OAMK OULUN AMMATTIKORKEAKOULU

Aleksi Ylitalo

AI assisted public cloud management

AI assisted public cloud management

Aleksi Ylitalo
AI assisted public cloud management
Spring 2023
MBA data-analytics and project management
Oulu University of applied sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Data Analytics and Project Management

Author(s): Aleksi Ylitalo
Title of thesis: AI assisted public cloud management
Supervisor(s): Ville Majava
Term and year when the thesis was submitted: 2023
Number of pages: 54 + 5

Artificial intelligence has been a hot topic and aim of the thesis is to study if generative artificial intelligence tools can help cloud professionals on managing cloud platforms and how artificial intelligence help. Specifically, interest is on the modern way of managing cloud using infrastructure as a code. Focus of the thesis is on the tools like ChatGPT and Github copilot.

Thesis is divided into different parts. First part is a theory part and it does have topics like cloud computing, infrastructure as a code and artificial intelligence. Artificial intelligence part also takes a closer look at studies on Github Copilot. Second part is the study and it was done using quantitative research methods by sending a survey to respondents. The last part is about conclusion when the findings of theory and study parts were combined.

Results of the study showed that tools are somewhat helpful. There are still a room for improvements with these tools. When respondents were asked to how helpful artificial intelligence is when writing infrastructure as a code on scale of 0-5, average was 2,6 and median was 3. Most helpful use cases for artificial intelligence tools were problem solving, learning, routine tasks and efficiency. These directly answer research question if artificial intelligence helps on problem solving and learning. In thesis there was also a hypothesis that infrastructure as code users use more artificial intelligence tools but there was no correlation when tested using chi square test.

Conclusion of the thesis is artificial intelligence can be a good tool for cloud professionals, but it is important for cloud professionals to have good foundational knowledge themselves. It is very important to notice if artificial intelligence's answer is wrong or if it is bad quality answer. This is the reason why artificial intelligence is like an assistant in today's world. It is also important to teach people more about artificial intelligence and how these tools work to increase knowledge.

Keywords: Cloud computing, Infrastructure as code, AI, Artificial intelligence, Cloud professionals

TABLE OF CONTENTS

1 LIST OF ABBREVIATIONS

AI	Artificial intelligence means intelligence of the machines in a way that machines could have intelligence like humans.
Branch	Separate version of the git main repository.
CapEx	Capital expenditure, money that is used to buy fixed assets.
CI/CD	Continuous integration and continuous deployment. Set of practices, tools and processes to deliver incremental code changes frequently and reliable way.
Cloud governance plan	A plan about company policies and rules that company is going to use to run services in cloud.
DevOps	It is a culture where development and operations are combined, and both are using same set of tooling and practices to develop and to operate.
Function as a service (FaaS)	Cloud computing service model which lets users to run a code in response to an event.
Infrastructure as code (IaC)	Provisioning and management of infrastructure by using code.
Infrastructure as a service (IaaS)	Cloud computing model which offers on-demand access to infrastructure services such as networking, storage, and computing.

JSON file

Javascript object notation is a file format used to store and transmit data. JSON file consists of attribute-value pairs and arrays.

Merge

Process of combining two branches together. For example bug fix branch to main branch.

MITRE 25

Common weakness enumeration lists 25 most dangerous software weaknesses that are currently most impactful.

OpEx

Operating expense, an ongoing expense on running business.

Platform as a service (PaaS)

Cloud computing model which offers hardware and software tools over the internet to create applications.

Principle of least privilege

Security concept in which only minimum amount of permissions needed to perform their job are granted to users.

Pull request

Proposal to merge changes from one branch to another. In pull request code changes are often reviewed and discussed with collaborators before merging.

RACI matrix

RACI matrix is a tool which helps to align responsibilities.

Software as a service (SaaS)

Software as a service is a delivery model where cloud-based applications are delivered over internet to users.

2 INTRODUCTION

2.1 Research methods

Thesis is divided to different parts, first part of the thesis is the theory part and it is further divided into three different topics. First one is about cloud computing because it is essential to understand cloud computing. Second part is about the new movement called infrastructure as code and DevOps. These are the new ways to manage infrastructure in the cloud. Last theory part is all about generative artificial intelligence and how it could help with cloud computing managed with the infrastructure as code. Last part also presents some good whitepapers about the artificial intelligence tools and theory about the tools that are currently widely used like ChatGPT and Github Copilot.

The study part of the thesis was conducted by using quantitative research methods. There was a survey and it was sent to Solita's internal channels and to LinkedIn feed. Survey was on the use of artificial intelligence tools by cloud professionals, how helpful they find the specific tool and what are the use cases where artificial intelligence tools are good at. There were also a lot more questions, but they were more generic.

In the study the main research question was: How AI will help cloud professionals in everyday work? Research question was divided into more specific questions:

1. Does artificial intelligence help to write better infrastructure as a code?
2. Do these tools help on problem solving?
3. Do these tools help professionals to better understand and learn cloud computing topics?

Also, there was a hypothesis that those cloud professionals who use infrastructure as code will use more artificial intelligence tools.

During thesis, author also created a test assignment using Github copilot. The task was to create a complex infrastructure as code file with the help of the artificial intelligence. It is important to get firsthand experience by using the tools.

3 CLOUD COMPUTING

It is important to understand what cloud computing is and its concepts. In cloud computing chapter, let's look at cloud computing more closely.

3.1 General information

There are many different definitions about cloud computing, but National Institute of Standards and Technology (NIST) uses following definition:

> *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models (National Institute of Standards and Technology, 2011).*

Let's further examine the cloud model, revisiting the meaning behind its five essential characteristics, three service models and four deployment models.

3.1.1 Five essential characteristics of cloud computing

Essential characteristics refers to a description of key characteristics that define cloud computing. The five essential characteristics in cloud computing are: On-demand self-service, broad network access, resource pooling, rapid elasticity and measured service (Lisdorf, 2021).

On-demand self-service

On-demand self-service is about provisioning computing capabilities without human interaction with service provider when these computing capabilities are needed (National Institute of Stantards and Technology, 2011).

Broad network access

Broad network access means that the capabilities are available over the network and are accessible through standard mechanisms that promote use by heterogeneous thick or thin client platforms

(National Institute of Stantards and Technology, 2011). Thick and thin client is a different software architecture. Thin client is an architecture where client software relies on server to do the computing and it requires internet connection to work. Thick client on the other hand is a software architecture where computing is done locally on the device where client resides, and it does not require internet connection to work (Spacey, 2023). Thin and thick client platforms are devices on which client software run. For example thick and thin client platforms are workstations, laptops, mobile phones and tablets (National Institute of Stantards and Technology, 2011).

Resource pooling

With resource pooling, cloud provider's computing resources are pooled to serve many consumers and it is done using multi-tenant model (National Institute of Stantards and Technology, 2011). Multi-tenancy model is a concept where multiple users share computational, storage and networking resources without seeing each other's data (Sandu, 2022).

There are different physical and virtual resources dynamically assigned and reassigned according to customer demand. The customer does not know exact locations of resource. Often there is a possibility to specify higher level location of resources like country or datacenter. In the context of cloud, these resources are storage, memory, processing, and network bandwidth (National Institute of Stantards and Technology, 2011).

Rapid elasticity

Rapid elasticity means that computing capabilities can be elastically provisioned and released. It can be done automatically to scale outward or inward with demand (National Institute of Stantards and Technology, 2011). Outward scaling is when demand is high, and scaling is done automatically to increase resources. Inward means when demand goes low, then the resources are automatically decreased. With elasticity, resources are always close to the amount that is currently needed automatically without human intervention (Microsoft, 2024). To consumer these capabilities are available for provisioning and often appear to be unlimited. These can be appropriated in any quantity at any time. (National Institute of Stantards and Technology, 2011).

Measured service

With measured service cloud services are monitored, controlled and reported. Measured service provides transparency for the consumer and to provider. With the help of measured service cloud systems automatically optimize resource usage for different services like processing or storage with help of some metering capability (National Institute of Stantards and Technology, 2011).

Today these previously listed characteristics are more blurred than decade ago, but these characteristics are still something one would expect to find from most cloud solutions. (Lisdorf, 2021)

3.1.2 Cloud computing service models

In cloud computing service models entails different ways cloud resources could be consumed. Service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Lisdorf, 2021).

Software as a Service (SaaS)

Software as a service is the most straightforward and in software as a service, everything is managed by vendor. The consumer or customer cannot program it; they can only use the service or do some configurations. These services are used on web browser. For example of these services are Microsoft 365 and Google Docs. (Lisdorf, 2021).

Platform as a Service (PaaS)

Platform as a service refers to platforms where consumer can write code to program applications and configure them. In platform as a service model vendor is responsible for managing underlying infrastructure. (Lisdorf, 2021).
For example of platform as a service is Azure App Service. With app service it is possible to create web applications to Azure without managing infrastructure (Microsoft, 2023).

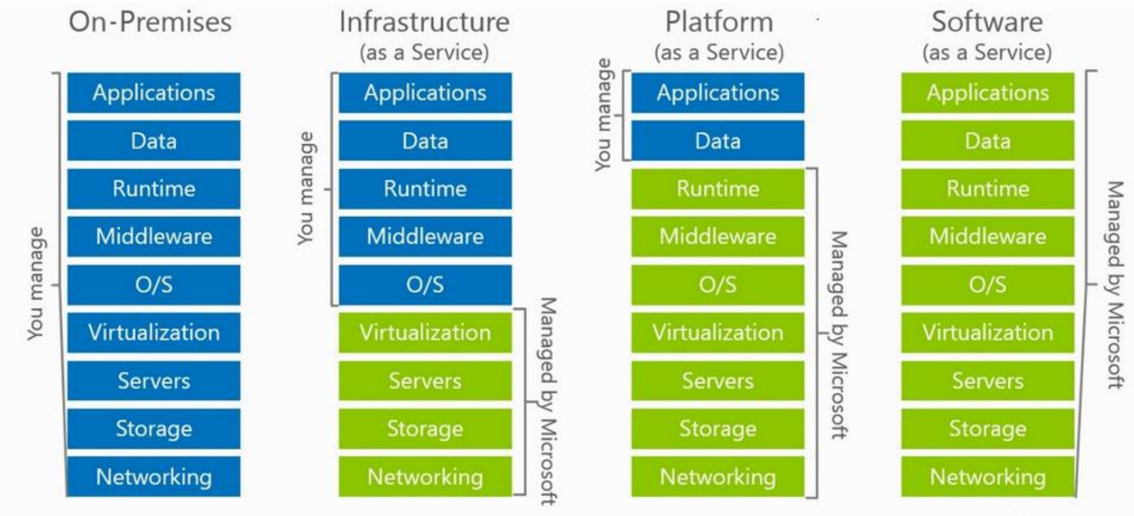
Infrastructure as a Service (IaaS)

In infrastructure as a service model, vendor provides computing resources for consumer. Consumer installs and manages all the software. Infrastructure as a service model provides most control but also requires most work from the consumer (Lisdorf, 2021). Consumer does not manage

underlying infrastructure but has access to operation system, applications, and virtual machine settings. (National Institute of Stantards and Technology, 2011)
An example of the infrastructure as a service is Azure virtual machine. With Azure virtual machine service it is possible to create a virtual machine to Azure (Microsoft, 2023). Microsoft manages physical infrastructure that is required virtual machine to run, but consumer manages software, patching of the virtual machine and configuration of virtual machine.

According to Lisdorf, in cloud computing there are some models that are in somewhere between these services. For example there are serverless computing functions and those are often called functions as a service (FaaS). These fall somewhere between platform as a service and software as a service. Containers and managed Kubernetes services like Azure Kubernetes service are also somewhere between infrastructure as a service and platform as a service (Lisdorf, 2021).

Thesis focuses on platform as a service, infrastructure as a service and function as a service because these service models are often considered as Azure infrastructure. Azure infrastructure covers all the Azure resources that can be used in solution architectures and are deployable resources (Toroman, 2018).



