from their peers to maximize their productivity." Abby Kearns, executive director of Cloud Foundry Foundation, told the audience.

Many member companies presented their latest features during the Summit, for example SUSE, who presented Project Stratos. Stratos is an open source user interface for the Cloud Foundry community, and it has, as of now, officially graduated from an incubating project to a core Cloud Foundry project. Or IBM: IBM gave a real surprise release at the summit, when they announced that they are working on bringing Cloud Foundry and Red Hat OpenShift together.

Moreover, *The Cloud Foundry Foundation* released the results from their most recent user survey. While the platform was only used by 24 percent of the companies surveyed in 2017, it is currently used by 45 percent. Especially enterprise companies seem to appreciate the benefits of the platform. "Over half the Fortune 500 are shaping the future of their companies on Cloud Foundry today." Said Abby Kearns.

Under the roof of "justice and peace" the topic of diversity, equali-



ty and justice in gender also played a major role. 25% of the speakers and 17% of the attendees were women.

As in previous events, a Diversity Luncheon was held again, with several talks and discussions. Especially impressive was the presentation on ungendered feedback, held by Nikita Rathi and Katrina Bakas. Both women showed what an impact feedback on employees has, and what good feedback should be like: "Feedback is key – at best it helps us grow, lets us know where we thrive, and makes us better teammates. At worst, it can stop careers in their tracks," Rathi and Bakas told the audience. They also pointed out that women were 1.4 times more likely to receive critical subjective feedback as "opposed to either positive feedback or critical objective feedback that are tied to outcomes".

To sum up: The Cloud Foundry Summit managed to show the audience the relevance of Open Source technologies such as Cloud Foundry for the digital transformation of companies worldwide. It also showed the relevance of working together in the community, exchanging experience. Moreover, it focused on including female developers and tried to reduce gender bias in the ecosystem.

Cloudical is definitely looking forward to the next CF Summit in 2020.

Julia Hahn.

CLOUDICAL

KubeCon + CloudNativeCon North America 2019

The de facto official Kubernetes developer and user conference or recruitment platform?

KubeCon – the Mecca for all cloudnative obsessive IT professionals opened its doors in San Diego in November. The KubeCon + CloudNativeCon North America 2019 was one of the biggest events of The Linux Foundation ever. Around 14,000 visitors from all over the world travelled to the golden state to meet, network, and everyone was full of expectations for the event.

One possible explanation for the astronomical growth in visitor numbers at KubeCon + CloudNativeCon (500 visitors in 2015 compared to 14,000 in 2019) is that the mergers and acquisitions in which Kubernetes played a major role in 2018 continued. According to the Linux Foundation, 2018 marked a turning point for Open Source, which invested more than $65 billion in mergers, acquisitions and IPOs (again, in which Kubernetes played a major role).

Another explanation for the success of KubeCon + CloudNativeCon is that the excitement for K8s has continued unabated, as developers love them, and companies try to integrate and tame them.

But what is the focus of this event? Technology? Well, from my point of view, no, it has shifted over time.

KubeCon + CloudNativeCon seems like the biggest recruitment platform ever. The sponsors seem to be joining forces under the guise of technology and products, but when you talk to the guys at the booth, you always hear the same story: the companies are present to be visible to potential employees. The same trend was observed at the last two CloudFoundry Summits. Most companies have recruitment stands that focus only on recruitment. This trend is understandable, but it kills the spirit and idea of a technology conference.

There smaller conferences feel more enjoyable but limited. Taking FOSDEM as an example, the developers, engineers, administrators, but not marketing people are giving talks. If you want to know about a project, there is a high chance developer of the project are attending and available to answer questions. It is good to

see that the CNCF seems to try to improve this by having the CNCF project pavilion. The CNCF project pavilion allowes maintainers to represent their projects and answer questions of the community in person.

The technological view of the presentations, sponsors and products is that we are all surfing the same wave. The wave to "Keep Cloud Native Classy". New technology is shown from many different projects and companies. Projects and products ranging from easily running Kubernetes to securing your containerized applications.

The project and product categories presented are container and/or Kubernetes security, surveillance, deployment, and some products for multi/hybrid orchestration. It is good to know that there is a diverse number of projects and products trying to solve today's and tomorrow's "problems" already.

The CNCF projects are presented by highly motivated upstream developers. During the discussions with these people we finally feel the fire for these technologies and projects. This is the spirit of Open Source.

Kim-Norman Sahm and Alexander Trost

CLOUDICAL

We are testing clouds

Cloud computing offerings are changing rapidly. Even the offerings of the individual providers are regularly being further developed. This makes it almost impossible to keep track of things. We, Cloudical, would like to remedy this situation and gradually examine the offers and evaluate them from an objective point of view. Our technicians have developed tests for this purpose. We test general information on onboarding, availability, SLAs, data centers, compute, storage, network, limitations, scaling, technologies, but also more internal information such as backup, security, image service, patch management, monitoring, CI/CD, as a Service offerings and, of course, the cost factor.

This results in rankings and tables help customers to inform themselves independently and to find the right provider for themselves. But not only readers of the cloud report receive comprehensive, independent datas, providers can also find out about their market, see where they stand and what their strengths are in comparison. They can also identify their possible weaknesses and potentials, see possible pent-up demand or discover approaches for further specialization and improvement. And of course, they present themselves to interested readers and potential customers.

Currently we have tested the providers AWS, Azure, Google Cloud, IBM Cloud, the Open Telekom Cloud and the OVH Cloud. On the following pages you will find the evaluations sorted by individual topics. In the topic "Databases as a Service" we wer not able to test the OVH cloud. You will find the complete evaluations here. We will gradually add more clouds, so in the next issues only exemplary test evaluations will be shown, you will find the detailed tables online.

- **the-report.cloud/testresults**

If you have any suggestions for supplementing the questions, please write to us at: press@cloudical.io.

