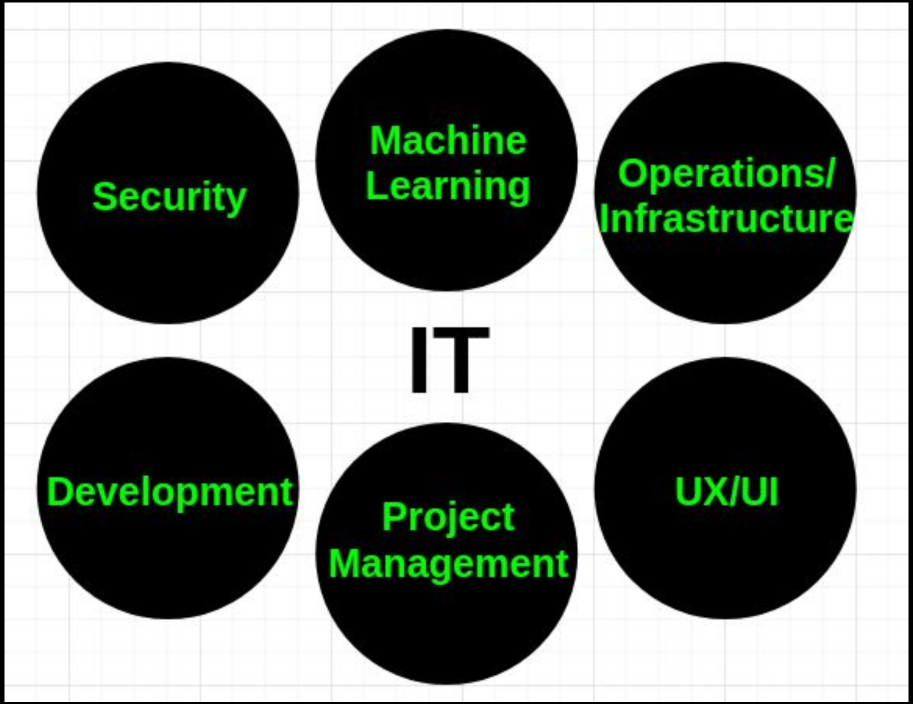
English title: Kubernetes cluster deployment for production environment
Polish title: Wdrażanie klastra Kubernetes w środowisku produkcyjnym

Student: Ewa Czechowska
Supervisor: Aneta Poniszewska-Marańda, DSc, PhD, MSc

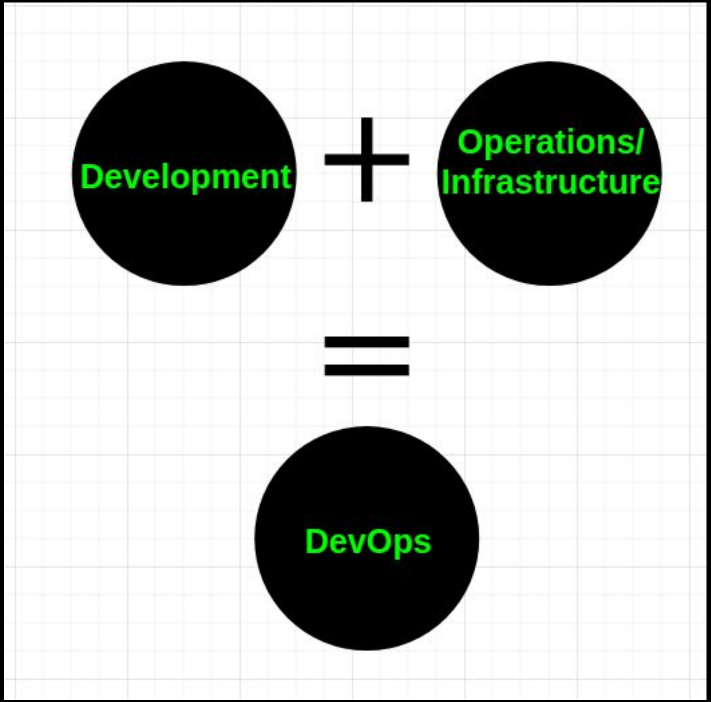
Lodz University of Technology
Łódź, 25th May 2020

IT - classification



Source: Own work

DevOps



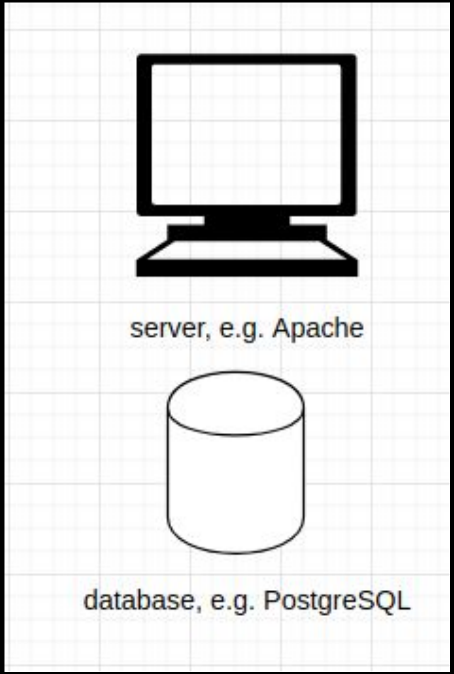
Source: Own work

DevOps

*“DevOps is **a movement t**o reduce barriers (...) [between] **development, operations,** and other stakeholders involved in planning, building, and running software.*