Cloud service models and shared responsibility (Toroman, 2018)

3.1.3 Cloud computing deployment models

Deployment models are representation of specific types of cloud environments. These are distinguished by access, ownership, and size (Thomas Erl, 2023).

Private cloud

Private cloud means cloud services that are offered from private infrastructure. According to the Lisdorf, private cloud misses most of the cloud benefits, and it is just another way to run on-premises data center (Lisdorf, 2021).

An example of private cloud solution is VMware, and it allows organizations to build their own private cloud using virtual machines and software defined networking (Bansal, 2023).

Community cloud

In community cloud deployment model there are consumers who form a group together and build a cloud they use together. The cloud they build is similar to a private cloud, but the group uses this cloud together and it is not assigned to a single consumer. It is a rare deployment model, but for example cloud vendors have so called government clouds are closest to a community cloud (Lisdorf, 2021).

Public cloud

In a public cloud, cloud infrastructure is provisioned for public use. Public cloud is owned, operated, and managed by academia, business or government. Infrastructure is on cloud providers on-premises but is accessible for public (National Institute of Stantards and Technology, 2011).

An example of public cloud providers includes Google Cloud Platform, Microsoft Azure and Amazon Web Services (Jebaraj, 2023).

Hybrid cloud

Hybrid cloud model is a combination of two or more of previous models. For example, a cloud consumer can run multiple cloud environments which are linked together, or a consumer can link private cloud with public cloud and run workloads in both (Lisdorf, 2021).

3.2 Features of cloud computing

There are many different benefits to choose cloud over on-premises datacenter. According to the Lisdorf, these benefits can be divided into seven categories: economy, security, scalability, focus, agility and sustainability (Lisdorf, 2021).

3.2.1 Economy

Big change in economy when comparing cloud and on-premises are capital expenditures in short CapEx and operating expenses in short OpEx. When using on-premises costs are CapEx but in cloud costs are OpEx (Lisdorf, 2021). In CapEx costs are up-front investments for the future use. Purchases often benefits the organization more than one tax year and therefore one of the defining features is the longevity. OpEx costs are those that incur from day-to-day operations and therefore they hold short-term value but not future value (Ross, 2023).

There are a few reasons why OpEx is considered better than CapEx in computing. When investing CapEx, then money is allocated exclusively to those particular investments. Another aspect is that servers are depreciating assets and therefore lose value over time. Lastly OpEx provides flexibility to an organization because costs are monthly fees and those are based on consumption rather than large upfront investment. With OpEx organization has an ability to scale down or up resources and directly affect the costs (Lisdorf, 2021).

Second economic aspect of the cloud is total cost of ownership. In cloud it is possible to reduce costs when comparing to on-premises. The reason because in on-premises when costs are paid upfront there are also yearly costs of support and upgrade plans. It is typically 10% to 20% of the upfront purchase price. In cloud there are no CapEx costs but in on-premises they exist together with some recurring costs (Lisdorf, 2021).

3.2.2 Security

According to the Lisdorf security is not the common driver for cloud because often cloud is considered insecure compared to on-premises, but security can be a powerful reason for cloud. (Lisdorf, 2021) An organization with own data center needs to be on top of all threats and attacks. It costs a lot of money and is difficult. The organization’s own IT department needs specialists on these fields. In cloud some of the threats are easier to mitigate. An example of such an attack is distributed denial of service (DDoS) attack. In the distributed denial of service attack bot network is sending so much traffic to website it cannot anymore serve real requests. In cloud these types of attacks are easier to mitigate because there are systems which can detect if system is attacked and machines are easier to scale in cloud to withstand demand. (Lisdorf, 2021).

Information security is also taken to account in cloud. Today encryption of data is offered by default in cloud platforms but those are hard to manage in on-premises. Another good aspect is identity and access management (IAM), in cloud IAM capabilities are enabled by default (Lisdorf, 2021).

3.2.3 Resilience

Companies and business operations are linked to IT and it is the reason why resilience is important for them these days. Resilience is important because disasters can take out entire data center and there needs to be a secondary data center. On-premises it is hard and expensive to create because one needs two different sites for data center. In cloud however resilience is easily done by a few clicks because cloud providers have many data centers in multiple regions and countries. For this same reason resilience is more efficient in the cloud because there are so many possible locations for secondary resources (Lisdorf, 2021).

3.2.4 Scalability

Scalability is one of the most important aspects of the cloud. In cloud, scalability is easy to do and scaling happens fast. Also computing resources can scale geographically so resources are close to the customers. In on-premises scaling requires a lot of upfront expenses but in cloud there are no high upfront costs (Lisdorf, 2021).

3.2.5 Focus

In cloud, companies and organizations do not need to put so much focus on managing IT infrastructure. It is an important aspect because it allows companies to focus on their expertise. For most organizations, managing on-premises infrastructure is not an advantage on the market. According to Lisdorf it can be compared to some real-world infrastructure, companies do not want to manage roads or power stations themselves (Lisdorf, 2021).

3.2.6 Agility

Agility means that it is possible to react changes fast. Covid-19 pandemic showed true importance of agility, most of the office workers worked remotely and cloud provided universal access for employees.

Elasticity is also part of the agility. Elasticity means that when demand lowers, it is possible to scale down and organization does not pay for unused resources. Elasticity is hard to create in on-premises because often there are licenses and hardware. It is not possible to just return them if demand is lower.

In cloud system resources are available in a matter of minutes when needed. There is no need to contact sales or customer service to buy licenses. One creates resource from cloud providers portal and licenses are included.

Agility drives innovation forward because it is easier to do tests and proof of concepts in the cloud. If proof of concept is discarded, resources can be easily deleted (Lisdorf, 2021).

3.2.7 Sustainability

Sustainability is rarely the primary driver for the cloud but in the current climate situation it is a good reason for cloud. With cloud it is possible to do better utilization of computing resources. The reason is because an organization with on-premises infrastructure and nine to five working hours, there are a lot of computing resources unused, but servers are still operational. In cloud these resources are shared between cloud consumers, and it provides much better utilization of resources.

Another aspect of sustainability is that with bigger data centers it is easier to build efficient energy consumption. According to Lisdorf, it is possible to reduce energy consumption for cooling about

40% when using big data centers. Also, hot air can be captured and used to heat water and because cloud data centers are big their roof can be used for solar panels to capture energy (Lisdorf, 2021).

3.3 Azure Cloud adoption framework

Focus of the thesis lies on Microsoft Azure because of author's expertise and knowledge about Azure, but also because in Finland, Microsoft Azure is the most used cloud platform according to Solita (Solita Oy, 2022). To properly start using Azure, the best practice to follow Microsoft Azure Cloud Adoption Framework about how to start using Azure and what to keep in mind when managing it.

Cloud adoption framework for Azure is a framework and it covers whole lifecycle to achieve cloud adoption goals. Cloud adoption framework consists of documentation, best practices, and tools. It will help to implement technology and business strategies for the cloud. Microsoft Cloud Adoption Framework for Azure consists of nine different methodologies. These methodologies are strategy, plan, ready, migrate, innovate, secure, manage, govern, and organize. Each of these methodologies are part of a cloud adoption lifecycle (Microsoft, 2023).

It is important to adopt cloud in the right way, because if it is not, then cloud adoption is a struggle for everyone in the organization. An organization with poorly done cloud adoption will miss benefits of the cloud (Sasa Kovacevic, 2023). For example, one of the worst things that could happen is that whole cloud adoption must be done again because first cloud adoption was poorly done and regulations were not taken care of. This could happen in highly regulated fields like banking or health care. By following Azure cloud adoption framework, it will help to do cloud adoption correctly (Sasa Kovacevic, 2023).

3.3.1 Strategy

In the strategy stage, one should define their business strategy and to document business objectives, evaluate and understand financial and technical considerations. After these, one should decide the aim of the cloud migration. These will help further to map cloud adoption strategy to cloud capabilities (Acuvate, 2022).

With cloud, there are multiple different business strategies to choose cloud. It can improve business agility, reduce costs, possibly accelerate time to market and enable expansion into new markets. But to take advantage of these, one should start by documenting business strategy in language which stakeholders and cloud technicians understand and reasons why to move infrastructure to cloud (Microsoft, 2023).

According to Microsoft most successful cloud transformation journeys start with a business outcome in mind. Transparency and cross-functional partnerships are important. Ability to speak in terms of business outcomes supports transparency and cross-functional partnerships in any cloud transformation. For example, business outcomes cloud enables are: Financial outcomes, agility outcomes, reach outcomes, customer engagement, performance outcomes and sustainability outcomes (Microsoft, 2023).

Financial considerations include issues like how cloud affects financial position, accounting key performance indicators and processes which finance teams need to understand. One of the most important aspects of cloud financial is the CAPEX to OPEX discussion that was discussed earlier in this thesis. But there are also other issues such as reduced data center footprint, increase productivity and sustainability.

Reduced data center footprint is about scaling and elasticity in the cloud. When demand is high resources can be easily scaled and when demand is low, resources can be scaled down. Often solutions which are hosted on-premises data centers are built around peak hours and there are a lot of unused resources other times (Microsoft, 2023).

Increased productivity and service delivery are possible because of the cloud. Cloud can help to unlock better DevOps processes and drive efficiency. These make it possible to hit market faster, increase productivity and ability to deploy more quickly.

Cloud migration can also help organizations to reduce carbon footprint because cloud providers are investing in green technologies. According to Microsoft, it is possible because when optimizing OPEX spending, it will also optimize carbon efficiency.

Technical considerations take account how cloud will improve how one manages and maintains cloud workloads and infrastructure. Technical benefits are those that were discussed earlier like scalability, availability, security and compliance and sustainability (Microsoft strategy, 2023).

3.3.2 Plan

After the strategy is done, then it is time to plan. In the plan stage it is important to create a roadmap and it will assist with objectives defined in strategy stage. At first one should review current infrastructure but also processes, people and other IT assets. It is important part because this review will determine what will be moved to the cloud, what is the best way to host applications in the cloud and to identify team that will drive the cloud adoption (Acuvate, 2022).

Microsoft provides four exercises on where to start with plan. These are digital estate, initial organizational alignment, skills readiness plan and cloud adoption plan.

Digital estate or a.k.a cloud rationalization is the process of evaluating assets to discover the best approach to host them in the cloud. The result for each asset should be one of the five Rs of rationalization. Rs of rationalization consists of refactor, rearchitect, rebuild, replace and rehost. There are two different models for cloud rationalization, traditional model, and incremental rationalization. Traditional rationalization works well with smaller organizations but for enterprise scale it is inefficient.

Traditional rationalization starts with inventory. Inventory is all about assets, software, hardware, applications, operating systems, and performance metrics. The inventory is a requirement, and it starts cloud rationalization. After the inventory is done, then comes quantitative analysis and there are questions like usage of assets, their dependencies and size of assets. The next phase is about qualitative analysis. Questions in qualitative analysis are often more complex and requires answers from power users and stakeholders. Qualitative analysis phase often slows the planning phase. Finally comes rationalization decision where qualitative and quantitative data creates decisions in rationalization team (Microsoft, 2022).

Incremental rationalization is more suited towards larger enterprises. Inventory in incremental rationalization is done but discovery data points are reduced. According to Microsoft one should use agentless solutions for an initial discovery to accelerate early decisions. But if environment is complex, agent-based solution might be required. In incremental rationalization outcomes are reduced and questions are also reduced.

Quantitative analysis in incremental rationalization is more reduced than in traditional rationalization. In quantitative analysis cloud adoption teams will reduce five Rs of rationalization, meaning that the outcome of the analysis is reduced to two outcomes. For instance, quantitative analysis might determine, if workload is going to be rehosted or retired (Microsoft rationalize, 2022).

Qualitative analysis part is also greatly reduced because of lesser number of outcomes and because of this there is a smaller number of questions, and qualitative analysis part does not take as much time as in traditional rationalization.

