AP HOGESCHOOL ANTWERPEN

EY

Bachelor in

Elektronica – ICT

Final document

Part of internship

Supported by

AP Hogeschool

Executed and guided by

EY

Zakaria El Morabit

Cybersecurity & Cloud

Begeleider: Similon Andie
Mentor: Ignace De Cock

Academiejaar 2023-2024
2de semester

Table

Version control

Nr.	Date	Spread	Status	Changes
0.1	22/05/2024	Zakaria El Morabit	Eerste draft	First draft research results
0.2	28/05/2024	Zakaria El Morabit	Verbetering	Paragraaf herschreven
0.3	31/05/2024	Zakaria El Morabit Ignace De Cock	Added content	Added what I learned from Mike Martin
1.0	31/05/2024	Zakaria El Morabit	Final draft	Terms and Abbrevations, preface, afterword, and combining all documentation.
2.0	19/08/2024	Zakaria El Morabit	Final draft summer	Changed onderzoeksplan, technical documentation, conclusions and template errors

Terms and abreviations

Term	Description
NIST (National Institute of Standards and Technology)	A U.S. federal agency that develops and promotes measurement standards and guidelines, including those for cloud computing and digital forensics.
Cloud Forensics	The application of digital forensic principles to cloud computing environments. This involves investigating security incidents, and collecting, analyzing, and preserving digital evidence stored in cloud services.
CPS (Cloud Service Provider)	A company that offers cloud computing services such as storage, databases, networking, and software over the internet. Examples include Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP).
Azure	A cloud computing platform and service created by Microsoft that provides a wide range of cloud services, including computing, analytics, storage, and networking.
SIEM (Security Information and Event Management)	A system that collects, analyzes, and stores security-related data from various sources to provide real-time analysis of security alerts generated by applications and network hardware.
PaaS (Platform as a Service)	A cloud computing service that provides a platform allowing customers to develop, run, and manage applications without dealing with the underlying infrastructure.
VM (Virtual Machine)	A software emulation of a physical computer that runs an operating system and applications just like a physical computer.
GUID (Globally Unique Identifier)	A unique reference number used as an identifier in computer software.
Entra Logging	Part of Microsoft Entra, a suite that provides identity and access management tools, including detailed logging for monitoring and security purposes.
AKS (Azure Kubernetes Service)	A managed container orchestration service provided by Azure that simplifies the deployment, management, and operations of Kubernetes clusters.
Application Insights	An application performance management service for web developers that provides insights into the performance and usage of their applications.
Defender for Cloud	An integrated security service by Microsoft that provides advanced threat protection for hybrid cloud workloads.
Soft-Delete	A feature that allows data to be marked as deleted but not permanently removed, providing a recovery option within a specified time frame.
Container Insights	A feature in Azure Monitor that provides performance monitoring and diagnostics for container workloads deployed to managed Kubernetes clusters hosted on Azure Kubernetes Service (AKS).
Log Analytics	A service in Azure Monitor that helps collect and analyze log data from various sources to provide insights into the operation and performance of cloud and on-premises resources.
Blob Storage	Azure's object storage solution for the cloud, optimized for storing massive amounts of unstructured data, such as text or binary data.

Forword

In this document, we explore the domain of cloud forensics, focusing on the unique challenges and methodologies relevant to investigating incidents in cloud environments. Given that 57% of organizations store their data in the public cloud, understanding these challenges is crucial. Our research is supported and guided by Ignace De Cock, an expert in digital forensics, whose expertise has been invaluable.

This thesis presents a comparative analysis of existing forensic tools and their effectiveness in cloud forensics. It emphasizes the need for tools that can handle the dynamic nature of cloud computing and provides insights into the adaptation of traditional forensic processes to the cloud model.

We hope this document contributes to the field of cloud forensics and assists investigators in effectively addressing the complexities of cloud environments.

Blueprint

Opdrachtgever

Ignace De Cock, onze opdrachtgever en stagementor, is een expert in het domein van digitale forensics. Zijn deskundigheid is een essentiële steun voor ons onderzoek. Hij biedt richting en inzicht, en corrigeert ons wanneer we op foutieve veronderstellingen stuiten.

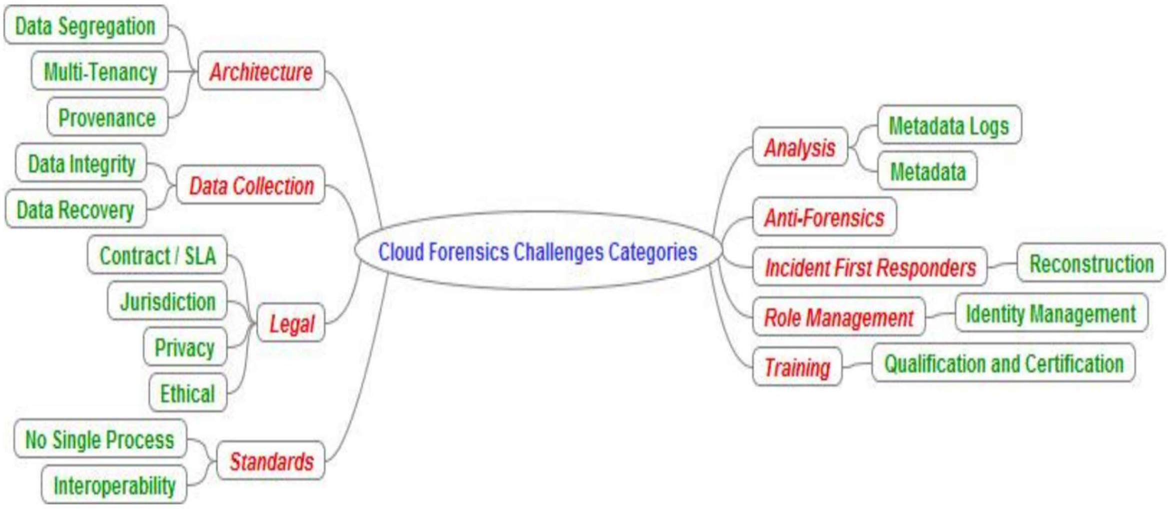
Samenvatting

In dit document zullen we het domein van cloud forensics verkennen, een cruciaal onderwerp gezien het feit dat 57% van de organisaties hun gegevens opslaan in de publieke cloud . We beginnen met het schetsen van de unieke uitdagingen die cloudomgevingen vormen voor forensische onderzoekers. Vervolgens verdiepen we ons in de verschillende methodologieën en hulpmiddelen die gebruikt kunnen worden voor een dergelijk onderzoek. We zullen ook de aanpassing van traditionele forensische processen aan het cloudmodel bespreken.

De thesis presenteert ook een vergelijkende analyse van bestaande forensische hulpmiddelen en hun effectiviteit voor het uitvoeren van cloud forensiek, waarbij de noodzaak wordt benadrukt voor hulpmiddelen die de dynamische aard van cloud computing aankunnen.

Probleemstelling

Om de huidige stand van wetenschap het best te beschrijven grijpen we naar het onderzoek van NIST in 2020, *NIST Cloud Computing Forensic Science Challenges.* Dit document brengt de verschillende uitdagingen die we vinden in cloud forensics in kaart, zo hebben ze er 62 samen gezet. Hieronder een afbeelding die de verschillende onderwerpen samenvat:



Voor dit onderzoek focussen we op de uitdagingen binnen architectuur, data verzameling, het analyseren en Incident first responders. Alles hierbuiten valt buiten de scope.

Architectuur

In cloud forensics is de compartimentalisatie en isolatie van huurdergegevens tijdens het toewijzen van bronnen een belangrijk aandachtspunt. Daarnaast speelt de verspreiding van systemen, locaties en eindpunten die data kunnen opslaan een grote rol. Ook is het van cruciaal belang om een nauwkeurige en veilige herkomst van gegevens te waarborgen, wat essentieel is voor het handhaven en bewaren van de 'chain of custody'.

Data verzameling

In cloud forensics is het lokaliseren van forensische artefacten in uitgebreide, verspreide en dynamische systemen een complexe taak. Het verzamelen van vluchtige data en het extraheren van gegevens uit virtuele machines zijn eveneens uitdagende aspecten. In een multi-tenant omgeving, waar data gedeeld wordt tussen meerdere computers op diverse locaties en toegankelijk is voor verschillende partijen, is het behouden van data-integriteit een cruciaal aandachtspunt. Forensisch experts staan voor de uitdaging om niet alle artefacten in de cloud volledig te kunnen imageren. Daarnaast is het toegankelijk maken van de data van één klant, zonder de vertrouwelijkheid van andere klanten te schenden, een delicate kwestie. Verder omvat cloud forensics ook de complexiteit van het herstellen van verwijderde data in een gedeelde en gedistribueerde virtuele omgeving.

Analyse

In cloud forensics omvat het werk de correlatie van forensische artefacten, zowel tussen verschillende cloudproviders als binnen één provider. Dit is essentieel voor het reconstrueren van gebeurtenissen uit virtuele afbeeldingen of opslag. De integriteit van metadata speelt hierbij een cruciale rol, aangezien deze essentiële informatie verschaft over de artefacten. Een ander belangrijk aspect is de tijdlijnanalyse van loggegevens, inclusief de synchronisatie van tijdstempels, wat helpt bij het nauwkeurig vaststellen van de chronologie van gebeurtenissen.

incident eerstelijns responders

Bij cloud forensics is de bekwaamheid en betrouwbaarheid van cloudproviders als eerste respondenten voor het verzamelen van data een cruciaal aspect. Het correct uitvoeren van een initiële triage blijkt vaak uitdagend, gezien de complexe aard van cloud omgevingen. Bovendien vormt het verwerken van de grote volumes aan verzamelde forensische artefacten een aanzienlijke uitdaging, door de uitgebreidheid en diversiteit van de gegevens.

State of the Art

In het kader van cloud forensics biedt Azure specifieke mogelijkheden die van cruciaal belang zijn voor forensisch onderzoek. Enkele van deze mogelijkheden zijn:

Azure Security Center:

Dit biedt geavanceerde beveiligingsanalyse en threat detection, waarmee verdachte activiteiten en potentiële inbreuken snel geïdentificeerd kunnen worden. Het helpt bij het verzamelen en analyseren van beveiligingslogs, wat essentieel is voor forensisch onderzoek .

Azure Monitor en Log Analytics:

Deze diensten bieden uitgebreide monitoring van applicaties, infrastructuur en netwerk, en stellen onderzoekers in staat om gedetailleerde logs en prestatiegegevens te verzamelen. Dit is nuttig voor het reconstrueren van gebeurtenissen en het identificeren van abnormaal gedrag .

Azure Sentinel:

Als een schaalbare, cloud-native, security information event management (SIEM) en security orchestration automated response (SOAR) oplossing, helpt Azure Sentinel bij het verzamelen, detecteren, onderzoeken en reageren op bedreigingen in de omgeving .

Azure Blob Storage Auditing:

Dit staat toe om toegang en activiteiten te loggen met betrekking tot opgeslagen data, wat essentieel is voor het traceren van toegang tot en wijzigingen in data .

Virtual Machine Forensics:

Azure ondersteunt het onderzoeken van virtuele machines, waaronder het maken van snapshots en het analyseren van virtuele harde schijven voor forensische doeleinden .

Het is belangrijk om rekening te houden dat het uitzoeken van alle mogelijkheden deel is van ons onderzoek en nog steeds loopt. Dit hoofdstuk beschrijft waar de huidige wetenschap staat maar is dus beperkt en omvat niet alle capaciteiten.

Gerelateerd werk

Het meest moderne en up-to-date kennis om aan cloud forensics te doen is de FOR509 cursus van Sans instituut [7]. Zij zijn een autoriteit op het vlak van certificatie en worden wereldwijd vertrouwd. De focus ligt op het begrijpen van forensische data in de cloud, toepassen van beste DFIR-praktijken in cloud logs, benutten van Azure, AWS en Google Cloud voor bewijsverzameling, inzicht in logs van Microsoft 365 en Google Workspace, en het verplaatsen van forensische processen naar de cloud voor snellere data-analyse. Cado security [4] heeft een samenvatting van verschillende tools en best practices om aan cloud forensics te doen.

NIST heeft een goed omvattende studie uitgebracht van alle uitdagingen in dit domein. Microsoft heeft gedocumenteerd hoe je je chain of custody bij houdt in Azure, deze bevat ook voorbeeld scenario's [9]. Forensics Labs heeft een artikel uitgebracht met wat praktische tips hoe je aan forensics kan doen in de cloud .