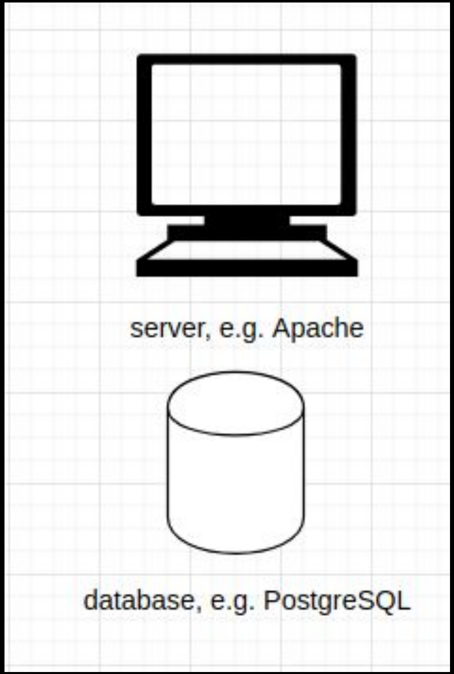
*Although technology is the most visible (...), it’s **culture, people, and processes** which have the most impact on flow and effectiveness“[2]*

An example system to deploy



Source: Own work

An example system to deploy - requirements

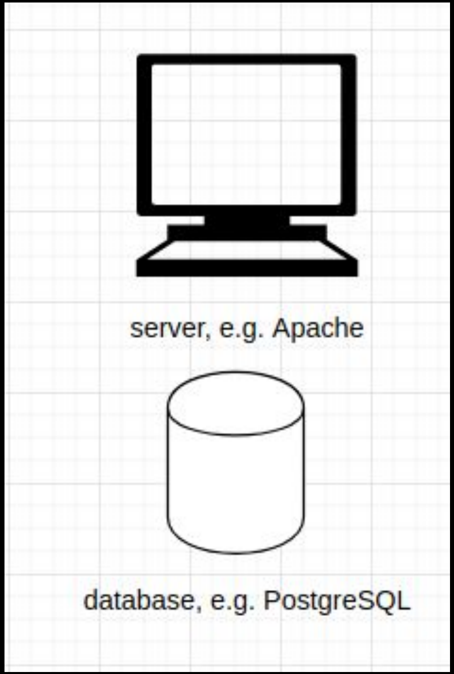


Source: Own work

Production environment requirements[2][3][6]

- It works (verified by tests)
- Central logging system
- Central monitoring system (CPU, Memory usage)
- Security (HTTPS, authorization, authentication)
- Autoscaling
- more…

Orchestration

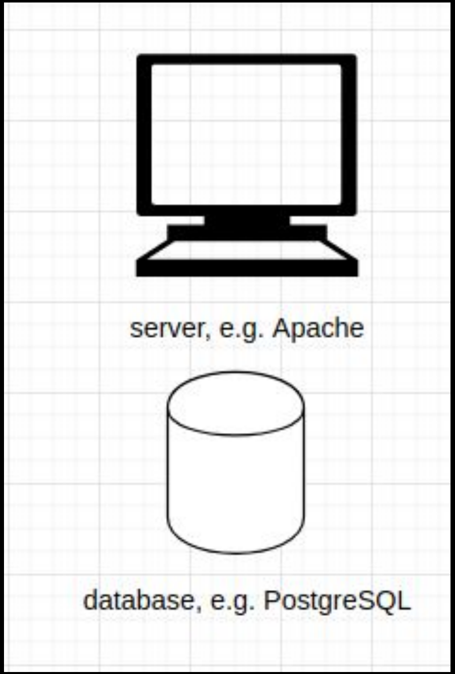


Source: Own work

Deploying multiple machines, which together create an application, can be optimized through automation.

This kind of automation is referred to as **orchestration**[8].