Organization alignment should not be overlooked because people make the plan reality. Full organization change is not needed but there should be some changes in team structures. There should be people who are responsible for the cloud governance and cloud adoption. It could be one team with people who share responsibilities. The team is important because it will balance speed and control. Cloud adoption people do the cloud adoption tasks and governance people are responsible for mitigating risks in the cloud by doing necessary checks in the adoption process. When risks are found then governance teams mitigate risks.

When starting cloud adoption, one should use automation from the beginning because it frees time and drives consistency. Automation can start from the small task and over time automation can evolve to more complicated tasks. For example, automation can be used to build resources like virtual machines to Microsoft Azure. (Microsoft organization alignment, 2023).

Cloud adoption will need new skills from the employees. It is important to capture employees concerns regarding cloud skills. If concerns are not addressed, it might cause employees to be slow to execute required changes. Another important aspect is to identify skill gaps, and this means to identify skills current organization does not have but is needed for cloud adoption. Finally in cloud adoption it is necessary to do cooperation between teams and it is important to have processes, which supports cooperation (Microsoft skill readiness plan, 2023).

After all these it is time to do the plan for cloud adoption. According to Microsoft, a cloud adoption plan is an iterative project plan and it helps organizations to move from traditional IT approaches to modern, agile approaches.

It is important to have a strategy before doing a plan and at minimum, the strategy should have outlined motivations, business outcomes and business justifications for the cloud adoption. It starts with digital estate and those translates the strategy into more concrete assets and workloads. One can then map these elements to more technical work. After that skilled people can execute the technical work. The cloud adoption plan combines all these topics into plan and now it is possible to budget, forecast, implement, and manage by the agile project-management practices.

3.3.3 Ready

Ready stage is about cloud adoption itself and Microsoft provides tools to help with ready stage. Azure setup guide is useful to find correct Azure Landing Zone which work best, and it also provides tools which will help to set up Azure Landing Zone. It is important to have complete knowledge about cloud adoption. Also be sure infrastructure, people, processes, and IT assets are cloud-ready (Acuvate, 2022).

Ready phase is essential before adopting, because in ready phase it is necessary to build landing zones that will host workloads. Microsoft has divided ready phase to four different phases and these phases are: cloud operating model, Azure landing zones, journey to the target architecture and Azure landing zone design areas.

According to Microsoft, *“a Cloud operating model is the collection of processes and procedures that define how you want to operate technology in the cloud”.* Cloud operating model is helpful to find the right operating model for the cloud adoption. Cloud operating model is like a traditional operating model, but its focus is on higher level of operations, and one still needs same people and processes. However, metrics change because people no longer focus on server uptime, so they need new metrics. Also, it will change the pace for changes, so it does not block innovation.

Azure landing zone is conceptual architecture for Azure. It follows key design principles on eight design areas. Landing zone uses Azure subscriptions for isolation. There are platform landing zones where important resources for the cloud environment are hosted. Application landing zone is for the application resources. Design areas for Azure landing zones are Azure active directory tenant, identity and access management, resource organization, network topology and connectivity, security, management, governance, and platform automation and DevOps. One can download landing zone infrastructure as code files from Microsoft to start easily use them. Files and instructions are available for ARM templates and for Terraform (Microsoft landing zone, 2023).

When an organization is at a starting point of using Azure landing zones, it is important to validate where the organization currently is. Is the organization at start of using cloud technologies? If the organization is at start with Azure, it is recommended to start using Azure landing zone conceptual architecture. The organization should explore Azure landing zone architecture's different design areas and choose those that are important for the organization and fits the organization's requirements.

If the organization already uses Azure but wants to align with Azure landing zone architecture, then Microsoft has guides on how to refactor and to change environments to be aligned with landing zones.

If environment is already following best practices but wants to enhance control or feature then Microsoft provides guidance for those who want to enhance security, governance, or management.

3.3.4 Adopt

Adopt is divided into four different adoptions, which are migrate, innovate, modernize and relocate (Microsoft adopt, 2023).

Migrate is about moving current workloads to cloud. Often these are different approaches: lift and shift, lift and optimize and lift and modernize. In the beginning, one should review workloads and to evaluate architecture, tools, and costs. After the review process it is time to deploy workloads to the cloud by replicating or improving workloads. Then after the deployment, release workloads. After the release workload should be tested and optimized. Finally review and document workloads and hand them over to support team. (Acuvate, 2022).

Innovate is another way to adopt. It is about using cloud innovation to maximize business value. But at first, it is important to understand customer needs, how customers interact with products. In this context customers are developers who are developing application on top of the Azure cloud platform. One should start with customer adoption and then start developing new cloud innovative solutions using cloud-based technologies (Acuvate, 2022).

The aim of the modernize is to enhance existing workloads to increase efficiency, reduce total cost of ownership and to maximize developer velocity. Often these are improved by using platform as a service solutions (Microsoft adopt, 2023).

Relocate is about moving workloads to different Azure region. It can be done any time after migration. The goal of the relocation is to respond to possible business changes, to expand global footprint and to provide services closer to end users to provide lower latency (Microsoft adopt, 2023).

3.3.5 Govern

Develop a good cloud governance plan and create benchmarks. These will help to keep track of the results at frequent intervals. Sufficient governance is about ensuring maximum return on investment and to identifying loopholes or gaps to avoid potential failures (Acuvate, 2022).

Cloud enables new models how technology is managed, adopted, and governed. It is possible to rebuild or destroy entire virtual datacenter with line of code which is executed by unattended process. It is the reason why traditional approaches do not work anymore and it is the reason why to rethink approaches (Microsoft govern, 2023).

If an organization does have existing policies for on-premises, then cloud governance should complement policies that are already in place. Cloud governance itself is an iterative process. Because cloud estate changes over time, cloud governance policies and processes should also change (Microsoft govern, 2023).

Cloud adoption is not a destination, it is a journey. There are milestones and business benefits, but the final state of cloud adoption is unknown when the journey begins. Cloud governance is important because it creates guardrails to keep organization safe through the cloud adoption journey (Microsoft govern methodology, 2023).

In governance it is important to have a rough vision about end state of cloud adoption governance. Cloud adoption framework gives a governance model that identifies key areas. Each of the areas cover different risks that a company must address when adopting cloud services. One could use these as an example end state.

Corporate policies are the first important aspect in governance. Corporate policies act like an early warning system to detect potential problems. Corporate policies consists of business risks, policy and compliance, and processes. Business risks are about understanding possible corporate risks and helps to identify them. Policy and compliance is about policy statements and statements are done by converting risks into these statements. Lastly processes are about following processes that are stated in the policies.

Discipline is the second important aspect in governance. Discipline helps organizations to create guardrails and to manage risks. There are five different areas of disciplines, cost management, security baseline, resource consistency, identity baseline and deployment acceleration (Microsoft govern methodology, 2023).

3.3.6 Manage

Manage is about managing cloud proactively and to ensure its performance and consistency. To achieve these, cloud platforms must be continually managed and monitored. Then it is possible to optimize cloud by using data and alerts from the monitoring. (Acuvate, 2022).

Cloud management is an ongoing operation and it must have a plan for well-managed operations of cloud. Microsoft provides four tools for planning management of cloud. The first tool is about defining business commitments. It is about documenting supported workloads and providing operational commitment for the business. Another aspect is to agree on management investments for each workload. Often in technical management there are highly technical terms, and it can lead to confusion for stakeholders. Due to this it is difficult to map management services to business value. But it is ideal time on cloud adoption to map these to make communication with stakeholders a lot easier. Critically mapping is about mapping workload to business processes and to rank criticality focus investments. Impact mapping on the other hand is about understanding potential outages and to aid in evaluating return on investment for cloud management. Final part is about commitment and commitment means documenting agreements with the business (Microsoft business alignment, 2022).

3.3.7 Secure

Secure stage in Microsoft cloud adoption framework provides complete vision of end state. It guides improvement of security program over time. It guides the cloud adoption journey by providing clarity for processes, models, best practices, and experiences.

According to Microsoft, security is a standalone organizational discipline and it is integrated or overlaid with other disciplines. Security is using different frameworks to plan, operate, and to capture risks. Disciplines in cloud adoption framework secure methodology relate to other security concepts. Concepts that Microsoft uses are zero trust, the open group and NIST cybersecurity framework.

When adopting cloud, organizations will discover that their static security processes cannot keep up with the cloud, because of cloud's pace of change, threat environments and change of the security technologies. The organizations security should shift to continuously evolving approaches to match pace with changes (Microsoft security, 2023).

3.3.8 Organize

When adopting cloud, it is necessary to have a well -organized organization with skilled people. Success in cloud adoption comes from people who do appropriate type of work. The work is then followed by clearly defined business goals. Microsoft provides exercises that will guide on how to organize in cloud adoption. There are four types of exercises: Structure types, cloud functions, mature team structures and RACI matrix.

The structure type helps to define organizational structure that is best for the ones operating model. At first, it is important to determine how these organizational structures are fulfilled: organization chart alignment, virtual teams and mixed teams. The organization chart alignment will map management hierarchies, manager responsibilities and staff alignment. Virtual teams are used if organizational chart alignment remains unchanged. Virtual teams will be created and are tasked to specific functions. Mixed model means involving a mixture of these two previous models.

Understanding of cloud functions is about what are the requirements to adopt and operate cloud. After one is familiar with cloud functions, it is possible to map them with organizational structure. List of functions are: cloud strategy, cloud adoption, cloud governance, central IT team, cloud operations, cloud center of excellence, cloud platform, cloud automation, cloud data and cloud security (Microsoft organize, 2023).

The third exercise is to determine the teams for earlier cloud functions. Teams can be developed organically or team can be intentionally designed to match defined team structure (Microsoft team structures, 2023).

RACI matrix is a tool provided by Microsoft to help align responsibilities across teams. It creates cross-team matrix and it identifies accountable, responsible, consulted and informed parties (Microsoft raci, 2023).

3.4 Challenges of cloud management

Cloud computing has its own challenges. Cloud computing enables rapid deployment of the resources, but it is also a problem. Many teams might be already struggling to manage existing IT infrastructure and cloud provides a faster way to deploy new resources. It will result in chaos because resource count will increase fast, and teams will be in bigger struggle than before. According to Kief Morris many organizations try to prevent chaos by tightening their change management processes and hope it will prevent chaos. But tightening change management will reduce the benefits of using cloud computing. Tight change management can also lead to so called “cowboy IT” according to Kief Morris. Cowboy IT means that people who are using cloud will try to bypass the rules and change management processes because they think they are not needed in the cloud (Morris, 2024).

Another big problem is inconsistent infrastructure in cloud because many people can create cloud resources, but their configuration can differ (Schults, 2023).

3.4.1 Approach to challenges

According to Kief Morris, modern cloud infrastructure needs a new “cloud age” approach and traditional “iron age” approach does not benefit cloud computing and it cuts benefits of cloud computing.

Cloud provides fast provisioning and configuration changes of resources, but it does not always make provisioning and configuration easier. When moving system with a lot of technical debt to cloud infrastructure it will result in chaos sooner or later.

One could use traditional governance methodologies to make cloud development less chaotic and reduce the speed of deployments. These methodologies could be rigorous change reviews, strict upfront planning and strictly segmentate responsibilities.

These models are from the iron age and they work there because according to Kief Morris, changes in traditional infrastructure and in iron age models are expensive and slow. In iron age a lot of work is put into upfront of changes because this way the change itself is not so time consuming.

In cloud age changes are cheap and fast and these should be exploited as much as possible to improve systems continuously. Iron age approach is huge tax on improvement and learning.

When using cloud, rather than using slow iron age processes with fast-moving cloud, one should adopt a whole new mindset of cloud age. Exploit fast paced technology to drive improvements, reduce risks and make better quality. But exploiting fast paced technology will need a whole new approach and new ways of thinking. The approach is called infrastructure as code, and it will drive continuous change for high reliability and quality.