Note: Three virtual machines of different sizes are used in the evaluations:

Small means:
- OS Ubuntu 16.04
- 2vCPUs
- 8GB RAM
- min. 50GB HDD
- Location: Germany, if not Western Europe, if not Europe

Medium means:
- OS Ubuntu 16.04
- 4vCPUs
- 16GB RAM
- min. 50GB HDD
- Location: Germany, if not Western Europe, if not Europe

Large means:
- OS Ubuntu 16.04
- 8vCPUs
- 32GB RAM
- min. 50GB HDD
- Location: Germany, if not Western Europe, if not Europe

And the winners are ...

As with every edition, we are testing several cloud vendors and look into many different aspects of their offerings. We analyzed their pros and cons, discussed our experiences, and decided for the winner in all categories.

The pattern we've already seen several times, continues this time as well: You should choose your own favourite based on your preferences, since all the tested cloud providers have their specific strengths. There is of course an overall winner, but this might look completely different to you considering your very own priorities, since each one has specific strengths and room for improvements.

We are especially impressed by the performance of smaller and / or not so-well-known cloud vendors, such as Open Telekom Cloud or OVH, which was tested for the first time in this edition: Often they offer comparable performance and sometimes even more options than their bigger competitors, combined with more personal support and very reasonable pricing. That being said, let's look into the winners. And don't forget to check out our detailed comparison tables on the next pages for more details!

The tests were ranked by Karsten Samaschke.

CLOUDICAL

Category	Winner Performance	Winner Price	Reason
Compute	AWS	OVH	AWS offers the best performance this time around and a complete feature-set. Although OVH does not offer the best performance everywhere and might be lacking some cutting edge features, it nowhere really falls off – and is considerably cheaper than the competition.
Storage	AWS for Object IBM for Block and File	Google	IBM offers the fastest Block-Storage with up to 240 MB/s. Price wise, Google wins that fight, but keep in mind that they have the slowest storage for that price. For Object Storage, there is no way around AWS as inventor of S3-Storage. Google might be one of the slower options, but it is considerably cheaper then the competition.
Backup, Recovery and Availability	Azure	IBM	Azure offers the most options within this category. IBM is not far off, but is considerably cheaper.
Databases (DBaaS)	AWS	Google	If you're looking for the full offering, then AWS is your undisputed champion here. They have most of the relevant databases and offer a complete package. For pure performance, IBM and OTC are very compelling with MySQL and PostgreSQL. When looking at a cheap offering with good performance, Google will be your platform of choice.
Network	Google	OVH	This time, Google has shown a considerable performance improvement, making it the fastest provider overall. Inter-Region connection speeds are still faster with AWS, though. OVH does not charge for traffic, so it is the natural winner here.
Security	Azure	n/a	Azure can still defend the first place in that category, as they are still the only one which do pen-tests against their platform.
Container-as-a-Service	Google	OVH	Google is the leading force behind Kubernetes – and this shows off here: Although other might offer more options, the Google Container Engine is so well integrated and easy to set up, while delivering good performance, it simply shines. OVH offers a compelling alternative, especially taking the pricing aspect into account.
IaaS / PaaS / SaaS-Patch Management	Google	n/a	GCP is the Winner, as they are offering a wide catalogue of common and container optimized OSes. As they are offering Patch-Management, too, Azure and IBM can be a good choice for you.
Software-as-a-Service	OVH	n/a	OVH offers an interesting Software-as-a-Service offering here with most features.
Logging-as-a-Service	Azure	n/a	Azure offers the most options.
Image Service	OTC	n/a	OTC offers all the major (and some minor) operating systems and distributions, plus they support the most imaging formats.

Compute

Questions	AWS	Azure
Small VM: OS Ubuntu 16.04; 2vCPUs; 8GB RAM; min. 50GB HDD; Location: Germany, if unavailable: Western Europe, if unavailable: Europe	yes	yes
Medium VM: OS Ubuntu 16.04; 4vCPUs; 16GB RAM; min. 50GB HDD; Location: Germany, if unavailable: Western Europe, if unavailable: Europe	yes	yes
Large VM: OS Ubuntu 16.04; 8vCPUs; 32GB RAM; min. 50GB HDD; Location: Germany, if unavailable: Western Europe, if unavailable: Europe	yes	yes
GPU support for the VM?	yes	yes
AutoScaling for VM?	yes	yes
Availability Zones (i.e Availability set) possible	yes	yes
Startup-time (till time of availability) – Small – Medium – Large	 35 sec 38 sec 37 sec	 110 sec 96 sec 98 sec
Count of steps until VM is created	7 steps	7 Steps
RAM throughput (sysbench, Block size 1k) – Read – Write	 858.76 MB/sec 824.77 MB/sec	 4429.93 MB/sec 3788.34 MB/sec
CPU speed (geekbench) – Small Single Core – Small Multi Core – Medium Single Core – Medium Multi Core – Large Single Core – Large Multi Core	 3292 6277 3251 11642 3335 21440	 3157 3583 3508 7504 3169 12244
– VM accessible via Console	no	yes
Total cost of VM per month (732hrs) – Small – Medium – Large	 € 71.23 / $ 78.48 € 142.46 / $ 156.95 € 284.91 / $ 313.89	 € 73.92 / $ 87.65 € 147.75 / $ 175.20 € 295.49 / $ 350.40
Supported disk formats / images	– OVA – VMDK – RAW – VHD/VHDX	– VHD – VMDK – VHDX – QCOW2 – RAW
Are there any limitations per VM?	Amount CPUs: 128 RAM size: 1952 GB Disk size: 2048 GB	Amount CPUs: 128 RAM size: 3892 GB Disk size: 4096 GB Amount Disk: 64
Can bare-metal servers be deployed via the cloud?	yes	no
Which hypervisor is used?	– KVM – Xen	– Hyper-V
Is autorecovery available	yes	yes