An example system to deploy - managed by Kubernetes

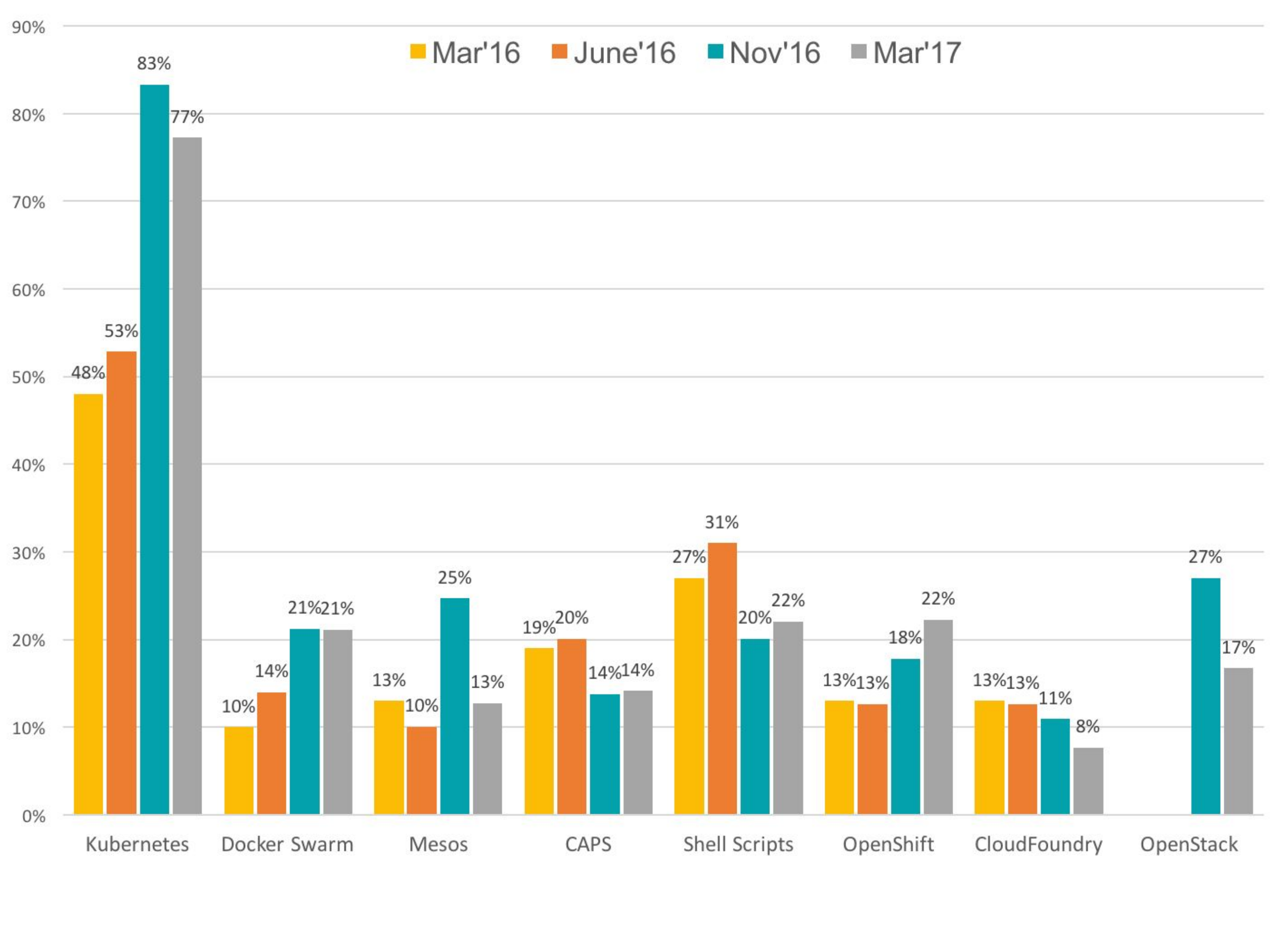


Source: Own work

Kubernetes provides a reliable and scalable **platform for running containerized workloads**[7].

Kubernetes (k8s)

- An open-source system for automating deployment, scaling, and management of containerized applications[1]
- A platform for managing application containers across multiple hosts[1]
- First released in 2014[1]
- Can be deployed on many clouds (AWS, GCP, Azure) or n-premises - unified experience[1]
- Based on 15 years of experience of running production workloads at Google[9]
- A CNCF graduated project[9]



Kubernetes alternatives,

based on CNCF survey 2017[17]

Fundamentals of effective cloud management for the new NASA Astrophysics Data System

Sergi Blanco-Cuaresma, Alberto Accomazzi, Michael J. Kurtz, Edwin Henneken, Carolyn S. Grant, Donna M. Thompson, Roman Chyla, Stephen McDonald, Golnaz Shapurian, Timothy W. Hostetler, Matthew R. Templeton, Kelly E. Lockhart, Kris Bukovi, and Nathan Rapport

Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138, USA `sblancocuaresma@cfa.harvard.edu`

Abstract.

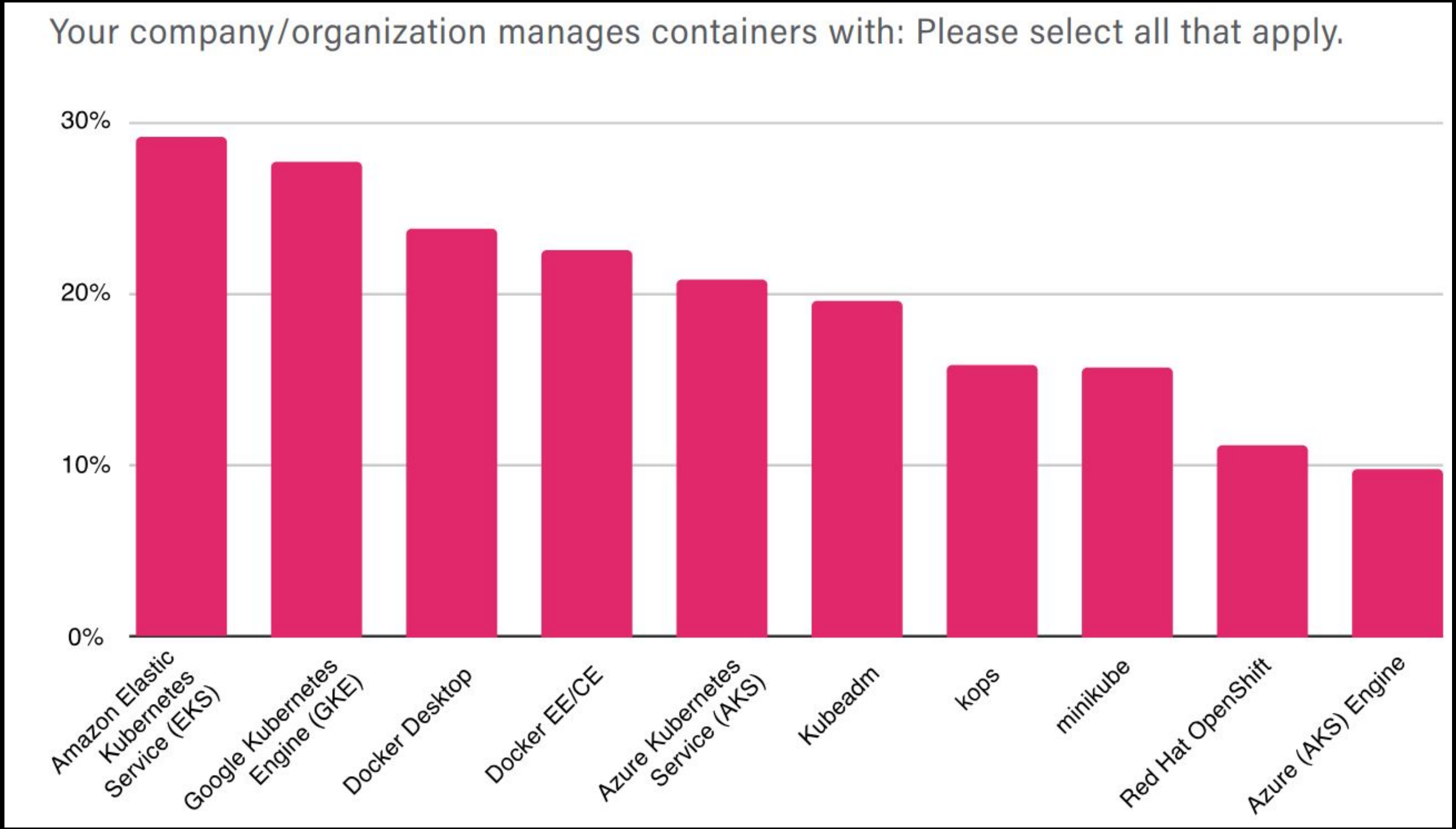
The new NASA Astrophysics Data System (ADS) is designed with a service-oriented architecture (SOA) that consists of multiple customized Apache Solr search engine instances plus a collection of microservices, containerized using Docker, and deployed in Amazon Web Services (AWS). For complex systems, like the ADS, this loosely coupled architecture can lead to a more scalable, reliable and resilient system if some fundamental questions are addressed. After having experimented with different AWS environments and deployment methods, we decided in December 2017 to go with Kubernetes as our container orchestration. Defining the best strategy to properly setup Kubernetes has shown to be challenging: automatic scaling services and load balancing traffic can lead to errors whose origin is difficult to identify, monitoring and logging the activity that happens across multiple layers for a single request needs to be carefully addressed, and the best workflow for a Continuous Integration and Delivery (CI/CD) system is not self-evident. We present here how we tackle these challenges and our plans for the future.