3.5 Infrastructure as code (IaC)

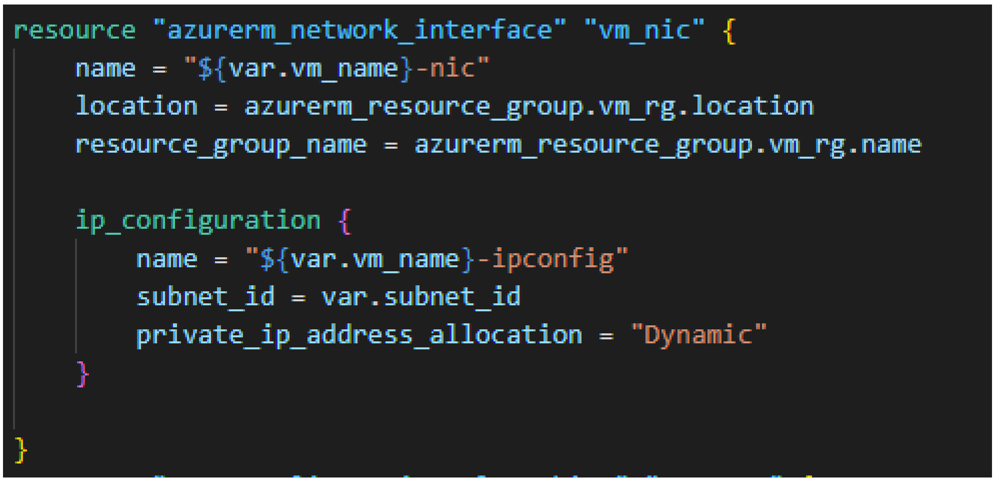
Infrastructure as a code is a way to allow automate infrastructure creation. It is based on approaches used in software development. Infrastructure as code emphasizes repeatable routines and consistency of provisioning, changing systems and configurations. For example, one makes changes to code and then uses automation to test and apply those changes to systems (Morris, 2024).

There two different types of infrastructure as code methods: imperative and declarative. In imperative method one states the steps that infrastructure as code will follow to achieve the desired outcome. In imperative method one will control how the automation happens and because of this it is possible to automate every detail and layer of command. For example, imperative tools are different scripting languages like Powershell and Bash.

In declarative method one describes the outcome that should be achieved. One will not describe how it should be done and thus leaving the process for the infrastructure as code to decide. Declarative method’s advantages are it is easier to understand, and it will provide the idempotent feature of infrastructure as code more than imperative method. Idempotent means that one can run code multiple times and the result is always the same (BasuMallick, 2022).

3.5.1 Terraform

Terraform is a declarative infrastructure as code provisioning tool created by HashiCorp and it is developed with Go programming language. The go code is then compiled to a single binary called terraform and binary is then used to provision infrastructure from computer or build server to a target provider. The provider is the platform where infrastructure is provisioned and it can be a different cloud provider such as AWS, Azure or GCP. Terraform does not need any additional infrastructure to run because terraform binary will make all the API calls to providers. With Terraform it is possible to create an entire infrastructure: servers, load balancers, databases and networks. Terraform consists of configuration files, and in these files one can write infrastructure as code to tell what terraform builds to chosen provider.



```
resource "azurerm_network_interface" "vm_nic" {
    name = "${var.vm_name}-nic"
    location = azurerm_resource_group.vm_rg.location
    resource_group_name = azurerm_resource_group.vm_rg.name

    ip_configuration {
        name = "${var.vm_name}-ipconfig"
        subnet_id = var.subnet_id
        private_ip_address_allocation = "Dynamic"
    }

}
```

Terraform code file with network interface code block

Terraform will parse code files and it will parse code into series of API calls to the cloud providers, which are specified in the code. Terraform does these API calls as efficiently as possible.

Terraform supports multiple cloud providers and often a question arises if it is possible to deploy the same infrastructure with the same code to another cloud provider. It is not possible because the provider is different, and infrastructure is not the same between different cloud providers. Terraform provides the same language to build infrastructure to different providers but every provider needs their own resource block (Brikman, 2022).

Terraform must track the current state of the environment that is managed through it. Terraform uses file called terraform state file and it is used to map resources. The state file is used to determine what changes terraform is going to make to infrastructure. Before applying any changes terraform refresh the state with real infrastructure. Terraform state file is a JSON file and it can be stored on local computer or in remote storage and it depends on configuration. Best practice is to store it in the remote storage because it can be encrypted and it is possible to share it in a team (Hashicorp, ei pvm). Terraform remote state should be secured by using encryption and following principle of least privilege on permissions. State file contains sensitive information like secrets in plain text and it is important to secure it correctly. Backups should also be taken from the state file regularly, but it is important to remember that state file contains only resources and their configurations, but it doesn't contain data from the resources (Hogget, 2023). It is advisable to logically divide terraform state file into multiple state files. This will create isolation and reduce blast radius if for example, *terraform destroy* is used accidentally (Roper, 2024).

Terraform has many different commands, but the most important ones to understand when starting to use Terraform are *Terraform init*, *Terraform validate*, *Terraform plan*, *Terraform apply* and *Terraform destroy*.

Terraform init is the first command, which should be run. It will prepare the working directory for Terraform usage and after that it is possible to use other commands in the working directory (Hashicorp, ei pvm).

Terraform validate command checks configuration files in directory and it only refers to configuration and it does not access any remote services like remote states. *Terraform validate* checks, that configuration files are syntactically valid and consistent. It is most useful for checking that value types and attribute names are correct but also for verification of reusable modules (Hashicorp, ei pvm).

Terraform plan command allows to preview changes one is about to do. The plan does the preview by creating an execution plan. Terraform does an execution plan by reading the current state of existing remote objects to ensure that the Terraform state file is up to date. Then it compares current configuration to prior state, and it will note differences. Lastly it will propose a set of change actions that should make the remote objects to match with the configuration files. One can also output plan to file and use it on apply. Usage of the file on apply will ensure changes are the same as terraform plan command output (Hashicorp, ei pvm).

Terraform apply will make the changes that are proposed in a Terraform plan. If one runs *Terraform apply* command without passing plan file from *Terraform plan*, then apply will run plan automatically before running apply (Hashicorp, ei pvm).

Terraform destroy command is a way to destroy all the remote objects that are managed through Terraform configuration. Using *terraform plan -destroy*, terraform will show what it will destroy if destroy command is used (Hashicorp, ei pvm). To be safe with this command, always review with *terraform plan -destroy* what *terraform destroy* is going to destroy. If *terraform destroy* command is used, it will automatically provide the plan of destruction, it should be reviewed carefully. One can also use targeted destruction to destroy only single resource if needed. This is done by using command *terraform destroy -target*=<*identifier of resource*>. It is important to backup Terraform state file if accidental deletion happens because this file contains the information about infrastructure that was deleted. (Chyhyryk, 2024).

3.6 Microsoft Cloud adoption framework and IaC

Microsoft Azure cloud adoption framework's ready stage has design areas, and one is about DevOps concepts including usage of version control and infrastructure as code methodology. According to Microsoft the goal of platform automation and DevOps practices are Azure landing zone management through infrastructure as code and usage of automation. Infrastructure as code and automation is used in provisioning, management, evolution, and operations.

Flexibility, scale, and agility of the cloud are enablers for the new ways of working and modern approaches to service delivery. Many of the traditional IT models are not suitable with cloud and it is mandatory to undergo operational transformation to meet the targets of the enterprise migration. According to Microsoft a good way is to use DevOps processes and tooling (Microsoft devops, 2023).

3.6.1 Version control

Code should be in version control system and more precisely Git is a recommended solution to provide team with flexibility in code sharing and management. Git provides these properties because it is a distributed version control system; in Git local copy of the code has the entire history. There are some design considerations on Git itself, for example should one use mono-repo or multi-repo? In mono-repo all source code is stored in a same repository, but in multi-repo specific codes are in different repositories. Another aspect is visibility because code is often stored in Github or Azure DevOps and there is a possibility to do private repositories and public ones. Infrastructure repositories should be private repositories (Microsoft development, 2022). The reason is that anyone can access public repositories but on private repositories it is possible to change who can access repositories. Infrastructure is critical part, and it should not be publicly accessible. For example, where public repositories can be used are documentations or open-source collaborations (Microsoft development, 2022).

3.6.2 Automated builds

Continuous integration (CI) should be used to automate builds and to test code every time it is committed to version control. There should be tests for infrastructure as code files as well as for application code. Tests should be unit tests and validation of code to ensure it is in correct format. Another important point is to keep secrets out of the version control, and it can be done by using Azure key vault and then build and release jobs read secrets from the Azure key vault (Microsoft development, 2022). There are many different CI tools but one of them is Azure DevOps and there it is possible to use Azure pipelines feature to run CI (Microsoft learn CI, 2022).

3.6.3 Deployment strategy

The best practice in deployment strategy is to use continuous delivery (CD). According to Microsoft, continuous delivery is the process of automating build, test, configuration, and deployment from build to a production environment. By using CD processes like release pipeline in Azure DevOps, it is possible to create a whole infrastructure and deploy new build from software to the infrastructure by using release pipelines. An important aspect to keep in mind is that it is possible to create infrastructure and build software to different environments like to development, testing and production. Often when developing software, it is done in development environment, and it is just used for development, not for real uses. After development is done, then deployment is going to test environment where software is tested before it goes to production. Production environment is environment where software is used. With infrastructure as code, it is possible to create whole infrastructure as a part of continuous delivery when creating these different environments (Microsoft CD, 2022).

According to Microsoft, continuous delivery ensures that code is always deploy ready because it is building, testing, and deploying code to production-like environments automatically. By combining CI/CD together, it will help to detect problems earlier and it ensures one can quickly release properly tested updates. When using IaC in CI/CD it is important to include review checks so one can preview changes if resources are created, deleted, or modified. Review checks are done by using pull requests. With pull requests it is possible to require reviews before changes are merged to important branches and after merge, CI/CD automation is triggered. (Microsoft development, 2022).

3.6.4 Rollback strategy

There should be a plan on how to use rollback with infrastructure as a code and version control system. Roll back means deployment is reverted to latest known good state and it is important because it allows to recover from failed deployment. In Git there are undo changes functions that one can use to do roll back (Microsoft development, 2022).

3.6.5 Security

When using infrastructure as code, best practice is to implement DevSecOps practices. Security should be implemented as early as possible to CI/CD pipelines. The recommendation is to create a strategy to implement static code analysis, secret scanning, dependency scanning and unit testing to CI/CD pipelines (Microsoft devsecops, 2022).

3.7 Motivation behind IaC

3.7.1 DevOps

Important note when talking about DevOps is that there is no single correct definition but often it is defined as a cultural change and its aim is to break down barriers between development and operations. In traditional culture, development and operations are separate teams with clear responsibilities. Another important note is that Azure DevOps is a tool and it should not be mixed to DevOps culture.

Development teams develop new features and fix bugs in the application and it is important for them to introduce as many changes as possible to meet requirements.

Operations on the other hand are responsible for infrastructure management, application deployments, monitoring infrastructure and reacting alerts. Focus of the operation team is to keep infrastrucure and software running and often it is better to have less changes to keep these running. It will lead to unhealthy situations because both team's responsibilities oppose each other.

DevOps culture is the answer to these problems because in DevOps teams are multi-skilled and teams consists of development and operations experts. They have the freedom and responsibility to manage the application and infrastructure, and they work together to meet their shared goal. The goal is to deliver high-quality application that brings business value and ensures stability. It requires knowledge from both development and operations.
It means in the practice, DevOps team is now responsible for creating and maintaining their infrastructure. Also, IT professionals often start to use development techniques to their work. They do not use anymore manual user interfaces, manual application, and verification of changes. Instead, they start to use advanced text editors, automated installation scripts, infrastructure as code and CI/CD pipelines to enable automation. Infrastructure as code allows developers and operations to describe and configure infrastructure needed by application deployment and to better work together (Eduard Keilholz, 2022).

3.7.2 Configuration drift prevention

A second big motivation for infrastructure as code is to prevent configuration drift. Configuration drift is when there are differences between different environments or within an environment. For example two virtual machines in environment can be with different configurations. Configuration drift often happens when configuring infrastructure manually because of unexpected, incomplete, or incorrectly executed change. There are many problems when configuration drift happens. Test environment will not represent production environment and can cause problems when deploying to production environment.

