

Azure reference environment - Terraform deployment automation on GitHub

Project Charter Document

Sepp Eyckmans
Student Bachelor in de Elektronica-ICT – Cloud & Cyber Security

Inhoudsopgave

1. INLEIDING	3
2. PROBLEEMOMSCHRIJVING	4
2.1. Functionele Vereisten	5
2.2. Niet-Functionele Vereisten	5
3. DOELSTELLING	6
3.1. Specifiek	6
3.2. Meetbaar	6
3.3. Acceptabel	6
3.4. Realistisch	6
3.5. Tijdsgebonden	6
4. SCOPE	7
4.1. In-Scope	7
4.2. Out-of-Scope	7
5. DELIVERABLES	8
5.1. Terraform Infrastructuurcode	8
5.2. Werkende Azure demo omgeving	8
6. STAKEHOLDERS	9
6.1. Stagiair	9
6.2. Solution Architect	9
6.3. Cloud Expert	9
6.4. Arxus	10
6.5. Referentieklant "Demotronix"	10
6.6. Potentiële klanten	10
7. RISICO'S	11
7.1. Complexiteit van de Azure Landing Zone architectuur	11
7.2. Aanleren van de Terraform module library	11
7.3. Tijdsbeperking van de stage	11
8. PLANNING	12
8.1. Sprint 1	13
8.2. Sprint 2	13
8.3. Sprint 3	13
9. SUCCESCRITERIA	14
9.1. Geautomatiseerde deployment van de omgeving	14
9.2. On-demand opzetten en verwijderen van de omgeving	14
9.3. Gebruik van gestandaardiseerde infrastructuurcode	14
9.4. Werkende demo omgeving in de Azure demo-tenant	14
9.5. Goedkeuring door de technische begeleiding	14
9.6. Project oplevering binnen de deadline	14
GLOSSARY	15
GENERATIEVE AI POLICY	16
REFERENTIELIJST	17

1. Inleiding

Dit document beschrijft de details van het probleem waaraan ik gedurende mijn stage aan zal werken. Het bevat een toelichting van de bedrijfscontext, de probleemomschrijving en de doelstellingen. Daarnaast wordt een scope gedefinieerd, wat zijn de deliverables die worden afgegeven op het einde van het project, wie alle stakeholders zijn die bij het project betrokken zijn, alle mogelijke risico's die zouden kunnen oplopen tijdens dit project, een projectplanning voor de komende weken en alle succes criteria om te evalueren wanneer het project afgerond is.

Het doel van dit document is om het project duidelijk te definiëren, de doelstellingen vast te leggen en alle details duidelijk te documenteren voordat het project van start gaat.