Kubernetes is used by NASA[19]

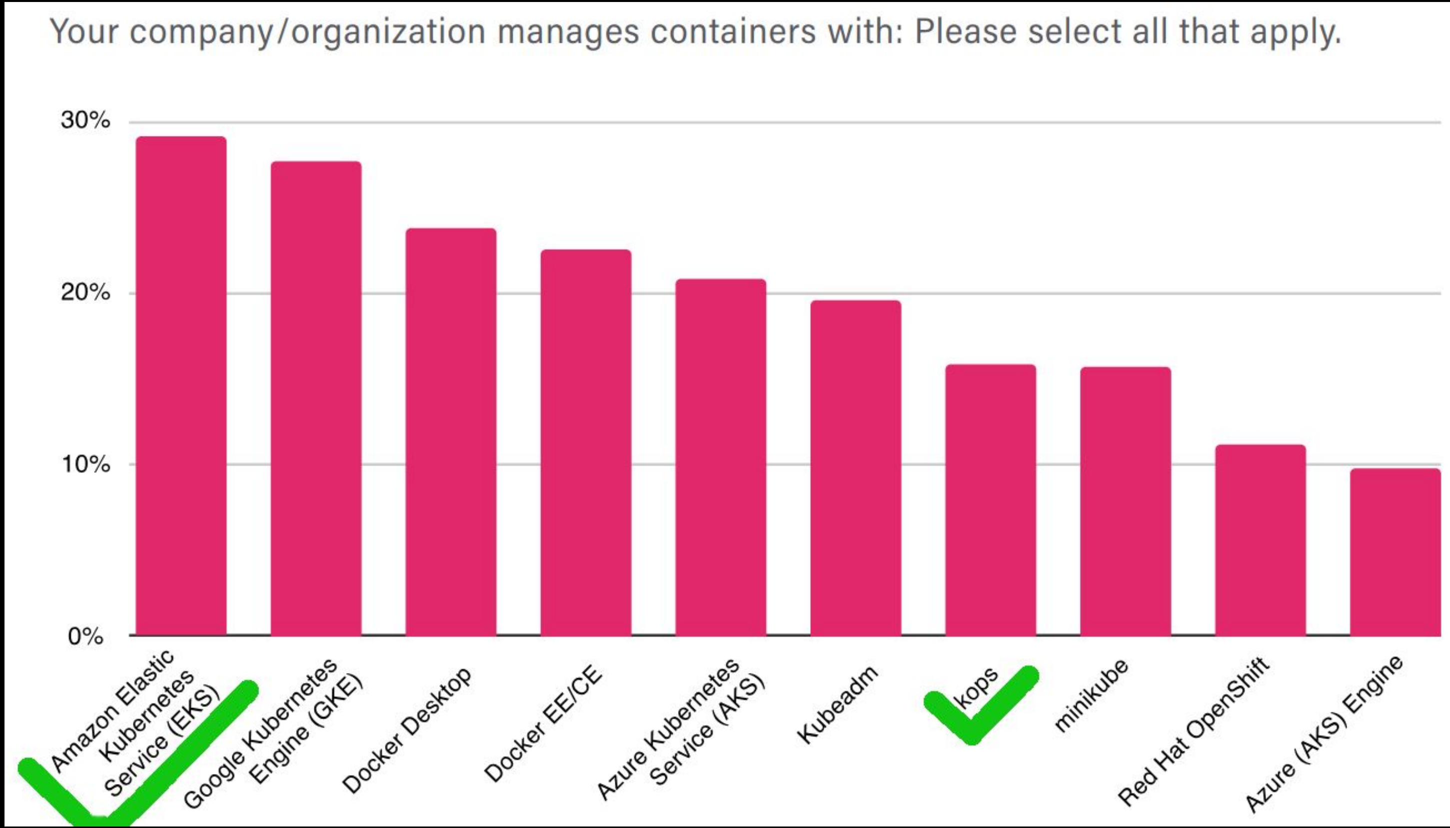
The problem is...

how to deploy Kubernetes.