	Google Cloud Platform	IBM Cloud	OTC	OVH
	yes	yes	yes	yes
	yes	yes	yes	yes
	yes	yes	yes	yes
	yes	yes	no	yes
	yes	yes	yes	no
	yes	yes	yes	no
	38 sec 40 sec 42 sec	365 sec 295 sec 265 sec	35 sec 38 sec 43 sec	36 sec 39 sec 40 sec
	3 Steps	4 Steps	4 Steps	5 Steps
	4526.86 MB/sec 3593.19 MB/sec	640.69 MB/sec 619.24 MB/sec	4260.23 MB/sec 3481.96 MB/sec	4213.27 MB/sec 3439.68 MB/sec
	3223 3878 3170 7158 3265 13712	2628 5006 2654 9173 3593 22974	2936 5444 2914 9925 2907 18081	3087 5589 3171 10336 3475 20095
	yes	yes	yes	yes
	€ 57.09 / $ 62.54 € 114.12 / $ 125.09 € 228.25 / $ 250.18	€ 74.45 / $ 82.28 € 142.00 / $ 156.93 € 278.06 / $ 307.30	€ 74.57 / $ 82.41 € 145.36 / $ 160.64 € 290.42 / $ 320.96	€ 23.92 / $ 26.40 € 45.66 / $ 50.40 € 92.41 / $ 102.00
	– VMDK – VDH – RAW	– VMDK – AKI – ARI – AMI – QCOW2 – RAW	– VMDK – VHD – VHDX – QCOW2 – RAW	– VMDK – ISO – ARI – AKI – AMI – VDI – VHD
	Amount CPUs: 160 RAM size: 3844 GB Disk size: 64 TB Amount Disk: 128	Amount CPUs: 64 RAM size: 512 GB Disk size: 12 TB	Amount CPUs: 60 RAM size: 940 GB	Amount CPUs: 128 RAM size: 1024 GB
	yes	yes	yes	yes
	– KVM	– PowerVM – VMware ESX Server – Xen – KVM – z/VM	– Xen – KVM	– KVM
	no	yes	yes	yes

Storage

Questions	AWS	Azure	
Which kinds of storage are available? – Object / Blob Storage – File Storage – Block Storage	 yes (S3 / Glacier) yes (EFS) yes (EBS)	 yes (Azure Blob Storage) yes (Azure Disk Storage) yes (Azure Files)	
Block - Different tier-classes? SATA, SSD, SAS	yes	yes	
Which objects storage-engines are offered?	– Amazon S3	– Azure Blob Storage	
File - Accessing file storage via (cluster) file system.	– EFS	– GlusterFS – BeeGFS – Luster	
Storage capacity limits	Overall size: Unlimited 5 TB per S3 object	2 PiB for US and Europe, and 500 TiB for all other regions (including the UK) ,250 Storage Accounts per Subscriptions per region	
Duration of provisioning?	8 sec	45 sec	
Throughput IOPS (only Block- and File-Storage)	– Random read: bw = 24.52 MB/s, iops = 3065 – Random write: bw = 128.92 MB/s, iops = 2014 – Random Read and write: – read : bw = 44.20 MB/s, iops = 2762 – write: bw = 5.04 MB/s, iops = 314 – Sequential read: bw = 24.55 MB/s, iops = 3068 – Sequential write: bw = 99.04 MB/s, iops = 3095	– Random read: bw = 1.95 MB/s, iops = 243 – Random write: bw = 15.59 MB/s, iops = 243 – Random read and write: – read: bw = 3.50 MB/s, iops = 218 – write: bw = 0.41 MB/s, iops = 25 – Sequential read: bw = 1.95 MB/s, iops = 243 – Sequential write: bw = 7.80 MB/s, iops = 243	
Costs per month – total price for 50 GB Disk which is mounted to the VM	€ 5.29 / $ 5.95	€ 7.65 / $ 8.58	

Backup, Recovery and Availability

Questions	AWS	Azure	
Are managed backups offered (Provider is responsible to take backups)	yes	yes	
Which types of backups are supported for VMs?	– Snapshots – Full backups – Incremental Backups	– Full Backups – Differential Backups – Incremental Backups – Snapshots	
Where will the backup be stored?	– Amazon S3 – Amazon Glacier – Different datacenter – Storage-Cluster	– Recovery Services Vault – Different Datacenter	
Can backups be scheduled?	yes	yes	
Usage costs per month–500 GB Backup Storage–Western Europe	€ 5.45 / $ 6.00	€ 17.88 / $ 21.20	
Is a managed Backup-Service for a VM provided?	no	yes	

	Google Cloud Platform	IBM Cloud	OTC	OVH
	yes (Google Cloud Storage) yes (Google Cloud Filestore / Google Drive) yes (Google Persistent Disk)	yes (IBM Cloud Object Storage) yes (IBM Cloud file storage) yes (IBM Cloud block storage)	yes (Object Storage Service) yes (Scalable File Service) yes (Elastic Volume Service)	yes (Swift) Yes (CephFS) yes (Based on Ceph)
	yes	yes	yes	yes
	– Buckets (like S3)	– S3 – Swift	– S3 – OpenStack Swift	-Swift
	– Google Cloud Storage FUSE – Beta: Google Cloud Filestore	– NFS	– NFS	-CephFS
	Overall size: Unlimited 5 TB per individual object	Object storage - Unlimited File / Block storage - 12 TB	50 TB of Object storage 32 TB of Block Storage 10 PB of File Storage	Overall size: Unlimited
	18 sec	60 sec	7 sec	7 sec
	– Random read: bw = 1.20 MB/s, iops = 149 – Random write: bw = 17.17 MB/s, iops = 268 – Random read and write: – read: bw = 2.24 MB/s, iops = 140 – write: bw = 0.26 MB/s, iops = 15 – Sequential read: bw = 33.40 MB/s, iops = 4175 – Sequential write: bw = 122.98 MB/s, iops = 3842	– Random Read: bw = 145.93 MB/s, iops = 18241 – Random Write: bw = 221.51 MB/s, iops = 3461 – Random Read and write: – read: bw = 171.15 MB/s, iops = 10696 – write: bw = 19.50 MB/s, iops = 1218 – Sequential Read: bw = 106.58 MB/s, iops = 13322 – Sequential Write: bw = 120.65 MB/s, iops = 3770	– Random Read: bw = 8.04 MB/s, iops = 1005 – Random Write: bw = 65.14 MB/s, iops = 1017 – Random Read and write: – read: bw = 14.45 MB/s, iops = 903 – write: bw = 1.65 MB/s, iops = 102 – Sequential Read: bw = 11.34 MB/s, iops = 1417 – Sequential Write: bw = 55.55 MB/s, iops = 1735	– Random Read: bw = 2.00 MB/s, iops = 250 – Random Write: bw = 15.91 MB/s, iops = 248 – Random Read and write: – read: bw = 4.00 MB/s, iops = 250 – write: bw = 0.47 MB/s, iops = 28 – Sequential Read: bw = 2.00 MB/s, iops = 249 – Sequential Write: bw = 7.78 MB/s, iops = 243
	€ 1.14 / $ 1.28	€ 10.08 / $ 10.8	€ 2.30 / $ 2.58	€ 2.04 / $ 2.25