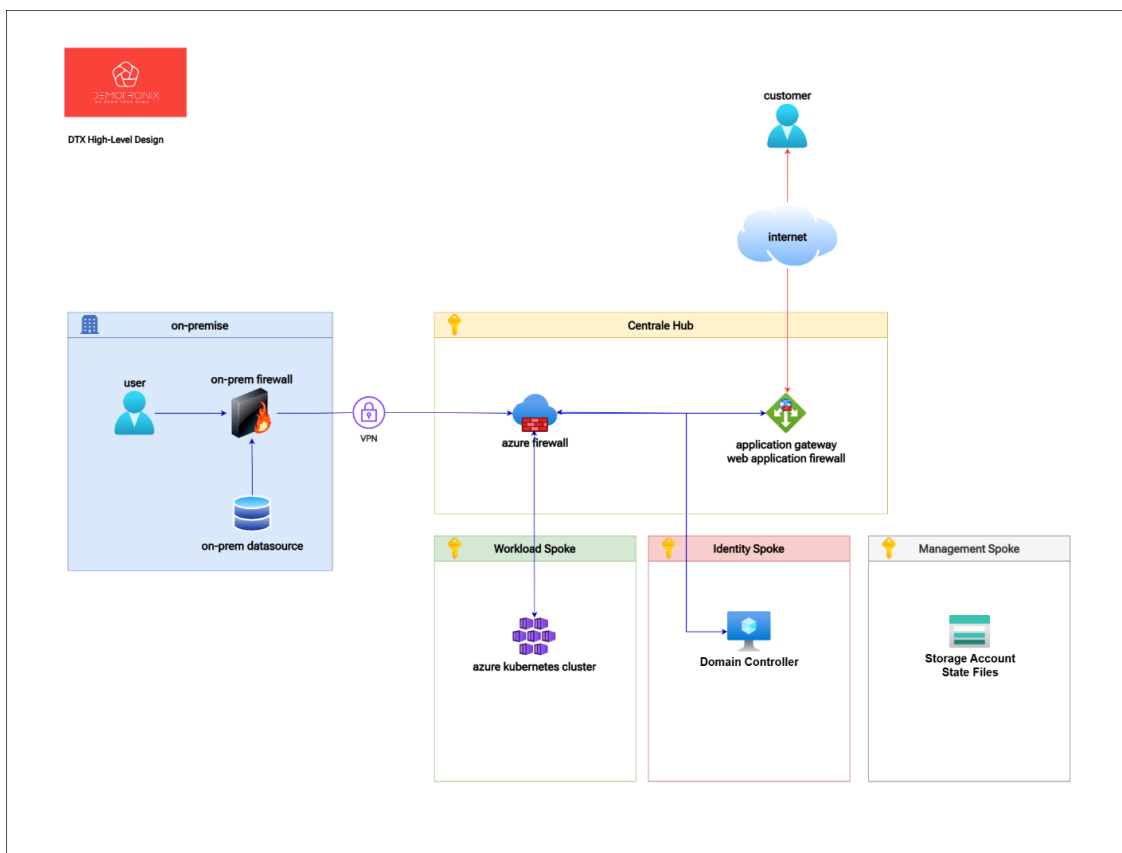
2. Probleemomschrijving

Arxus is een Cloud Service Provider en een Microsoft Azure Expert Managed Service Provider (MSP)-partner. Om klanten een consistente en betrouwbare service te bieden, zet Arxus sterk in op het standaardisatie van Azure Landing Zones, een volledige functionele cloud omgeving voor bedrijven, en op de automatisering van Azure deployments. Deze implementaties worden uitgevoerd volgens de richtlijnen van het Microsoft Cloud Adoption Framework (CAF), een Microsoft richtlijn voor het uitrollen van Landing Zones omgevingen.

Binnen deze context bestaat er voor de referentieklant "Demotronix" een standaard Azure Landing Zone architectuur. Het huidige probleem is dat deze architectuur voornamelijk gedefinieerd staat op conceptueel niveau en dient uitgewerkt te worden naar een geautomatiseerde infrastructuur implementatie. Bovendien zou deze omgeving on-demand moeten opgebouwd en afgebroken kunnen worden voor demonstratiedoeleinden aan klanten waarna de mogelijkheid bestaat om deze omgeving af te zetten om kostenbesparend te werken.

Het doel van dit project is om deze architectuur om te zetten naar Infrastructure as Code met behulp van Terraform, zodat de volledige uitrol van het infrastructuur omgeving geautomatiseerd wordt en in code gedefinieerd staat. Hiervoor wordt gebruikgemaakt van de interne Terraform module library van Arxus. De infrastructuur zal automatisch uitgerold worden naar een bestaande Azure demo tenant via GitHub Actions pipelines, dit laat toe om veranderingen in de omgeving continue en eenvoudig te implementeren en on-demand omgevingen te bouwen en af te breken.

(zie [Glossary](#))



(Arxus NV, 2026)

2.1. Functionele Vereisten

Dit is concreet samengevat wat het project specifiek moet doen:

- Het implementeren van een Azure Landing Zone-omgeving met Terraform
- Op aanvraag opbouwen en afbreken van de Azure Landing Zone-omgeving.
- Gebruikmaken van de interne Terraform module library van Arxus
- Automatische deployment van de Infrastructuur via GitHub Actions pipelines
- Implementatie van omgeving binnen een bestaande Azure demo-tenant

2.2. Niet-Functionele Vereisten

Dit vat kort samen de belangrijke voorwaarden en standaarden waaraan het project moet voldoen:

- De oplossing moet reproduceerbaar zijn via Infrastructure as Code
- De implementatie moet gestandaardiseerd zijn volgens interne Arxus standaarden
- De infrastructuur moet redundant en modulair zijn
- De oplossing moet voldoen aan de standaarden en richtlijnen van Microsoft CAF
- De uitrollen moet geautomatiseerd en betrouwbaar verlopen
- De volledige oplossing en beheer ervan is in code volgens de GitOps principes (zie [Glossary](#))

3. Doelstelling

Het doel van dit project is het ontwikkelen van een geautomatiseerde en gestandaardiseerde Azure omgeving die on-demand kan worden opgebouwd en afgebroken kan worden voor demonstratiedoeleinden aan klanten. Deze omgeving zal dienen als voorbeeldimplementatie voor klanten en zal de architectuurprincipes van een Azure Landing Zone, volgens het Microsoft CAF framework, volgen.

Dit SMART-doelstelling moet voldoen aan alle 5 vereisten, namelijk:

3.1. Specifiek

Het project heeft als doel een Azure Landing Zone omgeving te implementeren met behulp van Terraform en de interne Terraform module library van Arxus. De infrastructuur wordt uitgerold via GitHub Actions pipelines naar een Azure demo-tenant en volgt de beste praktijken van Microsoft.

3.2. Meetbaar

Het project wordt als succesvol beschouwd wanneer:

- De volledige Azure omgeving automatisch kan worden uitgerold via verschillende GitHub Actions workflows.
- De omgeving on-demand volledig opgebouwd en afgebroken kan worden via een automatisch proces.
- De implementatie maakt gebruik van de bestaande in-house Terraform module library voor het uitrollen van alle Azure componenten.
- De Azure Landing Zone omgeving wordt volledig opgezet volgens de Microsoft-richtlijnen en volledig gebaseerd op de vooropgestelde architectuur design.

3.3. Acceptabel

De oplossing sluit aan bij de bestaande standaarden, tools en werkwijzen die binnen Arxus worden gebruikt voor Azure implementaties en automatisering.

3.4. Realistisch

Het project maakt gebruik van bestaande architectuurdocumentatie- en diagrammen, een al bestaande Terraform module library en een bestaande Azure demo-tenant. Waardoor de implementatie binnen de beschikbare tijd van de stage haalbaar is.

3.5. Tijdsgebonden

De implementatie, demonstratie en resolutie documentatie (thesis) van de geautomatiseerde Azure Landing Zone omgeving worden afgerond binnen de looptijd van de stage.

4. Scope

4.1. In-Scope

De scope van dit project omvat het ontwikkelen van een geautomatiseerde Azure Landing Zone omgeving die kan worden gebruikt voor demonstraties aan klanten. De implementatie zal gebaseerd zijn op een bestaand referentiearchitectuur gemaakt voor de fictieve klant "Demotronix" en zal worden uitgewerkt volgens de richtlijnen van Microsoft.

Binnen dit project worden de volgende onderdelen gerealiseerd:

- Het omzetten van een bestaande Azure Landing Zone architectuur en documentatie naar Infrastructure as Code (IaC) met behulp van Terraform.
- Het gebruik van de interne Terraform module library van Arxus om de infrastructuur te definiëren en te beheren.
- Het deployen van de Azure Landing Zone naar een bestaande Azure demo-tenant.
- Het gebruiken van GitHub Actions pipelines voor de automatische en on-demand deployment van de infrastructuur.
- Het voorzien en opzetten van een GitHub Actions pipeline voor het on-demand verwijderen van de volledige Azure Landing Zone.
- Het implementeren van de omgeving volgens de beste praktijken en richtlijnen van Arxus en Microsoft.
- Het opleveren van een gestandaardiseerde demo omgeving die gebruikt kan worden voor demonstraties naar klanten.
- Alle code wordt beheerd en bewaard in de Demotronix GitHub Organisation en de repositories hierin volgens de GitOps principes.