Onderzoeksvraag

Welke methoden, artefacten en tools kunnen we het beste gebruiken om cloud forensisch onderzoek uit te voeren? Het belangrijkste doel van dit onderzoek is om de beschikbare informatie te centraliseren en te ontdekken welke mogelijkheden er zijn. De primaire keuze zal vallen op tools die door Azure worden aangeboden. Een methode, artefact of tool moet aantonen dat de informatie betrouwbaar en correct is. Het moet ook zijn nut in forensisch onderzoek bewijzen.

Hypothese

We verwachten te vinden dat de meeste hulpmiddelen en methoden om cloud forensics uit te voeren, geleverd zullen worden door de CPS, aangezien het moeilijk is om monitoring en scanning uit te voeren op een platform dat constant verandert en evolueert, met gebruik van externe software en hulpmiddelen. Echter, we vermoeden dat veel artefacten die gebruikt worden in traditionele digital forensics relevant zullen blijven, aangezien eindgebruikers het grootste veiligheidsrisico van organisaties blijven, zelfs wanneer ze cloudomgevingen gebruiken.

Doelstelling

De thesis zal de volgende resultaten opleveren: een samenhangende uitleg van de huidige problemen waar onderzoekers mee te maken krijgen bij cloud forensics, een analyse die digital forensics vergelijkt met cloud forensics, hulpmiddelen en artefacten uit de digital forensics die relevant blijven bij het onderzoeken van cloud computing en een analyse over op welke unieke manieren cloud computing een onderzoeker kan ondersteunen.

Scope

Voor deze scriptie zullen we ons richten op Azure. Deze beslissing is geïnspireerd door zowel het grote marktaandeel dat Azure heeft in deze ruimte als de afstemming van de expertise die door mijn collega's wordt geboden. We beperken ons tot het vinden en analyseren van artefact en bewijzen. Verder nog, focussen we ons op de meest gebruikte toepassingen op azure. Deze vinden we op de azure portal in de favorieten lijst. We onderzoeken “All resources”, “resource groups”, “virtual machines”, “Microsoft Entra ID”, “Monitor”, “virtual networks” en “microsoft Defender for Cloud”.

Niet in Scope

Omdat we in een geïsoleerde labo omgeving werken, laten we legale en ethische overwegingen ter zijde. Dus het forensisch aspect dat wordt beïnvloed door lokale wetten laten we ter zijde. OS specifiek onderzoek wordt ook terzijde gelaten aangezien dit overlapt met digital forensics. We maken ook geen vergelijkingen met andere CSP's en houden ons enkel bezig met Azure.

Planning

Voor onze stage werken we agile. We werken met 2-wekelijkse sprints, wat over de duur van onze stageperiode overeenkomt met 7 sprints waarvan elke sprint 8 werkdagen telt. Ik stel 1 storypoint gelijk aan een halve dag. Dit houd ik bij in Trello. Met deze gegevens kunnen we nu de sprint planning uittekenen.

Sprint 0:

- User story: Onderzoeken wat digital forensics is.
- Doel: Vertrouwd raken met het domein en de wetenschap ervan. Het bestuderen van publiek beschikbare cursussen en papers om de verschillende aspecten van forensics te begrijpen en een eerste idee te krijgen van de praktische toepassing ervan. Notities worden bijgehouden voor toekomstige referentie.
- Story points: 8.
- User story: Onderzoeken wat we al weten van cloud forensics.
- Doel: Aangezien informatie in dit sub domein verspreid en niet gemakkelijk te vinden is, is het doel zoveel mogelijk relevante informatie te verzamelen en samen te vatten. Dit zal het verdere onderzoek en de uitwerking vergemakkelijken.
- Story points: 8.

Sprint 1:

- User story: Verdiepen in cloud forensics.
- Doel: Specifieke kennis en technieken binnen cloud forensics begrijpen, vooral gericht op Azure. Dit omvat het leren over specifieke tools en methoden die gebruikt worden in Azure-omgevingen.
- Story points: 8.

- User story: Schrijven van onderzoeksplan.
- Doel: Compleet invullen en indienen van onderzoeksplan.
- Story points: 8.

Sprint 2:

- User story: Onderzoeken van Azure-specifieke forensische tools.
- Doel: Een lijst maken van beschikbare forensische tools die specifiek zijn voor Azure, inclusief hun mogelijkheden en beperkingen. Deze worden praktisch getest.
- Story points: 6.

- User story: Bestuderen van forensische artefacten in Azure.
- Doel: Identificeren en begrijpen van verschillende soorten artefacten die kunnen worden verzameld in Azure-omgevingen voor forensisch onderzoek.
- Story points: 10.

Sprint 3:

- User story: voorbereiden en het houden van de conversatie met Microsoft.
- Doel: Goed nadenken over de vragen die we kunnen stellen aan Microsoft om meer inzicht te vergaren over de mogelijkheden van de CPS bij een breach.
- Story points: 2.
- User story: Voorbereiding van de hack simulatie.
- Doel: Het opzetten van een gesimuleerde hack in een gecontroleerde Azure-omgeving om later te analyseren. Dit betreft ook de timeline van de gesimuleerde breach en hoe we
- Story points: 8.

- User story: Uitvoeren van de hack simulatie.
- Doel: Het daadwerkelijk uitvoeren van de gesimuleerde hack en het verzamelen van relevante data voor analyse.
- Story points: 6.

Sprint 4:

- User story: verder uitvoeren en opnemen van de hack simulatie.
- Doel: Het daadwerkelijk uitvoeren van de gesimuleerde hack en het verzamelen van relevante data voor analyse.
- Story points: 4.

- User story: Analyseren van de verzamelde data van de hack.
- Doel: Het gebruik van cloud forensics tools en technieken om de verzamelde data van de hack te analyseren en conclusies te trekken. We maken hier een tijdlijn gebaseerd op de gevonden gegevens en hoe de verschillende artefacten hier een rol spelen.
- Story points: 4.

- User story: Presentatie voor EY voorbereiden.

- Doel: Het bedrijf in een presentatie laten zien waar ik mee bezig ben geweest en wat het resultaat is van mijn stage.

- Story points: 4.

Sprint 5:

- User story: Samenstellen van een rapport over de bevindingen.

- Doel: Het documenteren van het hele proces, van simulatie tot analyse, inclusief de gebruikte tools, gevonden artefacten en conclusies.

- Story points: 10.

- User story: Voorbereiden van de demonstratie.

- Doel: Het opstellen van een presentatie of demonstratie om de bevindingen en het proces van de hack simulatie en analyse te laten zien.

- Story points: 6.

Sprint 6:

- User story: Afronden en verfijnen van de thesis.

- Doel: Het samenstellen, schrijven en redigeren van de thesis, waarin alle bevindingen, processen, analyses en conclusies van het stageproject worden gedocumenteerd. Dit omvat het integreren van feedback, het zorgen voor nauwkeurigheid en het waarborgen van academische kwaliteit.

- Story points: 12.

- User story: Voorbereiding voor de thesisverdediging.

- Doel: Het voorbereiden van een heldere en overtuigende presentatie van de thesis, waarin de belangrijkste punten en bevindingen worden belicht. Oefenen van de verdediging, anticiperen op mogelijke vragen en zorgen voor een grondige kennis van het onderwerp.

- Story points: 4.

Hoofdlijnen

Er zijn 3 groten mijlpalen:

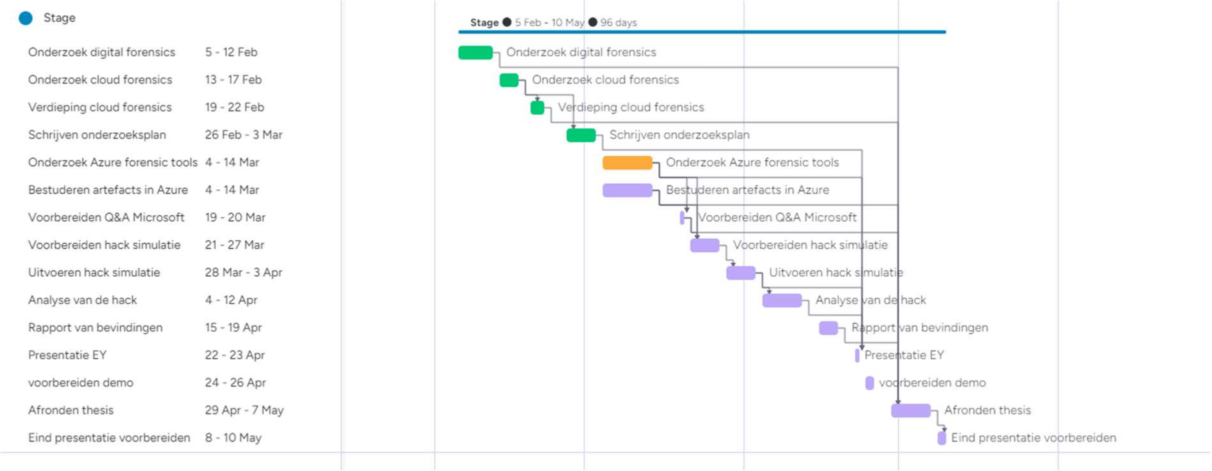
Onderzoek afronden over digital en cloud forensics. Hier is het doel om in kaart te brengen wat er allemaal beschikbaar is van literatuur en belangrijker nog, wat er nog niet is. Dit gebruiken we dan om te begrijpen wat er belangrijk is en welke richting we uit willen met dit onderzoek.

Vervolgens is er het toepassen van al wat ik heb geleerd in een gesimuleerde breach waar we forensics gaan gebruiken om uit te zoeken wat er is gebeurd. Dit is belangrijk om aan te tonen dat mijn onderzoek verder rijkt dan de theorie en het geeft een beeld over hoe dit in de realiteit zou kunnen worden toegepast.

Tenslotte hebben we als laatste mijlpaal het afronden van de bachelorproef. Hier komt alles samen wat we hebben gedaan tijdens het semester en laten we zien wat we hebben geleerd en hoe we kunnen bijdragen.

Detailplanning

De Gantt Chart toont de geschatte tijdlijn en alle afhankelijkheden voor elke taak.



● Stage

Stage ● 5 Feb - 10 May ● 96 days

Onderzoek digital forensics	5 - 12 Feb
Onderzoek cloud forensics	13 - 17 Feb
Verdieping cloud forensics	19 - 22 Feb
Schrijven onderzoeksplan	26 Feb - 3 Mar
Onderzoek Azure forensic tools	4 - 14 Mar
Bestuderen artefacts in Azure	4 - 14 Mar
Voorbereiden Q&A Microsoft	19 - 20 Mar
Voorbereiden hack simulatie	21 - 27 Mar
Uitvoeren hack simulatie	28 Mar - 3 Apr
Analyse van de hack	4 - 12 Apr
Rapport van bevindingen	15 - 19 Apr
Presentatie EY	22 - 23 Apr
voorbereiden demo	24 - 26 Apr
Afronden thesis	29 Apr - 7 May
Eind presentatie voorbereiden	8 - 10 May

Materiaal en methoden

Materiaal

Voor de uitvoering van het project werden de volgende materialen en hulpmiddelen ingezet. De hardware vereiste een computer met een quad-core processor, minimaal 8 GB RAM en 100 GB SSD opslag. Voor optimale prestaties worden 16 GB RAM en 250 GB SSD aanbevolen. Dit is om de garanderen dat het werk niet vertraagd zal worden vanwege de pc die gebruikt word. Daarnaast was een betrouwbare internetverbinding met een snelheid van ten minste 1 Mbps noodzakelijk, anders gaat het uploaden van code veel te traag verlopen. Wat betreft de software, werden de volgende versies gebruikt:

- Besturingssysteem: Windows 10 of Windows 11
- Code-ontwikkelingsomgeving: Visual Studio Code
- Beheer van Azure-resources: Azure CLI; Azure PowerShell
- Infrastructuurbeheer en automatisering: Terraform
- Containerbeheer: Kubernetes CLI (kubectl)

Deze tools en versies zijn gekozen om de betrouwbaarheid en functionaliteit van het project te waarborgen, en zijn gebaseerd op de meest recente aanbevelingen en best handelingen binnen de vakliteratuur.