Available methods of Kubernetes cluster deployment[18]



The master thesis will compare two methods



The aim of the thesis

- to deploy a Kubernetes cluster
- using two methods (comparison)
- satisfying production environment requirements

Table of Contents - part 1

Table of Contents - part 2

1.1 Topic and study scope

The choice of **comparing two deployment methods on AWS**:

- Kubernetes cluster deployment methods are described by blog posts and tutorials - not formal literature, but **practitioners use them**
- Comparison of two deployment methods on AWS is **not found in literature** (however usage of Kubernetes cluster in production environment was handled)
- **AWS is a broadly adopted cloud** with long history

1.1 Topic and study scope

The choice of focusing on **production environment**:

- To **facilitate others plan such deployment better** by letting them be aware upfront of its limitations and known issues
- To **apply a practical approach**
- Production environment is the one **creates value for businesses and customers**, it generates profit
- Because it is **challenging and interesting** to satisfy the requirements of a production environment

1.2 Aims and 1.3 Research Methodology

Theoretical approach:

- Search of existing literature (academic and other)
- Gather requirements of deployment in production environment
- Describe two deployment methods of Kubernetes cluster on AWS

Practical approach:

- Perform the two deployments while conforming to the DevOps best practices and Agile methodology
- Compare these methods in the context of production environment
- List encountered problems and try to provide solutions

1.4 Related work - Formal literature

Concern many clouds, practical approach, some production environment elements:

- **book**: Saito, Hideto. *DevOps with Kubernetes: Accelerating software delivery with container orchestrators*. Packt Publishing, 2017. ISBN-13: 978-1788396646.
- **book**: Morris, Kief. *Infrastructure as Code*. Edition 1. O'Reilly Media, 2016. ISBN-13: 978-1491924358.
- **book**: Sayfan, Gigi, et. al. *Mastering Kubernetes*. Edition: 2. Packt Publishing, 2018. ISBN-13: 978-1788999786.

book: Burns, Brendan, et. al. *Kubernetes: Up and Running: Dive into the Future of Infrastructure*. Edition: 2. O'Reilly Media, 2019. ISBN-13: 978-1492046530. - many clouds, practical approach, no production environment elements

1.4 Related work - Formal literature cont.

- **book**: Arundel, John, et. al. *Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud*. Edition: 1. O'Reilly Media, 2019. ISBN-13: 978-1492040767. - many clouds, theoretical approach
- **book**: Source: Uphill, Thomas, et. al. *DevOps: Puppet, Docker and Kubernetes*. Packt Publishing, 2017. ISBN 978-1788297615. - AWS, practical approach
- **papers** acknowledge using Kubernetes, but do not explain details of its deployment

1.4 Related work - Informal literature

Many clouds compared, only **theoretical approach**:

- Platform9. *Kubernetes Cloud Services: Comparing GKE, EKS and AKS*. [online]. 06.01.2020 [access: 20.04.2020]. Available in the Internet at: https://platform9.com/blog/kubernetes-cloud-services-comparing-gke-eks-and-aks/
- Klein, Evan. *Kubernetes as a Service: GKE vs. AKS vs. EKS*. [online]. 11.06.2019 [access: 20.04.2020]. Available in the Internet at: https://logz.io/blog/kubernetes-as-a-service-gke-aks-eks/
- Vasi, Ioana. *The State of Kubernetes Cloud Providers in 2019*. [online]. 29.05.2019 [access: 20.04.2020]. Available in the Internet at: https://www.presslabs.com/blog/kubernetes-cloud-providers-2019/

Practical approach, one method described:

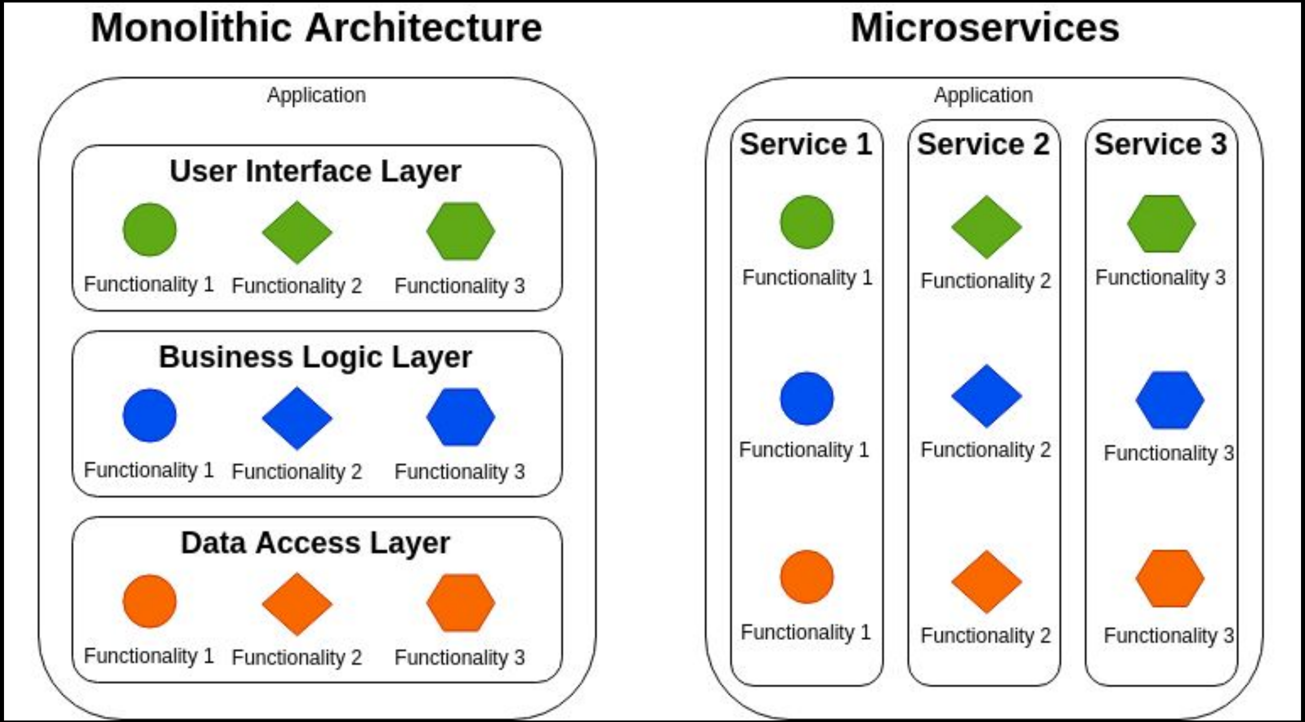
- Weaveworks. *eksctl - The official CLI for Amazon EKS*. [online]. 2020 [access: 20.04.2020]. Available in the Internet at: https://eksctl.io/
- Gruntwork. *How to deploy a production-grade Kubernetes cluster on AWS.* [online]. 2020 [access: 20.04.2020]. Available in the Internet at: https://gruntwork.io/guides/kubernetes/how-to-deploy-production-grade-kubernetes-cluster-aws
- Hightower, Kelsey. *Kubernetes The Hard Way*. [online]. 15.09.2019 [access: 20.04.2020]. Available in the Internet at: https://github.com/kelseyhightower/kubernetes-the-hard-way
- Berman, Daniel. *Deploying a Kubernetes Cluster with Amazon EKS*. [online]. 17.02.2020 [access: 20.04.2020]. Available in the Internet at: https://logz.io/blog/amazon-eks-cluster/

1.5 Structure of this thesis

- The 1st chapter serves as introduction and presents study topic, scope, aims, research methodology and related work
- The chapters: 2. and 3. are theoretical
- Chapter 2. focuses on definitions
- Chapter 3. presents available k8s cluster deployment methods
- The chapters: 4., 5. and 6. are practical
- Chapter 4. focuses on preparations before producing any code
- Chapter 5. summarizes the deployment methods performed for this study by the author
- Chapter 6. compares the deployment methods
- Chapter 7. provides a summary and future potential

2.1 Microservices

Microservices = A new **architecture for applications** which evolved as a solution to **monolith's** problems.

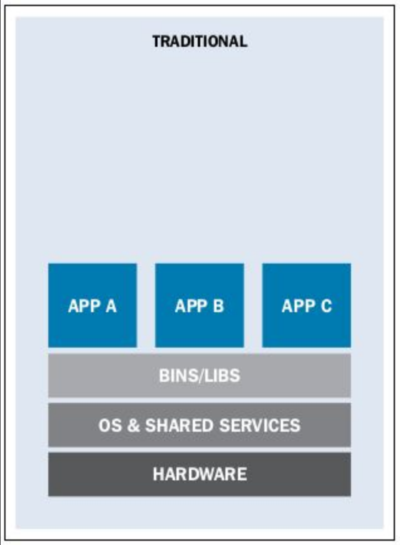


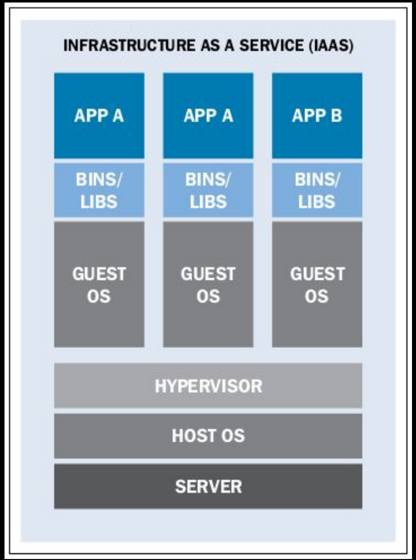
Source: Own work, based on a blog post[21]

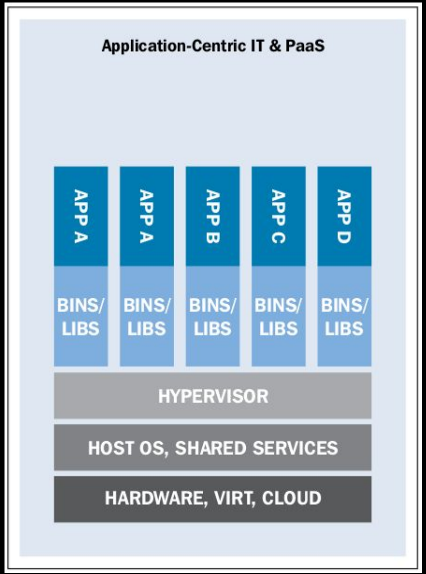
2.2 Production deployment requirements[2][3][6]