4.2. Out-of-Scope

De volgende onderdelen vallen buiten de scope van dit project:

- Het ontwikkelen van nieuwe Terraform modules voor de interne module library van Arxus.
- Het maken of aanpassen van de bestaande Azure tenant architectuur buiten de vereisten van de demo omgeving
- Het maken of aanpassen van project gerelateerde documentatie.
 - Deze documentatie kan in de toekomst verder aangepast worden indien nodig en dit wordt goedgekeurd door de verantwoordelijke Solution Architect.
- Het implementeren van productieomgevingen voor (non-fictieve) klanten.
- Het configureren van klant specifieke applicaties of workloads binnen de omgeving
 - Kan in de toekomst veranderen indien nodig en dit wordt goedgekeurd door de verantwoordelijke Cloud Expert.
- Het uitvoeren van operationeel beheer en langdurige monitoring van de omgeving na deployment.
- Het aanpassen van de algemene richtlijnen of architectuurprincipes.

Het project richt uitsluitend op het automatiseren en demonstreren van een gestandaardiseerde Azure Landing Zone omgeving binnen een demo context.

5. Deliverables

Binnen dit project worden een aantal concrete resultaten opgeleverd die bijdragen aan het realiseren van een geautomatiseerde en gestandaardiseerde demo Azure Landing Zone. De deliverables van dit project zijn de volgende:

5.1. Terraform Infrastructuurcode

Alle Terraform code nodig om de Azure demo omgeving op te zetten. Deze code definieert alle infrastructuur onderdelen in de Demotronix Azure omgeving. Deze architectuur is gebaseerd op de bestaande referentiearchitectuur voor Demotronix. Deze configuratie maakt gebruik van de interne Terraform module library van Arxus. Deze code wordt bewaard en beheerd in de Demotronix GitHub Organisation. Code repositories en de GitHub pipelines in de Organisation worden aan het einde van dit project overhandigd aan Arxus.

5.2. Werkende Azure demo omgeving

Een volledige functionele Azure Landing Zone omgeving die gedeployed kan worden in de bestaande Azure demo-tenant. Deze omgeving is gestandaardiseerd en kan gebruikt worden voor demonstraties aan klanten. De Landing Zone zal on-demand kunnen opgebouwd en afgebroken worden.

6. Stakeholders

Binnen dit project zijn verschillende stakeholders betrokken die een rol spelen in de uitvoering, begeleiding en het gebruik van de oplossing. Hieronder volgt een overzicht van de belangrijkste betrokken partijen.

6.1. Stagiair

Verantwoordelijk voor de uitvoering van het project. Dit omvat het implementeren van de Azure Landing Zone met Terraform, het deployen van infrastructuur in de demo-tenant via de deployment- en destroy workflows en het testen van deze oplossing.

6.2. Solution Architect

De solution architect (Jan) binnen Arxus fungeert als technische begeleider op vlak van de Azure architectuur van het project. Hij geeft richting en verduidelijking aan de architectuur, valideert architectuur keuzes, valideert juiste standaardisatie implementatie en keurt eventuele wijzigingen aan de architectuur.

De solutions architect verwacht van mij om de volledige oplossing te bouwen volgens de standaarden van zowel Arxus als Microsoft.

6.3. Cloud Expert

De Cloud Expert (Brent) binnen Arxus fungeert als technische begeleider op vlak van Terraform, GitHub pipelines en Azure technische expert. Hij geeft richting en verduidelijking aan de Terraform module library.

Bovendien valideert hij de Terraform code via pull-request approval. Hierbij wordt bij elke aanvraag om code te laten deployen in Azure, eerst een validatie test uitgevoerd om te controleren of de code werkt en wat het exact aan de infrastructuur verandert. Dit plan moet goedgekeurd worden door een Cloud Expert voordat het zal draaien in Azure.

Ten slotte geeft hij verduidelijking en richting aan het Terraform deployment proces in Azure, de GitHub pipelines workflow en volgt hij de algemene voortgang op. Zodat het project aansluit bij de interne werkwijze en standaarden van het bedrijf.

De cloud expert verwacht dat ik deze oplossing volledig kan bouwen via Infrastructure as Code en dat het uitrollen van de omgeving volledig geautomatiseerd gebeurt via het gebruik van de Terraform module library.