	Google Cloud Platform	IBM Cloud	OTC	OVH
	no	yes	yes	yes
	– Snapshots – Incremental Backups	– Snapshot – Full Backups – Incremental Backups	– Snapshot – Full Backups – Incremental Backups	– Snapshot – Full Backups – Incremental Backups
	– Google Cloud Storage – Storage Cluster	– Evault – dedicated backup space – IBM Cloud Backup, – IBM Object Storage archive	– different Data-Centers	– different Data-Centers – Cloud Repository
	yes	yes	yes	yes
	€ 4.48 / $ 5.00	€ 0.88 / $ 1.00	€ 5.00 / $ 5.63	€ 5.03 / $ 5.60
	no	yes	yes	yes

Databases (DBaaS)

Questions	AWS	Azure
Which DB engines are offered?	Relational DB – MySQL – PostgreSQL – MariaDB – Oracle – Microsoft SQL Server – Amazon Aurora Non-Relational DB – Amazon DynamoDB – Amazon ElastiCache – Amazon Neptune – Redis – MemCached Data Warehouse / Big Data – Amazon Redshift – Amazon Athena – Amazon EMR (Hadoop, Spark, HBase, Presto, etc.) – Amazon Kinesis – Amazon Elasticsearch Service – Amazon Quicksight	Relational DB – Azure SQL Database – Azure Database for MySQL – Azure Database for PostgreSQL – Azure Database for Maria DB – Microsoft SQL Server Non-Relational DB – Azure Cosmos DB – Azure Table Storage – Redis Data Warehouse / Big Data – SQL Data Warehouse – HDInsight (Hadoop, Spark, Hive, LLAP, Kafka, Storm, R.) – Azure Databricks (Spark) – Azure Data Factory – Azure Stream Analytics
– Performance of MySQL (MySQL Sysbench, table-size (row data): 1000000, Threads: 16) – Read – Write – Read / Write	Transactions: 39382 (656.14 / sec) Transactions: 75391 (1256.30 / sec) Transactions: 26755 (445.61 / sec)	Transactions: 15304 (254.77 / sec) Transactions: 12362 (205.98 / sec) Transactions: 7399 (123.03 / sec)
Provisioning time for a MySQL instance	561 sec	156 sec
Performance of PostgreSQL	Transactions: 756857 (12612.60 / sec) Transactions: 486204 (8101.82 / sec) Transactions: 26135 (435.37 / sec)	Transactions: 133651 (2219.42 / sec) Transactions: 34801 (579.61 / sec) Transactions: 6302 (104.62 / sec)
Provisioning time for a PostgreSQL instance	530 sec	204 sec
– Supported DB Versions	– MySQL 8.0,5.7, 5.6, 5.5 – MariaDB 10.3,10.2,10.1,10.0 – Microsoft SQL Server 2017 RTM, 2016 SP1, 2014 SP2, 2012 SP4, 2008 R2 SP3 – Oracle 18.0.0.ru, 12c (12.1.0.2, 12.1.0.1), Oracle 11g (11.2.0.4, 11.2.0.3, 11.2.0.2) – PostgreSQL 11.5, 11.4, 11.2, 11.1,10.6,10.5, 10.4, 10.3, 10.1, 9.6.x, 9.5.x, 9.4.x, 9.3.x,9.2.x – Amazon Aurora - compatible with MySQL 5.6.10a	– MySQL 8.0, 5.7, 5.6 – MariaDB 10.2 – Azure SQL Database: Microsoft SQL Server 2017 – Microsoft SQL Server 2017, 2016 SP1, 2014 SP2, 2012 SP4, 2008 R2 SP3 – PostgreSQL 11, 10.3, 9.6.x, 9.5.x – Azure Cosmos DB
Troubleshooting as a Service – Rollback – Support	 yes yes	 yes yes
Total price for the database per month – MySQL – 2 vCores – 100 GB Storage – Frankfurt / Western Europe – 100% active per month – No dedicated backup	€ 114.13 / $ 128.13	€ 142.29 / $ 159.50