- A healthy service/cluster
- Automation
- Security
- High Availability, failover, fault-tolerance
- Backup and restore, Disaster Recovery
- Centralized Monitoring
- Centralized Logging
- Centralized Audit
- The k8s cluster must be healthy
- Autoscaling

2.3 Docker containers[5]

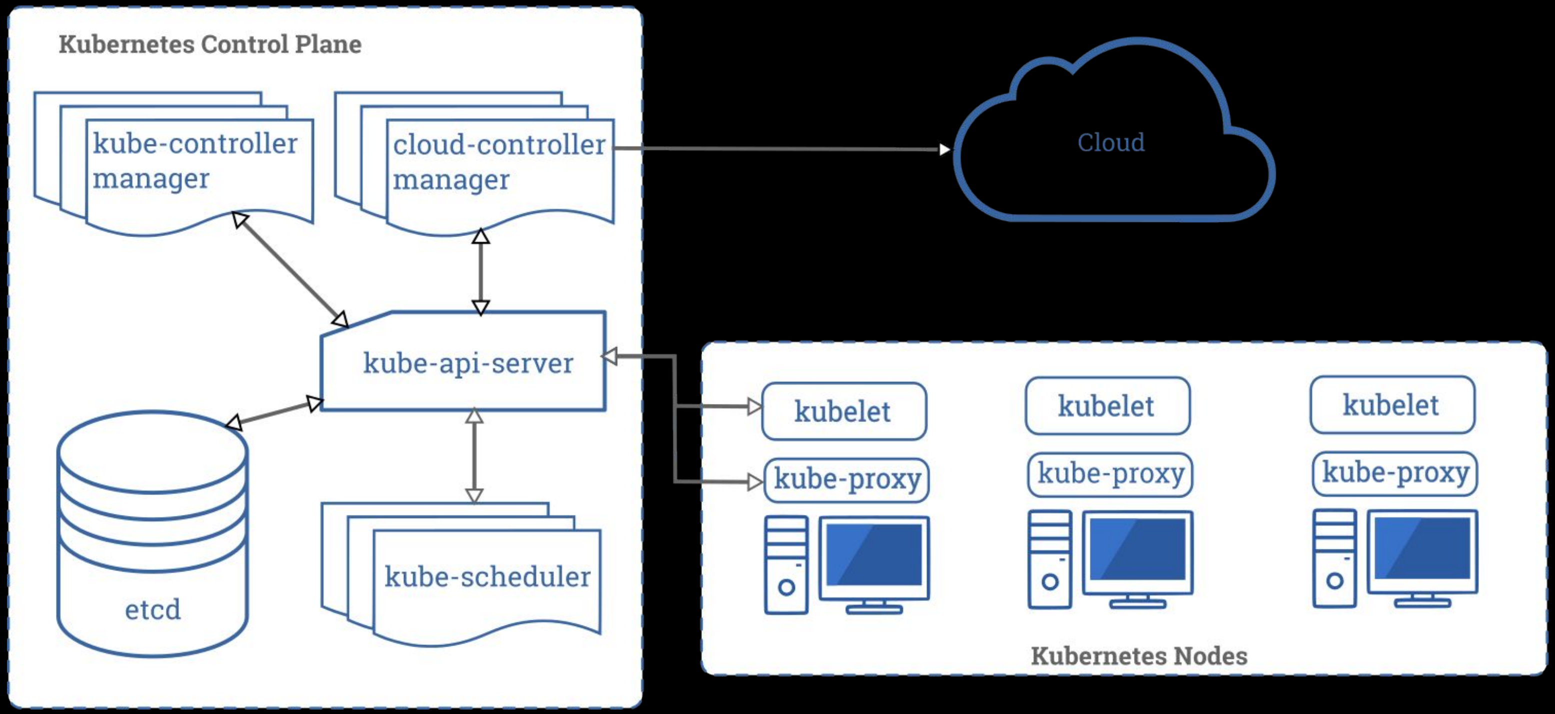






1. Applications were **installed on physical hardware** directly
2. Virtualization was invented. **Virtual Machines (VMs)** have separate OS and provide a safer environment
3. In 2013 **Docker was created** as a standard way to manage containers.

2.5 Kubernetes architecture[21]



3. k8s cluster deployment methods - Managed k8s

Amazon EKS

Azure Kubernetes Service (AKS)

Google Kubernetes Engine

3. k8s cluster deployment methods - custom solutions

	Watch	Star	Fork
digitalrebar / **provision**	9	92	15
kubernetes / **kops**	364	11.1k	3.5k
kubernetes / **kubeadm**	134	2k	366
kubernetes-sigs / **kubespray**	310	8.3k	3.5k

4.1 Chosen requirements of production deployment

- The k8s cluster must be healthy
- Automation, Continuous Integration, Continuous Development, Infrastructure as Code
- Security
- Centralized Monitoring
- Centralized Logging

4.2 Verifying that k8s cluster is healthy

- Check health status of all k8s services
- Use a status dashboard
- Check it every 1 minute
- Deploy a small app on k8s to test that k8s is usable
- Run these tests also in a CICD pipeline

4.3 Satisfying the Automation requirement

- A CICD pipeline
- Design 2 environments: testing and production

4.4 Satisfying the Security requirement

- Encryption in transit: HTTPS
- Use a static linter e.g. to check for hardcoded passwords or
- ssh keys
- RBAC, IAM - identity and access management on k8s AWS

4.5 Satisfying the Centralized Monitoring requirement

- Use a server like Nagios, Grafana, InfluxDB, Prometheus, etc.
- Use some AWS service e.g. CloudWatch

4.6 Satisfying the Centralized Logging requirement

- Use a server like Graylog, Fluentd, LogStash
- Use some AWS service e.g. CloudWatch

5.1 Production deployment - using method: AWS EKS

- Link to code
- Main steps summarized
- Overall how it went
- Some problems + solutions

5.2 Production deployment - using method: AWS with Kops

- Link to code
- Main steps summarized
- Overall how it went
- Some problems + solutions

6.1 Criterium 1: Cost

6.2 Criterium 2: The amount of problems

6.3 Criterium 3: The amount of not automated resources

6.4 - Criterium 4: Production deployment requirements met?