6.4. Arxus

Het bedrijf treedt op als organisatorische stakeholder. De resultaten van het project dragen bij aan de verdere standaardisatie en automatisering van Azure implementaties binnen het bedrijf en kunnen gebruikt worden voor demonstraties aan klanten.

Hierdoor verwacht Arxus dat deze oplossing modulair en redundant wordt opgebouwd zodat deze in de toekomst als template kan gebruikt worden voor toekomstige collega's en stagiairs om nieuwe omgevingen op te bouwen.

6.5. Referentieklient "Demotronix"

De architectuur waarop dit project gebaseerd is, is een referentiearchitectuur voor de fictieve klant "Demotronix". Hoewel er geen productieomgeving voor deze klant wordt gebouwd, dient deze architectuur als basis voor de implementatie van de demo omgeving.

6.6. Potentiële klanten

De uiteindelijke demo omgeving kan gebruikt worden door consultants en architecten van Arxus om Azure Landing Zone architecturen en automatiseringsmogelijkheden te demonstreren aan (potentiële) klanten. Deze potentiële klanten verwachten hierdoor een kwaliteitsvolle en waarheidsgetrouwe demo.

7. Risico's

Tijdens de uitvoering van dit project kunnen verschillende risico's optreden die een invloed kunnen hebben op de voortgang of het eindresultaat. Hieronder worden de belangrijkste potentiële risico's en hun impact beschreven.

7.1. Complexiteit van de Azure Landing Zone architectuur

De implementatie van een Azure Landing Zone volgens de richtlijnen van het Microsoft CAF richtlijnen kan technisch complex zijn. Dit kan leiden tot extra tijd die nodig is voor research, analyse, configuratie en troubleshooting.

7.2. Aanleren van de Terraform module library

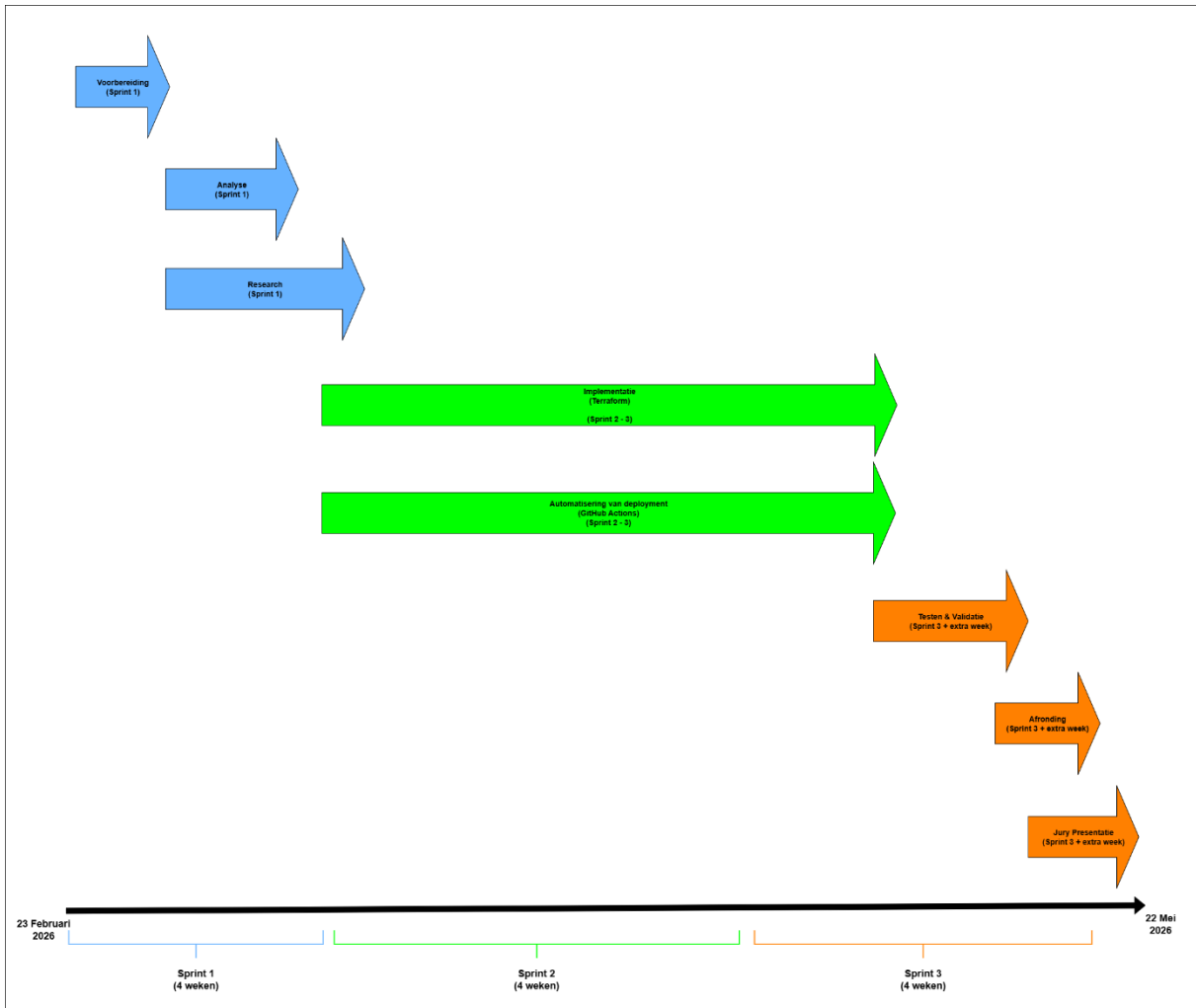
Het project maakt gebruik van de interne Terraform module library van Arxus. Het aanleren en fatsoenlijk toepassen hiervan kan leiden tot extra tijd die nodig is voor research, analyse, configuratie en troubleshooting. Dit kan de implementatie vertragen.