Infrastructure as code helps to remediate configuration drift because infrastructure code is held at version control and it is possible to re-apply infrastructure regularly (Eduard Keilholz, 2022).

There are many other benefits when using IaC. With infrastructure as code, it is possible to use automation and automation itself brings benefits like guaranteed outcomes and environment reproducibility.

3.7.3 Automation

Automatic infrastructure creation and configurations save time but also outcomes are guaranteed. It means that when one writes infrastructure code file, the same file can be used repeatedly, and it is guaranteed the outcome is always the same. This saves time and improves quality greatly. It is not the case if doing manual provisioning and configuration because it is very error prone to do them manually (Eduard Keilholz, 2022).

If infrastructure as code files are stored in version control system like cloud adoption framework suggest, then one can use automation tools to do automated deployment. There are multiple tools which can do this, but most common ones are Github actions and Azure DevOps pipelines. Another good ability these services give is that deployments can be run from a specific location, and it is secured by permissions rather than running them from an individual's computers (McKendrick, 2023).

3.7.4 Reproducibility of environments

If there are ready made infrastructure code files, then the cost of infrastructure creation is almost zero. It is possible because one needs to just start the tool that will take care of infrastructure creation.

It allows developers to be more flexible because it is fast and easy for them to provision new infrastructure for testing new features. In other words it helps developers to be more innovative (Krief, 2022).

There are also possible cost savings because when using infrastructure as code with automation, it is possible to delete environments in the evening and create them again on the morning. It is often case with testing environments (Eduard Keilholz, 2022).

3.7.5 Documentation

In declarative approach, infrastructure as code files are written in human readable format. It drives many benefits like version controllable, auditable, and reviewable changes.

Source control system for IaC file is important because source control is the single source of truth for the file. It includes full history of the changes and it is easy to see what changes have been made. If there is a problem with the latest change, it is possible to roll back easily to previous version. With these auditing is also easier because auditors or any other third party can easily see full log of changes.

Reviews are an important aspect of version control. With reviews it is possible to enforce standards before any change is final. Standards can be automated build checks, automated formatting checks and to enforce peer reviews (Eduard Keilholz, 2022).

3.8 Challenges of IaC

Even though IaC has many benefits over traditional manual infrastructure modification and configuration, it still has its own challenges that are good to keep in mind.
Using IaC requires a lot of knowledge because there are no UX when doing infrastructure code. Often in cloud provider portal there are tips about configuring and best practices. One will also see all the options available more easily using cloud providers' portal.

Another aspect is investments put into IaC. It will take more time to write code but it will pay off later on when modifying code, traceability of the code and if reproducing more environments. (Ronin, 2022)

4 ARTIFICIAL INTELLIGENCE

Could artificial intelligence tools help to manage cloud computing, and could these tools also help to answer the challenges of the infrastructure as code? Before looking at AI tools, let's look at generative artificial intelligence as a whole and how generative AI works.

4.1 Generative artificial intelligence

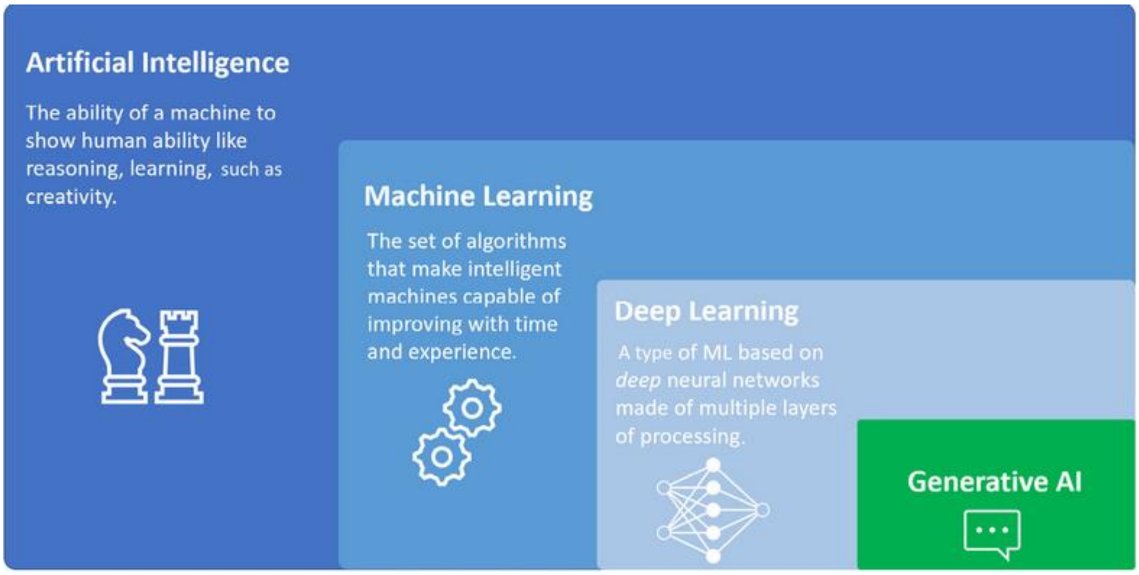
Artificial intelligence has developed forward a lot in recent years. One of the areas where it has improved a lot is generative artificial intelligence. Generative artificial intelligence is a subfield of artificial intelligence and deep learning, and it focuses on creating new content. Content can be music, art, images, videos, and text. It does this by using algorithms and models that have been trained on existing data using machine learning techniques. (Alto, 2023)

To better understand generative artificial intelligence, it must understand relationships between artificial intelligence, machine learning, deep learning and generative artificial intelligence.
Artificial intelligence is a broad field of systems, and it can perform various tasks, showing human intelligence and ability and it can interact with the ecosystem.

Machine learning is a branch of artificial intelligence, and it focuses on creating algorithms and models that enable artificial intelligence to learn and improve over time and training. These models learn from existing data and automatically update parameters as they grow.

Deep learning is a sub-branch of machine learning because it encompasses deep machine learning models. Those models are neural networks and they are suitable in computer vision or natural language processing. Machine learning and deep learning models are typically referred as discriminative models and aim of those is to make predictions and inferences patterns of data.

Generative artificial intelligence is sub-branch of the deep learning. It does not use deep learning neural networks to cluster, classify or make predictions on existing data. Generative artificial intelligence uses neural network models to generate new content like text, images, videos and music. (Alto, 2023)


Artificial Intelligence
The ability of a machine to show human ability like reasoning, learning, such as creativity.
Machine Learning
The set of algorithms that make intelligent machines capable of improving with time and experience.
Deep Learning
A type of ML based on *deep* neural networks made of multiple layers of processing.
Generative AI

Relationship between fields of AI (Alto, 2023)

Generative artificial intelligence models are often trained on vast amount of data and they generate new examples from zero using patterns found in the data. Process is different from discriminative models that try to predict label or class of given example (Alto, 2023). An example of discriminative AI is that it answers questions like "is this image drawing of rabbit or lion?". Generative AI can create a picture when told "draw me a picture of rabbit or lion". (Fruhlinger, 2023)

4.2 OpenAI

OpenAI has developed one of the most promising technologies regarding generative artificial intelligence, ChatGPT (Alto, 2023).

OpenAI was founded in 2015 and it is a research organization. Its mission is to provide artificial intelligence that benefits humanity according to Valentina Alto. OpenAI most notable achievements are generative models and their model called Generative Pre-trained Transformers (GPT).

GPT-2 was launched in 2019 and it was trained with WebText dataset. The dataset contained over 8 million documents and 40GB of text from URLs shared in Reddit that had atleast 3 upvotes. GPT-2 had 1,2 billion parameters.

After GPT-2 OpenAI launched GPT-3 in 2020 and it had dramatically improved benchmarks over GPT-2. The model had 175 billion parameters.

Then came GPT-3.5 and it is built upon GPT-3. The goal of the new model was to improve natural language understanding and its generation abilities. GPT-3.5 can perform complex tasks like composing coherent paragraphs and essays. It can also create computer programs in natural language. ChatGPT is one of the tool built using GPT-3.5 model (Alto, 2023).

OpenAI also created Codex and it is a set of models that can generate code in multiple programming languages. It can translate natural language to code (Alto, 2023). Codex training data contains code from publicly available sources like from public Github repositories. The model's training data contains billions of lines of code and natural language. Codex works same way as GPT-3's models meaning that only way it can affect world is through mind of the reader because reader issues a prompt to GPT-3's models. In codex one can prompt to codex and it will generate working code for you. It is a general-purpose programming model and it can do any programming task like transpilation, explaining code and refactoring code. Most notable software that has been built on top of OpenAI codex is Github Copilot (Wojciech Zaremba, 2021). Codex was deprecated on March 2023 because OpenAI's new model GPT-3.5-turbo does programming tasks as well. In benchmark results are same and, in some cases, even surpasses Codex models. At the time of writing, ChatGPT uses GPT-3.5-turbo model. (Alto, 2023).

4.3 Github copilot

In 2021 Github launched Github copilot with OpenAI. It is used within a code editor and it can suggest code for developers (Gershgorn, 2021). Another way to use Github copilot is to ask from Github copilot to create code by using prompt engineering techniques. Prompt engineering is about designing prompts to generative artificial intelligence in a way it gives wanted output for user (Mckinsey & Company, 2023). In a Github copilot these prompts are done by using code comments and then Github copilot will create code for the user. The study suggests that comments should be kept short because Github copilot performance degrades if the prompt is too long (Arghavan Moradi Dakhel∗, 2023).

According to Github, copilot analyzes code developer has created earlier and it will generate a matching code. It works best with Python, Ruby, Go, Javascript and Typescript. Github itself sees copilot as a pair programmer. Often in projects there are two programmers who review and fix each other's code but with the Github copilot pair can be virtual (Gershgorn, 2021).

4.3.1 Github copilot security

An important perspective that cannot be skipped when talking about Github copilot are security concerns. Even in context of the security, there are multiple perspectives, security of the code it suggests and security of the tool itself. The thesis focuses on both perspectives.

There is a study done on the code suggestion part of the Github copilot and if code is good quality in terms of security. The study was done by prompting Github copilot to create code in scenarios that are relevant to cyber security weaknesses like those which are from MITRE top 25. In the study, they produced 89 scenarios for the copilot to complete and it produced 1689 programs. 40% of those were vulnerable. (Asleep at the Keyboard? Assessing the Security of Github Copilot's Code Contributions, 2021). The reason why Github copilot generated code has many security flaws is that it is trained with Github's public repositories, which can contain a lot of potential security vulnerabilities. Another problem is that security field is constantly evolving and practices that are currently best practice can be obsolete in the future. It can lead to a problem where Github copilot training dataset can contain a lot of outdated code. Study suggests that when using Github copilot, it should be paired with a security aware tooling like CodeQL and the developer should be aware of the code it generates (Asleep at the Keyboard? Assessing the Security of Github Copilot's Code Contributions, 2021).

Concerns about the tool itself are secret leakage and code exposure. According to blog post by Vladimir Radchenko, Github copilot gets access to the developer's repositories and these are used as a large dataset to improve Github copilot. There are no options to modify what it can access. One should remember when starting to use Github copilot, one shares the code with Microsoft (Radchenko, 2022). In Github copilot webpage, it seems that even if one uses Github copilot for business, it still collects data from files in background. There are some policy settings for Github copilot for business that can be fine-tuned regarding of the data collection, processing, and retention of data (Github, ei pvm). According to Radchenko even if one disables telemetry and snippet sending, Github copilot will still send code to Microsoft. Secret leakages are also part of an earlier problem; this is because Github copilot can recommend secrets in the editor (Radchenko, 2022). Secret leakage is a problem if Microsoft gets breached and there is Github copilot data leakage, How can customers be sure that secrets have not been compromised?

If working with highly sensitive data, then one should disable Github copilot in their editor because there are no settings to deny code sending in the background. For organizations, it is important to disable “Allow Github to use my code snippets for product improvements”. Github says it does not use data to train Github copilot if the setting is disabled.