	Google Cloud Platform	IBM Cloud	OTC
	Relational DB - PostgreSQL - MySQL - SQL Server (Beta) - Google Cloud Spanner Non-Relational DB - Google Cloud Datastore - Google Cloud BigTable Data Warehouse / Big Data - Google Cloud BigQuery - Google Cloud Dataflow - Google Cloud Dataproc (Hadoop / Spark) - Google Cloud Datalab - Google Cloud Dataprep	Relational DB - Db2 on Cloud - PostgreSQL - MySQL Non-Relational DB - Cloudant - MongoDB - ScyllaDB - Redis - JanusGraph - etcd - Elasticsearch Data Warehouse / Big Data - Db2 Warehouse on Cloud	Relational DB - PostgreSQL - MySQL - Microsoft SQL Server Non-Relational DB - MongoDB - Redis
	Transactions: 7656 (127.35 / sec) Transactions: 18605 (309.83 / sec) Transactions: 5897 (98.09 / sec)	Transactions:42310 (704.90 / sec) Transactions:87176 (1452.64 / sec) Transactions:31427 (523.45 / sec)	Transactions: 27627 (460.27 / sec) Transactions: 90952 (1515.28 / sec) Transactions: 31583 (526.21 / sec)
	310 sec	184 sec	401 sec
	Transactions: 121018 (2016.60 / sec) Transactions: 89065 (1484.17 / sec) Transactions: 5844 (97.16 / sec)	Transactions: 788035 (13131.50 / sec) Transactions: 611756 (10194.23 / sec) Transactions: 33570 (559.19 / sec)	Transactions: 1260342 (20995.91 / sec) Transactions: 165319 (2753.82 / sec) Transactions: 22332 (367.25 / sec)
	280 sec	319 sec	363 sec
	- MySQL 5.7, 5.6 - PostgreSQL 9.11, 9.6.x - SQL Server (Beta) 2017	Db2-ge PostgreSQL 9.6.x, 9.5.x, 9.4.x MySQL 5.7.22, 5.7.20 Cloudant-h7 MongoDB 3.4.10, 3.2.18, 3.2.11, 3.2.10 ScyllaDB 2.0.3 Redis 4.0.10, 3.2.12 JanusGraph 0.1.1 beta etcd 3.3.3, 3.2.18 Elasticsearch 6.2.2 , 5.6.9 Db2 Warehouse-ef	PostgreSQL 10.0, 9.6.5, 9.6.3, 9.5.5 MySQL 5.7.20, 5.7.17, 5.6.35, 5.6.34, 5.6.33, 5.6.30 Microsoft SQL Server 2016 EE, 2016 SE2014 SE
	yes yes	yes yes	yes yes
	€ 121.43 / $ 138.75	N/A	€ 298.40 / $ 335.04

Total price for the database per month – PorstgreSQL – 2 vCores – 100 GB Storage – Frankfurt / Western Europe – 100% active per month – No dedicated backup	€ 121.64 / $ 136.34	€ 142.29 / $ 159.50	
Limitations: – How many simultaneous requests to the DB? – How much RAM? – How many users?	MySQL: – max Connections: 2540 PostgreSQL: – max Connections: 5696	MySQL: – max Connections: 10000 PostgreSQL: – max Connections: 1900	
How does backup/restore work?	Backups: – Automatic Backups. Restore: – Point-in-time restore	Backups: – Automatic Backups. Restore: – Point-in-time restore – Geo-restore	

Network

Questions	AWS	Azure	
Is network monitoring availble?	yes	yes	
Is a Content Delivery Network (CDN) available?	yes	yes	
Sample Measurements 1) Same AZ 2) Different AZ 3) Different Region	Iperf Result: 1) TCP: Bandwidth Sender: 4.70 Gbit/sec Receiver: 4.68 Gbit/sec UDP: Bandwidth: 7.21 Gbit/sec 2) TCP: Bandwidth Sender: 4.54 Gbit/sec Receiver: 4.52 Gbit/sec UDP: Bandwidth: 7.46 Gbit/sec 3) TCP: Bandwidth Sender: 454 Mbit/sec Receiver: 452 Mbit/sec UDP: Bandwidth: 7.69 Gbit/sec	Iperf Result: 1) TCP: Bandwidth Sender: 901 Mbit/sec Receiver: 899 Mbit/sec UDP: Bandwidth: 919 Mbit/sec 2) TCP: Bandwidth Sender: 899 Mbit/sec Receiver: 897 Mbit/sec UDP: Bandwidth: 920 Mbit/sec 3) TCP: Bandwidth Sender: 835 Mbit/sec Receiver: 834 Mbit/sec UDP: Bandwidth: 925 Mbit/sec	
Public IPs – Public IPs for VMs? – Available kinds of public IPs for VMs – Public IPs for Load Balancers? – Available kinds of public IPs for Load Balancers	 yes floating / static yes static	 yes floating / static yes static	
Is a dedicated network connection from datacenter to public cloud possible?	yes (AWS Direct Connect)	yes (Azure Express Route)	
Network Security features (Network Traffic analysis, Network Security Groups)	– AWS Web Application Firewall – Network security groups – Network Traffic analysis	– Azure Firewall – Azure Front Door – Azure Network Watcher – Azure Security Center – Azure DDoS protection – Network access control – Network layer control – Network security rules (NSGs)	
VPN as a Service	yes	yes	
Traffic costs per GB	€ 0.13 / $ 0.15	€ 0.009 / $ 0.01	

	€ 124.21 / $ 141.81	€ 103.04 / $ 136.00	€ 312.80 / $ 350.85
	MySQL: - max Connections: 4000 PostgreSQL: - max Connections: 1000	MySQL: - max Connections: 151 PostgreSQL: - max Connections: 1000	MySQL: - max Connections: 151 PostgreSQL: - max Connections: unlimited
	Backups: - Automatic Backups. Restore: - On-demand	Backups: - Automatic Backups. Restore: -On-demand	Backups: - Automatic Backups. Restore: - Point-in-time restore