7.3. Tijdsbeperking van de stage

Het project moet worden uitgevoerd binnen de beschikbare stageperiode. Onverwachte vertraging kan een invloed hebben op de planning. In het slechtste geval zouden project aspecten aangepast moeten worden om de deadline te halen.

8. Planning

Het project wordt uitgevoerd gedurende de stageperiode en is opgedeeld in verschillende fasen. Elke fase richt zich op een specifiek onderdeel van het project en bouwt verder op de resultaten van de vorige fase. De planning hieronder geeft een overzicht van de belangrijkste activiteiten.



(Eyckmans, 2026)

8.1. Sprint 1

In deze fase wordt de bestaande Azure Landing Zone architectuur voor de referentieklient Demotronix geanalyseerd. Hierbij worden de verschillende Azure resources en services onderzocht. Daarnaast wordt het Microsoft CAF framework verder bestudeerd en wordt de interne Terraform module library van Arxus verkend. Het doel van deze fase is het zich eigen maken van de architectuur, de vereisten en de gebruikte tools.

8.2. Sprint 2

In deze fase wordt de Azure Landing Zone architectuur effectief geïmplementeerd in Terraform. De configuraties worden opgebouwd met behulp van de interne Terraform module library van Arxus.

Hierbij worden GitHub Actions pipelines gebruikt die het mogelijk maken om de infrastructuur automatisch te deployen en opnieuw te verwijderen. Hiermee wordt het mogelijk om de demo omgeving on-demand op te starten en af te breken.

8.3. Sprint 3

De geïmplementeerde oplossing wordt getest in de Azure demo-tenant. Hierbij wordt gecontroleerd of de infrastructuur correct wordt gedeployed, de omgeving correct functioneert en of de implementatie in overeenstemming gebracht is met de richtlijnen.

In de laatste fase wordt het project opgeleverd en gevalideerd door de betrokken Solution Architect en Cloud Expert. Bij feedback, kunnen we dit nog verder aanpassen en/of implementeren. Enkel bij goedkeuring van alle stakeholders, wordt het project pas afgerond.

Na de afronding van het project, zal er tijdens de werkuren binnen Arxus aan het jury presentatie gewerkt worden. Dit zal ook gevalideerd worden door een betrokken Stakeholder. Bij feedback, zal dit verder aangepast worden.

Na alle goedkeuring, wordt er op kantoor een dry-run presentatie gehouden om dit in te oefenen en te valideren of de Stagiair klaar is voor te presenteren in Juni.