Mogelijke interfaces

Azure Portal

De Azure Portal is een webgebaseerde interface die een uniforme console biedt voor het beheren van Azure-resources. Het biedt een grafische gebruikersinterface (GUI) waarmee gebruikers interactie kunnen hebben met verschillende Azure-services, waaronder App Services, Virtual Machines en Kubernetes-clusters. Binnen dit project werd de Azure Portal gebruikt voor diverse doeleinden. Het bood een intuïtieve GUI voor het creëren, configureren en beheren van Azure-resources zoals App Services en Virtual Machines. De portal vereenvoudigde taken zoals resource provisioning en configuratie. Daarnaast werd de Azure Portal ingezet voor diagnostiek en monitoring. Hiermee konden gebruikers diagnostische instellingen configureren, metrics bekijken en prestatiegegevens analyseren. Voor loganalyse en visualisatie van gegevens gebruikte het project de Log Analytics-interface van de Azure Portal, die tools biedt voor het uitvoeren van queries en het genereren van rapporten op basis van verzamelde logs en metrics.

Azure CLI

De Azure Command-Line Interface (CLI) is een cross-platform command-line tool die gebruikers in staat stelt om Azure-resources te beheren en diverse taken uit te voeren met gescripte commando's. In het project werd de Azure CLI gebruikt voor resource deployment en configuratie. Specifieke commando's zoals az vm create werden ingezet voor het uitrollen van Virtual Machines, terwijl az aks create werd gebruikt om een AKS-cluster op te zetten. De CLI speelde ook een rol bij configuratiemanagement, waar commando's zoals az monitor diagnostic-settings create werden gebruikt om diagnostische instellingen voor monitoring te configureren. Bovendien werd de Azure CLI gebruikt om interactie met AKS te beheren, waarbij het commando az aks get-credentials werd gebruikt om kubectl te configureren met de juiste referenties voor toegang tot en beheer van het AKS-cluster.

Azure PowerShell

Azure PowerShell is een set van cmdlets voor het beheren van Azure-resources vanuit de PowerShell-opdrachtregel. Het biedt automatiserings- en scriptingmogelijkheden voor Azure-operaties. Binnen het project werd Azure PowerShell gebruikt voor VM-configuratie, zoals het installeren van IIS (Internet Information Services) op Windows Server Virtual Machines. PowerShell-scripts werden ingezet om repetitieve taken te automatiseren en VM-instellingen efficiënt te configureren. Voorbeelden hiervan zijn scripts die automatisch diagnostische instellingen en andere configuraties instellen.

Visual Studio Code

Visual Studio Code (VS Code) is een broncode-editor die ondersteuning biedt voor verschillende programmeertalen en tools via extensies. Het biedt functies voor codebewerking, debugging en versiebeheer. In het project werd VS Code gebruikt voor code deployment, waarbij het hielp bij het schrijven en beheren van configuratiebestanden en scripts, zoals Terraform-configuraties en Kubernetes YAML-manifests. Met behulp van extensies voor Azure en Terraform werden de ontwikkelings- en implementatieworkflows verbeterd. Daarnaast stelde VS Code ontwikkelaars in staat om applicatiecode lokaal te testen voordat deze naar Azure App Service werd gedeployd. Het werd gebruikt om repositories te clonen, virtuele omgevingen in te stellen en applicaties lokaal uit te voeren.

Azure Cloud Shell

Azure Cloud Shell is een online shellomgeving die door Azure wordt aangeboden en vooraf geïnstalleerde command-line tools en scriptingcapaciteiten bevat. In het project werd Azure Cloud Shell gebruikt voor het beheren van Azure-resources rechtstreeks vanuit de Azure Portal. Het bood een toegankelijke omgeving voor het uitvoeren van Azure CLI-commando's en het beheren van AKS-clusters. Cloud Shell werd ook ingezet voor Kubernetes-beheer, waarbij gebruikers toegang kregen tot kubectl voor interactie met het AKS-cluster. Dit stelde hen in staat om applicaties te deployen en clusterbronnen te beheren met behulp van de geïntegreerde tools van Cloud Shell.

Kubernetes CLI (kubectl)

Kubectl is een command-line tool voor interactie met Kubernetes-clusters. Het stelt gebruikers in staat om administratieve taken uit te voeren en containerized applicaties te beheren. In het project werd kubectl gebruikt voor clusterbeheer, waarbij commando's zoals kubectl apply werden ingezet om YAML-manifests met applicatieconfiguraties te deployen. Monitoring en troubleshooting van applicaties die in het cluster draaiden werden uitgevoerd met kubectl-commando's, zoals kubectl get pods en kubectl logs, om de status van pods, services en deployments te controleren en problemen op te lossen.

Impact op de infrastructuur

Azure App Service

De implementatie van Azure App Service vereist een actieve Azure-abonnement om de App Services te kunnen inzetten en beheren. Voor het draaien van de App Service is minimaal een Basic tier App Service Plan nodig, hoewel een Standard of Premium tier noodzakelijk kan zijn, afhankelijk van de schaal van de applicatie en het volume van de logs. Indien logbestanden worden opgeslagen in Azure Storage, is een General-purpose v2 (GPv2) opslagaccount vereist. Om log analytics en monitoring uit te voeren, is Azure Monitor nodig, inclusief Azure Log Analytics, dat essentieel is voor het analyseren van de AppServiceHTTPLogs tabel en andere telemetriegegevens. Dit houdt ook in dat een Azure Log Analytics Workspace moet worden ingesteld om de logs van de App Service op te slaan en te doorzoeken. Voor software- en configuratiebehoeften wordt toegang tot de Azure Portal gebruikt voor het configureren van logging- en monitoringinstellingen. Visual Studio Code 2022 speelt een rol bij het pushen van code naar de App Service, wat helpt bij het simuleren van verkeer. De diagnostische logs worden ingesteld via de Azure Portal om applicatie-, webserver-, gedetailleerde fout- en implementatielogs te verzamelen.

Azure Virtual Machines

Voor de inzet van Azure Virtual Machines is een actieve Azure-abonnement nodig, met de benodigde rechten om VMs te implementeren en beheren. In dit project werd een Azure Enterprise-abonnement gebruikt, dat maandelijks €140 aan credits verstrekt; voor soortgelijke implementaties wordt een maandelijkse krediet van ten minste €80 aanbevolen. De vereiste Virtual Machine is de Standard_D4s_v3 met Windows Server 2019 Datacenter, voorzien van 4 vCPUs en 8 GB geheugen. Voor dit onderzoek volstaat een standaard HDD-managed disk. Een opslagaccount is nodig voor het opslaan van diagnostische gegevens, back-ups en snapshots. Bij het opzetten van de machine wordt gevraagd om een opslagaccount te maken of te kiezen. Voor licenties werd Windows Server 2019 Datacenter Edition gebruikt. Azure Monitor met Log Analytics was inbegrepen in het Enterprise-abonnement en werd ingezet voor het monitoren van de VM-prestaties en het verzamelen van logs. Beheer van de VMs en bijbehorende diensten werd uitgevoerd met Azure CLI of Azure PowerShell. Diagnostische instellingen werden ingeschakeld op de virtuele machines om logs te verzamelen en te verzenden naar Azure Monitor of een opslagaccount, wat een belangrijk aspect van dit onderzoek was.

Azure Kubernetes Service

Azure Kubernetes Service (AKS) is een beheerde Kubernetes-service van Azure die de implementatie, het beheer en de schaalvergroting van containerized applicaties met Kubernetes vereenvoudigt. Voor de opzet van een AKS-cluster is een actieve Azure-abonnement met de juiste rechten nodig. Voor minimale setups is een abonnement vergelijkbaar met het Azure Enterprise-plan geschikt. Een minimaal AKS-cluster met basisnode-configuraties dient te worden geïmplementeerd. Voor kostenbesparing worden VM-groottes zoals Standard_B2s of Standard_D2s_v3 voor de worker nodes gebruikt. Een Azure Storage Account is vereist voor het opslaan van logs en diagnostische gegevens van het AKS-cluster. Tijdens de clusterinstelling is het nodig om een opslagaccount te maken of te selecteren. Kubernetes zelf is open-source en vereist geen aparte licentie, maar de onderliggende Azure-resources moeten wel worden betaald. Azure Monitor met Log Analytics wordt gebruikt voor het monitoren van het AKS-cluster en het verzamelen van logs, en deze service is inbegrepen bij het Azure-abonnement. Voor software- en configuratiebehoeften moeten Kubernetes CLI-tools zoals `kubectl` worden geïnstalleerd om het AKS-cluster te beheren en applicaties te deployen. Diagnostische instellingen moeten in AKS worden ingeschakeld om logs te verzamelen en naar Azure Monitor of een opslagaccount te verzenden. Deze configuratie is essentieel voor het analyseren en monitoren van logs en metrics van het AKS-cluster.

Test plan

Introduction

This document has been crafted with the intention of serving individuals who are keen on validating the findings presented herein, replicating the tests conducted, or engaging in comparable research in cloud forensics. The significance of a well-structured test plan cannot be overstated, as it provides a roadmap for understanding the processes involved in validating our results.

Since our research is on cloud forensics in Azure, a great deal of testing is no longer needed since it will be Microsoft handling most of the software and tools we will be interacting with. Details on them are well documented on their official support pages. Although testing will be limited, it doesn’t mean they’re not important. Wherever Azure doesn’t provide assistance we will use third party tools such as Volatility. This is a tool primarily used in digital forensics which is a different field. This will need to be tested properly to confirm that it can be used in cloud forensics as well.

In this document you will find a short project description that will describe what our research is about. The stakeholders entail those directly involved with this test plan. You will also find a risk analysis describing all possible risks we may encounter and how pressing it may be. Our actions to handle these risks can also be found here. A risk strategy can be found towards the end of this document, outlining the different ways we will be testing our project.

This document is intended for those that wish to verify the validity of my findings, reproduce my tests, or wish to perform similar research into cloud forensics. Having a good test plan is important so that anyone can understand how we have validated our results. If they were to execute the tests the way we have laid them out in this document, they should get the same results and this would confirm the thoroughness of this thesis.

Project description

In this project, we will explore the domain of cloud forensics, a crucial topic given the fact that 57% of organizations store their data in the public cloud. We begin by outlining the unique challenges that cloud environments pose for forensic investigators. Then, we delve into the various methodologies and tools that can be used for such an investigation. We will also discuss the adaptation of traditional forensic processes to the cloud model. The thesis also presents a comparative analysis of existing forensic tools and their effectiveness for conducting cloud forensics, highlighting the need for tools that can handle the dynamic nature of cloud computing.

Stakeholders

Name	Contribution/Role
Zakaria El Morabit	Writing test plan
Ignace De Cock	Share input on risks and tools to be used

Risk Analysis

Trouble accessing relevant data

Description	Risk	Impact	Prior.
We might face difficulties accessing data in Azure due to the control being exercised by Microsoft.	5	3	3

Action	Action type
We will try to limit the possibility of this occurring by first making sure we have explored every possibility to access specific data we are looking for. Beyond that, we will attempt to ask Microsoft through contacts provided by EY to see if they would be able to provide information that we wouldn't have access to in case of an incident.	Mitigate

Infrastructure cost

Description	Risk	Impact	Prior.
To perform my research, I will need to deploy mock infrastructure which may cost money.	4	1	1

Action	Action type

Description	Risk	Impact	Prior.
When you run out of credit on Azure, all your services will be paused till next billing, so you won't lose any data. Through EY we have been granted about 140 euro a month which is plenty for our needs. To make sure we never run out, we pause or delete our resources when we are not using them.		Mitigate	

No/little collaboration with the CSP

Description	Risk	Impact	Prior.
The Cloud Service Provider, in our cause Azure, may decide that they will not assist in any forensics in case of an incident.	3	4	2

Action	Action type
Since public cloud is hosted on Microsoft's servers, we cannot perform any digital forensics on the physical infrastructure. However, this means that Azure may have access to the physical storage to recover deleted data. In the case that they can't/won't share this with us, there is not much we can do.	Accept

Data Integrity

Description	Risk	Impact	Prior.
When collecting data, it may be possible data gets altered.	2	2	1

Action	Action type
For example, when you make an image of your memory in a virtual machine, actions made during that process will be part of that image. This could happen with other artefacts as well. We can mitigate this risk by clearly documenting all steps taken before acquiring the artefact and researching how this can impact the evidences.	Mitigate

Resource de provisioning

Description	Risk	Impact	Prior.
Cloud resources can quickly scale up or down, which can result in loss of forensic evidence.	1	2	0

Action	Action type
When setting up cloud resources, you are provided with multiple options on how you would like to scale your resource. We make sure to turn all of these off, and if this is not possible we will make sure that your usage will not pass the threshold for it to upscale.	Avoid

Tools not being applicable/usable.