	Google Cloud Platform	IBM Cloud	OTC	OVH
	yes	yes	yes	no
	yes	yes	yes	yes
	Iperf Result: 1) TCP: Bandwidth Sender: 5.99 Gbit/sec Receiver: 5.98 Gbit/sec UDP: Bandwidth: 639 Mbit/sec 2) TCP: Bandwidth Sender: 5.83 Gbit/sec Receiver: 5.82 Gbit/sec UDP: Bandwidth: 588 Mbit/sec 3) TCP: Bandwidth Sender: 224 Mbit/sec Receiver: 223 Mbit/sec UDP: Bandwidth: 410 Mbit/sec	Iperf Result: 1) TCP: Bandwidth Sender: 925 Mbit/sec Receiver: 919 Mbit/sec UDP: Bandwidth: 1.81 Gbit/sec 2) TCP: Bandwidth Sender: 917 Mbit/sec Receiver: 911 Mbit/sec UDP: Bandwidth: 2.06 Gbit/sec 3) TCP: Bandwidth Sender: 836 Mbit/sec Receiver: 831 Mbit/sec UDP: Bandwidth: 1.57 Gbit/sec	Iperf Result: 1) TCP: Bandwidth Sender: 3.09 Gbit/sec Receiver: 3.08 Gbit/sec UDP: Bandwidth: 1.48 Gbit/sec 2) TCP: Bandwidth Sender: 3.00 Gbit/sec Receiver: 2.99 Gbit/sec UDP: Bandwidth: 2.18 Gbits/sec 3) N/A	Iperf Result: 1) TCP: Bandwidth Sender: 245 Mbit/sec Receiver: 244 Mbit/sec UDP: Bandwidth: 5.10 Gbit/sec 2) N/A 3) TCP: Bandwidth Sender: 244 Mbit/sec Receiver: 243 Mbit/sec UDP: Bandwidth: 5.05 Gbit/sec
	yes floating / static yes static	yes floating/static yes static	yes static yes static	yes static yes static
	yes (Google Cloud Interconnect)	yes	yes (Direct Connect - MPLS)	yes (OVHcloud connect)
	- Firewall - Network security groups - Network Traffic analysis	- Network Security Groups - Firewalls (Multi VLAN, Single VLAN and Web App) - DDOS mitigation	- Network Security Groups - Firewalls (Multi VLAN, Single VLAN and Web App)	- Network Firewall - Failover IP - vRack (private network) - OVHCLoud Connect - Bandwidth - Load Balancers - Anti-DDoS protection
	yes	yes	yes	yes
	€ 0.073 / $ 0.082	€ 0.078 / $ 0.087	€ 0.06 / $ 0.067	included

Security

Questions	AWS	Azure
Integration to a SIEM possible? (Security Information and Event Management) Security Groups Disk Encryption Network Traffic Analyse	yes yes yes yes	yes yes yes yes
Protection against Denial of Service Attacks	yes	yes
Firewall - Does the cloud provider provide additional integrated security features i.e. a Next Generation Firewall?	yes	yes
Does the cloud provider keep an eye on current threats and take action?	yes	yes
Does the cloud provider support additional integrated security features for cloud resources using 3rd party tools: IDS (Intrusion Detection System) IPS (Intrusion Prevention System) ATP (Advanced Threat Protection)	 yes yes yes	 yes yes yes
– Does the provider carry out regular penetration tests against the platform?	No	yes

Container as a Service

Questions	AWS	Azure
Which technologies are being provided/supported?	– Kubernetes – Docker	– Kubernetes – Mesosphere – DV/OS – Docker
Is a managed container service available?	yes (EKS)	yes (AKS)
Can worker nodes be accessed directly by customers?	yes	yes
Can master nodes be accessed directly by customers?	no	yes
Which version of the technologies/Kubernetes is being offered?	1.13, 1.12, 1.11, 1.10	1.14.3, 1.14.1, 1.13.7, 1.13.5, 1.12.8, 1.12.7, 1.11.10, 1.11.9
How much time does it take to provide the container service for four nodes	9 min	< 2 min
Costs for a Kubernetes Cluster (4 Nodes) – Managed service – 732hrs per month – small flavor – hosted in Frankfurt or Western Europe – Storage / IPs not included * Prices in USD have been converted to EUR	€ 201.19 / $ 224.88*	€ 344.74 / $ 383.82
– Shared or dedicated Container Engine Cluster?	dedicated	shared
Do predefined StorageClasses exist in Kubernetes? Name - Provisioner	gp2 - kubernetes.io/aws-ebs	default (default) - kubernetes.io/azure-disk managed-premium - kubernetes.io/azure-disk
Limitations - What is the maximum cluster size?	max. 50	max. 100
Do you have full access to all K8s ressources (no RBAC restriction)?	yes	no
Does the Container Service provide a Load-Balancer Service?	yes	yes
Is a Storage Class useable?	yes	yes

	Google Cloud Platform	IBM Cloud	OTC	OVH
	yes yes yes yes	yes yes yes yes	yes yes yes yes	no yes yes yes
	yes	yes	yes	yes
	yes	yes	yes	yes
	yes	yes	yes	yes
	yes yes yes	yes yes yes	no no no	yes yes yes
	no	no	no	no

	Google Cloud Platform	IBM Cloud	OTC	OVH
	– Kubernetes – Mesosphere	– Kubernetes – OpenShift	– Kubernetes – Docker – Cloud Container Engine	– Kubernetes
	yes (GKE)	yes	yes	yes
	yes	yes	yes	yes
	yes	yes	yes	yes
	1.13.7-gke.8, 1.13.6-gke.13, 1.12.9-gke.7, 1.12.8-gke.8, 1.12.7-gke.25, 1.11.10-gke.5	1.15.5, 1.14.8, 1.13.12	1.11.3, 1.9.10	1.15, 1.14, 1.13, 1.12, 1.11
	< 3 min	< 2 min	14 min	< 3 min
	€ 222.53 / $ 247.78	€ 247.68 / $ 275.78	€ 285.69 / $ 318.07	€ 105.60 / $ 117,53
	shared	dedicated	shared	dedicated
	standard (default) - kubernetes.io/gce-pd	ibmc-file-bronze (default) - ibm.io/ibmc-file ibmc-file-custom - ibm.io/ibmc-file ibmc-file-gold - ibm.io/ibmc-file ibmc-file-retain-bronze - ibm.io/ibmc-file ibmc-file-retain-custom - ibm.io/ibmc-file ibmc-file-retain-gold - ibm.io/ibmc-file ibmc-file-retain-silver - ibm.io/ibmc-file ibmc-file-silver - ibm.io/ibmc-file	cce-evs - cce-evs cce-sfs - cce-sfs	cinder-high-speed cinder-classic
	max. 5000	n/a	Max nodes/cluster: 1000	100
	yes	no	yes	no
	yes	yes	yes	yes
	yes	yes	yes	yes