9. Succescriteria

Het project wordt als succesvol beschouwd indien de in hoofdstuk 3 geformuleerde [Doelstellingen](#) zijn behaald en de gerealiseerde oplossing voldoet aan de vooropgestelde technische en functionele vereisten. De volgende criteria dienen als basis voor de evaluatie van het projectsucces.

9.1. Geautomatiseerde deployment van de omgeving

De Azure Landing Zone omgeving kan volledig automatisch worden deployed via een GitHub Actions pipeline zonder manuele configuratiestappen.

9.2. On-demand opzetten en verwijderen van de omgeving

De demo omgeving kan op aanvraag opgezet en opnieuw verwijderd worden via automation. Waardoor deze eenvoudig gebruikt kan worden voor demonstraties aan klanten.

9.3. Gebruik van gestandaardiseerde infrastructuurcode

De implementatie maakt gebruik van Terraform en de interne Terraform module library van Arxus, waardoor de infrastructuur op een gestandaardiseerde en reproduceerbare manier wordt beheerd. Deze implementatie volgt de Microsoft-richtlijnen op.

9.4. Werkende demo omgeving in de Azure demo-tenant

De infrastructuur kan succesvol worden opgezet naar de bestaande Azure demo-tenant en functioneert zoals verwacht binnen deze omgeving.

9.5. Goedkeuring door de technische begeleiding

De oplossing (en de bijbehorende documentatie) word(en) gevalideerd en goedgekeurd door de betrokken Solution Architect en Cloud Expert begeleiders binnen Arxus.

9.6. Project oplevering binnen de deadline

De oplevering van het voltooid project met alle nodige goedkeuring gebeurt voor de deadline verstrekt en de stageperiode eindigt.