Description	Risk	Impact	Prior.
Since cloud environments are constantly evolving, it may be possible that tools found in research papers, forums, or documentation may not work (properly) anymore.	3	2	1

Action	Action type
To mitigate this risk, we will be focusing on information and tools provided by Azure. This will ensure the results will be reliable. It will remain useful however to still test the tools, since knowing whether these still work or not is important to know in forensics.	Mitigate

Technical skill gap

Description	Risk	Impact	Prior.
To perform forensics into the different services provided by Azure, we could be held back by a lack of technical skill and understanding.	4	2	2

Action	Action type
We can rely on help from different colleagues at the office that are experts in cloud infrastructure, they will be able to explain what certain services can or can't offer. If needed they will also be able to help set test environments up.	Mitigate

Data from different zones

Description	Risk	Impact	Prior.
It is not unusual for a cloud environment to be hosted in different zones. In these cases it can be difficult to correlate different logs and evidences	1	2	0

Action	Action type
This can be easily avoided through two methods. Firstly, all timestamps will be converted to UTC to ensure it remains uniform across time zones. Secondly, for my research we will host all resources in western Europe wherever possible.	Avoid

Not having the right configuration

Description	Risk	Impact	Prior.
When setting up any sort of resource or service in Azure, most if not all settings that have to do with monitoring and logging will be turned off.	4	4	3

Action	Action type

Description	Risk	Impact	Prior.
To mitigate this risk, we will keep good track of what the default settings are and document what difference the relevant settings make. We will also investigate deploying example environments based on scripts to avoid making human errors when configuring this. Not configuring this		Mitigate	

Cyber attacks on our environment

Description	Risk	Impact	Prior.
Since we have services that are open to the internet, there may be attempts to hack those services.	3	1	1

Action	Action type
Any environment will only be running when we are actively performing research on them. Since we use templates to easily spin up a resource, removing the one that was already running is no issue in case of a breach. Unplanned downtime would be the only impact.	Mitigate

Test strategy

In this project, we aim to validate our research findings through a structured testing approach. Initially, sanity testing will ensure our simulated environments reflect real-world cloud forensic scenarios. We'll then employ cross-browser testing to confirm that browser choice does not skew forensic artifact discovery. Additionally, cross-tool testing will be vital for gathering specific artifacts like memory dumps, which Azure does not directly provide; we'll use third-party tools for this purpose, comparing outputs to assess reliability.

Our primary objective is to ensure that our research results are reliable and reproducible. This testing regimen is designed to verify that the forensic techniques, tools, and artifacts we identify are applicable and effective in real investigative situations, thus providing a framework for cloud forensic investigations.

Test type	Planned?	Coverage and criteria
Unit testing	No	This involves testing individual components or pieces of code to ensure they function correctly in isolation. For our topic we will not be developing any tools, so this kind of testing is not relevant.
Integration testing	No	Integration testing focuses on the interaction between different component to make sure they work as intended.
System testing	Yes	This is a holistic approach that tests the complete system to evaluate its compliance in its requirement. This will become relevant when we make

Test type	Planned?	Coverage and criteria
		the simulated environment. Azure has a lot of health checks built in that we can use to verify whether the system works properly or not.
Sanity testing	Yes	With this kind of testing, we would perform a quick run through of the functionalities to ensure that the application/unit under test is working after a minor change. This will be performed with the system testing. If we change the system to probe different kinds of data, a quick sanity test to make sure the health checks aren't showing different results.
Interface testing	No	Interface testing focuses on checking the communication and data transfer processes between different modules or systems. This includes testing web services, REST APIs, and other interfaces to ensure that data is correctly sent and received. Since we will be testing Azure services in isolated environments, this will not be relevant.
Regression testing	No	Regression testing is conducted to ensure that recent updates or modifications to the software/system haven't negatively impacted the functionality that was previously in place. It verifies that existing features continue to operate as expected. Since we will not be developing any software or infrastructure, this kind of testing will not be relevant.
Beta testing	No	Beta testing is a pre-release testing phase where a sample of the intended audience uses the product in a real environment to uncover any defects that were not found in previous tests. Since the point of this research is to report on my findings an not developing a product, this testing will not be relevant.
Performance testing	No	Performance testing involves testing the speed, response time, reliability, resource usage, and scalability of a software under a particular workload. It includes several sub-types such as load testing or stress testing. We aren't developing a product, so this testing is not relevant.
Security Testing	No	Security testing detects potential vulnerabilities and threats in software to prevent unauthorized access, data theft, and other security breaches.
Cross-browser and cross-system testing	Yes	This type of testing ensures that the software works across different web browsers (Chrome, Firefox, Safari, etc.) and operating systems (Windows, macOS, Linux, etc.). It helps to identify browser-specific or system-specific compatibility issues. For our research this is out of scope and not relevant to cloud forensics.
Usability testing	No	Usability testing evaluates the software's user interface and user experience (UI/UX) from the perspective of the end user. It aims to identify any aspects of the software that could be confusing, frustrating, or difficult for users, with the goal of making the software as user-friendly

Test type	Planned?	Coverage and criteria
		as possible. We aren't developing a product and neither do we have some sort of user interface, so this testing is relevant.
Cross-tool testing	Yes	This type of testing is where we use different tools to verify, they yield the same result. This is a useful approach to verify the consistency, accuracy, and reliability of our results. Although this test isn't always applicable, we won't need to verify tools provided by Azure themselves, it is still useful to include wherever we can.

Tools

Name	Version	Supplier	Description
Visual Studio Enterprise Subscription	latest	EY	Subscription that provides the needed credits to perform research.
Volatility	3	Volatility foundation	A python tool that helps analyse memory dumps.
Azure Storage Explorer	Latest	Azure	Allows to manage azure cloud storage resources from your desktop.
VMware Workstation pro	17	VMware	Desktop hypervisor to run virtual machines on Windows or Linux PCs.
Everything	1.4.1	Voidtools	Quicker and efficient way to find files on windows.

Test results

Test results overview

Code	Short description	Tester	Created on	Executed on	Result
Cross_tool_01	Cross-tool testing NotMyFault	30-04-2024	2-05-2024	OK	Cross_tool_01
Cross_tool_02	Cross-tool testing Serial Dump	30-04-2024	2-05-2024	OK	Cross_tool_02
Cross_browser_01	Cross browser test Firefox	2-05-2024	10-05-2024	OK	Cross_browser_01
Cross_Browser_02	Cross browser test Chrome	2-05-2024	10-05-2024	OK	Cross_Browser_02
Cross_Browser_03	Cros browser test Edge	2-05-2024	10-05-2024	OK	Cross_Browser_03
Health_test_01	Health test AKS	1-05-2024	12-05-2024	OK	Health_test_01
Health_Test_02	Health test VM	1-05-2024	12-05-2024	OK	Health_Test_02
Health_Test_03	Health test Web App	2019-03-12	12-05-2024	NOK	Health_Test_03

Test result details

System testing

Health_Test_01

Short description	Tester	Created on	Executed on	Result
Resource Health in Azure monitors the status of your resources, such as virtual machines and Kubernetes clusters, to ensure they are running as expected. It provides real-time insights into the health of your resources, alerting you to any issues. If problems arise, it offers diagnostic tools and recommendations for troubleshooting.	Zakaria El Morabit	1-05-2024	12-05-2024	OK

Expected result	Actual result
All the checks should show that there are no issues affecting this Kubernetes cluster.	The Kubernetes cluster is healthy, and Azure is not providing any recommendations.

Health_Test_03

Short description	Tester	Created on	Executed on	Result
The Availability Risk Alerts in Azure provide insights and warnings about potential risks affecting the availability and performance of your web app. These alerts highlight critical configurations, such as the number of instances your app runs on, the status of health checks, and the use of Auto-Heal features. They offer recommendations to mitigate downtime risks, such as scaling out to multiple instances and monitoring the health of your instances.	Zakaria El Morabit	1-05-2024	12-05-2024	NOK

Expected result	Actual result
The alert would show that we pass all the health checks such as Instance Distribution, Health Check configuration, Auto-Heal configuration etc.	Didn’t pass the instance distribution check, meaning the web app only runs on one instance. Not an issue for the sake of performing research.

Sanity testing

Short description	Tester	Created on	Executed on	Result
With this kind of testing, we would perform a quick run through of the functionalities to ensure that the application/unit under test is working after a minor change. This will be performed with the system testing. If we change the system to probe different kinds of data, a quick sanity test to make sure the health checks aren't showing different results.	Zakaria El Morabit	N/A	N/A	NA

Justification
This was not executed due to lack of relevance. My environment has little moving parts of which none contained changes that impact my research.

Cross-browser testing

Cross-browser_01

Short description	Tester	Created on	Executed on	Result
We run all our environments and health tests using different browsers to make sure they give the same results. In this case we will be looking at Firefox.	Zakaria El Morabit	1-05-2024	12-05-2024	OK

Expected result	Actual result
Perform all the tasks will yield the same results as it does on Microsoft Edge.	We were able to perform all the tasks as we would on Edge.

Cross-browser_02

Short description	Tester	Created on	Executed on	Result
We run all our environments and health tests using different browsers to make sure they give the same results. In this case we will be looking at Chrome.	Zakaria El Morabit	1-05-2024	12-05-2024	OK

Expected result	Actual result
Perform all the tasks will yield the same results as it does on Microsoft Edge.	We were able to perform all the tasks as we would on Edge.

Cross-browser_03

Short description	Tester	Created on	Executed on	Result
We run all our environments and health tests using different browsers to make sure they give the same results. In this case we will be looking at Microsoft Edge.	Zakaria El Morabit	1-05-2024	12-05-2024	OK

Expected result	Actual result
Perform all the tasks will yield the same results as it does on Google Chrome and Firefox.	We were able to perform all the tasks as we would on Chrome and Firefox.

Cross-tool testing

Cross_tool_01

Short description	Tester	Created on	Executed on	Result
In terms of 3rd party tools, the one applicable to my research are the ones that those that can dump the memory in virtual machines.	Zakaria El Morabit	30-04-2024	2-05-2024	OK

Expected result	Actual result
If the memory dump is the same size as the virtual machine's memory, we will know it did so successfully. Furthermore using Volatility to analyse its contents will show that the dump was done so correctly.	The dump is the same size the memory and can be analysed with Volatility.

Cross_tool_02

Short description	Tester	Created on	Executed on	Result
In terms of 3rd party tools, the one applicable to my research are the ones that those that can dump the memory in virtual machines.	Zakaria El Morabit	30-04-2024	2-05-2024	OK

Expected result	Actual result
If the memory dump is the same size as the virtual machine's memory, we will know it did so successfully. Furthermore using Volatility to analyse its contents will show that the dump was done so correctly.	The dump is the same size the memory and can be analysed with Volatility.

Conclusion

The system testing in Azure aimed to ensure resources, such as Kubernetes clusters and virtual machines, were running as expected. Health_Test_01 confirmed the Kubernetes cluster was healthy, though a placeholder text needed clarification. Health_Test_02 found that only 48% of activity was focused on the first screen, suggesting a UI adjustment was needed, and marked the test as NOK.

Health_Test_03 identified a failure in instance distribution as the web app ran on only one instance, also marked as NOK, but noted this did not impact the research.

Sanity testing, which checks functionalities after minor changes, was not executed due to lack of relevance, as the environment had no impactful changes. Cross-browser testing verified that the web app performed consistently across different browsers, including Firefox, Chrome, and Microsoft Edge, with all tests confirming expected results and marked as OK.

Cross-tool testing involved using third-party tools for memory dumping in virtual machines. Both tests confirmed that the memory dumps matched the virtual machine's memory size and were analyzable with Volatility, marking them as OK. Overall, the testing highlighted a few configuration issues but ensured the core functionalities performed reliably across various environments.

The tests did not address the automatic scaling of cloud resources, which could result in the loss of forensic evidence. This involves setting up cloud resources to prevent scaling that could disrupt data. The risk of cyber attacks on services open to the internet was not tested. This involves ensuring environments are only active during research and can be quickly decommissioned and restored if compromised.

The tests did not address the risk of automatic scaling of cloud resources, which could lead to the loss of forensic evidence. Mitigating this risk involves configuring cloud resources to prevent scaling actions that might disrupt data collection. Additionally, the tests did not cover the risk of cyber attacks on services exposed to the internet. Addressing this involves ensuring that environments are only active during active research periods and can be swiftly decommissioned and restored in case of a security breach. When performing forensics in an actual breach, it would be important to make sure bad actors no longer have the ability to mess with the artefacts.