Secrets leakage can be mitigated by using environment variables when using secrets or by using password manager to store secrets and to then retrieve secrets from API (Radchenko, 2022).

4.3.2 Github copilot with programmers

Programming and infrastructure as code are quite similar but infrastructure as code does not have as much logic as programming. It is important to take a look at results on how Github copilot performed in the study.

There was a study conducted in York university, Toronto in which it studied if Github copilot is a liability or asset in the software development. The study found that Github copilot can be both. In hands of a novice programmer Github copilot can be a dangerous because they might not understand coding context or problem that well and Github copilot can suggest buggy and bad code quality. On the other hand, study also found that Github copilot is an asset when more experienced developers were using it because they can easily see problems in the code Github copilot generated and often code that was generated is easily fixable by experienced developers. The overall finding in the study was that Github copilot can compete with humans in programming, but Github copilot has lower diversity and correct ratio for solutions than humans. Github copilot is not good at understanding when combining different solutions together. These are the reasons why it is important to understand context and to carefully read what Github copilot has generated to avoid problems in the programming projects (Arghavan Moradi Dakhel∗, 2023).

The study found that Github copilot does not necessarily improve time spent on programming, but developers still want to use Github copilot because it can provide a good starting point when starting a project. An Interesting finding in the study was that it is easier to fix Github Copilot’s buggy code than human generated code (Arghavan Moradi Dakhel∗, 2023).

5 STUDY

5.1 Research questions

Main research question on the thesis was how AI will help cloud professionals on everyday work? The question was broad so there are additional accurate questions to help in research. These questions were:

1. Does artificial intelligence help to write better infrastructure as a code?
2. Do these tools help on problem solving?
3. Do these tools help professionals to better understand and learn cloud computing topics?

Also, one important hypothesis was if someone uses infrastructure as code, then the people using IaC uses artificial intelligence tools more often because AI can provide whole infrastructure as code blocks when asked.

5.2 Data gathering

Data was gathered using quantitative methods by sending a survey to LinkedIn and to large finnish IT-consulting company's internal communication channel. There were 32 respondents to the survey and it is a minimum number of respondents to do a research. Error margin of the study is 0.3464. Population size is 121 000 and sample size 32. Confidence level is 95%. Population size is the IT worker count in Finland according to Ficom (Ficom, 2023)

5.2.1 Survey questions

There were 16 questions in the survey but five of them were made with conditional rules. These questions will only appear if a respondent selects specific AI tool. For example if respondents answer they have used ChatGPT, then ChatGPT question will appear to respondents to review how helpful ChatGPT is and other AI tool questions will be hidden. There was also a question if respondents have used infrastructure as code and if respondent selects yes, then there will be question if AI helps to write infrastructure as code.

Questions were:

1. What is your age? (Select one)
2. From what continent are you from? (Select one)
3. What cloud provider do you use? (Multiple choices)
4. Do you use infrastructure as code to manage cloud? (Select one)
5. Would you use AI tools at your work? (Select one)
6. How important do you see AI tools in the future? (Scale 0-5)
7. Have you used AI tools at your work? (Select one)
8. Do you find AI tools helpful when writing infrastructure as code? (Scale 0-5)
9. What AI tools you have used at your work? (Multiple choices)
10. Do you find Google Bard helpful? (Scale 0-5)
11. Do you find ChatGPT helpful at work? (Scale 0-5)
12. Do you find Bing Chat helpful at work? (Scale 0-5)
13. Do you find Github Copilot helpful at work? (Scale 0-5)
14. Do you find AI tool chosen on option other helpful at work? (Scale 0-5)
15. In your opinion, what are the most helpful use cases for AI tools you have used? (Multiple choices)
16. What are two keywords that come to your mind about AI tools? (Write two words)

The first two questions were generic questions about age and the continent where the respondents are from. Age question is there to find out if age matters if using AI tools. With continent, it is possible to find out if the continent has impact on the results.

The third and fourth questions are more generic questions about what cloud providers respondents use and if respondents use infrastructure as code to manage cloud. It is important to find out if AI tools are more helpful for specific cloud providers and if respondent uses infrastructure as code to manage cloud.

Next three questions are if respondents would use AI tool at work, how important respondents find AI tools in the future and if one has used AI tools. These questions are there because it will show what the attitude towards artificial intelligence is and if people who haven’t used AI tools would use them in the future. Then there is a simple question if respondent has used AI tools. If respondent hasn’t used them then it will end the survey.

Respondent will encounter question number 8 if they indicate the use of infrastructure as code for cloud management in their response. The question is important because it will answer the one additional research question if AI will help to write better infrastructure as code.

Then comes questions about what AI tools respondent has used and every tool respondent chooses, will open additional question about if chosen tool is helpful at work. These questions are important for the study to find out what AI tools most used and what AI tool are most helpful for cloud professionals.

The second last question is most important question in the survey and it is about the helpful use cases for AI are and how it helps. This is one of the research questions in the thesis.

Final question investigates what are the two keywords respondents think about AI. This will also map mood of the respondent and how they see AI. AI can be scary for some, and question is here to map how respondents think of AI.

5.3 Results to research questions

One of the questions was if artificial intelligence will help to write infrastructure as code. In survey there was a question "*Do you find AI tools helpful when writing infrastructure as code?*". Scale is 0- not at all and 5 – very helpful. There were 25 responses to this question. Average value is 2,6 and median 3. These results show that AI tools are somewhat helpful when writing infrastructure as code, but there are still a lot to be improved.

To answer a research question *"How AI will help cloud professionals on everyday work?"* let's look at 2nd last question. According to results most helpful use case for AI is that it helps with problem solving. Second question also answers one of the sub questions on *"Do these tools help on problem solving?"*. It is the most important use case by a large margin because problem solving was selected by 63% of respondents.

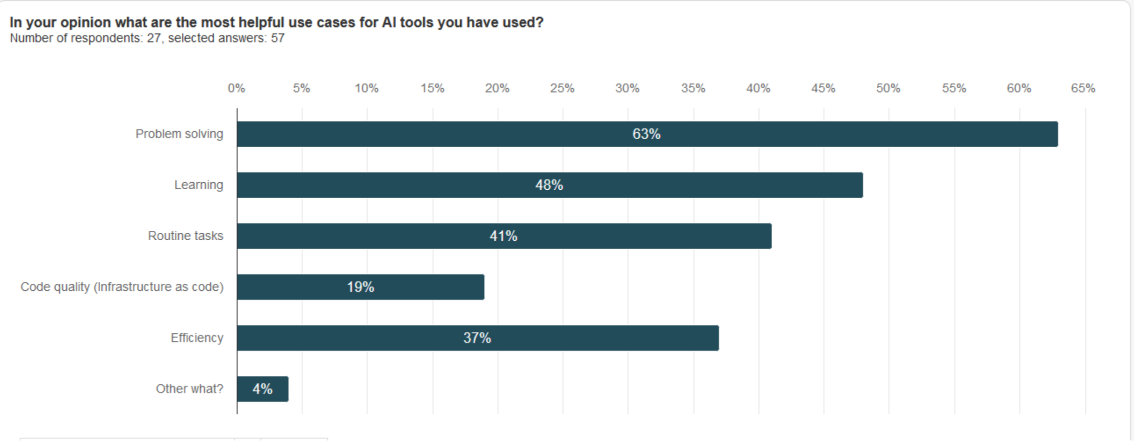
Next helpful use case is learning, and it was selected by 48% of respondents. This answers sub question: *"Do these tools help professionals to better understand and learn cloud computing topics?"*. It seems like AI is a good tool to help learning about cloud computing.

Third most selected use case was AI is good at routine tasks. It was picked up by 41% of the respondents. It means AI can help with routine tasks that cloud professionals perform in their everyday work. An example could be writing same IaC code blocks over again and AI can help to write these code blocks faster.

Efficiency was the fourth selected answer chosen by 37% of the respondents. It could be that AI is helping to write code faster. Efficiency was also an important use case because it was selected by many respondents.

Last use case was code quality in infrastructure as code and it received only 19% of votes. It shows that AI tools are not helpful in code quality and can cause problems with code quality. The study used earlier in the thesis in chapter 4.3.2 supports this. Code quality can be bad with AI tools and the developer must be experienced to understand possible problems AI generates in the code.

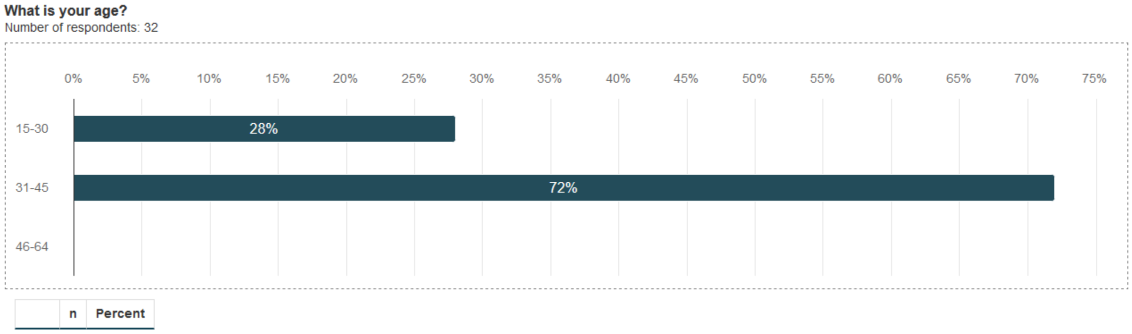
Lastly there was the option to write use case and it received one response. The respondent wrote that AI is offering new ideas and viewpoints. New ideas and viewpoints are an important aspect because with generative AI it is possible to chat with AI and receive ideas and new viewpoints. This is valid but it received only one response, it received so few because it was free text box where one can write. If it had been one of the options in the survey, it could have received more responses. These are the reasons why this is out of scope in the thesis but it is important to note.



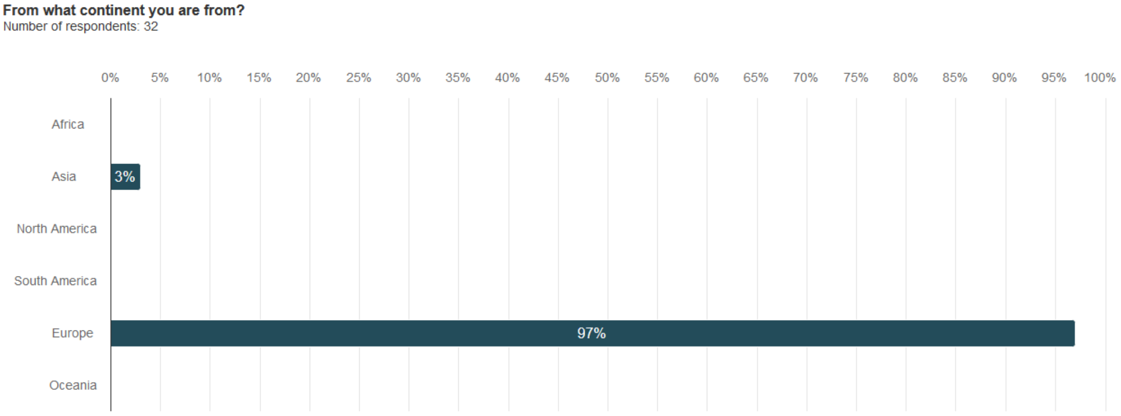
Results of question: “How AI will help cloud professionals on everyday work?

5.4 Other findings

72% of the respondents were 31-45 years old. 28% of the respondents were 15-30 years old and none were 46-64 years old. Nearly all the respondents were from Europe and one answer came from Asia.