Glossary

Term	Definitie
Azure demo-tenant	Het afgebakend stuk van Microsoft Azure waarin alle Clouddiensten, gebruikers en instellingen van een organisatie beheerd worden.
Azure Kubernetes Service (AKS)	Een managed service binnen Azure voor het implementeren, beheren en schalen van Kubernetes clusters. AKS vereenvoudigt het beheren van gecontaineriseerde applicaties in Azure zonder de onderliggende controle plane te onderhouden.
Azure Landing Zone	Gestructureerde en geoptimaliseerde Azure omgeving. Een omgeving ontworpen om zo veilig, flexibel en optimaal mogelijk te zijn. Kan modulair aangepast worden aan de behoeften van een organisatie dankzij het hub-spoke design.
Continuous Integration & Continuous Delivery (CI/CD)	Verschillende werkwijzen om het software ontwikkeling proces te automatiseren en te versnellen. Bij nieuwe code veranderingen in de code repository zou dit automatisch deze code testen, integreren met de bestaande code en het uitrollen van de infrastructuur. Dit laat toe om frequente veranderingen, reparaties en nieuwe implementaties snel door te voeren. Deze automatie implementatie wordt soms ook een deployment pipeline of gewoon een pipeline genoemd.
GitHub Actions pipeline	Het Continuous Integration & Delivery dienst van GitHub. Het volledige proces van code migreren, testen en uit te rollen gebeurt rechtstreeks vanuit de code repository in GitHub en gebeurt automatisch met minimale manuele interactie zodra er veranderingen zijn in de code.
GitHub Organisation	Een privé stuk in GitHub waarin alle code repositories van een organisatie kan bewaart en beheerd worden. Allemaal privé en geïsoleerd.
GitOps	Een werkwijze waarbij je infrastructuur en applicatie configuratie beheert via Git repositories, die als enige bron van waarheid dienen. Deployments gebeurt automatisch op basis van wijzigingen in Git.
Infrastructure as Code (IaC)	Het automatisch opzetten en beheren van IT-infrastructuur via code in plaats van handmatige interacties. Hierdoor krijg je consistente, herhaalbare en snel uitrolbare omgevingen.
Ingress	Een unieke Kubernetes onderdeel die inkomende HTTP/HTTPS verkeer van buiten de cluster naar interne services routeert via regels (zoals host- of pad gebaseerde routing). Meestal wordt dit beheert door een ingress controller.
Microsoft Azure Expert MSP partner	Het officiële Microsoft certificaat om als Cloud Solution Provider bedrijf samen te werken met Microsoft als partners. Dit certificaat bewijst dat dit bedrijf uitblinkt in het verkopen en beheren van Azure services bij klanten. Met deze partnership krijgt dit bedrijf verschillende voordelen zoals kortingen op Azure licenties & subscriptions en prioriteit in Microsofts "referral engine", dat het bedrijf helpt om nieuwe klanten te kunnen krijgen. Maar een partner worden is een complex proces. Dit omvat verschillende audits en allerlei hoge verwachtingen. Dit certificaat moet ook jaarlijks hernieuwd worden.
Microsoft Cloud Adoption Framework (CAF)	Richtlijnen en best practices om bedrijven te helpen Azure Cloud te implementeren in hun huidige IT-omgeving.
Pods	De kleinste uitvoerbare eenheid in Kubernetes. Een Pod bevat een of verschillende containers die samen draaien, dezelfde netwerkconfiguratie delen en toegang hebben tot gedeelde opslag. Pods worden gebruikt om applicaties of onderdelen van applicaties te draaien in een Kubernetes cluster.
SKU (Stock Keeping Unit)	Een identificatiecode voor specifieke productvarianten binnen Azure te onderscheiden. Een SKU is een specifieke configuratie of versie van een Azure service die bepaalt welke functies, prestaties en prijs van toepassing zijn. Elke SKU komt overeen met een bepaalde capaciteit of service niveau binnen een Azure product. Zoals VM size, opslagtype of database tier.
Terraform	Terraform is een Infrastructure as Code tool waarmee infrastructuur automatisch wordt opgezet en beheert via code, grotendeels onafhankelijk van de Cloud provider.
Terraform module library van Arxus	De interne en zelfgemaakte Terraform module collectie. Deze collectie wordt gebruikt om de volledige infrastructuur omgeving in code te definiëren en te bouwen. Het gebruik van deze collectie standaardiseert het proces van Azure resource deployment via Terraform.
Weighted Ranking Method	De Weighted Ranking Method is een gestructureerde beslissingsmethode waarbij verschillende alternatieve systematisch met elkaar vergeleken wordt op basis van vooraf gelegde criteria. Elk criteria krijgt een bepaald gewicht gebaseerd op het belang ervan. Elk alternatief wordt per criteria gescoord, om zo uiteindelijk een totaalscore te berekenen. Hierdoor kan er constructief en systematisch keuzes onderbouwd worden.

Generatieve AI Policy

Tijdens het schrijven van deze thesis, is er generatieve AI gebruikt om het geschreven werk te verbeteren. Specifiek de Copilot tool van Arxus is gebruikt om mijn geschreven werk te verbeteren. Zoals controleren op grammatica, spelling, formaliteit en algemene zinsbouw. Alle prompts die gebruikt zijn bij het schrijven van deze stagedocumenten zijn allemaal in de aard van "Verbeter deze tekst op grammatica, spelling, formaliteit en algemene zinsbouw." geformuleerd. Waarna ik deze output zelf nog gecontroleerd heb op kwaliteit, accuraatheid en aangepast aan mijn persoonlijke voorkeur. Dit allemaal om tot de definitieve uitkomst te geraken.

AI is altijd voorzichtig gebruikt en met een korrel zout opgenomen, het is nooit blindelings geïmplementeerd in het document. Want AI is niet perfect, het maakt regelmatig fouten of geeft een zelfverzonnen antwoord zonder bron. Bovendien is het partijdig (Copilot heeft voorkeur naar Microsoft producten bijvoorbeeld) en beperkt tot bronnen die dateren tot en met 2025. Hierdoor is AI niet een definitieve bron van waarheid.

Referentielijst

Arxus NV. (2026). *Demotronix-Arxus-SolutionDesign_v1.0*. Kontich: Arxus.

Eyckmans, S. (2026). Project Planning Diagram.

Microsoft. (2026, Juny 3). *Cloud-adoption-framework*. Opgehaald van [learn.microsoft.com](https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/):
<https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/>