Research Results

Client

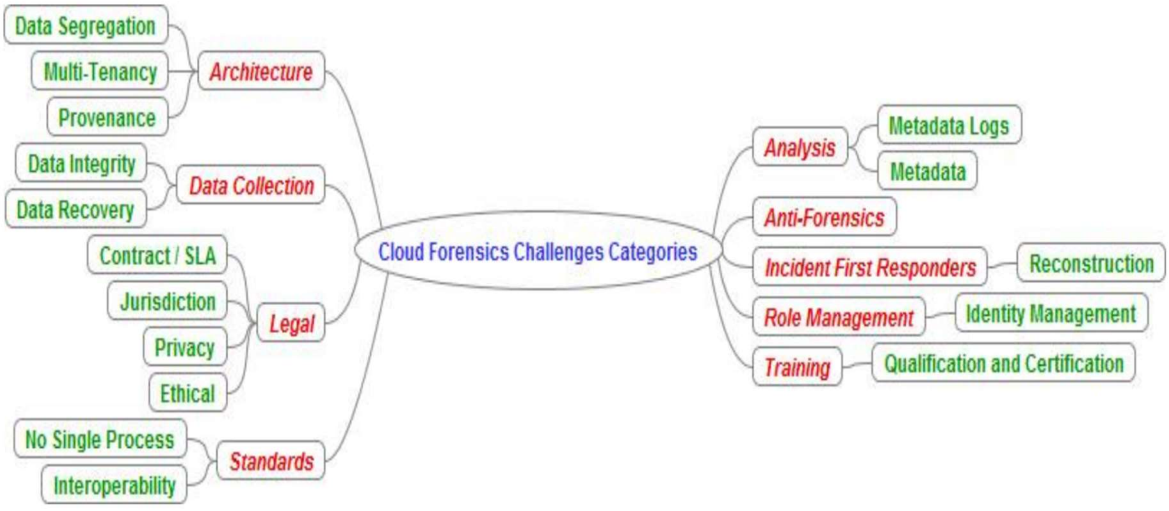
Ignace De Cock, our client and internship mentor, is an expert in the field of digital forensics. His expertise is an essential support for our research. He provides guidance and insight and corrects us when we encounter erroneous assumptions.

Summary

This document is a research report focused on cloud forensics within the Azure environment. The study explores different Azure services, including Azure Virtual Machines, Azure App Service, and Azure Kubernetes Service, to understand how they can be used for forensic investigations. Key findings include the importance of logging and memory dumps for gathering evidence and the complexity of managing Kubernetes. The report concludes that organizations should design their infrastructure expecting breaches and recommends further research into using Application Insights for enhanced forensic capabilities.

Problem statement

To best describe the current state of science, we refer to the NIST research from 2020, "NIST Cloud Computing Forensic Science Challenges." This document outlines the various challenges encountered in cloud forensics, identifying a total of 62 challenges. Below is an image that summarizes the different topics. For this research, we focus on the challenges within architecture, data collection, analysis, and incident first responders. Anything outside of these areas falls outside the scope of our study.



In cloud forensics, the compartmentalization and isolation of tenant data during resource allocation is a key concern. Additionally, the distribution of systems, locations, and endpoints that can store data plays a significant role. It is crucial to ensure the accurate and secure provenance of data, which is essential for maintaining and preserving the chain of custody.

Locating forensic artifacts in extensive, distributed, and dynamic systems is a complex task in cloud forensics. Collecting volatile data and extracting information from virtual machines are also challenging aspects. In a multi-tenant environment, where data is shared among multiple computers across diverse locations and accessible to various parties, maintaining data integrity is crucial. Forensic experts face the challenge of not being able to fully image all artifacts in the cloud. Furthermore, making one customer's data accessible without compromising the confidentiality of other customers is a delicate issue. Cloud forensics also involves the complexity of recovering deleted data in a shared and distributed virtual environment.

The work in cloud forensics includes the correlation of forensic artifacts between different cloud providers and within a single provider. This is essential for reconstructing events from virtual images or storage. The integrity of metadata plays a crucial role, as it provides essential information about the artifacts. Another important aspect is the timeline analysis of log data, including the synchronization of timestamps, which helps in accurately establishing the chronology of events.

The capability and reliability of cloud providers as first responders for data collection are crucial in cloud forensics. Correctly performing an initial triage is often challenging due to the complex nature of cloud environments. Moreover, processing the large volumes of collected forensic artifacts poses a significant challenge due to the vastness and diversity of the data.

Research question

What methods, artifacts, and tools can we best use to conduct cloud forensics? The main goal of this research is to centralize information available and to learn what possibilities are out there. The primary choice will be tools made available to us by Azure. A method, artifact or tool will have to prove that the information is reliable and correct. It also has to proof its benefit in forensic research.

Hypothesis

We expect to find that most tools and methods for conducting cloud forensics will be provided by the Cloud Service Providers (CPS), as it is challenging to perform monitoring and scanning on a platform that is constantly changing and evolving using external software and tools. However, we suspect that many artifacts used in traditional digital forensics will remain relevant, as end-users continue to be the greatest security risk to organizations, even when using cloud environments.

Results

Azure App Service

In this section, the results related to the AppServiceHTTPLogs table of Azure App Service are presented. The table contains the following columns:

Log	**Type**	**Description**
_BilledSize	real	The record size in bytes
CIp	string	IP address of the client
ComputerName	string	The name of the server on which the log file entry was generated.
Cookie	string	Cookie on HTTP request
CsBytes	int	Number of bytes received by server
CsHost	string	Host name header on HTTP request
CsMethod	string	The request HTTP verb
CsUriQuery	string	URI query on HTTP request
CsUriStem	string	The target of the request
CsUsername	string	The name of the authenticated user on HTTP request
_IsBillable	string	Specifies whether ingesting the data is billable. When _IsBillable is `false` ingestion isn't billed to your Azure account
Referer	string	The site that the user last visited. This site provided a link to the current site
_ResourceId	string	A unique identifier for the resource that the record is associated with

Result	string	Success / Failure of HTTP request
ScBytes	int	Number of bytes sent by server
ScStatus	int	HTTP status code
ScSubStatus	string	Substatus error code on HTTP request
ScWin32Status	string	Windows status code on HTTP request
SourceSystem	string	The type of agent the event was collected by. For example, `OpsManager` for Windows agent, either direct connect or Operations Manager, `Linux` for all Linux agents, or `Azure` for Azure Diagnostics
SPort	string	Server port number
_SubscriptionId	string	A unique identifier for the subscription that the record is associated with
TenantId	string	The Log Analytics workspace ID
TimeGenerated	datetime	Time when event is generated
TimeTaken	int	Time taken by HTTP request in milliseconds
Type	string	The name of the table

We have found that we can split app service logs into different categories: application logging, web server logging, detailed error messages, failed request tracing, deployment logging.

Application logging in Azure App Service supports both Windows and Linux environments. It allows you to capture log messages generated by your application code. These messages can originate from the web framework you select or directly from your application code using the standard logging patterns of your programming language. Each log message is categorized into one of the following levels: Critical, Error, Warning, Info, Debug, and Trace. The verbosity of the logging can be adjusted by setting the severity level when enabling application logging. Logs are stored either in the App Service file system or in Azure Storage blobs.

For Windows environments, web server logging captures raw HTTP request data in the W3C extended log file format. Each log entry includes various details such as the HTTP method, resource URI, client IP address, client port, user agent, and response code. These logs are stored in the App Service file system or in Azure Storage blobs. Web server logging provides insights into HTTP requests and responses, which can be useful for troubleshooting and performance monitoring.

In Windows environments, detailed error messages are saved in the App Service file system. These messages are copies of the .htm error pages that would have been sent to the client browser. Although detailed error pages should not be exposed to clients in a production environment for security reasons, App Service retains these error pages for occurrences where the HTTP status code is 400 or greater. The information in these pages can assist in diagnosing why the server returned the error code.

Failed request tracing is available in Windows environments and provides detailed tracing information on failed requests. This includes a trace of the IIS components used to process the request and the time taken by each component. This type of logging is useful for improving site performance or isolating specific HTTP errors. For each failed request, a separate folder is generated, containing an XML log file and an XSL stylesheet for viewing the log file.

Deployment logging supports both Windows and Linux environments. It records logs for when content is published to an app. Deployment logging occurs automatically and does not have configurable settings. It helps in determining why a deployment might have failed. For example, if a custom deployment script is used, deployment logging can assist in identifying why the script is not functioning as expected.

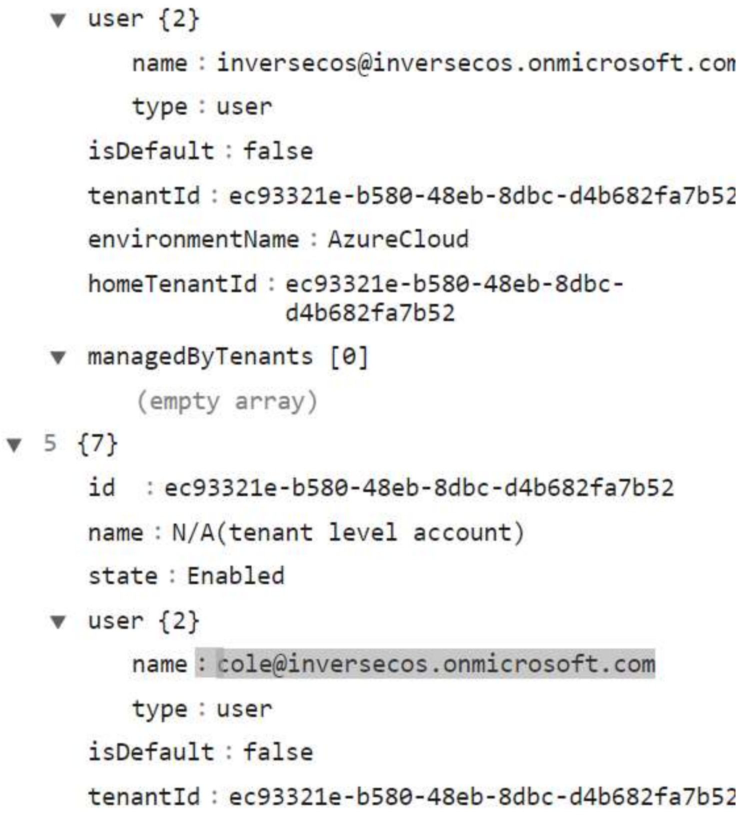
Azure virtual machines

When Azure CLI is used on a host, a specific folder is created under the user account at the path location “C:\Users<username>.azure”. This folder contains a file named azureProfile.json, which stores information about the properties of Azure subscriptions and users accessed on the host. The timestamps and user accounts in this file can be used to check the relevant Azure AD / Sign-in and UAL (User Access Log) logs. This helps in constructing a timeline of malicious activities.

Azure Profile JSON Files

- **Location:** C:\Users\<username>\.azure\azureProfile.json

- **Content:** Information about Azure subscriptions and user accesses, including timestamps. An example is shown in the image below.



```
    ▼ user {2}
          name : inversecos@inversecos.onmicrosoft.com
          type : user
      isDefault : false
      tenantId : ec93321e-b580-48eb-8dbc-d4b682fa7b52
      environmentName : AzureCloud
      homeTenantId : ec93321e-b580-48eb-8dbc-
                     d4b682fa7b52
    ▼ managedByTenants [0]
          (empty array)
▼ 5 {7}
      id  : ec93321e-b580-48eb-8dbc-d4b682fa7b52
      name : N/A(tenant level account)
      state : Enabled
    ▼ user {2}
          name : cole@inversecos.onmicrosoft.com
          type : user
      isDefault : false
      tenantId : ec93321e-b580-48eb-8dbc-d4b682fa7b52
```

Azure Active Directory (Azure AD) administrative activities are stored in the Azure AD Audit Log. By default, these logs flow to the Azure AD portal for a period of 30 days and to the Office 365 Unified Audit Log for either 90 days (for E3 subscriptions) or 1 year (for E5 subscriptions). These logs are for understanding the scope of any administrative compromise of a tenant and include information on:

- Policy changes
- User / Group / Device changes
- Application changes
- Authentication method changes
- Administrative role changes
- Hybrid authentication changes

From my conversations with Mike Martin from we learned that virtual machines are recoverable within 24h by Azure even though they seem completely gone on the client side. It does require that the system administrator has not made any new machines using the same name, this will reduce the chances of recovering what was lost to zero since they rely on the GUID. This is however not a reliable way of recovering lost evidence and artefacts.