IaaS / PaaS / SaaS Patch Management

Questions	AWS	Azure	
Does the cloud provide a managed patch service?	no	yes (Azure Automation)	
Which operating systems are available?	Linux: – Red Hat Enterprise Linux (RHEL) 7.3, 6.8 – SUSE Linux Enterprise Server (SLES) 15, 12 – Amazon Linux 2, 2018.03 – CentOS 7.x, 6.x – Debian 9.x, 8.x – Ubuntu Server 18.04, 16.04, 14.04 – Oracle Linux 7.x, 6.8 Windows: – Windows Server 2008 – Windows Server 2012 – Windows Server 2016 including R2 Versions – Windows Server 2019	Linux: – CentOS 7.6, 7.5, 7.4, 6.9 – Clear Linux – Container Linux – Debian 10, 9, 8 – Red Hat Enterprise 7.x – SUSE Linux Enterprise Server 11 SP4, 12SP3 – Ubuntu 14.04, 16.04, 18.04, 19.04, 19.10 Windows: – Windows Server 2008 R2 SP1, SP2 – Windows Server 2012, 2012 R2 – Windows Server 2016 – Windows Server 2019 – Windows 10	
Is the operating system from the deployed VM at a current patch level?	yes	yes	

Software as a Service

Questions	AWS	Azure	
Is a mobile office suite offered? Is it deeply integrated with other services?	no n/a	yes no	
Managed App Services	– Amazon Outpost – AWS Step Functions – Amazon API Gateway – Amazon Elastic Transcoder – Amazon SWF	– Azure Stack – Security and Compliance – Backups and Archives – Disaster Recovery – Cosmos DB – Networks – Active Directory Services – Development and Testing Services – Mobile Services	
Mobile App Services – Push Notifications – User Management – NoSQL-Datenbase – File Storage – Messaging – Social Networks	AWS Mobile yes yes yes yes yes no	Azure Mobile App Service yes yes yes yes yes yes	
Application Environments – Websites – Microservices – Messaging – Serverless	 yes (AWS Lightsail) yes (AWS Elastic Beanstalk) yes (AWS SQS) yes (AWS Lambda)	 yes (Azure Web Sites) yes (Azure Service Fabric) yes (Azure Service Bus) yes (Azure Functions)	
Rollback to a previous application version?	yes	yes	

	Google Cloud Platform	IBM Cloud	OTC	OVH
	yes (Google App Engine)	yes (IBM BigFix Patch Management)	no	no
	Linux: – Centos 7, 6 – Container-Optimized OS from Google cos-69-lts, cos-stable, cos-beta, cos-dev – CoreOS coreos-stable, coreos-beta, coreos-alpha – Debian 9 – Red Hat Enterprise Linux (RHEL) 8, 7, 6 – RHEL for SAP, rhel-7-6-sap-ha, rhel-7-4-sap – SUSE Enterprise Linux Server (SLES) 15, 12 – SLES for SAP sles-15-sap, sles-12-sp4-sap, sles-12-sp3-sap, sles-12-sp2-sap, sles-12-sp1-sap – Ubuntu 19.04, 18.10, 16.04, 14.04 Windows: – Windows Server 2019, 2016, 2012 R2, 2008 R2 – Windows Server Core 2019 – Windows Server Core 2019 for containers	Linux: – CentOS 7, 6 – RedHat Enterprise 7, 6 – SUSE Linux Enterprise Server 12, 11 – Ubuntu Minimal 18.04, 16.04, 14.04 Windows: – Windows Server 2016 R2, 2012 R2, 2008 R2	Linux: – openSUSE 42.x, 15.x – CentOS 6.x, 7.x – Debian 9.x, 8.x – Fedora 29, 28, 27, 26 – EulerOS 2.x – Ubuntu 18.04, 16.04, 14.04 – RedHat Enterprise Linux 7.x, 6.x – SUSE 15.x, 12.x, 11.x – Oracle Linux 7.x, 6.x Windows: – Windows Server 2019, 2016, 2012 R2, 2012, 2008	Linux: – CentOS 6.x, 7.x – Debian 9.x, 8.x, 7.x – Fedora 29, 27, 26 – Ubuntu 19.04, 18.10, 18.04, 17.10, 16.04 – FreeBSD – CoreOS – Archlinux Windows: – Windows Server 2019, 2016, 2012 R2, 2012, 2008
	yes	yes	yes	yes

	Google Cloud Platform	IBM Cloud	OTC	OVH
	yes yes	yes yes	yes no	yes no
	– Google App Engine – GSuite	– IBM Cloud API – CI/CD – Database as a Service – Network as a Service – Function as a Service, – Webserver as a Service, – Monitoring as a Service, – Backup as a service, – AI as a service	– BigData MapReduce – Database as a service, – Workspace management, – Backup as a service, – Network as a service, – Monitoring as a service	– Database as a service, – Backup as a service, – Network as a service, – Data & Analytics as a service, – Workspace Management (Horizon), – Project Management as a service, – Monitoring as a service – Telecom services – Server services
	Google Firebase / App Engine yes yes yes yes yes yes	IBM Mobile Foundation yes yes yes yes yes yes	yes yes no yes yes no	yes yes yes yes yes yes
	no yes (App Engine) yes (Cloud Pub/Sub) yes (Cloud Functions)	yes yes yes (IBM message Hub) yes (Cloud Functions)	yes yes yes no	yes (OVH Web, SSL gatway/CDN) yes (Infrastructure as Code/terraform) yes (OMNI) yes (OpenFaaS)
	yes	yes	no	yes

Logging as a Service

Questions	AWS	Azure	
Does the cloud platform provide a Logging as a Service functionality?	yes	yes	
Is the data stored in encrypted form?	yes	yes	
Which logging technology is used?	- AWS Cloudwatch - AWS Cloudtrail - AWS VPC flow logs - Amazon Cloudfront access logs - Amazon S3 access logs	- Activity logs - Azure Log Analytics - Activity diagnostics logs - Azure AD Reporting - Virtual machines and cloud services - Azure Storage Analytics - Network Security Group (NSG) flow logs - Application insight	