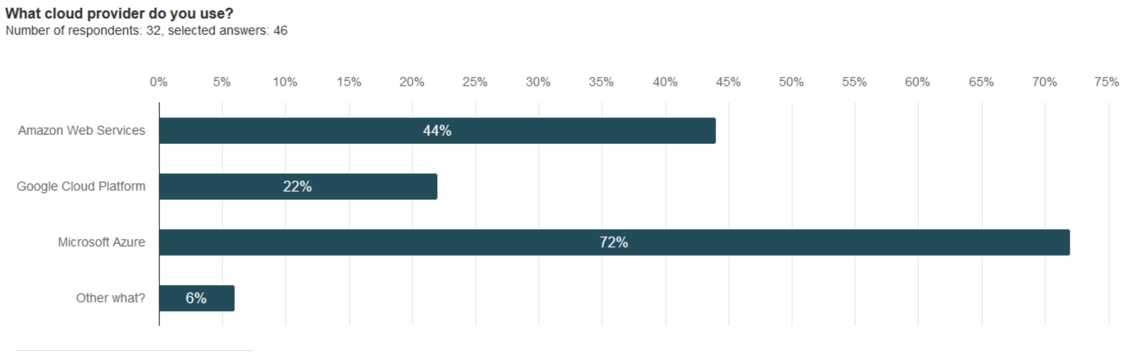


What is your age?



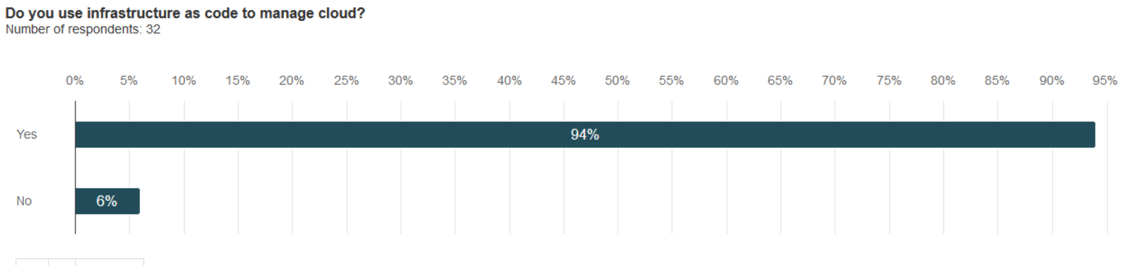
From what continent you are from?

Respondents utilized multiple cloud providers, but Microsoft Azure was the most popular with the 72%. Amazon Web Services came second by a 44% and Google Cloud Platform was 22%. One respondent utilized OVH cloud, and one answered prefer not to say.



What cloud provider do you use?

Most of the respondents uses infrastructure as code to manage cloud. 94% of respondents answered they use infrastructure as code and 6% answered no.

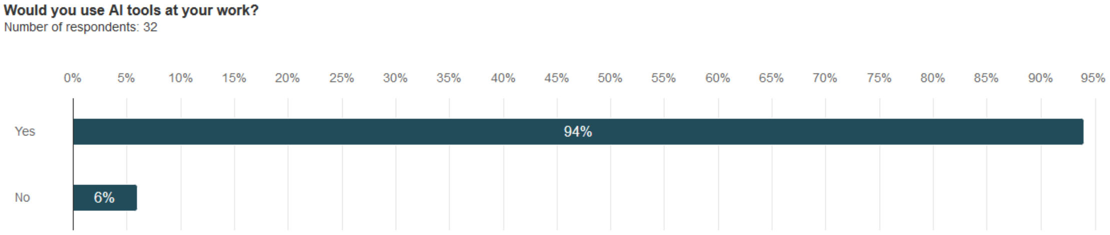


Do you use infrastructure as code to manage cloud?

When comparing results by grouping respondents by the cloud provider, it seems that the results are very similar if respondents find artificial intelligence tools helpful. In ChatGPT median is three and group average values are: For AWS it is 3.1, for Azure it is 3.3 and for GCP it is 3.4.

For Github Copilot results are more divided. AWS and GCP user group median is 4.0, but Azure got 3. Average values for groups are: For AWS it is 4, for Azure it is 3.3 and for GCP it is 3.7. Here it seems that Azure got worst score, but the fact is Azure got most respondents so it might affect the result. It seems helpfulness of artificial intelligence is same for major cloud providers.

Most of the respondents would use AI tools at work, 94% answered yes and only 6% answered no.

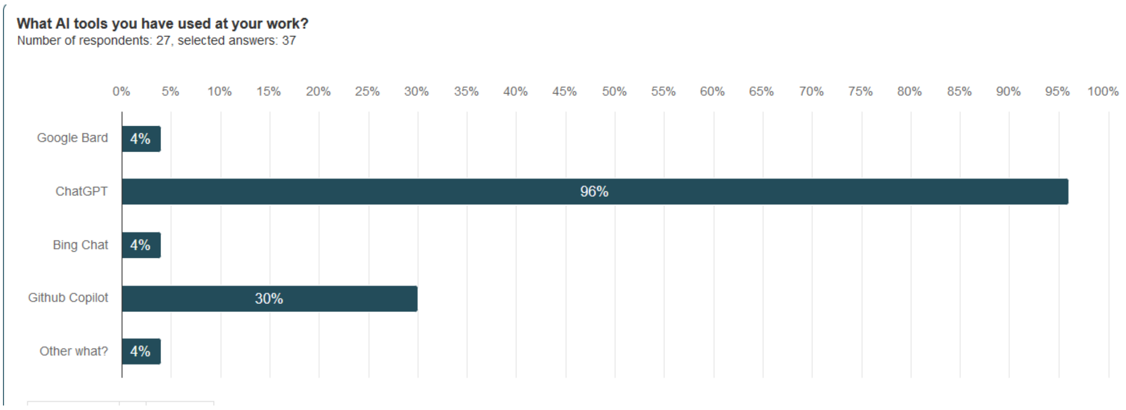


Would you use AI tools at your work?

84% of the respondents answered they have used AI tools at work and 16% haven’t.

What was the most used AI tools in the survey? According to the survey ChatGPT was the most used AI tool by a large margin because 96% of respondents had used ChatGPT. Second was Github Copilot by 30%. Google bard and Bing Chat were at 4% and other what was 4% with Azure AI studio.

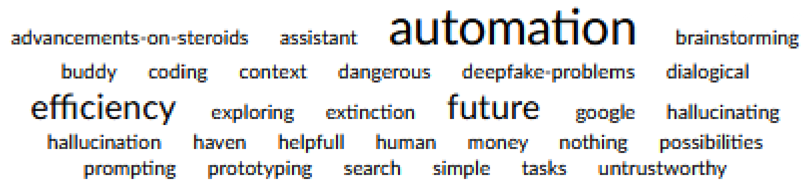
Follow -up questions for these were how helpful respondents found a tool they chose. Google Bard got one response and respondents reviewed it as a 1 in scale of 0-5 on how helpful it is. It is too low number of respondents to say anything about Google Bard. ChatGPT got 26 responses on how helpful it is, and average value is 3,2 and median is 3. It seems ChatGPT is helpful but there are also a lot to improve. Bing Chat also got one respondent and respondent valued it as 3. Again, too low number of responses to say anything about it but it is based on OpenAI model and because of that it makes sense the score is so close to ChatGPT. Github copilot got 8 responses and respondents found it somewhat helpful and it got an average score of 3,4 and median is 3,5. Its score is slightly more than ChatGPT despite having a smaller number of respondents. It seems Copilot is helpful but there are also improvements that must be made to make it more helpful. Azure AI studio received one response and it was valued as 4, but response amount is too low.



What AI tools have you used at your work?

Last question in the survey was "*what are two keywords that come to your mind about AI tools?*" The purpose of the last question was to study what are the respondent's mindset on artificial intelligence. There were 17 respondents on the last question, and it seems the mentality is mostly positive but there are also some negative sounding words, but those words are something I think most of us think about in AI. Most words whom respondents generated were automation, efficiency, and future. Most used "negative" sounding word was hallucination. Hallucination in context of AI means that AI generates false facts or information, and it creates challenge for its users to know what real fact is. Hallucination is a common problem with generative AI tools (Keary, 2023). Keyword untrustworthy should be counted here because meaning is the same. The word cloud is interesting, and it shows that AI can be a good tool but again, it should not be blindly trusted. The word cloud is along the same lines with the study on Github copilot with programmers. Tool can be dangerous if users blindly trust its results, but if user has knowledge and understanding around topic asked from AI, then it can be a good tool.

The word cloud show that some respondents are not in hype and understand the great responsibility that comes when using AI tools. Clearly it shows that it is the future that these tools are used more and more. Respondents also think AI is going to be a good business and it also creates a lot of possibilities for corporations themselves to automate routine tasks and to help employees to do their everyday tasks.



Word cloud of the question "what are two keywords that come to your mind about AI tools?*"*

Hypothesis about IaC users use more artificial intelligence at work is not true regarding the survey. There is no correlation between IaC users and usage of artificial intelligence tools. Calculation was done by using chi square test found onf Scipy library in Python.

Results of the test are following:

```
Chi2 Stats: 0.0
P-value: 1.0
Degrees of Freedom: 1
Expected Values:
 [[ 0.3125  1.6875]
 [ 4.6875 25.3125]]
```

6 CONCLUSION

Cloud management and administering can be a difficult task and it needs new kind of thinking and processes to manage it effectively. This is where infrastructure as code steps into the picture because with it cloud is managed through code, and it allows greater quality and better processes through automation. Infrastructure as code brings its own difficulties because it can be hard to learn, and it can take more time to write infrastructure as code. This is where artificial intelligence could help cloud professionals. It could help with learning, writing infrastructure as code and to help with problem solving.

With the rise of generative artificial intelligence tools, it was very important to study if cloud professionals use these kinds of tools and what they think about them. There is a lot of hype in the media about artificial intelligence, but it is not so black and white if artificial intelligence tools always help. Study results show these tools are somewhat helpful, but there are a lot of concerns if the results of the tools are correct. Whitepaper that was discussed earlier in thesis shows it is crucial to understand what one is doing if using artificial intelligence tools because these tools can give a lot of false information and quality can be bad. But there are reasons why one should use artificial intelligence and it is that it works like an assistant, and it can give new viewpoints to problems, help to learn and in programming it can give good starting point where to start. Results show there are helpful use cases for artificial intelligence.

Regarding data gathering, there should have been more responses to the survey because 32 is too low number. Most of the answer might have been from the company employees because infrastructure as a code might not be popular in companies but answers say that nearly all used infrastructure as code. It does not accurately answer to hypothesis on whether infrastructure as code users use more artificial intelligence tools. Hypothesis originates from the idea that with infrastructure as code, it is possible to ask the whole code from the artificial intelligence. However, it is not possible to ask artificial intelligence to create infrastructure from the user interface.

Important aspect that should be done in today's world is to teach how these artificial intelligence tools work, ethics of the artificial intelligence and to speak more about the risks of using artificial intelligence tools. It is important to increase knowledge on articial intelligence and it also decreases hype around artificial intelligence. Hype is dangerous because often it will hide the risks tools provide. Knowledge also make it easier for people to notice hallucinations artificial intelligence might produce.

On study “Github Copilot AI Pair programmer: asset or liability” they discovered that Github copilot is a good tool for seniors but not for juniors. The conclusion from study is that it is also important to increase knowledge of the employees to their own working field. Like in the thesis, it is important to increase knowledge about the cloud and infrastructure as code in the addition to the artificial intelligence. Knowledge is the best way to make artificial intelligence a safer and better tool and it will not end up being burden.

Another study from GitClear showed that “code churn” is on the rise. It means that code that is authorized to be a part of code base is thrown away after two weeks. This is expected to be doubled in 2024. This leads to problem where more mistakes could be introduced to production. According to study artificial intelligence is good at adding code, but it might cause technical debt. In another study from Mckinsey found out that massive surge in software production is possible by using artificial intelligence, but it depends on how experienced programmer is and how complex the problem is. Programmers must understand their attributes that creates good quality code and to use correct prompt engineering methods to get best possible outcome (Soper, 2024).

Users should acknowledge that using artificial intelligence tools brings a lot of responsibility for the user. One must understand what they are doing and the risks that come with it. Organizations cannot control the usage of these tools because it is like Googling and thus the responsibility is more on the shoulders of the users.