Azure Kubernetes Service

By going to the control panel of your Kubernetes cluster, you will find “Diagnose and solve problems” on the left-hand side of the screen. By clicking on that you will find the “Collect memory dump” this will automatically save the memory of the VM on which the container was running. The same can be done for the image of the virtual machine.

Discussion and conclusion

In this chapter we will discuss the conclusions that can or cannot be drawn from our paper. To reiterate, the point of this research was to learn what cloud forensics is, why you would do it, what challenges it brings with it and what different techniques or technologies are possible to perform it. Furthermore, during this conclusion we will provide recommendations for organizations on how to provide the tools and information needed for a forensic researcher to provide the best possible results.

What we’ve learned

One of the main conclusions that can be drawn is that the amount of information available to a researcher is not necessarily dependent on the level of abstraction but rather on the specific service being used. We had assumed that services with the least amount of abstraction would provide the most information, but that is not necessarily the case. For example, we initially thought that Azure Virtual Machines (AVM) would yield the most artifacts since this service is closest to the physical layer of the machines that the hypervisor runs on. This is surprising because adding layers of abstraction usually distances the user from the underlying infrastructure. Just as you cannot access all the information that Windows provides through its graphical user interface—some things require the user to go into the terminal and/or open PowerShell—our results showed that different services, such as Azure App Service, can provide a comparable number of artifacts to those found with virtual machines. The difference between the services primarily lies in the type of information available, rather than the quantity.

Azure App Service

To continue with Azure App Service, Azure provides an AppServiceHTTPLogs table, which can be accessed through the Azure Portal. This table offers a plethora of different artifacts and evidence. When hosting an App Service, logs like these provide information on the clients visiting your website. One of the more important pieces of evidence is the CIP value, which stands for Client IP address. This can provide information on the location or even the Internet Service Provider. The address can also be compared against lists of known bot addresses to determine if the request was legitimate.

Another important log is the "TimeGenerated" field, which records the time when each entry was created. This is crucial because, without it, the forensic researcher would not be able to maintain a consistent timeline and link all the events together.

Finally, the Referer field indicates which website the client visited before arriving at yours. This can be useful for identifying the source of suspicious traffic. For example, if a phishing campaign directs users to your website, the referer log can reveal the URL of the malicious webpage. This information can also help reconstruct the attack vector. These are just a few examples of how these records can be utilized.

Azure Kubernetes Service

Azure Kubernetes Service (AKS) was by far the most difficult service to research. This difficulty primarily stemmed from the lack of technical knowledge regarding this technology. Managing it on a cloud platform further added to the complexity of the task. One of the most significant findings was the ability to collect memory dumps. This capability is crucial for forensic analysis because it provides detailed information about all processes running on the machine, open network connections, and loaded libraries. Such data is essential for pinpointing the perpetrator and understanding exactly what happened.

Azure Virtual Machines

This service is the closest to the physical machine, allowing us to even dump hypervisor memory from the serial console. This means we don't need to access the machine directly and can instead obtain information through the Azure Portal. This is important because directly interacting with the VM might tamper with the artifacts found on it.

Azure provides a range of logs divided into three categories: Application logs, Security logs, and System logs. The choice of which logs an organization should enable depends on the criticality of the system and its use case. Some logs should be active regardless of the VM's importance. For example, the 'Audit failure' log, categorized under Security logs, indicates if the machine has been altered and is no longer compliant with security policies. This log also provides a timestamp indicating when the change occurred.

Beyond logs, having access to the machines and their environment is crucial, as this is where an analyst can extract the necessary evidence. Azure is often capable of restoring deleted content within twenty-four hours, but this is not always reliable. Additionally, restoration is only possible if the names assigned to the machines have not been reused. If the same GUID is used after the original has been wiped, Azure will not be able to restore it. This applies to all components in the environment, such as disks, snapshots, and memory.

Final thoughts

As shown in the previous paragraphs, there are many ways to perform cloud forensics, and as hypothesized, this is primarily achieved through tools provided by Azure. What I did not account for when writing the blueprint was the vast scope of Azure. I had assumed that I could cover most of the services provided by the platform, but the scope turned out to be too large to research comprehensively. This is why I opted to focus on the three services discussed in this document, each representing a different layer of abstraction: Azure Virtual Machine, being closest to IaaS, and Azure App Service, representing the other end of the spectrum, closer to SaaS.

From our paper, we can draw several conclusions. Firstly, organizations should design their infrastructure with the expectation that it will be breached. As Mike Martin, a Senior Cloud Solution Architect at Microsoft, aptly put it: failure is not an option; it’s imminent. Therefore, when determining what information and logs to store, this expectation should be kept in mind. Organisations should also avoid using VMs in their infrastructure as much as possible and instead rely on the PaaS solutions available in azure instead. Manually maintaining virtual machines is very difficult and increases the chances of machines not being well updated making them vulnerable for breaches.

Further research would be especially useful if it focused on Application Insights. This tool collects and stores a vast array of telemetry data, including traces, metrics, and events, which could be invaluable for forensic investigations. Exploring how this telemetry can be effectively used to track malicious activity or reconstruct the sequence of events leading to a security incident could provide organizations with a new tool in their forensic toolkit. Additionally, Application Insights is integrated within Azure, meaning it works well with Azure DevOps and Azure Security Center. Further research could help in understanding how to build a more comprehensive forensic framework by leveraging these integrations.

Techincal documentation

Summary of the assignment

The research focused on exploring cloud forensics, particularly within the Azure platform, to understand its complexities, challenges, and the tools available for effective forensic analysis. The study examined three Azure services—Azure Virtual Machines, Azure App Service, and Azure Kubernetes Service—each representing different layers of abstraction.

In this research, we will outline various methods for conducting forensic analysis across three key Azure services. Additionally, we will detail the research process to ensure that others can easily replicate the steps and achieve similar results.

Impact on the infrastructure

Azure App Service

Infrastructure Requirements

- Azure Subscription: An active Azure subscription is needed to deploy and manage App Services.

- App Service Plan: At least a Basic tier plan, though Standard or Premium tiers might be required depending on the scale of the app and the volume of logs.

- Storage Account: If you are saving logs to Azure Storage, you'll need a General-purpose v2 (GPv2) storage account.

Licenses and Subscriptions

Azure Monitor: Required for log analytics and monitoring. Azure Monitor includes Log Analytics, which is essential for analysing the AppServiceHTTPLogs table and other telemetry data.

- Azure Log Analytics Workspace: This is needed to store and query the logs collected from your App Service.

Software and Configuration

- Azure Portal Access: For configuring the logging and monitoring settings.

- Visual Studio Code 2022: This is used to push code to your app service. This way we can simulate traffic.

- Diagnostic Logs Configuration: Set up through the Azure Portal to collect application, web server, detailed error, and deployment logs.

Azure Virtual Machines

Infrastructure Requirements

- Azure Subscription: An active Azure subscription with permissions to deploy and manage VMs. We had an Azure Enterprise subscription providing 140euro of credits every month. We recommend at least 80 euro monthly.

- Virtual Machines: Deploy the Standard_D4s_v3 Azure VM, running Windows Server 2019 Datacenter, featuring 4 vCPUs and 8 GB of memory. For the purpose of this thesis the standard HDD managed disk is enough.

- Storage Account: Needed for storing diagnostic data, backups, and snapshots. When setting up the machine you will be prompted to make or choose one.

Licenses and Subscriptions

- Windows Server License: For our VM we used the Windows Server 2019 Datacenter Edition.

- Azure Monitor with Log Analytics: For monitoring VM performance and gathering logs. This was included in the Enterprise subscription provided by EY.

Software and Configuration

- Azure CLI or Azure PowerShell: For managing VMs and associated services.

- Diagnostic Settings: Enabled on virtual machines to collect and send logs to Azure Monitor or a storage account. This feature was the focus of the research.

Azure Kubernetes Service

Azure Kubernetes Service (AKS) is a managed Kubernetes service provided by Azure. It simplifies the deployment, management, and scaling of containerized applications using Kubernetes.

Infrastructure Requirements

- Azure Subscription: An active Azure subscription with permissions to deploy and manage AKS resources. For minimal setups, a subscription similar to the Azure Enterprise plan.

- AKS Cluster: Deploy a minimal AKS cluster with basic node configurations. For cost efficiency, we use VM sizes like Standard_B2s or Standard_D2s_v3 for the worker nodes.

- Azure Storage Account: Required for storing logs and diagnostic data from the AKS cluster. During the cluster setup, you will need to create or select a storage account.

Licenses and Subscriptions

- Kubernetes License: Kubernetes itself is open-source and does not require a separate license, but you need to pay for the underlying Azure resources.

- Azure Monitor with Log Analytics: For monitoring the AKS cluster and gathering logs. This service is included in the Azure subscription and provides insights into the cluster's performance and health.

Software and Configuration

- Kubernetes CLI Tools: Install kubectl for managing the AKS cluster and deploying applications. This tool is for interacting with the Kubernetes environment.

- Diagnostic Settings: Enable diagnostic settings in AKS to collect and send logs to Azure Monitor or a storage account. This setup ensures that logs and metrics from the AKS cluster are available for analysis and monitoring. Since this is the focus of the research, the diagnostics turned on or off will constantly change to see the effects and the contents of it.

Release plan

Azure App service

1. **Clone the Application**

    1. Open your terminal or command prompt on your local machine.

    2. Clone the Flask application repository:

```
git clone https://github.com/Azure-Samples/msdocs-python-flask-webapp-quickstart
```

2. **Set Up the Application Locally**

   This step is done for testing to make sure the application works as expected. This part can be skipped.

   1. Navigate to the application folder:

   ```
   cd msdocs-python-flask-webapp-quickstart
   ```

   2. Create a virtual environment for the application:

      - **Windows:**

        ```
        py -m venv .venv

        .venv\Scripts\activate
        ```

      - **Linux:**

        ```
        python3 -m venv .venv

        source .venv/bin/activate
        ```

   3. Install the dependencies listed in requirements.txt:

      ```
      pip install -r requirements.txt
      ```

   4. Run the Flask application:

      ```
      flask run
      ```

   5. Open a web browser and navigate to http://localhost:5000 to see the Flask app running locally.

3. **Create a Web App in Azure**

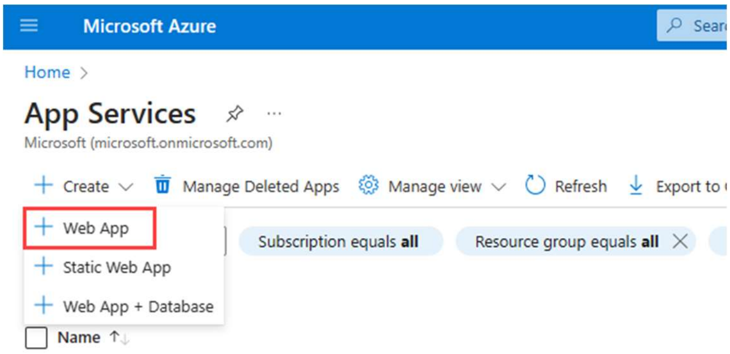
   1. Sign in to the Azure portal.

   2. Create a new Azure App Service:

      - Enter ‘app services’ in the search bar at the top of the Azure portal.