Image Service

Questions	AWS	Azure	
Which operating systems are offered by the provider with which versions?	Windows: - Windows Server 2008, 2012, 2016 Build 1809, 2019 Linux: - Amazone Linux 2, 2018.03 - CentOS 6.x , 7.x - Debian 8.x 9.x - Fedora 26, 27, 28, 29 - Ubuntu 14.04.x , 16.04.x , 18.04.x - SUSE Enterprise Linux 12, 15 - Oracle Linux 6.8, 7.2 - Red Enterprise Linux 6.8, 7.3	Windows: - Windows Server 2008 R2 SP1, 2008 SP2, 2012 R2, 2016 , 2019 - Windows Server 2016 Build 1709, 1803, 1809 - Windows 10 Linux: - CentOS-based 6.9 , 7.4, 7.5, 7.6 - ClearLinux - Container Linux - Debian 8, 9, 10 - Red Hat Enterprise Linux 7.x - SLES 11SP4 , 12SP3 - Ubuntu 14.04, 16.04, 18.04, 19.04, 19.10	
Can own images be uploaded?	yes	yes	
Can existing licenses be used to minimize costs?	yes	yes	
Is there an image build service?	no (But image upload service i.e import/export service can be used) Supported Formats: - OVA File - VMDK - VHD - RAW	yes Supported formats: - VHD - VMDK - VHDX - QCOW2 - RAW	
Can images be created from existing cloud instances?	yes	yes	
Are different patch levels of images available?	yes	yes	

	Google Cloud Platform	IBM Cloud	OTC	OVH
	yes	yes	yes	yes
	yes	yes	yes	yes
	– Stackdriver Logging	– IBM Log Analysis with LogDNA – Bluemix UI – Cloud Foundry Line Interface(CLI) – External logging	– cloud trace	– Logs Data Platform

	Google Cloud Platform	IBM Cloud	OTC	OVH
	Windows: – Windows Server 2008, 2012, 2016, 2019 – Windows Server 2016 Build 1709, 1803, 1809 – Windows Server 2019 Build 1903 Linux: – CentOS 6.x , 7.x, 8.x – Container-optimised OS 69-x, 73-x, 77-x, 78-x, 79-x (beta) – CoreOS 2247.5.0, 2275.2.0, 2296.0.0 – Debian 9.x, 10.x – Ubuntu 14.04.x, 16.04.x, 17.04.x, 18.04.x, 19.04.x, 19.10.x – SLES 12, 15 – SLES for SAP 12-sp2-sap, 12-sp3-sap – Oracle Linux 6.8, 7.x , – RedHat Enterprise Linux 6, 7, 8 – RHEL for SAP 7-4-sap, 7-6-sap-ha, 7.7	Windows: – Windows Server 2012 , 2016, 2019 Linux: – CentOS- Minimal 6.X, 7.x – CentOS-LAMP 6.X, 7.X – Debian Minimal Stable 8.X, 9.x – Debian LAMP Stable 8.X – Red Hat Minimal 6.x, 7.x – Red Hat LAMP 6.x, 7.x – Ubuntu Minimal 16.04, 18.04 – Ubuntu LAMP 16.04, 18.04	Windows: – Windows Server 2008, 2012, 2016, 2019 Linux: – openSUSE 15.x, 42.x – CentOS 6.x, 7.x – Debian 9.x, 10.x – Fedora 28, 29, 30 – EulerOS 2.x – Ubuntu 14.04.x, 16.04.x, 18.04.x – SUSE Enterprise Linux 12, 15 – SUSE SAP 12 – Oracle Linux 6.8 , 7.2 – Red Enterprise Linux 6.8 , 7.3	Windows: – Windows Server 2012, 2016, 2019 Linux: – CentOS 6, 7 – Debian 10, 9, 8, 7 – Fedora 29, 27, 26 – Ubuntu 16.04, 17.10, 18.04, 18.10, 19.04, 19.10 – FreeBSD – CoreOS – ArchLinux
	yes	yes	yes	yes
	yes	yes	yes	yes
	yes Supported Formats: – VMDK – VHD – VDI – VPC – QCOW2 – RAW	yes Supported formats: – VHD – VMDK – QCOW2 – AKI – ARI – AMI	yes Supported Formats: – VHD – ZVHD – VMDK – VHDX – QCOW – QCOW2 – RAW – ZVHD2 – VDI – QED	yes Supported formats: – AKI – ARI – AMI – ISO – QCOW2 – RAW – VDI – VHD – VMDK
	yes	yes	yes	yes
	yes	yes	yes	yes

Monitoring

Questions	AWS	Azure	
Dashboard	yes	yes	
Which cloud resources will be monitored?			
VMs	yes	yes	
Apps	yes	yes	
Network	yes	yes	
Load Balancer	yes	yes	
Storage	yes	yes	
Connection/Usage of external monitoring solutions	yes	yes	

	Google Cloud Platform	IBM Cloud	OTC	OVH
	yes	yes	yes	no
	yes	yes	yes	no
	yes	yes	yes	no
	yes	yes	yes	no
	yes	yes	yes	no
	yes	yes	yes	no
	yes	yes	no	no

HAMBURG CONTAINER days

The ContainerDays Journey
From 2016 until today!

More than four years ago, when we first launched ContainerDays as a community conference, there was only a small group of cloud native early adopters and evangelists in attendance. Since then, the number of attendees and speakers has been increasing every year, reflecting the growth of the container ecosystem as a whole. In 2019, our fourth edition, over 1,000 attendees came from around the world to Hamburg to discuss the past, present, and future of cloud native technologies.

In 2020, ContainerDays will again take place in the Hamburg Harbor Museum. With its rustic charm, the venue is the ideal place for attendees to network, enjoy a relaxing ride through the harbor, and refuel at a variety of local food trucks. The shipping industry has already experienced that "the container is the package of globalization" and we can't imagine a better place for the IT industry to learn the same lesson at the next edition of ContainerDays.

Join us on our exciting journey and save your ticket now!

HAMBURG
Save the Date
June 22–24
2020

www.containerdays.io