These findings came from the multiple sources when doing the thesis. One was the study results that show that there are a lot to be improved with artificial intelligence tools. Results mixed with the reality, there are a lot of hype around artificial intelligence currently so these both worlds collide. Third one was that during the thesis I was doing my own infrastructure as code test assignment using Github Copilot and findings were that most of the suggestions from the Github Copilot were incomplete or wrong. Some of the code blocks could open very bad security vulnerabilities for example, incomplete networking settings. I still found Github Copilot helpful because it provided good starting point for infrastructure coding. It is still essential to understand the code and what configurations must be added to make code complete and then to fix these vulnerabilities generative artificial intelligence generated code blocks have. My own finding underlines how important human knowledge is and the importance of the studying about my own field of work and artificial intelligence. Another helpful assistance Github copilot provided was the naming conventions because it will remember the variable names and code block names, so Github copilot helps to keep code easy to read. For example, if using naming abbrevations from the Azure cloud adoption framework, Github copilot will learn to use those abbrevations after a few code blocks.

Generative artificial intelligence has gained a lot popularity within a year because tools have developed a lot, and they became more user friendly in a way everyone can use them. The world is just starting to use artificial intelligence, but these tools are going to develop are lot in the coming years. Study shows it is not artificial intelligence that will replace worker, but it is person who knows how to use artificial intelligence. However, everyone should keep in mind that artificial intelligence might generate wrong answers and one must understand its answer. Artificial intelligence when used as an assistant, can be a powerful tool to have in a toolbox but use it wisely.

7 REFERENCES

Acuvate. (28. 6 2022). *LinkedIn*. Noudettu osoitteesta LinedIn: https://www.linkedin.com/pulse/azure-cloud-adoption-framework-caf-stages-benefits-acuvate/

Alto, V. (2023). *Modern Generative AI with ChatGPT and OpenAI Models.* Packt.

Arghavan Moradi Dakhel∗, V. M. (2023). *GitHub Copilot AI pair programmer: Asset or Liability?* Noudettu osoitteesta https://arxiv.org/pdf/2206.15331.pdf

(2021). *Asleep at the Keyboard? Assessing the Security of Github Copilot's Code Contributions.*

Bansal, M. (05. 08 2023). *Knowledgehut*. Noudettu osoitteesta Knowledgehut: https://www.knowledgehut.com/blog/cloud-computing/what-is-private-cloud#reasons-for-popularity-of-private-cloud-environments%C2%A0

BasuMallick, C. (26. 08 2022). *Spiceworks*. Noudettu osoitteesta Spiceworks: https://www.spiceworks.com/tech/cloud/articles/what-is-infrastructure-as-code/

Brikman, Y. (2022). *Terraform: Up and Running, 3rd Edition.* O'Reilly Media.

Chyhyryk, O. (01 2024). *Marketsplash*. Noudettu osoitteesta Marketsplash: https://marketsplash.com/tutorials/terraform/terraform-destroy/

Courtemanche, M. (ei pvm). *Techtarget*. Noudettu osoitteesta Techterget: https://www.techtarget.com/searchitoperations/definition/DevOps

Eduard Keilholz, E. S. (2022). *Azure Infrastructure as Code.* Manning Publications.

Ficom. (16. 01 2023). *Ficom*. Noudettu osoitteesta Ficom: https://ficom.fi/ajankohtaista/uutiset/ict-ala-on-suuri-kokoisekseen/

Fruhlinger, J. (7. 8 2023). *InfoWorld*. Noudettu osoitteesta InfoWorld: https://www.infoworld.com/article/3689973/what-is-generative-ai-artificial-intelligence-that-creates.html

Galante, L. (27. 7 2023). *Humanitec*. Noudettu osoitteesta Humanitec: https://humanitec.com/blog/infrastructure-as-code-the-good-the-bad-and-the-future

Gershgorn, D. (29. 6 2021). *The Verge*. Noudettu osoitteesta The Verge: https://www.theverge.com/2021/6/29/22555777/github-openai-ai-tool-autocomplete-code

Github. (ei pvm). *Docs.github*. Noudettu osoitteesta Docs.github: https://docs.github.com/en/enterprise-cloud@latest/copilot/overview-of-github-copilot/about-github-copilot-for-business

Hashicorp. (ei pvm). Noudettu osoitteesta https://developer.hashicorp.com/terraform/language/state

Hashicorp. (ei pvm). *Developer Hashicorp*. Noudettu osoitteesta Developer Hashicorp: https://developer.hashicorp.com/terraform/cli/commands/destroy

Hashicorp. (ei pvm). *Developer Hashicorp*. Noudettu osoitteesta Developer Hashicorp: https://developer.hashicorp.com/terraform/cli/commands/apply

Hashicorp. (ei pvm). *Developer Hashicorp*. Noudettu osoitteesta Developer Hashicorp: https://developer.hashicorp.com/terraform/cli/commands/plan

Hashicorp. (ei pvm). *Developer Hashicorp*. Noudettu osoitteesta Developer Hashicorp: https://developer.hashicorp.com/terraform/cli/commands/validate

Hashicorp. (ei pvm). *Developer Hashicorp*. Noudettu osoitteesta Developer Hashicorp: https://developer.hashicorp.com/terraform/cli/commands/init

Hashicorp. (ei pvm). *Hashicorp*. Noudettu osoitteesta Hashicorp: https://developer.hashicorp.com/terraform/language/state/sensitive-data

Hogget, W. (08. 06 2023). *Pluralsight*. Noudettu osoitteesta Pluralsight: https://www.pluralsight.com/resources/blog/cloud/what-is-terraform-state

Jebaraj, K. (08. 08 2023). *Knowledge hut*. Noudettu osoitteesta Knowledge hut: https://www.knowledgehut.com/blog/cloud-computing/public-cloud

Keary, T. (22. 09 2023). *Techopedia*. Noudettu osoitteesta Techopedia: https://www.techopedia.com/definition/ai-hallucination

Krief, M. (2022). *Learning DevOps - Second Edition.* Packt.

Lean, S. (1. 11 2022). *Techieclass*. Noudettu osoitteesta Techieclass: https://www.techielass.com/infrastructure-as-code-the-benefits-and-the-tools/

Lisdorf, A. (2021). *Cloud Computing Basics: A Non-Technical Introduction.* Apress.

McKendrick, R. (2023). *Infrastructure as Code for Beginners.* Packt.

Mckinsey & Company. (22. 08 2023). *Mckinsey & Company*. Noudettu osoitteesta Mckinsey & Company: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-prompt-engineering

Microsoft. (12. 01 2022). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/digital-estate/rationalize

Microsoft. (27. 3 2023). *Learn.Microsoft*. Noudettu osoitteesta Learn.Microsoft: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/overview

Microsoft. (28. 2 2023). *Microsoft Learn*. Noudettu osoitteesta Microsoft Learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/strategy/

Microsoft. (28. 02 2023). *Microsoft Learn*. Noudettu osoitteesta Microsoft Learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/strategy/financial-considerations/

Microsoft. (31. 08 2023). *Microsoft learn app service*. Noudettu osoitteesta Microsoft learn app service: https://learn.microsoft.com/en-us/azure/app-service/getting-started?pivots=stack-net

Microsoft. (01. 03 2023). *Microsoft learn virtual machine*. Noudettu osoitteesta Microsoft learn virtual machine: https://learn.microsoft.com/en-us/azure/virtual-machines/overview

Microsoft. (2024). *Microsoft.com*. Noudettu osoitteesta Microsoft.com: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-elastic-computing

Microsoft adopt. (3. 8 2023). *Microsoft learn*. Noudettu osoitteesta https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/adopt/

Microsoft business alignment. (12. 1 2022). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/manage/considerations/business-alignment

Microsoft CD. (28. 11 2022). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/devops/deliver/what-is-continuous-delivery

Microsoft development. (19. 10 2022). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/considerations/development-strategy-development-lifecycle

Microsoft devops. (28. 2 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/design-area/platform-automation-devops

Microsoft devsecops. (19. 10 2022). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/considerations/security-considerations-overview

Microsoft govern. (24. 4 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/govern/

Microsoft govern methodology. (29. 3 2023). *Microsoft learn*. Noudettu osoitteesta https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/govern/methodology

Microsoft landing zone. (31. 5 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/

Microsoft learn CI. (28. 11 2022). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/devops/develop/what-is-continuous-integration

Microsoft organization alignment. (28. 04 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/plan/initial-org-alignment

Microsoft organize. (28. 02 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn.

Microsoft raci. (28. 2 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/organize/raci-alignment

Microsoft rationalize. (1. 12 2022). *Microsoft Learn*. Noudettu osoitteesta Microsoft Learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/digital-estate/rationalize

Microsoft security. (18. 8 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/secure/

Microsoft skill readiness plan. (28. 2 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/plan/adapt-roles-skills-processes

Microsoft strategy. (28. 2 2023). *Microsoft Learn*. Noudettu osoitteesta Microsoft Learn: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/strategy/technical-considerations/

Microsoft team structures. (28. 2 2023). *Microsoft learn*. Noudettu osoitteesta Microsoft learn.

Morris, K. (2024). *Infrastructure as code, 2nd Edition.* O'Reilly Media, Inc.

National Institute of Standards and Technology. (09 2011). *Nist*. Noudettu osoitteesta Nist: https://csrc.nist.gov/pubs/sp/800/145/final

National Institute of Stantards and Technology. (28. 09 2011). Noudettu osoitteesta Computer Security Resource Center: https://csrc.nist.gov/publications/detail/sp/800-145/final

Outsystems. (ei pvm). Noudettu osoitteesta https://www.outsystems.com/glossary/cloud-scalability-vs-elasticity/

Radchenko, V. (22. 5 2022). *Medium*. Noudettu osoitteesta Medium: https://vlad-rad.medium.com/github-copilot-security-conserns-d4209f0d5c28

Ronin, V. (12. 12 2022). *aembit*. Noudettu osoitteesta aembit: https://aembit.io/blog/infrastructure-as-code-is-more-complicated-than-you-think

Roper, J. (28. 02 2024). *Spacelift*. Noudettu osoitteesta Spacelift: https://spacelift.io/blog/terraform-state

Ross, S. (27. 4 2023). *Investopedia*. Noudettu osoitteesta Investopedia: https://www.investopedia.com/ask/answers/112814/whats-difference-between-capital-expenditures-capex-and-operational-expenditures-opex.asp

Sandu, S. (27. 5 2022). *medium*. Noudettu osoitteesta https://medium.com/@sudheer.sandu/multi-tenant-application-68c11cc68929

Sasa Kovacevic, D. D. (2023). *Azure Cloud Adoption Framework Handbook.* Packt.

Schults, C. (18. 08 2023). *Stackify*. Noudettu osoitteesta Stackify: https://stackify.com/what-is-infrastructure-as-code-how-it-works-best-practices-tutorials/

Solita Oy. (09. 02 2022). *Solita*. Noudettu osoitteesta Solita: https://www.solita.fi/news/suomalaiset-suuryritykset-hyodyntavat-aktiivisesti-pilvipalveluita-kasvumahdollisuuksia-yha-paljon-kayttamatta/

Soper, T. (24. 01 2024). *Geekwire*. Noudettu osoitteesta https://www.geekwire.com/2024/new-study-on-coding-behavior-raises-questions-about-impact-of-ai-on-software-development/

Spacey, J. (14. 8 2023). Noudettu osoitteesta https://simplicable.com/IT/thin-client-vs-thick-client

Statista. (28. 4 2023). Noudettu osoitteesta Statista: https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/

Thomas Erl, E. B. (2023). *Cloud Computing: Concepts, Technology, Security, and Architecture, 2nd Edition.* Pearson.

Toroman, M. (2018). *Hands-On Cloud Administration in Azure.* Packt.

Wikipedia. (2023). *Wikipedia*. Noudettu osoitteesta Wikipedia: https://en.wikipedia.org/wiki/GitHub_Copilot

Wojciech Zaremba, G. B. (10. 8 2021). *OpenAI*. Noudettu osoitteesta OpenAI: https://openai.com/blog/openai-codex