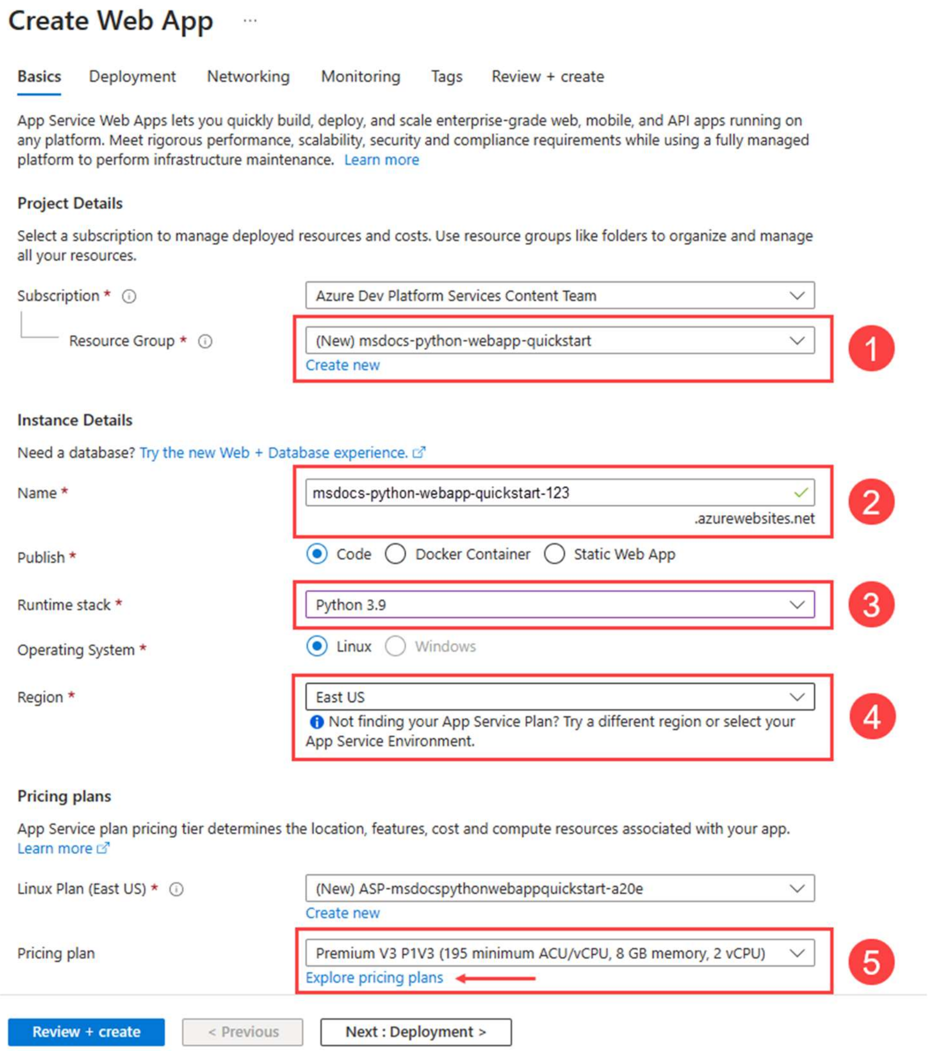
      - Select ‘App Services’ from the drop-down menu.

      - Click + Create, then select + Web App.

- 

3. **Fill out the Create Web App form:**

   - **Resource Group:** Select Create new and use the name msdocs-python-webapp-quickstart.
   - **Name:** Enter msdocs-python-webapp-quickstart-XYZ, where XYZ are any three random characters (must be unique across Azure).
   - **Runtime stack:** Choose Python 3.9.
   - **Region:** Select any Azure near you, however we opted for the US east since this was the cheapest region at the time.
   - **App Service Plan:** Click Explore pricing plans, then select the Basic B1 plan under Dev/Test for better performance compared to the Free F1 plan.

- 

Create Web App ···

App Service Web Apps lets you quickly build, deploy, and scale enterprise-grade web, mobile, and API apps running on any platform. Meet rigorous performance, scalability, security and compliance requirements while using a fully managed platform to perform infrastructure maintenance. Learn more

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure Dev Platform Services Content Team

Resource Group * : (New) msdocs-python-webapp-quickstart
Create new

Instance Details

Need a database? Try the new Web + Database experience.

Name * : msdocs-python-webapp-quickstart-123 .azurewebsites.net

Publish * : ● Code ○ Docker Container ○ Static Web App

Runtime stack * : Python 3.9

Operating System * : ● Linux ○ Windows

Region * : East US
Not finding your App Service Plan? Try a different region or select your App Service Environment.

Pricing plans

App Service plan pricing tier determines the location, features, cost and compute resources associated with your app. Learn more

Linux Plan (East US) * : (New) ASP-msdocspythonwebappquickstart-a20e
Create new

Pricing plan : Premium V3 P1V3 (195 minimum ACU/vCPU, 8 GB memory, 2 vCPU)
Explore pricing plans

- Click Select to apply changes.

4. **Review and Create the Web App:**
   - On the main Create Web App page, click Review + create.
   - Review your configuration and click Create to deploy the App Service.

Azure Virtual Machines

1. **Sign In to Azure**
   1. Open your web browser and sign in to the Azure portal.
2. **Create a Virtual Machine**

1. **Search for Virtual Machines:**
   - Enter virtual machines in the search bar at the top of the Azure portal.
   - Under the Services section, select Virtual machines.

2. **Start VM Creation:**
   - On the Virtual machines page, click + Create and then select Azure virtual machine. This will open the Create a virtual machine page.
3. **Configure Instance Details:**
   - **Virtual Machine Name:** Enter myVM.
   - **Image:** Choose Windows Server 2022 Datacenter: Azure Edition - x64 Gen 2.
   - Leave other default settings as is.
4. **Set Administrator Account:**
   - **Username:** Enter azureuser.
   - **Password:** Create a password that is at least 12 characters long and meets complexity requirements.
5. **Configure Inbound Port Rules:**
   - Select Allow selected ports.
   - Choose RDP (3389) and HTTP (80) from the drop-down menu.
6. **Review and Create:**
   - Click Review + create at the bottom of the page.
   - After validation, click Create to deploy the VM.
7. **Go to Resource:**
   - Once deployment is complete, click Go to resource to access your VM.

3. Connect to Virtual Machine

1. **Create RDP Connection:**
   - On the VM’s overview page, click Connect > RDP.
   - In the Connect with RDP tab, ensure the default settings are used to connect via IP address and port 3389.
   - Click Download RDP file.
2. **Connect via RDP:**
   - Open the downloaded RDP file.
   - Click Connect when prompted.
   - In the Windows Security window, click More choices, then Use a different account.

- Enter localhost\azureuser for the username, input the password you created, and click OK.

Note: You may see a certificate warning during sign-in. Click Yes or Continue to proceed.

4. Install IIS Web Server

1. **Open PowerShell:**
    1. Once connected to the VM via RDP, open a PowerShell prompt.
2. **Install IIS:**
    1. Run the following command to install the IIS web server:

    ```
    Install-WindowsFeature -name Web-Server -IncludeManagementTools
    ```
3. **Close the RDP Connection:**
    1. After installation is complete, close the RDP session.

5. View IIS Welcome Page

1. **Access IIS Welcome Page:**
    - In the Azure portal, select your VM and, in the overview section, hover over the IP address to show the Copy to clipboard option.
    - Copy the IP address and paste it into a web browser tab.
2. **Verify Installation:**
    - The default IIS welcome page should appear, confirming the successful installation.

Azure Kubernetes Service

Just like the other services, the point of the cluster we are building is to have something to interact with so we can evaluate the different monitoring tools Azure provides and to see how they react against certain interactions.

1. Sign In to Azure

1. Open your web browser and sign in to the Azure portal.

2. Create an AKS Cluster

Navigate to Create a Resource:

- On the Azure portal home page, select Create a resource.

Select AKS:

- In the Categories section, select Containers > Azure Kubernetes Service (AKS).

3. **Configure Basic Settings:**

    1. **Subscription:** Select the Azure subscription you want to use.

    2. **Resource Group:** Select Create new, enter a resource group name such as myResourceGroup, and then select OK. This is recommended for testing or evaluation to avoid impacting production workloads.

    3. **Cluster Preset Configuration:** Select Dev/Test. You can compare presets by selecting Compare presets.

    4. **Kubernetes Cluster Name:** Enter myAKSCluster.

    5. **Region:** Choose a region like East US 1 since this will be the cheapest.

    6. **Availability Zones:** Select None.

    7. **AKS Pricing Tier:** Select Free.

    8. Leave other defaults and select Next.

4. **Configure Node Pools:**

    1. Click Add node pool.

    2. **Node Pool Name:** Enter nplinux.

    3. **Mode:** Select User.

    4. **OS SKU:** Select Ubuntu Linux.

    5. **Availability Zones:** Select None.

    6. **Node Size:** Click Choose a size, select D2s_v3, and then select Select.

    7. Leave the default values for the remaining settings and select Add.

5. **Review and Create:**

    1. Click Review + create to validate the configuration.

    2. After validation, select Create.

6. **Access the AKS Resource:**

    1. Once deployment is complete, select Go to resource or navigate to the AKS cluster resource group and select the AKS resource.

3. Connect to the AKS Cluster

1. **Open Cloud Shell or Configure Local Environment:**

- If using Azure Cloud Shell, open it by clicking the >_ button on the top of the Azure portal.
- For local PowerShell, connect using Connect-AzAccount.
- For local Azure CLI, connect using az login.

2. **Configure kubectl:**
   - Run the following command to configure kubectl with your cluster credentials:

```
az aks get-credentials --resource-group myResourceGroup --name myAKSCluster
```

3. **Verify Connection:**
   - Run ‘kubectl get nodes’ to ensure your nodes are available and ready.
4. **Deploy the Application**
   2. Create a YAML Manifest File:
   3. In Cloud Shell, create a file named aks-store-quickstart.yaml.
   4. Paste the Manifest Content:
   5. Copy and paste the provided YAML manifest into the file. This includes configurations for RabbitMQ, order service, product service, and store front.
   6. Deploy Using kubectl:
   7. Apply the YAML manifest with:

```
kubectl apply -f aks-store-quickstart.yaml
```

   8. Check the deployment status. You should see outputs confirming the creation of deployments and services.

5. **Test the Application**

1. Check Pod Status:
   - Run kubectl get pods to ensure all pods are in Running status.
2. Monitor Service Status:
   - Use kubectl get service store-front --watch to monitor the external IP assignment. Initially, it may show as <pending>.
3. Access the Application:
   - Once the EXTERNAL-IP changes from <pending> to an actual IP address, use that IP address to open the application in a web browser.

Technical design

Local Machine Requirements

Development Environment:

- **Operating System**: Windows 10/11
- **Development Tools**:
  - **Visual Studio Code**: For writing and managing Terraform configurations and scripts.
  - **Azure CLI**: Command-line tool for managing Azure resources.
  - **Azure PowerShell**: Alternative to Azure CLI for managing Azure resources.

Hardware Specifications:

- **CPU**: Minimum 4 cores
- **RAM**: Minimum 8 GB
- **Storage**: Minimum 100 GB SSD
- **Network**: Stable internet connection with a minimum of 1 Mbps download/upload speed

Infrastructure and Component Interactions

Azure App Service

Software and Configuration:

- **Azure Portal Access**: For configuring logging and monitoring settings.
- **Visual Studio Code 2022**: Used for code deployment and traffic simulation.
- **Diagnostic Logs Configuration**: Set up through Azure Portal to collect application, web server, and deployment logs.

Components:

- **App Service Plan**: Defines the region, instance size, and scaling options for your app. It determines the resources allocated to your App Service and affects performance, cost, and scalability. There are different tiers (e.g., Basic, Standard, Premium) that provide various features and capabilities, such as autoscaling and more storage.
- **App Service**: Hosts your web application or API. It provides the runtime environment (such as Python, .NET, or Node.js) for your application and includes features like custom domains, SSL certificates, and built-in load balancing.
- **Storage Account**: General-purpose v2 (GPv2) storage accounts are used to store logs and diagnostic data. This can include application logs, web server logs, and detailed error logs, which are essential for forensic analysis.

- **Azure Monitor**: Used to collect and analyze performance metrics and logs. Azure Monitor integrates with App Service to provide insights into application health and performance, including metrics such as response times and error rates.
- **Log Analytics Workspace**: A central repository for log data collected from various sources. For App Service, it stores logs and metrics that are sent from the Azure Monitor, is used for querying, and visualizing the data.

Interactions:

1. **Application Deployment**: Developers push code changes to the App Service. The App Service Plan dictates the resources and scaling options for the application.
2. **Logging and Diagnostics**: Diagnostic settings are configured to collect logs and metrics from the App Service. These logs are sent to Azure Monitor, which then forwards them to Log Analytics Workspace for further analysis.
3. **Monitoring and Analysis**: Azure Monitor tracks performance and availability metrics, and logs collected data in the Log Analytics Workspace. This data is analyzed to ensure application health and to troubleshoot issues.