6.5 Comparison summary

	AWS EKS	AWS Kops
Cost		
Problems count		
Not automated resources count		
Production requirements met		

7.1 Lessons learned and achieved results

- Choose which method was better, applying specified criteria
- Maybe some more preparations were needed - more planning, design

7.2 Future work potential

- Compare more methods of deployment
- Use more comparison criteria
- Use more production requirements, e.g. automated upgrades
- Apply more load and run load and performance tests
- Instead of satisfying production requirements, test enterprise and big data deployments requirements - create a survey to get lessons learned based on long running clusters and their administration
- Test available Kubernetes tools like Velero for backup

8. Literature

1. Saito, Hideto. *DevOps with Kubernetes: Accelerating software delivery with container orchestrators*. Packt Publishing, 2017. ISBN-13: 978-1788396646.
2. Morris, Kief. *Infrastructure as Code*. Edition 1. O'Reilly Media, 2016. ISBN-13: 978-1491924358.
3. Sayfan, Gigi, et. al. *Mastering Kubernetes*. Edition: 2. Packt Publishing, 2018. ISBN-13: 978-1788999786.
4. Burns, Brendan, et. al. *Kubernetes: Up and Running: Dive into the Future of Infrastructure*. Edition: 2. O'Reilly Media, 2019. ISBN-13: 978-1492046530.
5. Uphill, Thomas, et. al. *DevOps: Puppet, Docker and Kubernetes*. Packt Publishing, 2017. ISBN 978-1788297615.
6. Humble, Jez, et. al. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Edition 1. Addison-Wesley Professional, 2010. ISBN: 978-0321601919.
7. Arundel, John, et. al. *Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud*. Edition 1. O'Reilly Media, 2019. ISBN: 978-1492040767.
8. Diouf, Gor, et. al. *On Byzantine Fault Tolerance in Multi-Master Kubernertes Clusters*. ArXiv abs/1904.06206. 2019.
9. Kubernetes Community. *Kubernetes Documentation*. [online]. 06.03.2020 [access: 20.04.2020]. Available in the Internet at: https://kubernetes.io/.

8. Literature

10. Platform9. *Kubernetes Cloud Services: Comparing GKE, EKS and AKS*. [online]. 06.01.2020 [access: 20.04.2020]. Available in the Internet at: https://platform9.com/blog/kubernetes-cloud-services-comparing-gke-eks-and-aks/.
11. Klein, Evan. *Kubernetes as a Service: GKE vs. AKS vs. EKS*. [online]. 11.06.2019 [access: 20.04.2020]. Available in the Internet at: https://logz.io/blog/kubernetes-as-a-service-gke-aks-eks/.
12. Vasi, Ioana. *The State of Kubernetes Cloud Providers in 2019*. [online]. 29.05.2019 [access: 20.04.2020]. Available in the Internet at: https://www.presslabs.com/blog/kubernetes-cloud-providers-2019.
13. Weaveworks. *eksctl - The official CLI for Amazon EKS*. [online]. 2020 [access: 20.04.2020]. Available in the Internet at: https://eksctl.io/.
14. Gruntwork. *How to deploy a production-grade Kubernetes cluster on AWS.* [online]. 2020 [access: 20.04.2020]. Available in the Internet at: https://gruntwork.io/guides/kubernetes/how-to-deploy-production-grade-kubernetes-cluster-aws.
15. Hightower, Kelsey. *Kubernetes The Hard Way*. [online]. 15.09.2019 [access: 20.04.2020]. Available in the Internet at: https://github.com/kelseyhightower/kubernetes-the-hard-way.
16. Berman, Daniel. *Deploying a Kubernetes Cluster with Amazon EKS*. [online]. 17.02.2020 [access: 20.04.2020]. Available in the Internet at: https://logz.io/blog/amazon-eks-cluster/.
17. CNCF. *CNCF Survey* [online]. 2017. [access 20.04.2020]. Available in the Internet at: https://www.cncf.io/blog/2017/06/28/survey-shows-kubernetes-leading-orchestration-platform/.
18. CNCF. *CNCF Survey* [online]. 2019. [access 20.04.2020]. Available in the Internet at: https://www.cncf.io/blog/2020/03/04/2019-cncf-survey-results-are-here-deployments-are-growing-in-size-and-speed-as-cloud-native-adoption-becomes-mainstream.
19. Blanco-Cuaresma, Sergi, et. al. *Fundamentals of effective cloud management for the new NASA Astrophysics Data System*. ArXiv. Cambridge, 2019.
20. Simon, John. *A Transition From Monolith to Microservices*. [online]. 22.05.2018 [access: 15.05.2020]. Available in the Internet at: https://dzone.com/articles/a-transition-from-monolith-to-microservices.
21. Source: Kubernetes. *Cloud Controller Manager.* [online]. 18.03.2020 [access: 20.06.2020]. Available in the Internet at: https://kubernetes.io/docs/concepts/architecture/cloud-controller/