Azure Virtual Machines

Components:

- **Virtual Machine**: A compute resource that runs an operating system (e.g., Windows Server 2019 or Ubuntu). VMs can be customized in terms of CPU, memory, and storage to match workload requirements.
- **App Service Plan**: Not used in VMs but important for comparison; the equivalent in VMs would be the VM size and configuration, which defines the resources available.
- **Storage Account**: Used for storing VM disks, diagnostic data, backups, and snapshots. Managed disks are a type of storage used to persist the VM's operating system and data.
- **Azure Monitor**: Monitors the performance and health of VMs. It collects metrics and logs from VMs, such as CPU usage, memory consumption, and disk I/O.
- **Log Analytics Workspace**: Stores diagnostic data from VMs. Logs and performance data from Azure Monitor are aggregated here for querying and analysis.

Interactions:

1. **VM Deployment**: VMs are created and configured using Azure Portal or CLI. They are set up with required operating systems and software.
2. **Diagnostics and Monitoring**: Diagnostic settings are enabled on VMs to collect and send logs to Azure Monitor. These logs include performance metrics and system events.
3. **Log Aggregation and Analysis**: Azure Monitor collects data from VMs and sends it to the Log Analytics Workspace. This data is then used to analyze VM performance and diagnose issues.

Azure Kubernetes Service

Components:

- **AKS Cluster**: The core component of AKS, which includes the master nodes (managed by Azure) and worker nodes (user-configurable). The cluster handles the deployment and management of containerized applications.

- **Node Pools**: Collections of worker nodes with specific configurations. Node pools allow you to run different types of workloads on different sets of nodes, optimizing performance and cost.

- **Azure Storage Account**: Used for storing logs and diagnostic data related to the AKS cluster. This includes container logs, application logs, and metrics.

- **Azure Monitor**: Provides monitoring capabilities for AKS. It collects metrics and logs from the Kubernetes cluster, such as resource utilization and container performance.

- **Log Analytics Workspace**: Aggregates and analyzes logs and metrics collected from the AKS cluster. This workspace is used to query and visualize data to understand cluster performance and diagnose issues.

- **Kubernetes CLI Tools (kubectl)**: Command-line tool used to interact with the AKS cluster. It allows users to deploy applications, manage cluster resources, and inspect logs.

Interactions:

1. **Cluster Deployment**: AKS clusters are provisioned and configured via Azure Portal or CLI. Node pools and storage accounts are set up to manage and store logs.

2. **Application Deployment**: Applications are deployed to the AKS cluster using Kubernetes manifests. The deployment is managed through kubectl, and applications run on the worker nodes.

3. **Monitoring and Diagnostics**: Azure Monitor collects metrics and logs from the AKS cluster. These logs are sent to Log Analytics Workspace for analysis, helping to track the health and performance of applications and infrastructure.

4. **Log Aggregation and Analysis**: Logs from the AKS cluster are aggregated in the Log Analytics Workspace. This data is used for in-depth analysis and troubleshooting of containerized applications.

External system interfaces

Azure Portal

The Azure Portal is a web-based interface that provides a unified console for managing Azure resources. It offers a graphical user interface (GUI) for interacting with various Azure services, including App Services, Virtual Machines, and Kubernetes clusters.

Usage in the Project:

- **Resource Management**: Used to create, configure, and manage Azure resources such as App Services, Virtual Machines, and AKS clusters. The portal's GUI simplifies resource provisioning and configuration tasks.

- **Diagnostics and Monitoring**: Accessed for configuring diagnostic settings and monitoring metrics. It allows users to set up logging, view metrics, and analyze performance data.

- **Log Analytics**: Used to query and visualize logs and metrics collected from various Azure services. The Azure Portal's Log Analytics interface provides tools for creating queries and generating reports.

Azure CLI

The Azure Command-Line Interface (CLI) is a cross-platform command-line tool that allows users to manage Azure resources and perform various tasks using scripted commands.

Usage in the Project:

- **Resource Deployment**: Commands were used to deploy and configure resources, including creating Virtual Machines and AKS clusters. For example, az vm create was used to deploy VMs, and az aks create was used to set up an AKS cluster.

- **Configuration Management**: The CLI was used to configure resources, manage settings, and perform administrative tasks. Commands such as az monitor diagnostic-settings create were used to configure diagnostic settings for monitoring.

- **Interaction with AKS**: az aks get-credentials was used to configure kubectl with the appropriate credentials for accessing and managing the AKS cluster.

Azure PowerShell

Azure PowerShell is a set of cmdlets for managing Azure resources from the PowerShell command line. It provides automation and scripting capabilities for Azure operations.

Usage in the Project:

- **VM Configuration**: PowerShell was used to perform configuration tasks on Azure Virtual Machines. For instance, commands were used to install IIS (Internet Information Services) on Windows Server VMs.

- **Automation**: PowerShell scripts were employed to automate repetitive tasks and configure VM settings efficiently. For example, scripts could automate the setup of diagnostic settings and other configurations.

Visual Studio Code

Visual Studio Code (VS Code) is a source code editor that supports various programming languages and tools through extensions. It provides features for code editing, debugging, and version control.

Usage in the Project:

- **Code Deployment**: Used to write and manage configuration files and scripts, such as Terraform configurations and Kubernetes YAML manifests. Extensions for Azure and Terraform were utilized to enhance development and deployment workflows.

- **Local Testing**: Enabled local testing of application code before deploying it to Azure App Service. Developers used VS Code to clone repositories, set up virtual environments, and run applications locally.

Azure Cloud Shell

Azure Cloud Shell is an online-based shell environment provided by Azure that includes pre-installed command-line tools and scripting capabilities.

Usage in the Project:

- **Cluster Management**: Used for managing Azure resources directly from the Azure Portal. It provided an accessible environment for executing Azure CLI commands and managing AKS clusters.

- **Kubernetes Management**: Provided access to kubectl for interacting with the AKS cluster. Users could perform tasks such as deploying applications and managing cluster resources using Cloud Shell's integrated tools.

Kubernetes CLI (kubectl)

Kubectl is a command-line tool for interacting with Kubernetes clusters. It allows users to perform administrative tasks and manage containerized applications.

Usage in the Project:

- **Cluster Management**: Used to deploy and manage applications in the AKS cluster. Commands like kubectl apply were used to deploy YAML manifests containing application configurations.

- **Monitoring and Troubleshooting**: kubectl commands were used to check the status of pods, services, and deployments. Commands such as kubectl get pods and kubectl logs were employed to monitor and troubleshoot applications running in the cluster.

Documentation

This document contains all the required documentation to fully understand and reproduce the project. It includes comprehensive details on setup, configuration, and deployment procedures for each component. This way, every aspect of the project can be replicated. With the information provided, readers should be able to follow the outlined steps and methodologies to achieve the same results.

Afterword

Completing this research on cloud forensics has been an insightful and rewarding journey. Given that 57% of organizations now store their data in the public cloud, understanding the complexities and challenges of cloud forensics is more critical than ever. This thesis aimed to explore these challenges, methodologies, and tools necessary for conducting thorough forensic investigations in cloud environments, specifically focusing on Azure.

Throughout this research, we have highlighted the unique challenges posed by cloud environments, such as data compartmentalization, integrity, and the complexity of analyzing vast and distributed systems. By examining various forensic tools and their effectiveness, we have emphasized the need for dynamic and reliable tools tailored to cloud computing.

This work has benefited greatly from the resources and support provided by AP Hogeschool and EY, whose contributions have been essential in facilitating this research. The practical application of theories through simulated breaches has also provided valuable insights into how cloud forensics can be applied in real-world scenarios.

I hope that this thesis will serve as a valuable resource for future researchers and professionals in the field of cloud forensics. It aims to bridge the gap between traditional digital forensics and the evolving needs of cloud environments, offering a comprehensive understanding of the current state of the art and the direction for future developments.

Sources

EC-Council, “What is Digital Forensics | Phases of Digital Forensics | EC-Council”, *EC-Council*, 30 augustus 2023. https://www.eccouncil.org/cybersecurity/what-is-digital-forensics/ (geraadpleegd 23 februari 2024).

Flexera, “2023 STATE OF THE CLOUD REPORT”, *flexera.com*, 2023. Geraadpleegd: 28 februari 2024. [Online]. Beschikbaar op: https://info.flexera.com/CM-REPORT-State-of-the-Cloud

M. Herman *e.a.*, “NIST cloud computing forensic science challenges”, aug. 2020. doi: 10.6028/nist.ir.8006.

“PB: Ultimate Guide to Incident Response in Azure”. https://offers.cadosecurity.com/ultimate-guide-to-incident-response-in-azure (geraadpleegd 29 februari 2024).

N. Bruijn, “Wat is het verschil tussen public, private en hybrid cloud?”, *OGD*, 26 januari 2024. https://blog.ogd.nl/wat-het-verschil-tussen-public-private-en-hybride-cloud (geraadpleegd 3 maart 2024).

“Integriteit bewijsmateriaal | Forensicon”, *Forensicon*. https://www.forensicon.nl/integriteit-bewijsmateriaal/ (geraadpleegd 3 maart 2024).

"FOR509: Enterprise Cloud Forensics and Incident Response | SANS Institute". https://www.sans.org/cyber-security-courses/enterprise-cloud-forensics-incident-response/ (geraadpleegd 25 februari 2024).

P. Purnaye en V. Kulkarni, "A comprehensive study of cloud forensics", *Archives Of Computational Methods in Engineering*, vol. 29, nr. 1, pp. 33–46, mrt. 2021, doi: 10.1007/s11831-021-09575-w.

Simonesavi, "Computer forensics chain of custody in Azure - Azure Example Scenarios", *Microsoft Learn*. https://learn.microsoft.com/en-us/azure/architecture/example-scenario/forensics/ (geraadpleegd 1 maart 2024).

Forensic Labs, "Azure Forensics and Incident Response - Forensic Labs - medium", *Medium*, 21 juni 2023. Geraadpleegd: 3 maart 2024. [Online]. Beschikbaar op: https://cloudyforensics.medium.com/azure-forensics-and-incident-response-c13098a14d8d

Sonia Cuff, "What's the difference between Azure Security Center, Azure Defender and Azure Sentinel?", *TECHCOMMUNITY.MICROSOFT.COM*. https://techcommunity.microsoft.com/t5/itops-talk-blog/what-s-the-difference-between-azure-security-center-azure/ba-p/2155188 (geraadpleegd 2 maart 2024).

Microsoft, "Azure Blob Storage". https://azure.microsoft.com/en-us/products/storage/blobs (geraadpleegd 3 maart 2024).

P. Mell en T. Grance, "The NIST definition of cloud computing", jan. 2011. doi: 10.6028/nist.sp.800-145.

J. R. Lyle, B. Guttman, J. M. Butler, K. Sauerwein, C. Reed, en C. E. Lloyd, "Digital Investigation Techniques", *NIST*, mei 2022, doi: 10.6028/nist.ir.8354-draft.

"Digital Forensics and Incident Response (DFIR) Training, Courses, Certifications and Tools | SANS Institute". https://www.sans.org/digital-forensics-incident-response/ (geraadpleegd 3 maart 2024).

"Wat is SIEM? | Microsoft Beveiliging". https://www.microsoft.com/nl-be/security/business/security-101/what-is-siem (geraadpleegd 3 maart 2024).

"Wat is SOAR? Technologie en oplossingen | Microsoft Beveiliging". https://www.microsoft.com/nl-be/security/business/security-101/what-is-soar (geraadpleegd 3 maart 2024).

IEvangelist, "Gegevensintegriteit garanderen met hashcodes - .NET", *Microsoft Learn*, 10 mei 2023. https://learn.microsoft.com/nl-nl/dotnet/standard/security/ensuring-data-integrity-with-hash-codes (geraadpleegd 3 maart 2024).

Wikipedia-bijdragers, "Active Directory", *Wikipedia*, 29 januari 2021. https://nl.wikipedia.org/wiki/Active_Directory (geraadpleegd 3 maart 2024).

Wikipedia contributors, "Logging (computing)", *Wikipedia*, 20 februari 2024. https://en.wikipedia.org/wiki/Logging_(computing) (geraadpleegd 3 maart 2024).

"chatGPT", *Openai*. https://chat.openai.com/ (geraadpleegd 3 maart 2024).

Msangapu-Msft, "Enable diagnostics logging - Azure App Service", *Microsoft Learn*, 2 februari 2024. https://learn.microsoft.com/en-us/azure/app-service/troubleshoot-diagnostic-logs

Genlin, “Capture a TCP dump from a Linux node in an AKS cluster - Azure”, *Microsoft Learn*, 10 april 2024. https://learn.microsoft.com/en-us/troubleshoot/azure/azure-kubernetes/logs/capture-tcp-dump-linux-node-aks